Genomics-Based Security Protocols: From Plaintext to Cipherprotein

Harry Shaw
Microwave and Communications Systems Branch
NASA/Goddard Space Flight Center
Greenbelt, MD, USA
harry.c.shaw@nasa.gov

Sayed Hussein, Hermann Helgert
Department of Electrical and Computer Engineering
George Washington University
Washington, DC, USA
drsay@gwu.edu, hhelgert@gwu.edu

Abstract— **The evolving nature of the internet will require continual advances in authentication and confidentiality protocols. Nature provides some clues as to how this can be accomplished in a distributed manner through molecular biology. Cryptography and molecular biology share certain aspects and operations that allow for a set of unified principles to be applied to problems in either venue. A concept for developing security protocols that can be instantiated at the genomics level is presented. A DNA (Deoxyribonucleic acid) inspired hash code system is presented that utilizes concepts from molecular biology. It is a keyed-Hash Message Authentication Code (HMAC) capable of being used in secure mobile Ad hoc networks. It is targeted for applications without an available public key infrastructure. Mechanics of creating the HMAC are presented as well as a prototype HMAC protocol architecture. Security concepts related to the implementation differences between electronic domain security and genomics domain security are discussed.**

Keywords- HMAC; keyed Hash Message Authentication Code; Cryptography; DNA; PKI; public key infrastructure; MANET; cipherprotein; epigenetics

I. INTRODUCTION

The ability to authenticate the identity of participants in a network is critical to network security. Bimolecular systems of gene expression "authenticate" themselves through various means such as transcription factors and promoter sequences. They have means of retaining "confidentiality" of the meaning of genome sequences through processes such as control of protein expression. These actions occur independently of a centralized control mechanism. The overall goal of the research is to develop practical systems of authentication and confidentiality such that independence of authentication and confidentiality can occur without a centralized third party system.

Genes are capable of expressing a wide range of products such as proteins based upon an alphabet of only four symbols. This research implements a keyed-HMAC system using a DNA-based code and certain principles from molecular biology. The system will permit Mobile Ad hoc Networks (MANET) to distinguish trusted peers, yet tolerate the ingress and egress of nodes on an unscheduled, unpredictable basis. The system allows for authentication without a Public Key Infrastructure (PKI), X.509 certificates, RSA and nonce exchanges, etc. It also provides for a biological authentication capability.

This paper is organized as follows:

- A description of the elements of the prototype genomic HMAC architecture
- A description of the DNA code encryption process, genome selection and properties
- The elements of the prototype protocol architecture and its concept of operations
- A short plaintext to ciphertext encryption example.
- Description of the principles of gene expression and transcriptional control to develop protocols for information security. These protocols would operate in both the electronic and genomic domains.

This paper will move between the electronic and genomics contexts when discussing the protocols and their potential instantiation. This scheme can be used to create encrypted forms of gene expression that express a unique, confidential pattern of gene expression and protein synthesis. The ciphertext code carries the promoters (and reporters and regulators) necessary to control the expression of genes in the encrypted chromosomes to produce cipherproteins. Unique encrypted cellular structures can be created that can be tied to the electronic hash code to create biological authentication and confidentiality schemes.

I. ELEMENTS OF THE GENOMICS HMAC ARCHITECTURE

Plaintext is mapped into a reduced representation consisting of an alphabet of q letters, where q = 4 for a genomic alphabet such as DNA or Ribonucleic acid (RNA), q = 20 for proteomic alphabet, or other values when representing other functions in molecular biology, e.g., histone code. The actual HMAC requires additional base representations beyond the four DNA bases, but the minimum requirement is shown in (1) and (2). B is the set of DNA bases A, T, C and G, which represent the molecules adenine, thymine, guanine and cytosine and represent the entire alphabet of the genomic hash code. DNA bases have the property that the only permitted pairs are Watson-Crick matches (A-T), (C-G), thus, the binary representations of B and B' sets are complimentary such that a r-bit length sequence of B_q and B'_q maintain the identity property shown in (3). Assignment of letter to DNA base sequences is

performed. Letters with greater frequency can be assigned shorter DNA sequences to reduce the code size.

A. *Lexicographic and DNA representation of plaintext*

Plaintext words, P are converted into a numerical form suitable for subsequent coding into the cryptographic alphabet of the required code. Plaintext words are coded such that a lexicographic order is maintained between words, i.e., the numerical forms may take either integer or floating point representations. F is a function that converts the plaintext to lexicographic numerical form. D represents the numerical form of the dictionary (lexicographically ordered set) such that $D_{1,..n}$ represents the set of all words. The subset of $D_{1,..i}$ represents the subset of words in the plaintext message. The function U assigns the DNA base sequence corresponding to the D_i as shown in (4), (5) and (6). L is the plaintext message coded into the DNA alphabet found in sets B and B'.

B. *Sentence-message order coding*

A system of linear equations codes the lexicographic position of each word relative to the sentence position of each word. This complicates detection of words based upon frequency analysis. Multiple appearances of the same word are uniquely coded. As a minimum requirement, if there are i DNA representations in the message, and n represents a numerical sequence related to the number of DNA representations in the message (the simplest case being i = 1, 2, 3, ..., n), then the system of linear equations shown in (7) provide the solutions for sentence-message order coding.

$$B_q = \{A, T, C, G\} \tag{1}$$

$$B'_q = \{T, A, G, C\} \tag{2}$$

$$1 = B^r_q \oplus B'^r_q \qquad \forall\, r = 1, \ldots, q. \tag{3}$$

Equations 1 and 2 define the sets containing the DNA bases that comprise the alphabet for the HMAC code. Equation 3 defines the complimentary relationship required for the binary representations of the members of that space. For example: the XOR product of the r^{th} bit of A and T is a one as is true for T and A, C and G, G and C.

$$D_i = F(P_i) \ni D_i < D_{i+1} \;\forall\, i < n \tag{4}$$

$$L = U(D_1, B_q) \,\|\, U(D_2, B_q) \,\|\, \ldots \,\|\, U(D_i, B_q) \tag{5}$$

$$L' = U(D_1, B'_q) \,\|\, U(D_2, B'_q) \,\|\, \ldots \,\|\, U(D_i, B'_q) \tag{6}$$

Equation 4 defines each word in the message, P_i as a member of a set of all words in a lexicographically ordered dictionary. Equations 5 and 6 show the operation of the function that assigns a DNA sequence using the members of the set of DNA bases to a coding of concatenated sequences labeled L and L'. L and L' maintain the same complimentary relationship that is a property of the individual DNA bases in the sets B_q and B'_q.

This yields a series of coefficients $x_1, x_2, \ldots, x_i$ that are concatenated as shown in (8). The binary representation of each coefficient undergoes bit expansions such that only B_q or B'_q codes are represented in the bit stream created by (8). X represents the relationship between lexicographic coding of the words and their position in the message.

C. *Message coding*

DNA coding on the message is completed by XOR and bit expansions to maintain the DNA base coding in the binary sequence in the operation shown in (9). M is the plaintext message coded into the DNA alphabet and coded again with the sentence-message coefficients. This sequence will be subjected to encryption.

$$\begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_i \end{bmatrix} = \begin{bmatrix} D_1 & D_2 & D_3 & \ldots & D_i \\ D_i & D_1 & D_2 & \ldots & D_{i-1} \\ & & & & \\ D_2 & D_3 & D_4 & \ldots & D_1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \\ r_i \end{bmatrix} \tag{7}$$

$$X = x_1 \,\|\, x_2 \,\|\, \ldots \,\|\, x_i \tag{8}$$

$$M = L \oplus X \tag{9}$$

The set of linear equations in equation 7 provide the process of sentence-message order coding using the r^{th} position in the message to code each word of the message. The resulting coefficients are concatenated and XOR'd with the coded plaintext message to produce the ciphertext message.

II. Encryption Process

The use of DNA as a cryptographic medium is not new. Systems using DNA as a one-time code pad [1], image compression–encryption system [2], encryption utilizing dummy sequences of DNA [3]-[4] and DNA watermarks [5] have been published. The approach described herein is a new implementation of DNA-based cryptography targeted at network security.

A. *Chromosome encryption keys*

Approximately 800 genomes have been sequenced [6]. The human genome alone has approximately 3.2 million base pairs. The sets of genomes provides for the possibility of "security by obscurity". Additionally, there is an infinite number of ways to use genome sequences as cryptographic keys. However, genomes have high degrees of redundancy and sequence conservation across species. Consequently, sections of genomes used as keys should be treated as one-time pads. The first step is to select a genome and a sequence from that genome and encode it with the binary representations of B_q and B'_q.

DNA consists of two complimentary sequences, referred to as the sense and antisense strands as shown in Figure 1 [7]. A DNA sequence has a start point called the five-prime end (5') and an endpoint called the three-prime (3'). In biochemistry, the 5' and 3' designations refer to orientation of each strand necessary for proper replication and transcription. The complements are bonded to each other base by base to create base pairs. The antisense strand is oriented in the 3' to the 5' direction, relative to the sense strand. For a DNA encryption key, both sense and antisense strands can be encoded and utilized. Figures 2 and 3 demonstrate two ways of implementing the chromosome encryption key in the HMAC scheme. Figure 2 represents the simplest scheme in which successive bases from the key and message are XOR'd and a single ciphertext message is produced. Encryption proceeds in the 5' to 3' direction using the sense strand. Figure 3 represents a more complex scheme in which both sense and antisense bases from key and message are XOR'd. Encryption proceeds in the 5' to 3' direction in both strands.

B. *Mismatches and Annealing*

The encryption process generates base pair mismatches that do not conform to the A-T, C-G pairing rule. These mismatches are central to creating a one-way hash code. Subsequent to the encryption step, the mismatches are resolved through an annealing process that results in an irreversible transformation of the encryption sequence not directly traceable to the original ciphertext.

III. PROTOTYPE DNA-BASED, KEYED HMAC SYSTEM

Assume a network such as the one shown in Figure 4. Jack, Jill, JoAnn and Lisa wish to form a secure MANET. In the same wireless transceiver space can be found X and Y whose intentions are unknown, but are capable of sending and receiving messages. Jack, Jill, JoAnn and Lisa possess all of the required authentication tools:

- A common genome, C, to use as an HMAC key.
- A pre-shared secret, pss, unique to each party.
- The DNA-based HMAC algorithm.

Consider two authentication scenarios. In the first scenario Jack, Jill, JoAnn and Lisa send and receive cleartext messages using the DNA-based HMAC authentication. If the receiver is not the intended destination, the receiver rebroadcasts
the message with their hash and the process continues until the message reaches the intended receiver or until a message time-out period elapses. X and Y also receive the cleartext messages and hash codes. X and Y may possess the algorithm. However, if X and Y wish to substitute a new message with a valid hash code, or forward the message and have it accepted by the network members, they have to create a valid hash code and checksum, which requires knowledge of the chromosome sequence and valid pre-shared secrets known to the other MANET nodes. The MANET members change their pre-shared secrets on a pre-established basis to thwart a brute force attack to derive the pre-shared secret from the hash code.

In the second scenario, Jack, Jill, JoAnn and Lisa wish to establish a trust relationship before exchanging sensitive information across a MANET. In this case, the participants utilize a confidentiality (encryption) protocol for the messages and establish a chain of custody using keyed HMAC authentication. A hash chain of hash codes is established such that each recipient can determine the origin and subsequent hops of the message. In this case, X and Y cannot read the plaintext and the hash code transcript may be encrypted and compressed with the ciphertext.

A. *Genomic hash code properties*

Table 1 summarizes the properties of the prototype hash code against the requirements for an ideal hash code [8]. Figure 5 provides a flow chart of the genomic hash coding process.

B. *Initialize and Perform Lexicographic and DNA assignments*

The plain text message is read and parsed into 3-word blocks (3WB). Take each word in the string, assign it a lexicographic value of x.yyyy....y where x = 1, ... ,26 corresponding to the first letter of the word and subsequent letters are assigned to each successive decimal place until the entire word is coded in a rational number. Assign a DNA letter code to each letter. Most common English alpha characters use 2-letter codes, the rest use a 3-letter code as shown in table 2. The column labeled 'α' is the English alphabetic character adjacent to its DNA code equivalent. As an example, the short phrase 'jump out windows' is shown in its lexicographic and DNA assigned forms in table 3.

TABLE 1. GENOMIC HASH CODE PROPERTIES.

Property	*Compliance*
Produces a fixed length output.	2560 bits
Can be applied to a block of data of any length	Yes.
H(x) is relatively easy to compute for any message x.	Yes. 12 step process for hash code.
One-way property. For any h, it is computationally infeasible to find H(x)=h	To be determined
Weak collision resistance. For a set of x_i messages, with $y \neq x_i$ for all i, no $H(y)=H(x_i)$ for all i.	Yes.
Strong collision resistance. For any x, with $y \neq x$, no H(y)=H(x)	No. Messages ≤ 128 bits require padding

TABLE 2. SAMPLE OF ALPHA TO DNA CONVERSION CODES.

α	DNA	α	DNA	α	DNA	α	DNA
0	CGG	G	TT	N	TG	U	CT
A	GC	H	AC	O	AG	V	CTG
B	TGT	I	AA	P	GA	W	CAC
C	TC	J	AAG	Q	CCT	X	GTA
D	GT	K	ACT	R	CC	Y	GTT
E	TA	L	AT	S	GG	Z	TAG

TABLE 3. PLAINTEXT TO LEXICOGRAPHIC ORDER AND DNA LETTER CODES.

Plain Text	Lexicographic conversion	DNA conversion
jump	10.211316	AAGCTCGGA
out	15.2120	AGCTCA
windows	23.9144152319	CACAATGGTAGCACGG

C. *Binary representation of the DNA bases*

The four DNA bases (A, T, C, G) are represented by binary sequences (0011, 1100, 1001, 0110). The remaining 12 four-bit sequences code for transitional base sequences that are used to anneal mismatches in the encryption process as shown in table 4. The 'Key' column represents the base in the chromosome encryption key. The 'M' column represents the corresponding base in the DNA coded message.

TABLE 4. ENCRYPTION AND ANNEALING TABLE.

Key	M	Result	Anneal	Key	M	Result	Anneal
A	T	T	G	C	G	G	A
A	A	gA	C	C	A	aA	C
A	C	gC	T	C	C	aC	G
A	G	gG	A	C	T	aT	T
T	A	A	T	G	C	C	C
T	G	cC	G	G	A	tA	G
T	C	cG	A	G	G	tG	A
T	T	cT	C	G	T	tT	T

The 'Result' column represents the results of encrypting the key onto message. The 'Anneal' column represents the final ciphertext base. In an operational system, all codes would be significantly lengthened to thwart brute force attacks.

D. *Encryption, Mismatches and Annealing*

Figure 5 also provides a short example of the encryption and annealing process. Each base in the chromosome is XOR'd against the corresponding base in the message. If the base in the message is the complement of the base in the chromosome, the base in the message is copied to the encrypted output string and then altered to a new base in the annealed output string If the base in the message is not the complement of the base in the chromosome, a transitional base, whose value depends upon the mismatch is written to the encrypted output string. The 5' base always determines the change in the other strand; consequently, a 5' G mismatch always codes for a 3' transitional base. This feature allows tracking of point mutations and provides a future expansion capability for mutations. The annealing process also alters the encrypted result by transforming the positions that are not mismatches.

E. *Cryptographic Genome*

Mycoplasma genitalium G37 (NCBI accession number NC000908.2) is the bacterial genome used as an encryption key in the prototype system. There are a number of characteristics of *M. genitalium* that make it a good candidate as an encryption key base. It may be the smallest, self-replicating genome. It has 580,070 base pairs with 470 predicted coding regions. *M. genitalium* has a low G+C content of 34% (random distribution of basepair content would provide for 50% G-C pairs and 50% A-T pairs). This feature provides some testability advantages. The genome contains 470 predicted protein coding regions, which is also a manageable number of potential cipherproteins [9]. Knowledge of the genome coding characteristics is important in selecting and utilizing genomes as cryptographic keys. Approximately 62,000 base pairs are being utilized from the *M. genitalium* genome for the prototype HMAC.

F. *Protocol for Message Authentication.*

The process is as follows:

- Encode the plaintext message into DNA code (Pre-sense message) 3 words at a time (3 word blocks – 3WB)
- Encrypt with pre-shared secret chromosome key and generate sense and antisense strands.
- Different chromosome segments are used to encrypt each 3WB for increased key confidentiality.
- Combine sense and antisense strands to create a checksum (S).
- Anneal the sense strand (Sender) or the antisense strand (Receiver) removing the transitional bases in the 3WBs.
- Concatenate the first 64 DNA bases from the first nine 3WBs to create the Promoter (P).
- Append the checksum to the Promoter. The Promoter || checksum is the Hash Code, K (2560 bits long). The sender and receiver processes are summarized in Figure 6.

The receiver extracts the Promoter and checksum from the message. The hash code computed at the receiver must have the complement of the Promoter sequence and an exact match of the checksum. Sender and receiver must have the pre-shared secret of the genome, and the location of the first base of the sequence. A sample of the output for the test message 'jump out windows' is shown in Figure 7. The hash code has been truncated for test and presentation purposes.

G. *Short Message Performance*

A critical factor in determining the goodness of a hash code is the ability to satisfy criteria four and five from table 1. A hash code algorithm should not produce identical hash code outputs for two or more different messages. Performance of short messages was evaluated for soft and hard collision resistance. The number of MAC verifications, R, required to perform a forgery attack on a m-bit MAC by brute-force verifications [10] is shown in equation 10:

$$R = 2^{m-1} + (2^{m-1} - 1)/2^{m} \approx 2^{m-1} \qquad (10)$$

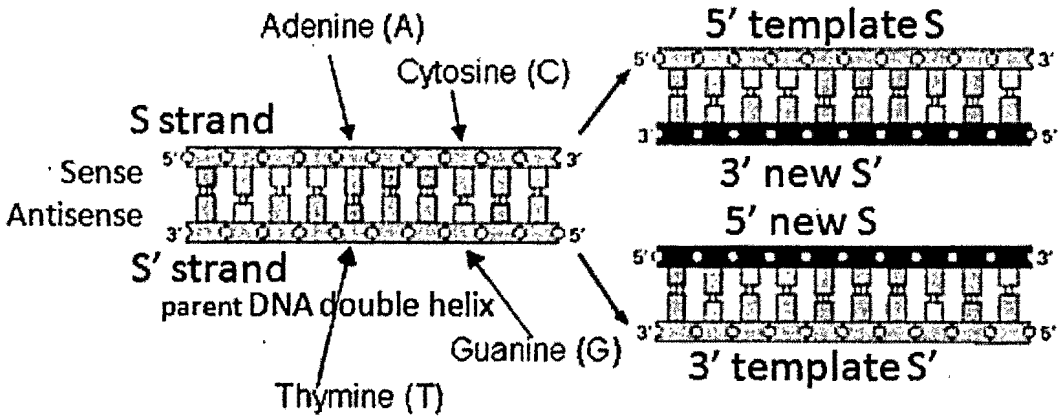


Figure 1. Strand Sequence specification in DNA. A binds with T, C binds with G.

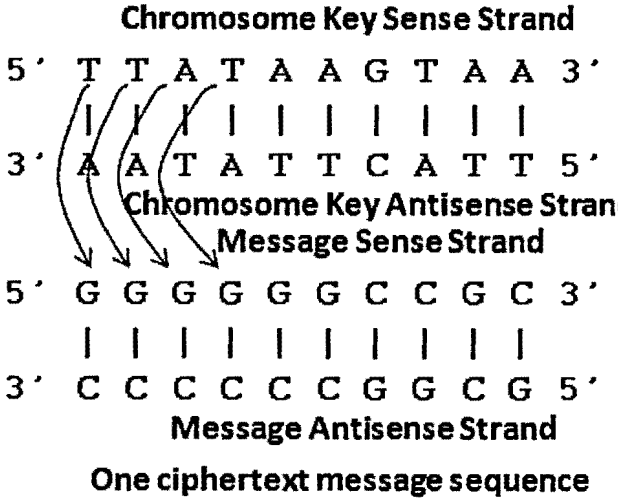


Figure 2. Single strand chromosome encryption utilizing yielding a single ciphertext message sequence.

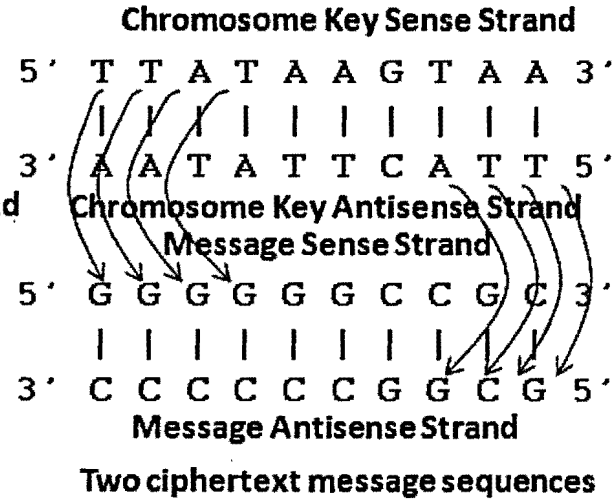


Figure 3. Dual strand chromosome encryption yielding two ciphertext message sequences

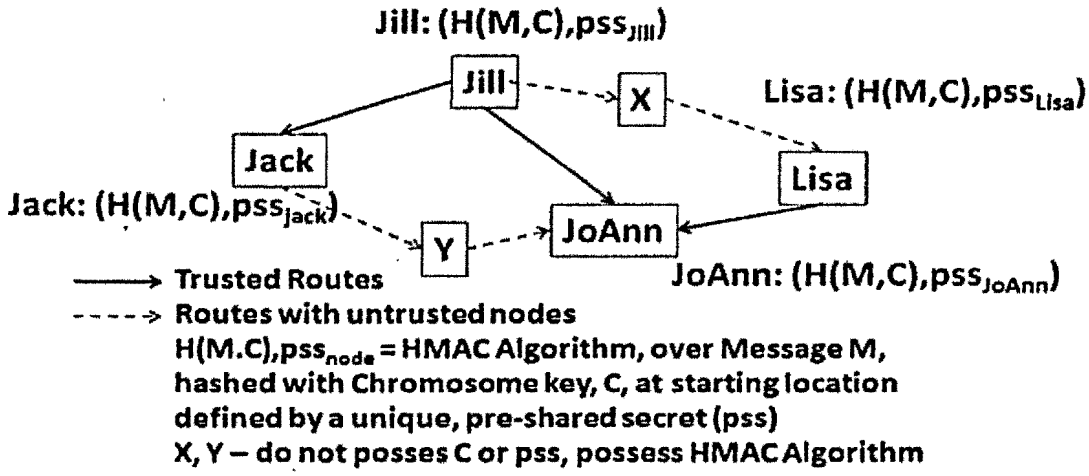


Figure 4. Mobile Ad-hoc Network with trusted and untrusted nodes and routes.

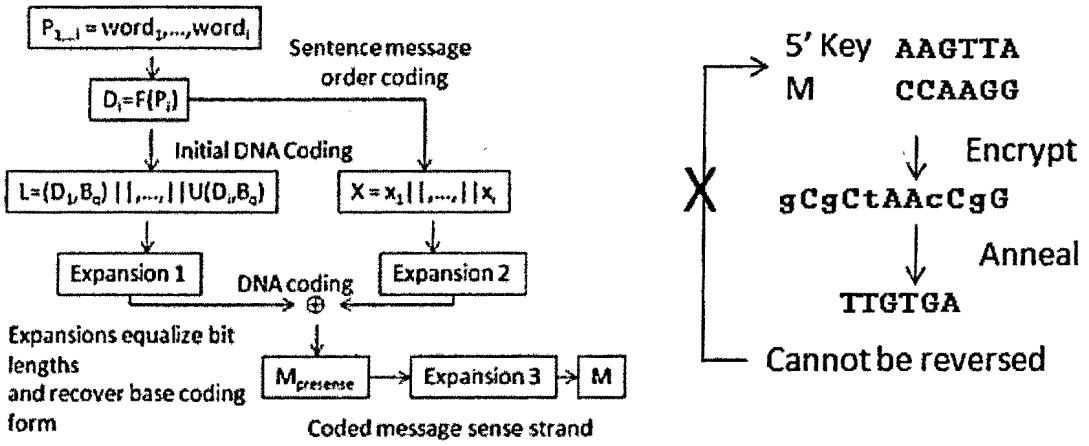


Figure 5. Plaintext coding process, Encryption and Annealing process

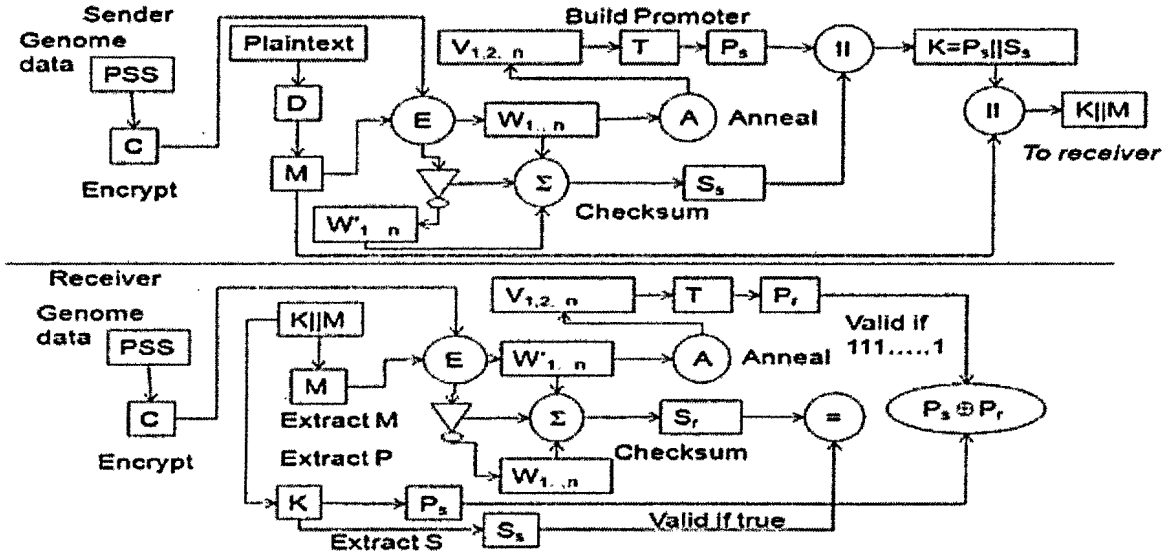


Figure 6. Sender and Receiver Protocol.

DNA Sense strand coded message
AAGCTCGGAAGCTCACACAATGGTAGCACGG

M genitalium key
TTTAGTTATAAGTTATTATTTAGTTAATAAGTTATT
ATT..... GTTATTATTTAGT

checksum = 1871221
'----------Sender copy of hash----------
Sender Hash =
AAAAAAGTATTCTTTGTCAGTGTTTGTGCGTTTCAACCCCTC
AATTAGATTTATAGAGTCCTTC
Plain Text =
jump out windows /
Message Builder 4 - for DNA Hash Code System ended at:
5/21/2009 1:24:47 PM

'----------Receiver copy of hash----------
Sender Hash =
AAAAAAGTATTCTTTGTCAGTGTTTGTGCGTTTCAACCCCTC
AATTAGATTTATAGAGTCCTTC
Receiver Hash =
TTTTTTCATAAGAAACAGTCACAAACACGCAAAGTTGGGGA
GTTAATCTAAATATCTCAGGAAG
Sender Checksum = 1871221
Receiver Checksum = 1871221
Plain Text = jump out windows /
Checksum matched
hash code matched
User and Message Authenticated!
Message Reader 4 for DNA Hash Code System ended at:
5/21/2009 1:32:09 PM

Figure 7. Sample output of sender and receiver performing authentication on the cleartext phrase 'jump out windows'.

The variable R is an upper bound to the brute-force verification limit. Short messages were repeatedly hashed using over different cryptographic sequences to look for collisions. The process is shown in figure 8. Table 5 summarizes the results of those tests.

The single letter message exhibited 403 checksum collisions and 466 hash code collisions. Chromosomes have a high degree of redundancy and repetition; therefore short messages will require padding to eliminate hash code collisions. These statistics utilize different transcripts on the same message to identify potential collisions. These statistics should be indicative of the potential for multiple messages to produce the same hash code from a single transcript. For secure authentication purposes, this code must be implemented with higher level protocols that would block a brute force attack and not reuse genome sequences for authentication. It must also move the starting point in the genome to widely separated start positions to prevent an attacker from guessing the encryption sequence.

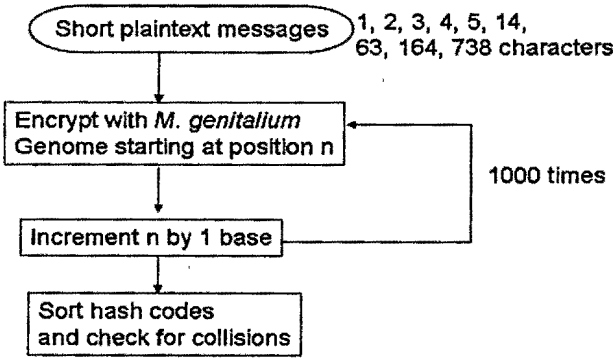


Figure 8. Collision resistance tests for short messages

TABLE 5. SAMPLE OF HASH CODE COLLISIONS

Plain Text	Msg Length	Hash Code Length	Total Hash Code Collisions	Total C/S Collisions	R
z	1	22	466	403	2097152.5
ty	2	30	255	214	536870913
cat	3	36	136	109	3.436E+10
vent	5	64	0	0	9.223E+18
aeiou	6	64	0	0	9.223E+18
jump out windows	17	64	0	0	9.223E+18
jump out windows jump out windows jump out windows jump out	60	256	0	0	5.79E+76
large please require all personnel to take their equipment with them for the work to be performed in 365777 small increments it will be good to get practice on these tasks	202	576	0	0	1.24E+173

A hash code must be secure against the possibility that the cryptographic key, in this case the original genome sequence cannot be recovered from the hash code. Figure 9 represents a small MANET example for developing trust metrics. Assume Jack is broadcasting forward requests to establish a link with Lisa and Lisa is broadcasting return route requests to Jack to establish a return link. Jill is relaying route requests in both directions. Felix wishes to join the MANET. Each node is capable of dynamically appearing and disappearing from the network at will via application of a dynamic source routing protocol. Each node can also take the role untrusted/unknown trust or trusted depending upon the situation. Source and Destination must determine the trustability of a potential route through some quantitative means. In this case successful forward and return route requests (FREQ, RREQ) and route delays are used to create the trust metrics. The sources and destination can set the minimum level of trust for a route via a dynamic fitness algorithm.

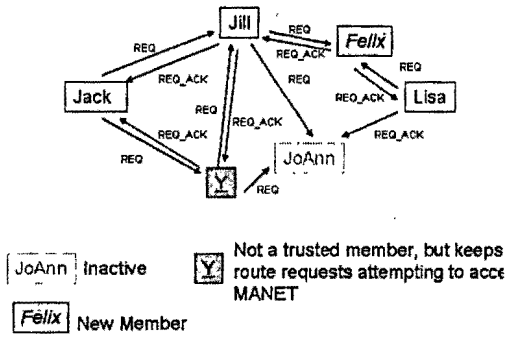


Figure 9. MANET route establishment at a slice in time.

To establish Felix as a trusted member, he relays forward REQs from Jack destined for Lisa and return REQ from Lisa destined for Jack with his DNA HMAC authentication attached. JoAnn, does not respond to route requests and those requests time-out. Y is a malfeasor attempting to breach the network by sending route requests with counterfeit DNA HMAC authentication and analyzing received DNA HMACs for vulnerabilities. Assume that when Y sends a counterfeit route request, genuine nodes respond with negative acknowledgement attached to a genuine authentication code.

The questions to be answered are:

Can Y establish a counterfeit authentication code (hash + checksum) for the current session (however a session is defined)?

Can Y utilize the stolen information to recover information that might be useful for a future network breach?

If Y can recover the original cryptographic sequence, or determine the genome and genome location a cryptographic key was taken from, Y may be able to forge a valid hash code. This could be problematic for a cryptographic sequence due to the high degree of redundancy in the all genomes. For this application, the hash code must be evaluated against the cryptographic key to ensure it has the proper characteristics of diffusion and confusion.

IV. MUTATION EFFECTS, FITNESS, DIFFUSION AND CONFUSION

Life is intolerant of a high mutation rate in its genetic code. Ribonucleic acid (RNA) viruses have the highest mutation rate of any living species, 10^{-3} to 10^{-5} errors/nucleotide and replication cycle [11]. The human DNA mutation rate has been approximated to be on the

order of 10^{-8} errors/nucleotide and generation [12]. Injection of mutations into DNA encrypted messages is an approach to improving the encryption process. Because of the dynamic, evolutionary nature of this approach, potential intruders must continually intercept decoding instructions between source and destination. Missing one generation of genome decryption information seriously corrupts the analysis process. Missing multiple generations eventually renders previous decryption analyses useless.

In evolutionary biology, fitness is a characteristic that relates to the number of offspring produced from a given genome. From a population genetics point of a view the relative fitness of the mutant depends upon the number of descendants per wild-type descendant[13]. In evolutionary computing, a fitness algorithm determines whether candidate solutions, in this case encrypted messages, are sufficiently encrypted to be transmitted. This DNA encryption method uses evolutionary computing principles of fitness algorithms to determine which encrypted mutants should be selected as the final encrypted ciphertext. Two parameters, Confusion and Diffusion are being used as the basis of the fitness criteria. Diffusion and Confusion are fundamental characteristics of ciphers. Shannon [14] describes them as:

a) Diffusion: any redundancy or patterns in the plaintext message are dissipated into the long range statistics of the ciphertext message.

b) Confusion: make complex the relationship between the plaintext and ciphertext. A simple substitution cipher would provide very little confusion to a code breaker.

The challenge is to create a set of FREQ and RREQ messages that hash into codes with a high degree of diffusion and confusion. One strategy for attacking the authentication message is to generate long strings of zeros and identify the correct code for the non-zero positions. If a message generates long strings of zeros it is particularly vulnerable to a key recovery attack because the attacker can reduce the number of bit matches required by the length of zero bit blocks. Table 7 summarizes test results of 1000 trials on messages consisting of zeroes and spaces against the genome. No collisions were identified. The hash code will be tested against all other single character strings to identify patterns. A sample hash code of a string of 217 zeros is shown below in table 6.

TABLE 6. SAMPLE HASH CODE OF STRING OF 217 ZEROS

AATTCTAAGTTCCCGCCCGTCGGTCCGCCGCCCGTCCGGTC
CGCCGCCCGTCCCGGTCCGCCGCAATCTCAATTCTCGCCCG
TCGGTCCGCCGCCCGTCCGGTCCGCCGCCCGTCCCGGTCCG
CCGCCAACTCCAATCTTGCCCGTCGGTCCGCCGCCCGTCCG
GTCCGCCGCCCGTCCCGGTCCGCCGCCCAATCCGAACTTCC
CCGTCGGTCCGCCGCCCGTCCGGTCCGCCGCCCGTCCCGGT
CCGCCGCCCGAACCGTAATTCTCCGTCGGTCCGCCGCCCGT
CCGGTCCGCCGCCCGTCCCGGTCCGCCGCCCGTAACGTTAA
TCTTCGTCGGTCCGCCGCCCGTCCGGTCCGCCGCCCGTCCC
GGTCCGCCGCCCGTCAAGTTCAACTTTAATCCGAACTTCAA
TCGTAACGTTAATCTTTCGTTTAAGTTCAACTTTAATTAATT
CTAATTTCAACCGTAATTCTAACGTTAAGTTCAACTTTCGTT
TCAATTCTAATTTCAATC 10437404

Next the hash codes were compared to the original cryptographic keys to evaluate diffusion and confusion. Table 8 displays four mutation samples from 50 combinations of hash codes on the message 'jump out windows' with encryption keys from the genome. The process was run on 1000 message combinations at a time.

TABLE 7. TEST RESULTS ON REDUNDANT STRINGS OF ZEROES MESSAGES

Plain Text	Hash length	Number of collisions
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	73	0
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	109	0
00000000 00000000	217	0

Mutant 4, for example would be a particularly poor fit due to the number of consecutive matches between the hash code and encryption key. Mutant 10 has only one match of two consecutive bases and a fewer than ¼ of the bases are identical between the hash code and key. Each position in the hash code has 1 of 4 chance of randomly matching the same location in the encryption key.

TABLE 8. SAMPLE MUTANT ENCRYPTIONS FOR HASH CODES AND DNA ENCRYPTION KEY FOR MESSAGE 'JUMP OUT WINDOWS'

ID		64 base pair hash code	Cryptographic key
Mutant	4	AAAAAATGATGG TCCGCCAGTGCTC CGGCTCTCCAAT GCCTGAATCAGA TGGAGAGATTCT GGC	TAAGTTATTATTTAG TAAGTTATTATTTAG TTAAGTTATTATTTA GTTTAAGTTATTATTT AGT
Mutant	10	AAAAAACGATGG CTGGCGATCTCTC CGTTCCCGTAACT CCTGAAGGATAG CTATAGATTCCCT C	TTATAAGTTATTATTT AGTAAGTTATTATTT AGTTAAGTTATTATT TAGTTTAAGTTATTA TTT
Mutant	23	AAAAAAGGAGGG CGGGCCAGTGCT CCGGCTCTTCAAT CGCGTAAGTAGA TCCACAGAGTGT CTG	AAGTTATTATTTAGT TAAGTTATTATTTAG TTTAAGTTATTATTTA GTTATAAGTTATTAT TTA
Mutant	25	AAAAAAGGAGGT TTGTGTAGCGTTT GGGCCCTCGAAC CGGCGAAGGAGA GGGAGATATCTT CCC	GTTAAGTTATTATTT AGTTTAAGTTATTAT TTAGTTATAAGTTAT TATTTAGTTAATAAG TTAT

The confusion metric counts the number of 2-base, 3-base, 4-base and 5-base consecutive matches between the hash code and the key. Each combination actually represents a mutant message which can be further evaluated via a genetic algorithm. One of the major advantages of this system of a conventional encryption system is the ability to provide a set of encrypted outputs from which the most fit (best) member can be selected.

TABLE 9. SAMPLE DIFFUSION AND CONFUSION SCORES FOR HASH CODE FOR MESSAGE 'JUMP OUT WINDOWS'

ID		Diffusion - matching base pair positions	Confusion - consecutive match positions			
			2	3	4	5
Mutant	10	11	1	0	0	0
Mutant	23	11	1	0	0	0
Mutant	25	21	5	1	0	0
Mutant	4	25	9	5	3	2

A. Intronic sequence padding and potential frameshift mutations can increase cryptographic hardness

Padding short messages and short words has been previously discussed as a means to decrease collisions and reduce the likelihood of successfully forging messages. Adding padding to the front of messages as well as the end and padding short words makes it more difficult for an attacker to find the start of the coded message sequence. The analogy in molecular biology is the frameshift mutation in which changing the starting position for a single nucleotide can result in a completely different protein sequence as shown in figure 10. The mechanics of DNA transcription in cells relies on a number of properties to identify the nucleotide triplet sequence that actually transcribes to mRNA which translates to a protein. Some of the mechanics are thermodynamic and biochemical in nature such as DNA folding, binding to transcription factors, and chromatin relaxation in eukaryotes. Some of the mechanics are sequence related. Four types of sequences and mechanisms from molecular biology are directly relevant to this discussion:

a. *Start codon* (e.g. ATG) to specify the transcription start site (three letter sequence that ultimately specifies the first amino acid in the protein to be translated.)
b. *Stop codon* (TAA, TGA, TAG) to end transcription
c. *Promoters.* The function of promoters is different in prokaryotes and eukaryotes, but as a general statement, the promoter is sequence of nucleotides necessary to locate the transcription starting point. In eukaryotic genes that contain a promoter, the sequence often contains the letters 'TATA' hence the term 'TATA box'.
d. *Enhancers.* In eukaryotes, a variety of sequences upstream and downstream from the transcription site provide binding sites for transcription factors (proteins) necessary to enhance protein expression.

The transcription (decryption) of DNA uses these sequences as markers for process control. But the sequences can have multiple interpretations. ATG within a gene codes for the amino acid methionine; at the start of a gene it is a start codon. All instances of TATA do not signify a promoter. These ambiguities provide DNA with its own version of adding diffusion and confusion, and the analyst must fully understand the rules and mechanisms of transcription. In fact, research in gene expression starts with unambiguously identifying the actual gene sequence that codes for proteins (in eukaryotes this is called the exon region) from intervening sequences that are untranslated regions that do not code for proteins (intron regions) as shown in figure 11 for the human gene *hspB9*, which codes for heat shock protein B9 (Ensembl ENSG00000197723). Referring back to figure 10, transcription from a different start site would yield a different outcome, one that is possibly fatal to the organism. Padding creates introns spread throughout the message (exon).

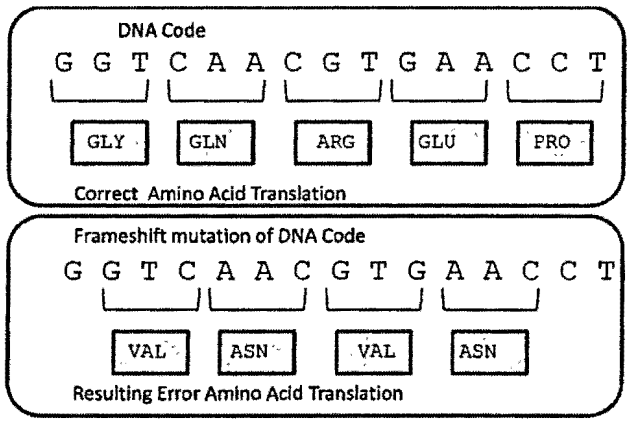


Figure 10. Frameshift Mutations

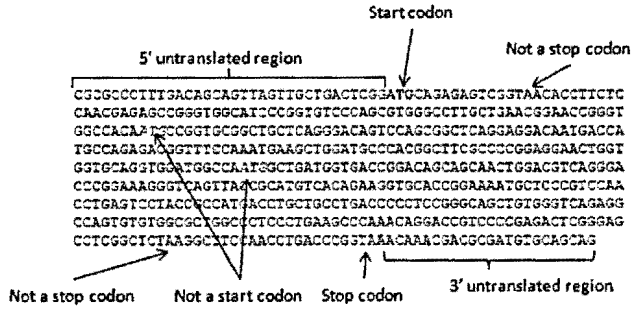


Figure 11. Confusion factors in actual DNA genome

The same confusion and diffusion factors would apply when crafting DNA coded messages for the electronic domain that

will be later instantiated into actual genomes. The ciphertext must be capable of meeting the requirements of the cryptographic hardness in the electronic domain while producing a ciphertext that can be reliably integrated into a cellular genome via standard techniques, transcripted into RNA, and translated into the appropriate cipherprotein. Decryption (expression) of the cipherprotein gene occurs in response to specific decryption instructions hidden within the electronic domain ciphertext .

V. RELATIONSHIP BETWEEN CRYPTOGRAPHY AND GENE EXPRESSION

The following relationships can be observed between the cryptographic treatment of messages and control of gene expression. In the case of gene expression, the message is genomic (DNA or RNA sequence).

- Cryptography transforms messages between two states: plain and encrypted.
- Cryptography uses operations such as circular shifts, bit expansions, bit padding, arithmetic operations to create ciphertext. These operations have analogs in molecular biology, e.g. transposable elements
- Cells transform DNA sequences in genes between two states: Expressed (decrypted) and Silent (encrypted)
- In prokaryotes a simple system involving operators and repressors can be described in terms of encryption and decryption, but prokaryotes have fewer mechanisms available for a rich set of cryptographic protocols. Figure 12 provides an example from *Escherichia coli* using *lacZ* gene expression.

In this prokaryotic example from *E. coli*, the *lacZ* gene expresses the β-galactosidase enzyme when lactose is present and the simple sugar glucose is absent. β-galactosidase metabolizes lactose into glucose and galactose. It would be inefficient to express the enzyme above a trace level if glucose is present. Figure 12 provides a cryptographic analogy to the states of the *lacZ* gene under the various conditions of glucose and lactose present, lactose present, and lactose absent. The *lacZ* gene is encrypted when lactose is absent or both lactose and glucose are present. A repressor protein (rep) authenticates (binds) to the encryption site (*lacZ* operator) on the *lacZ* gene with lactose is absent. A catabolite activator protein (CAP) authenticates (binds) to the decryption site (CAP site) allowing RNA polymerase to decrypt (express) the *lacZ* gene when glucose is absent. All of these operations are shown as analogies to elements of cryptographic message traffic in operations shown in figure 12. It is possible to write the description of the gene expression sequence in figure 12 in terms of a series of messages between a sender and receiver.

Figure 13 shows the architecture of the DNA HMAC (without all the required control regions) described in detail in this paper and its comparison between gene transcriptional control structures for a typical mammalian gene, and a simple, yet important eukaryote, yeast (*S. Cerevisiae*). The DNA HMAC structure preserves the intent of the design to mimic a genomic transcriptional control structure.

A successful, in vivo instantiation of a DNA HMAC system will require specific stop codons, start codons, promoters and enhancers sequences. An in vivo DNA encryption system should be multi-dimensional, utilize primary, secondary and tertiary structural information and include up/downstream regulators such that a single sequence can be seamlessly implemented at the genomic level and have multiple levels of encryption at the message or data level, depending upon the context (only known between sender and receiver). This approach also permits generation of mutant hash codes which can be evaluated for fitness such that only the best hash code is selected for authentication purposes.

A. Epigenetic relationships between cryptography and gene expression.

Epigenetics involves heritable control of gene expression that does not involve modifications of the underlying DNA sequence[15]. Examples of epigenetic effects include: DNA methylation of cytosine residues[16], and control of gene expression via the higher order structures of DNA. In eukaryotes, DNA is packed into a higher order nucleosome structure which is in turn packed into a higher order structure called chromatin[17]. Chromatin states can also be utilized as a form of encryption and decryption by exposing or not exposing genes for transcription. Examples include:

- Heterochromatin form (encrypted) and Euchromatin form (decrypted)
- Post-translational control of chromatin states[18] Histone Code[19]. Histone lysine acetylation by histone acetyl transferase – open chromatin (decrypted); Histone lysine deacetylation by histone deacetylase – closed chromatin (encrypted).

Expansion of the cryptographic protocols to include epigenetic operations will increase the richness of the protocols and the options for producing combinations of cipherproteins.

VI. CONCLUSION

A cryptographic hash code based upon a DNA alphabet and a secure MANET authentication protocol has been presented. These codes can be utilized at the network level or application level and can also be implemented directly into genomes of choice to provide a new level of ciphertext communication at the genomic and proteomic level. The DNA inspired cryptographic coding approach is an option in developing true MANET architectures and developing novel forms of biological authentication to augment those architectures.

ACKNOWLEDGMENT

Thanks to the NASA Space Network, NASA/GSFC Exploration and Space Communications Projects Division,

and the NASA Space Communications and Navigation Program office for supporting this research.

REFERENCES

[1] A. Gehani, T. LaBean and J. Reif, "DNA-based Cryptography, Aspects of Molecular Computing", Springer-Verlag Lecture Notes in Computer Science, pp. 167—188, vol. 2950, 2004.

[2] N.G. Bourbakis, "Image Data Compression-Encryption Using G-Scan Patterns", Systems, Man, and Cybernetics, IEEE International Conference on Computational Cybernetics and Simulation, pp.1117—1120, vol.2, October 1997.

[3] A. Leier, C. Richter, W. Banzhaf, H. Rauhe, "Cryptography with DNA binary strands", BioSystems, vol. 57, pp 13-22, December 1999.

[4] C.T. Clelland, V. Risca, C. Bancroft, "Hiding Messages in DNA microdots", Nature, vol. 399, pp 533—534, June 1999 doi:10.1038/21092 Scientific Correspondence

[5] D. Heider, A. Barnekow, "DNA-based watermarks using the DNA-Crypt algorithm", BMC Bioinformatics, vol. 8: 176, May 2007, doi:10.1186/1471-2105-8-176

[6] *Functional and Comparative Genomics Fact Sheet*, Human GenomeProject, September 19, 2008, Available: http://www.ornl.gov/sci/techresources/Human_Genome/faq/compgen.shtml

[7] B. Alberts, A. Johnson, J. Lewis, L. Raff, K. Roberts, Pl Walter, "*Molecular Biology of the Cell*", 4th edition , New York, NY: Garland Science, 2002

[8] W. Stallings, "*Cryptography and Network Security*", 4th edition, Upper Saddle River, NJ: Pearson Prentice-Hall , 2006

[9] Claire M. Fraser, et.al., "The Minimal Gene Complement of *Mycoplasma genitalium*", Science, Vol. 270, No. 5235, pp. 397-403, Oct. 20, 1995

[10] C. J. Mitchell, "Truncation attacks on MACs", Electronics Letters- IEE, Vol 39, Part 20, pages 1439-1445, IEE,2003

[11] Santiago F Elena, Purificación Carrasco, José-Antonio Daròs, Rafael Sanjuán, "Mechanisms of genetic robustness in RNA viruses",2006, Vol 7:2, pp 168-173, EMBO Report

[12] M.W. Nachman, S.L. Crowell., "Estimate of the mutation rate per nucleotide in humans.", Genetics, Vol. 156, 297-304, September 2000, Genetics Society of America

[13] TBD

[14] C. Shannon, Communication Theory of Secrecy Systems, Bell System Technical Journal, p. 623, July 1948

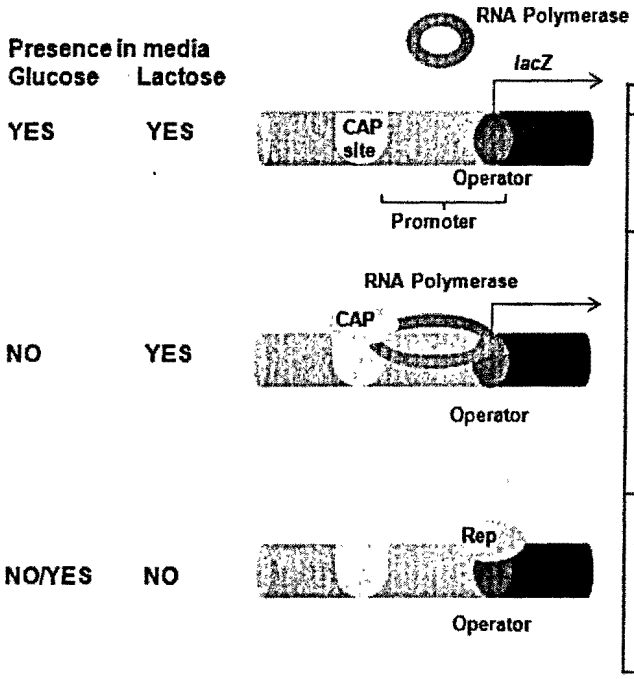
[15] TBD

[16] TBD

[17] TBD

[18] TBD

[19] TBD



State	Operations	Comments
Basal (Encrypted) *lacZ* low level of expression	D(key $_{Lactose}$, *lacZ*) E(key $_{Glucose}$, *lacZ*) E(key $_{CAP}$, *lacZ*)	lacZ gene is encrypted by presence of glucose and lack of CAP at the *lacZ* promoter binding site
Activated (Decrypted) *lacZ* expressed	D(key $_{Lactose}$, *lacZ*) D(key $_{Glucose}$, *lacZ*) D(key$_{CAP}$, *lacZ*) MAC(key$_{CAPsite}$, *lacZ*) MAC($_{keyRNA\ Polymerase\ site}$, *lacZ*)	Absence of glucose and presence of CAP at *lacZ* promoter decrypts the *lacZ* gene. CAP and RNA polymerase MAC authenticate the binding sites so that RNA Polymerase can bind to *lacZ* promoter
Off (Encrypted) *lacZ* repressed by Rep protein	E(key $_{Lactose}$, *lacZ*) MAC (key$_{Rep\ site}$, *lacZ* E($_{keyRep}$, *lacZ*)	Absence of lactose allows Repressor protein to authenticates binding site to *lacZ* operator and encrypts *lacZ* gene by binding to operator site

Figure 12. Conceptual example of Confidentiality and Authentication in *E. coli* using *lacZ* expression

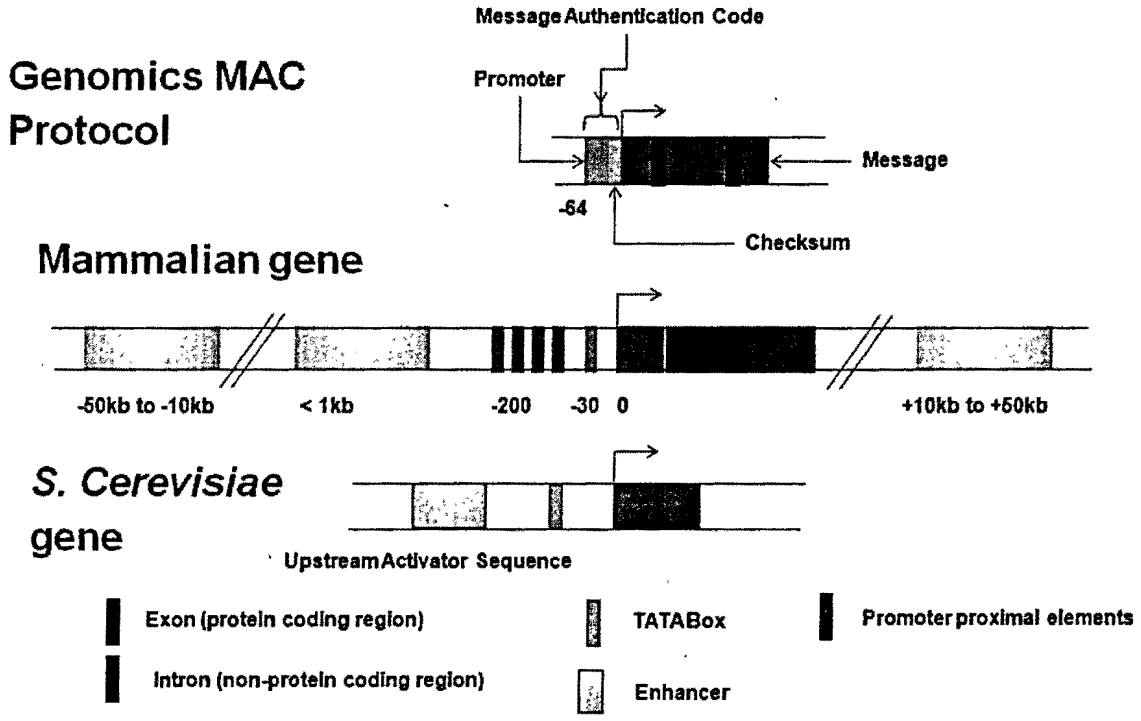


Figure 13. Simplified comparison between gene transcription control regions and MAC protocol