Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100427

Secure Blind Data Hiding into Pseudo DNA Sequences Using Playfair Ciphering and Generic Complementary Substitution

Amal Khalifa[1,2], *Ahmed Elhadad*[3,*] *and Safwat Hamad*[2]

[1] College of Computer and Information Sciences, Princess Nora Bint Abdulrahman University, Riyadh, KSA
[2] Department of Scientific Computing, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt
[3] Department of Mathematics and Computer Science, Faculty of Science, South Valley University, Qena, Egypt

Received: 9 Apr. 2016, Revised: 26 May 2016, Accepted: 27 May 2016
Published online: 1 Jul. 2016

Abstract: Steganography provides unconventional solutions to protect communication as well as copyright of intellectual property. In this paper, we propose a steganographic method that exploits some characteristics of the Deoxyribonucleic Acid (DNA) for hiding other types of digital content. The proposed work is an extension to [1], as it provides a solution to the problem that the sender and the receiver have to secretly communicate both the stego-DNA and the reference sequence. Communicating such information could be suspicious and reveals the secrecy of the steganographic channel itself.
Thus, the proposed hiding method is implemented in two main stages: the first one hides the secret message into some reference DNA sequence using a generic substitution technique. The next phase employs a self embedding algorithm that randomly inserts the stego-DNA sequence into the reference one. In this way, the extraction process can be done blindly and the communicating parties don't actually have to exchange anything in advance but the secret key. Furthermore, a DNA-based playfair ciphering is applied on the secret data before embedding in order to increase the security of the hiding algorithm. When compared with other hiding methods, the proposed method showed an outstanding performance providing high security and embedding capacity.

Keywords: Steganography, DNA, Playfair cipher, blind extraction, palindrome, self-embedding

1 Introduction

The great advances in computers and their applications simplified the creation, manipulation, copying, and modification of digital media. As valuable and secret information are massively communicated through public channels like the Internet, secure and yet creative information protection techniques are in great demand. One way to achieve this is to use cryptography. Cryptography protects information by transforming it into an incomprehensive format (cipher text) which can then be deciphered using some kind of a secret key [2].

Although cryptography protects the data during the transmission stage, this protection cannot be guaranteed after subsequent decryption. On contrast, Steganography techniques hide the information into some innocent looking “cover media”in such a way that the resultant “stego media”is perceptually indistinguishable from the original one. Exactly like a leaf insect when it exploits the natural surroundings of leaves to camouflage itself. Thus, it would not be easy for an enemy to identify the insect or even recognize its existence.

In fact, steganographic techniques are growing rapidly providing unconventional solutions to protect the communication of a wide range of digital media. Digital images attracted a lot of steganographic efforts mainly because of their popularity on the Internet [3]. Other media formats include: audio tracks [4], video streams [5], file systems [6], networks [7] and more interestingly 3D Objects [8]. Thanks to the Human genome project, large amounts of genetic data are now available through public access databases. Thus, Deoxyribonucleic Acid (DNA) sequences are now considered the new candidates added to the list of media for hiding other types of digital content.

Encoding information into DNA sequences can have a variety of applications. These techniques can be used for

[*] Corresponding author e-mail: ahmed.elhadad@sci.svu.edu.eg

copyright protection of genetically engineered organisms, gene therapy, transgenic crops, tissue cloning and DNA computing. It can also be used to facilitate long-term data storage in the genome of a living organism such as Bacteria[9]. DNA-steganography can also be utilized to improve hybrid cryptosystems. An innovative approach was proposed by [10] to achieve a secure communication using a combination of cryptography and DNA-based steganography. According to suggested protocol, symmetric crypto-key is embedded into a DNA stream by one party and communicated to the other one. Once the receiver successfully extracts the key, it can be used to establish a symmetric connection reducing the need for public cryptography.

In this paper, we present an innovative scheme for DNA-based Steganography. As shown in figure 1, the proposed method is divided into two main processes: the embedding process which is carried out by the sender, while the extraction process is carried out by the receiving party. The sender follows a number of steps in order to hide his secret message into some “cover”DNA sequence. One of these steps uses a DNA-based Playfair cipher [11] to transform the secret message into an encrypted DNA sequence. By doing this we actually achieve two goals: first increasing the security of the hiding algorithm, and secondly making advantage of the similarity between the encrypted data and the cover media. The embedding step is carried out using a complementary substitution algorithm. On the other hand, the receiver can extract the hidden message by simply reversing the steps of the embedding process. Notice that, the extraction process can be done “blindly”without the need to the reference sequence. Therefore, the sender and the receiver do not actually have to exchange anything in advance but the secret key.

The rest of the paper is organized as follows: section 2 gives an overview on the literature of DNA-based hiding techniques. Next, section 3 provides a quick glimpse on some preliminary properties of DNA sequences. In section 4, the steps of the proposed algorithm are explained in detail including the ciphering step as well as the embedding and the extraction modules. Later in section 5, we will provide a theoretical analysis of the proposed approach in terms of capacity as well as robustness against attacks. Finally, section 6 introduces the experimental results as well as a comparative study with other existing techniques.

2 Related Work

One of the earliest tries in applying information hiding schemes on biological DNA appears in [12]. The authors synthesized a DNA strand that encrypts the secret message. The message sequence is then copied and camouflaged within a huge number of similarly sized fragments of human DNA. In their proof-of-principle experiment, they succeeded to send the DNA-containing message on a paper in a micro-dot fashion. Later in another publication [13] they proved that the amount of time required to crack DNA-based steganography is long enough to qualify the technique as essentially unbreakable. More watermarking techniques were introduced to protect R&D investments in DNA computing [14][15][16].

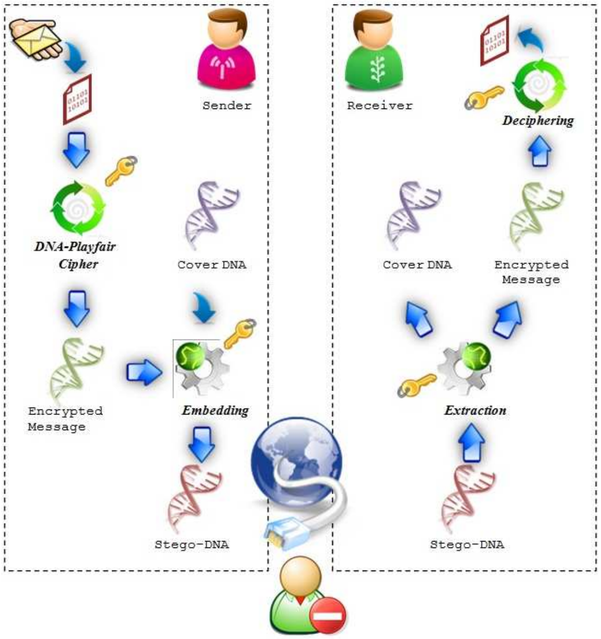


Fig. 1: The proposed DNA-based secure steganographic channel

All of these techniques were mainly developed for steganographically hide information into live molecular DNA. Obviously, the implementation of such methods must support by the availability of different sequencing and hybridization facilities to conduct experiments on biological samples. In addition, these approaches suffer from natural biological errors such as mutation. On the other hand, few methods regarded DNA as an information coding medium that can be stored in a digital form, collected in databases, and easily distributed through discussion groups just like any other kinds of files. In this case, and due to the high randomness of DNA space, it will be really hard to distinguish between a real DNA sequence and a fake or “pseudo”one.

The work presented in [17] addressed “arithmetic encoding”as a way of hiding data into DNA sequences. The idea of the algorithm was based on the feature of codon redundancy. That is, different codons can be mapped to the same amino acids at the translation stage of the central dogma. Thus, it starts by converting the secret binary sequence into a decimal number between 0 and 1. Then in order to hide that number into the cover DNA, arithmetic encoding is employed to parse through the different codon tables. The length of the resultant stego-DNA depends on the precision of the embedded fraction that obviously affects the accuracy of the blind retrieval process.

Three more theoretical methods were introduced in [1]. The proposed techniques select a reference sequence upon which the sender and the receiver agree before the transmission takes place. The sender then embeds the secret message into that sequence producing another DNA sequence. The embedded sequence can then be communicated between the sender and the receiver through public networks. However, the receiver can not recover the secret message without the help of the reference sequence. The problem with this scenario is that communicating such information could be suspicious and reveals the secrecy of the in [1]steganographic channel itself. A modification to the insertion technique presented in [1] was proposed in[18].

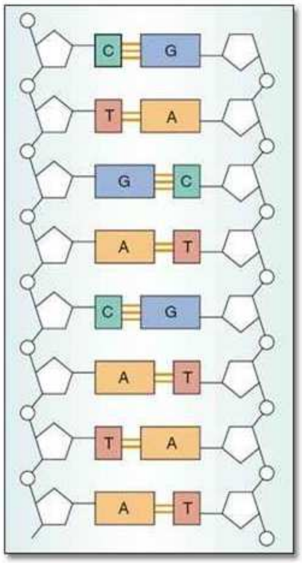


Fig. 2: The complementary base pairing structure of the DNA

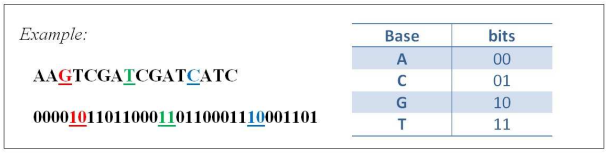
3 Biological Preliminaries

Genes carry the physical and functional traits that are passed on from one generation of biological organism to the next. The genetic information is stored in the form of DNA molecules. According to the *Watson-Crick* model, a DNA molecule consists of two polynucleotide strands coiled around each other in a double helix structure [19]. In effect, each strand of DNA is made up of building blocks called nucleotides or bases. There are only four kinds of bases: adenine (A), guanine (G), thymine (T) and cytosine (C). As shown in figure 2, these bases pair up in a unique complementary way, where A pairs with T and G pairs with C. Hence, a DNA sequence can be represented as a linear set of characters representing nucleotides such as: **AAGTCGATCGATCATCGA**. Furthermore, every three adjacent nucleotides constitute a single unit known as the codon. These codons are "read"and eventually translated into chains of amino acids, which form a protein in a long and complex process called Central Dogma [20].

Looking at DNA as a coding medium makes it convenient to adopt some coding rule to convert this string of bases into binary form and vice versa. One of these rules actually maps each base to a 2-bits number. For example, according to the rule shown in figure 3, the bases A; C; G and T are mapped into 00, 01, 10, and 11 respectively. In addition, figure 4 shows how the complementary of a given DNA sequence can be computed. The Watson-Crick complementary rule states that A pairs only with T and C pairs only with G and vice versa. However, another interesting complementary rule was proposed in [1]. They suggested that each base can be assigned a complement according to any generic complementary rule as the one shown in Figure 3. According to this rule, the complementary base of A, G, C, and T are C, T, A, and G respectively. The most interesting thing about this rule is that for every base (b) in the DNA sequence, the following property holds:

$$C(b) \neq C(C(b)) \neq C(C(C(b))), \quad b \in \{A, G, C, T\}.$$

Where *C(b)* is the complementary of the base *b*.



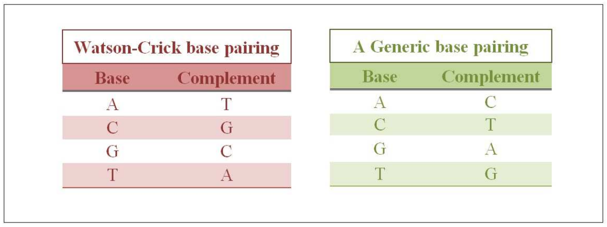
Example:

AAGTCGATCGATCATC

0000101101100011011000111000 1101

Base	bits
A	00
C	01
G	10
T	11

Fig. 3: A digital coding of DNA bases



Watson-Crick base pairing	
Base	Complement
A	T
C	G
G	C
T	A

A Generic base pairing	
Base	Complement
A	C
C	T
G	A
T	G

Fig. 4: Complementary rules for the nucleotide bases of DNA

Of course, we can generically propose different complementary rules that satisfy the above property. In this case, one can define his own rule and utilize it within the hiding technique as will be illustrated shortly in the embedding module.

4 The Hiding Method

In this section, we are going to give a detailed description of the main steps of the proposed steganographic approach as they appear in figure 1. Then, a more formal and structured pseudo code is given and listed in Algorithm 1. Finally, a step-by-step illustration of the hiding process is given using “Rev:25Jan”as the message and “EGYPT”as a key. For the sake of simplicity, the cover is accessed in a sequential fashion. However; in the actual implementation (as shown in **Algorithm 1**), the order by which the cover bases are selected for embedding is determined by a random permutation function that depends on the secret key. This ensures that; even if the embedding algorithm is known, only recipients who know the corresponding secret key will be able to extract the message correctly.

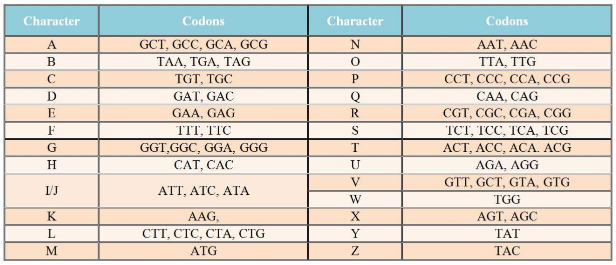
4.1 The DNA Playfair Cipher

This step can be considered as a preprocessing step that can be used to encrypt the secret data before the embedding process takes place. A number of cryptography techniques inspired by DNA have been recently proposed [21] ,[22]. In this paper, we adopt a DNA-based implementation of the very well-known Playfair cipher as proposed in [11]. Unlike the traditional Playfair cipher which limits the plaintext to the form of alphabets, this new implementation allows the ciphering of any kind of binary data such as text, audio, or even images. That is, assuming that the secret message M consists of a binary sequence, it can be transformed into sequences of DNA nucleotides through some DNA digital encoding rule such as the one shown in figure 3.

This coded DNA sequence forms the input to the ciphering algorithm which proceeds as follows. First, a chain of codons is constructed from the DNA sequence according to the standard universal table of amino acids [23]. However, although we have 64 different codons, they only code for 22 unique amino acids, each one of them is given an abbreviation, and a single character symbol, except the START and the STOP codons. Thus, the second step in the ciphering algorithm maps each codon to its corresponding character symbol in order to construct and use the Playfair matrix.

However, in the universal table of amino acids the letters **B, O, U, X,** and **Z** are missing, so the authors in [11] suggested a redistribution of the codons over the complete set of alphabets. As illustrated in table 1, the letter (**B**) is assigned to the 3 stop codons, the letters (**O, U, X**) share two codons from the (**L, R, S**) amino acids respectively, and the letter (**Z**) will take one codon from (**Y**). Finally, the start codon is neglected since it is typically the amino acid (**M**).

Table 1: A Mapping of the 64 codons of DNA onto the 26 English alphabets



Character	Codons	Character	Codons
A	GCT, GCC, GCA, GCG	N	AAT, AAC
B	TAA, TGA, TAG	O	TTA, TTG
C	TGT, TGC	P	CCT, CCC, CCA, CCG
D	GAT, GAC	Q	CAA, CAG
E	GAA, GAG	R	CGT, CGC, CGA, CGG
F	TTT, TTC	S	TCT, TCC, TCA, TCG
G	GGT,GGC, GGA, GGG	T	ACT, ACC, ACA. ACG
H	CAT, CAC	U	AGA, AGG
I/J	ATT, ATC, ATA	V	GTT, GCT, GTA, GTG
		W	TGG
K	AAG,	X	AGT, AGC
L	CTT, CTC, CTA, CTG	Y	TAT
M	ATG	Z	TAC

In this case, the rules of the classic Playfair cipher can be applied on the coded codon character symbols. Eventually, a header is added to resolve the ambiguity caused by the codon redundancy. Now, the ciphered DNA sequence can be converted back into a bit stream that represents the cipher text. However, here we prefer to keep the encrypted data in the DNA format making advantage of the similarity between the encrypted message and the cover media.

4.2 The Embedding Module

In the first step of the hiding process, the encrypted DNA-message is hidden into the cover DNA sequence by means of a novel substitution procedure. Next, the resultant stego-DNA is randomly inserted into the cover DNA to guarantee that the extraction process can be performed blindly. Finally, instead of adding some header information about the size of the embedded message, a very unique palindromic DNA structure is used to signal the end of the hidden message.

4.2.1 The Substitution Phase:

As the name implies, the **G**eneric **C**omplementary **B**ase **S**ubstitution (**GCBS**) phase replaces some bases of the cover sequence with their complementary nucleotides depending on the contents of the message and some generic complementary rule. As shown in figure 1, two DNA sequences are input to the embedding process; the encrypted message $(S_{msg} = m_1;m_2;m_3;...;m_p)$, and the cover sequence $(S = s_1;s_2;s_3;...;s_n)$, to produce the stego-DNA sequence $(S' = s'_1;s'_2;s'_3;...;s'_n)$, where $p = \mid S_{msg} \mid$, $n = \mid$ S $\mid$, and $p < n$.

That is, any selected cover base (s_j) is substituted by its complement depending on the value of secret base

(m_i) to be hidden. More specifically, if m_i is A, then the cover base s_i is left unchanged, otherwise si is replaced by it complement $C(s_j)$,$CC(s_j)$ or $CCC(s_j)$ according to the following rules:

$$Messagebase \left\{ \begin{array}{l} A \rightarrow s_j \\ C \rightarrow C(s_j) \\ G \rightarrow CC(s_j) \\ T \rightarrow CCC(s_j) \end{array} \right\} Stegobase$$

where $C(s_j)$ is computed using the generic complementary rule illustrated previously in figure 4. This substitution mechanism is capable of embedding one message base into another cover base doubling the hiding capacity achieved in [1].

4.2.2 The End-of-Message Signal:

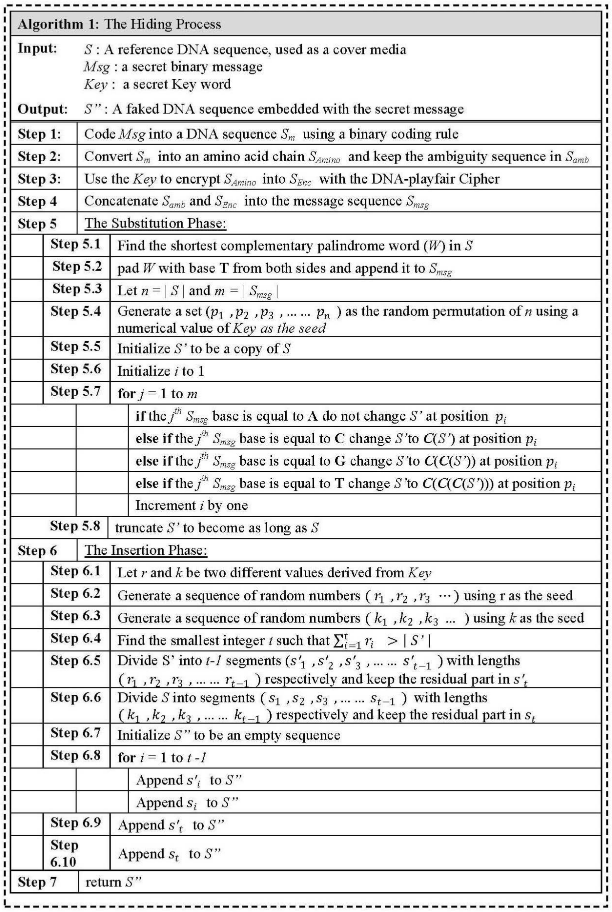
At this point, there is an important question need to be answered: what about the size of the embedded message? One trivial answer would be to use a 16-bit header; for example, to record the size of the message. However, here we suggest another solution that is very specific to the DNA structure of the cover media: *palindromic motifs*. In fact, the meaning of palindrome in the context of genetics is slightly different from the definition used for words and sentences. That is, since the two strands of DNA always pair according to a fixed biological complementary rule, a single-stranded sequence of DNA is said to be a palindrome if it is equal to its complementary sequence read backwards [24]. For example, the sequence **ACCTAGGT** is palindromic because its complement is **TGGATCCA**, which is equal to the original sequence in reverse complement.

Hence, we suggest searching for a specific *palindromic sequence* in the cover DNA and use it as a signal to the end of the embedded message. The proposed scheme suggests using the shortest palindromic sequence (W) and embeds it right after the message to indicate its termination. Furthermore, the selected (W) is padded with the nucleotide base **T** from both sides to become (T W T). This makes sure that the newly inserted palindrome will not interact with the surrounding nucleotides producing another, even longer palindrome. Furthermore, for reasons that will be declared shortly, **S’** must be of the same length as the Cover sequence (**S**). Therefore, the resultant **S’** must be truncated since the length of the resultant sequences has been changed due to adding the palindromic end-of-message signal.

It is important to highlight that, even though we choose the padding nucleotide base to be (T), it can be replaced by any other nucleotide instead. In addition, the shortest palindrome can be replaced by any other particular one, such as the longest one or it can be randomly chosen. In fact, this decision is based on the fact that in some real DNA sequences, palindromes can be more that 100 bases long. Therefore, we choose to minimize this space in order to be able to hide as much information as possible.

4.2.3 The Insertion phase:

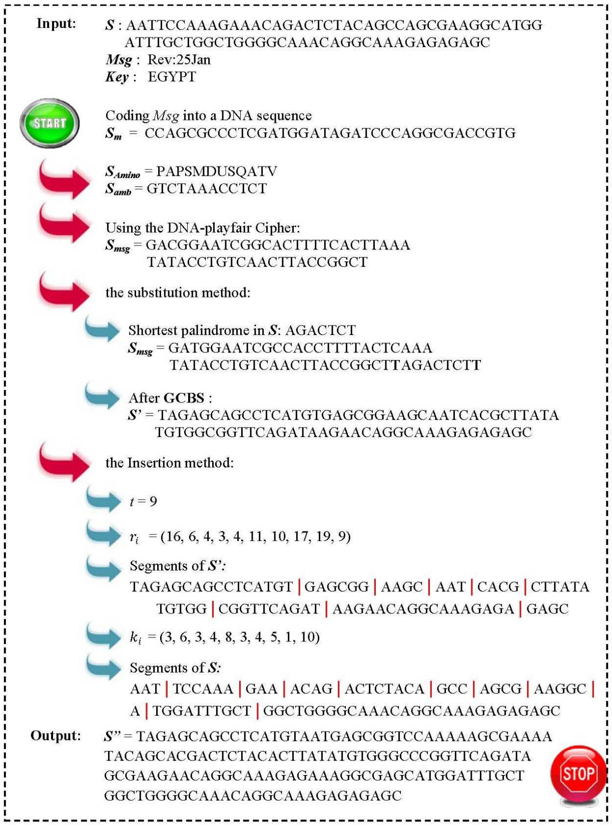
Now, it is clear that the recovery of the originally embedded message bases (m_i) can’t be done without a reference to the original cover sequence. Thus, we suggest a self embedding strategy in which both sequences the (S'); resulted from the substitution phase, and the cover (S) are packaged into one sequence (S'') using an insertion method similar to the one introduced in [1]. That is, the cover and the message sequences are chopped into segments of random lengths that are eventually concatenated producing a sequence (S'') whose length equals to $2* \mid S' \mid$. In this case, it will be possible to inverse this process to separate (S) from (S'') in order to blindly carry out the extraction process as will be illustrated shortly.



```
Algorithm 1: The Hiding Process
Input:    $S$ : A reference DNA sequence, used as a cover media
          $Msg$ : a secret binary message
          $Key$ :  a secret Key word
Output:   $S''$ : A faked DNA sequence embedded with the secret message
Step 1:   Code $Msg$ into a DNA sequence $S_m$ using a binary coding rule
Step 2:   Convert $S_m$ into an amino acid chain $S_{Amino}$ and keep the ambiguity sequence in $S_{amb}$
Step 3:   Use the $Key$ to encrypt $S_{Amino}$ into $S_{Enc}$ with the DNA-playfair Cipher
Step 4    Concatenate $S_{amb}$ and $S_{Enc}$ into the message sequence $S_{msg}$
Step 5    The Substitution Phase:
    Step 5.1  Find the shortest complementary palindrome word ($W$) in $S$
    Step 5.2  pad $W$ with base T from both sides and append it to $S_{msg}$
    Step 5.3  Let $n = | S |$ and $m = | S_{msg} |$
    Step 5.4  Generate a set ($p_1, p_2, p_3, \dots\dots p_n$) as the random permutation of $n$ using a
              numerical value of $Key$ as the seed
    Step 5.5  Initialize $S'$ to be a copy of $S$
    Step 5.6  Initialize $i$ to 1
    Step 5.7  for $j = 1$ to $m$
                  if the $j^{th}$ $S_{msg}$ base is equal to A do not change $S'$ at position $p_i$
                  else if the $j^{th}$ $S_{msg}$ base is equal to C change $S'$to $C(S')$ at position $p_i$
                  else if the $j^{th}$ $S_{msg}$ base is equal to G change $S'$to $C(C(S'))$ at position $p_i$
                  else if the $j^{th}$ $S_{msg}$ base is equal to T change $S'$to $C(C(C(S')))$ at position $p_i$
                  Increment $i$ by one
    Step 5.8  truncate $S'$ to become as long as $S$
Step 6    The Insertion Phase:
    Step 6.1  Let $r$ and $k$ be two different values derived from $Key$
    Step 6.2  Generate a sequence of random numbers ($r_1, r_2, r_3 \cdots$) using r as the seed
    Step 6.3  Generate a sequence of random numbers ($k_1, k_2, k_3 \dots$) using $k$ as the seed
    Step 6.4  Find the smallest integer $t$ such that $\sum_{i=1}^{t} r_i > | S' |$
    Step 6.5  Divide S’ into $t$-$1$ segments ($s'_1, s'_2, s'_3, \dots\dots s'_{t-1}$) with lengths
              ($r_1, r_2, r_3, \dots\dots r_{t-1}$) respectively and keep the residual part in $s'_t$
    Step 6.6  Divide $S$ into segments ($s_1, s_2, s_3, \dots\dots s_{t-1}$) with lengths
              ($k_1, k_2, k_3, \dots\dots k_{t-1}$) respectively and keep the residual part in $s_t$
    Step 6.7  Initialize $S''$ to be an empty sequence
    Step 6.8  for $i = 1$ to $t$ -$1$
                  Append $s'_i$ to $S''$
                  Append $s_i$ to $S''$
    Step 6.9  Append $s'_t$ to $S''$
    Step 6.10 Append $s_t$ to $S''$
Step 7    return $S''$
```

4.3 The Extraction module

The extraction process is the inverse of the embedding process. Therefore, it starts with the inverse of the



Input: S : AATTCCAAAGAAACAGACTCTACAGCCAGCGAAGGCATGG ATTTGCTGGCTGGGGCAAACAGGCAAAGAGAGAGC
Msg : Rev:25Jan
Key : EGYPT

Coding *Msg* into a DNA sequence
S_m = CCAGCGCCCTCGATGGATAGATCCCAGGCGACCGTG

S_{Amino} = PAPSMDUSQATV
S_{amb} = GTCTAAACCTCT

Using the DNA-playfair Cipher:
S_{msg} = GACGGAATCGGCACTTTTCACTTAAA TATACCTGTCAACTTACCGGCT

the substitution method:

Shortest palindrome in S: AGACTCT
S_{msg} = GATGGAATCGCCACCTTTTACTCAAA TATACCTGTCAACTTACCGGCTTAGACTCTT

After **GCBS** :
S' = TAGAGCAGCCTCATGTGAGCGGAAGCAATCACGCTTATA TGTGGCGGTTCAGATAAGAACAGGCAAAGAGAGAGC

the Insertion method:

$t = 9$

r_i = (16, 6, 4, 3, 4, 11, 10, 17, 19, 9)

Segments of S':
TAGAGCAGCCTCATGT | GAGCGG | AAGC | AAT | CACG | CTTATA TGTGG | CGGTTCAGAT | AAGAACAGGCAAAGAGA | GAGC

k_i = (3, 6, 3, 4, 8, 3, 4, 5, 1, 10)

Segments of S:
AAT | TCCAAA | GAA | ACAG | ACTCTACA | GCC | AGCG | AAGGC | A | TGGATTTGCT | GGCTGGGGCAAACAGGCAAAGAGAGAGC

Output: S'' = TAGAGCAGCCTCATGTAATGAGCGGTCCAAAAAGCGAAAA TACAGCACGACTCTACACTTATATGTGGGCCCGGTTCAGATA GCGAAGAACAGGCAAAGAGAAAGGCGAGCATGGATTTGCT GGCTGGGGCAAACAGGCAAAGAGAGAGC

Fig. 5: A detailed example illustrating the steps of the proposed hiding process

insertion process in order to separate the original cover-DNA from the embedded stego-DNA. Then it proceeds to the next phase where the inverse of the substitution process is carried out to recover the encrypted message in DNA format. Finally, a decryption step is needed to reveal the original binary message content. The details of the whole extraction process are also demonstrated using the same example as shown in figure 6.

4.3.1 The Reference Recovery Phase

In this step, the insertion process is reversed to separate the original cover-DNA (S) from the embedded stego-DNA (S') and hence we will be able to compare them in the coming step. As shown in Algorithm 2, $(S?)$ is divided into augmented segments of (S') and (S) whose respective lengths are regenerated using the same seed values adopted during the hiding process. Eventually, the extracted segments are concatenated back into two sequences of the same length.

4.3.2 The Message Recovery Phase

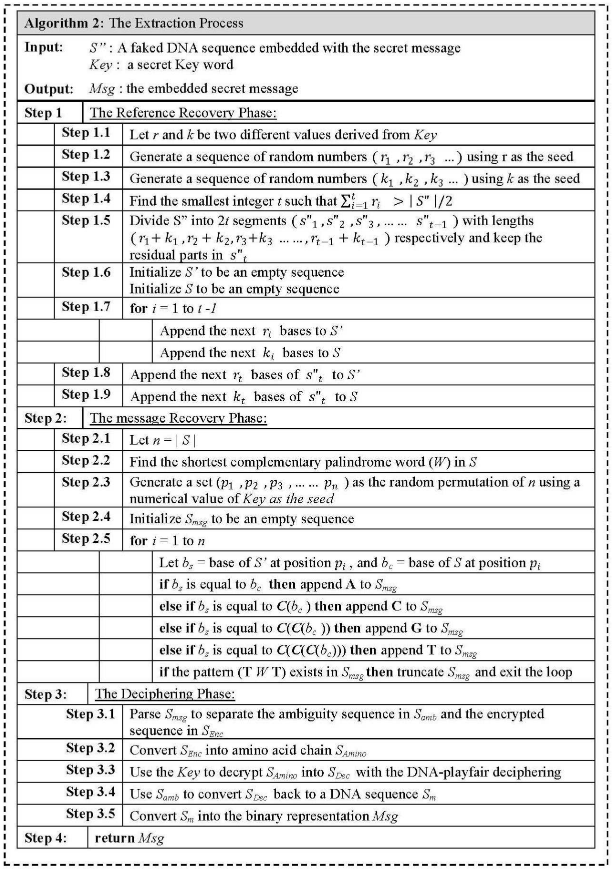
The cover-DNA (S) is now available in separation from stego-DNA (S'). However, not all the bases of S are actually carrying secret information. So, we need to search first for the end-of-message signal that was inserted previously during the hiding process. As shown in Algorithm 2, this is done by searching for the shortest palindrome word (W) in (S) followed by detecting the structure (TWT) in (S') in order to identify the end of the embedded sequence.

The inverse of the *GCBS* method can then be carried out by comparing each stego-base (s'_j) with its corresponding cover base (s_j) to reveal the value of the hidden message base (m_i) and append it to the (S_{msg}) sequence. For example, if s'_j is equal to s_i , then m_i must be **A**. Otherwise, if s'_j is equal to the complement of s_i , then m_i can be extracted according to the following rules:

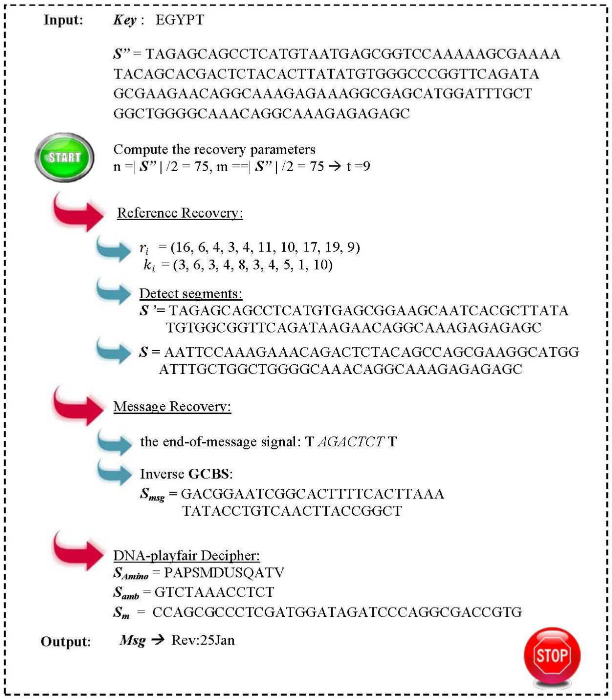
$$Stegobase \left\{ \begin{array}{l} s_j \rightarrow A \\ C(s_j) \rightarrow C \\ CC(s_j) \rightarrow G \\ CCC(s_j) \rightarrow T \end{array} \right\} Messagebase$$

4.3.3 The playfair Decryption Phase

Finally, the extracted DNA sequence (S_{msg}) should be decrypted using the play-fair deciphering module. As shown in Algorithm 2, the deciphering process starts by separating the ambiguity sequence (S_{amb}) from the encrypted sequence (S_{enc}). After that, (S_{enc}) is divided into 3-bases long codons and converted to the chain of amino acids (S_{amino}) using table 1. Next, the playfair matrix is constructed using the same key to carry on the decryption process. The resultant deciphered chain (S_{dec}) can then be converted to its corresponding DNA codons (S_m) with the help of (S_{amb}) [11]. At the end, S_m in converted into binary representation to recover the content of the original secret message.



```
Algorithm 2: The Extraction Process
Input:   $S''$ : A faked DNA sequence embedded with the secret message
         $Key$ : a secret Key word
Output:  $Msg$ : the embedded secret message
Step 1   The Reference Recovery Phase:
    Step 1.1  Let $r$ and $k$ be two different values derived from $Key$
    Step 1.2  Generate a sequence of random numbers ( $r_1$ , $r_2$ , $r_3$ ... ) using r as the seed
    Step 1.3  Generate a sequence of random numbers ( $k_1$ , $k_2$ , $k_3$ ... ) using $k$ as the seed
    Step 1.4  Find the smallest integer $t$ such that $\sum_{i=1}^{t} r_i > | S'' |/2$
    Step 1.5  Divide S'' into 2$t$ segments ( $s''_1$ , $s''_2$ , $s''_3$ , ... ... $s''_{t-1}$ ) with lengths
              ( $r_1 + k_1$ , $r_2 + k_2$, $r_3 + k_3$ ... ..., $r_{t-1} + k_{t-1}$ ) respectively and keep the
              residual parts in $s''_t$
    Step 1.6  Initialize $S'$ to be an empty sequence
              Initialize $S$ to be an empty sequence
    Step 1.7  for $i$ = 1 to $t$ -1
                  Append the next $r_i$ bases to $S'$
                  Append the next $k_i$ bases to $S$
    Step 1.8  Append the next $r_t$ bases of $s''_t$ to $S'$
    Step 1.9  Append the next $k_t$ bases of $s''_t$ to $S$
Step 2:  The message Recovery Phase:
    Step 2.1  Let $n = | S |$
    Step 2.2  Find the shortest complementary palindrome word ($W$) in $S$
    Step 2.3  Generate a set ($p_1$ , $p_2$ , $p_3$ , ... ... $p_n$ ) as the random permutation of $n$ using a
              numerical value of $Key$ as the seed
    Step 2.4  Initialize $S_{msg}$ to be an empty sequence
    Step 2.5  for $i$ = 1 to $n$
                  Let $b_s$ = base of $S'$ at position $p_i$ , and $b_c$ = base of $S$ at position $p_i$
                  if $b_s$ is equal to $b_c$ then append A to $S_{msg}$
                  else if $b_s$ is equal to $C(b_c)$ then append C to $S_{msg}$
                  else if $b_s$ is equal to $C(C(b_c))$ then append G to $S_{msg}$
                  else if $b_s$ is equal to $C(C(C(b_c)))$ then append T to $S_{msg}$
                  if the pattern (T $W$ T) exists in $S_{msg}$ then truncate $S_{msg}$ and exit the loop
Step 3:  The Deciphering Phase:
    Step 3.1  Parse $S_{msg}$ to separate the ambiguity sequence in $S_{amb}$ and the encrypted
              sequence in $S_{Enc}$
    Step 3.2  Convert $S_{Enc}$ into amino acid chain $S_{Amino}$
    Step 3.3  Use the $Key$ to decrypt $S_{Amino}$ into $S_{Dec}$ with the DNA-playfair deciphering
    Step 3.4  Use $S_{amb}$ to convert $S_{Dec}$ back to a DNA sequence $S_m$
    Step 3.5  Convert $S_m$ into the binary representation $Msg$
Step 4:  return $Msg$
```



Input: ***Key*** : EGYPT

S'' = TAGAGCAGCCTCATGTAATGAGCGGTCCAAAAAGCGAAAA
TACAGCACGACTCTACACTTATATGTGGGCCCGGTTCAGATA
GCGAAGAACAGGCAAAGAGAAAGGCGAGCATGGATTTGCT
GGCTGGGGCAAACAGGCAAAGAGAGAGC

Compute the recovery parameters
n =| ***S''*** | /2 = 75, m ==| ***S''*** | /2 = 75 → t =9

Reference Recovery:

r_i = (16, 6, 4, 3, 4, 11, 10, 17, 19, 9)
k_i = (3, 6, 3, 4, 8, 3, 4, 5, 1, 10)

Detect segments:
S'= TAGAGCAGCCTCATGTGAGCGGAAGCAATCACGCTTATA
TGTGGCGGTTCAGATAAGAACAGGCAAAGAGAGAGC

S = AATTCCAAAGAAACAGACTCTACAGCCAGCGAAGGCATGG
ATTTGCTGGCTGGGGCAAACAGGCAAAGAGAGAGC

Message Recovery:

the end-of-message signal: **T** *AGACTCT* **T**

Inverse **GCBS**:
S_{msg} = GACGGAATCGGCACTTTTCACTTAAA
TATACCTGTCAACTTACCGGCT

DNA-playfair Decipher:
S_{Amino} = PAPSMDUSQATV
S_{amb} = GTCTAAACCTCT
S_m = CCAGCGCCCTCGATGGATAGATCCCAGGCGACCGTG

Output: ***Msg*** → Rev:25Jan

Fig. 6: detailed example showing the steps of the proposed extraction process

5 Performance Analysis

5.1 Hiding Capacity:

The payload provided by a steganographic technique represents the maximum hiding capacity offered by this algorithm. In other words, it measures the maximum size of bits that can be embedded into the cover media. Here, in the case of DNA media, the hiding capacity is measured in bit-per-nucleotide (bpn).

Once more, assuming that the length of the cover DNA (S) equals to $| S |$; which reflects the number of nucleotides composing its sequence, the proposed (GCBS) algorithm can hide one message base per cover base. In other words, any given DNA sequence can hide a secret sequence that is as long as itself. However, according to the Playfair cipher explained above, only $\frac{3}{4}$ of these bases represent the actual message bits, since the remaining $\frac{1}{4}$ should be reserved for the ambiguity bases. While each nucleotide base actually represents two bits of the binary message (**M**), the overall hiding payload of the algorithm can be expressed as:

$$Capacity = \frac{sizeofmessageinbits}{sizeofcoverinbases} = \frac{\frac{3}{4} \times |S| \times 2}{|S|} = \frac{3}{2} bpn$$

5.2 Robustness

For an attacker to discover the secret message, the following information must be known:

1. The random number generator and the two seeds used in the insertion phase.
2. The complementary rule.
3. The Binary Coding Scheme.
4. The Playfair ciphering technique.

Regarding the first point, the attacker will be faced with the problem of finding the sequence of numbers generated by random seeds r and k denoted as $r_1; r_2; ...; r_p$ and $k_1; k_2; ...; k_p$ respectively, which are used to separate the secret sequence (S') and the cover sequence (S) in the insertion phase. An attacker may have to try all the possible combinations. The authors of [1] showed that the total number of guesses needed to achieve that can reach:

$$\binom{m}{m-1} + \binom{m}{m-2} + \binom{m}{m-3} + \ldots + \binom{m}{0} = \sum_{i=0}^{m-1} \binom{m}{m-1-i} = 2^m - 1$$

Where m represents the length of the message and $\binom{m}{i}$ is the set of all i-combinations of m. Since the summations of r_i's and k_i's are equal to $\mid S \mid$, the probability of an attacker making a successful guess at this stage is $\frac{1}{(2^{|S|}-1)^2}$.

For an attacker to guess the complementary rule used, he/she has to check all of the possibilities. However, there are only six legal complementary rules that actually maps each nucleotide x to a complement $C(x)$ such that $C(x) \neq CC(x) \neq CCC(x)$. Therefore, the probability of making a correct guess at (2) is $\frac{1}{6}$. Similarly, since there are only 4 nucleotides, the the probability of an attacker making a successful guess at coding rules is $\frac{1}{24}$ since the number of possible coding is $4! = 24$. Thus, the probability of an attacker making a successful guess to crack only the hiding phase of the proposed method can be formulated as follows:

$$P_{bf} = \frac{1}{(2^{|S|}-1)^2} \times \frac{1}{6} \times \frac{1}{24}$$

With $\mid S \mid$ can exceed hundreds of thousands, it is almost impossible for an attacker to extract the hidden message. Therefore, no matter how easy or difficult it is to conduct a frequency analysis on the ciphertext in order to break the Playfair cipher, the security of the proposed scheme actually relies on the secrecy of the information hiding itself to avoid any suspension that may lead the attacker to further analysis.

6 Results and comparisons

The purpose of this set of experiments is to evaluate the performance of the proposed method. As shown in Table 2, twelve DNA sequences were used as a test sample. Each sequence is identified by its accession number as drawn from the database of the Genebank. In addition, a 30k bytes of randomly selected textual data is used as the secret message. In each case, the shortest palindrome used as the end of message signal is shown. Furthermore, the maximum capacity offered by the cover sequence well as the actual payload occupied by the message are shown in the table.

In the following set of experiments, the proposed method was compared with other DNA-based hiding schemes. This comparison spotted a number of differences between these methods with respect to two data hiding parameters: capacity and blindness. With blindness we mean that the hidden message can be retrieved without the need to the cover sequence used originally in the embedding stage at the sender side. As shown in table 3, both the proposed method and the methods suggested by [25] allow blind extraction of the hidden data. However, the proposed approach outperforms all the other techniques in terms of capacity. In fact, it offers nearly double the hiding capacity of the substitution method suggested by [1]. Furthermore, it outperforms the method proposed in [18] in terms of both capacity and blind extraction.

Table 2: Results of hiding **30K bytes** of text messages into different DNA sequences

Sequence	*Length (bp)*	*Shortest Palindrome*	*Max Capacity(Kb)*	*Actual Payload(%)*
AC153526	200,117	TATATA	36.64	81.87
AC167221	204,841	TAATTA	37.51	79.98
AAEX02030934	255,827	GGATCC	46.84	64.04
AAEX02030944	281,970	AAGCTT	51.63	58.11
AAEX02030967	220,557	TGTACA	40.39	74.28
AAEX02030982	237,468	CCTAGG	43.48	68.99
AAEX02030999	229,935	GGCGCC	42.10	71.25
ADDN01000038	221,439	CAGCTG	40.54	73.99
ADDN01000040	202,059	ACTAGT	36.99	81.09
ADDN01000058	225,376	CGGCCG	41.27	72.7
ADDN01000102	214,545	GCCGGC	39.28	76.36
ADDN01000119	223,765	GTCGAC	40.97	73.22

Furthermore, table 4 gives an overview of the robustness of the proposed method against brute force attacks in comparison with the three hiding techniques suggested in [1]. It actually combines the advantages of both the substitution as well as the insertion methods.

Table 3: A comparison between the proposed hiding approach and similar methods

Provider	*Approach*	*Capacity (bpn)*	*Blind?*
Chang [25]	Lossless compression-based	0.78	√
	Difference expansion-based	0.11	√
Shiu [1]	Insertion method	0.58	×
	Complementary method	0.07	×
	Substitution method	0.82	×
Atito [18]	Playfair-Insertion method	0.14	×
The authors	Generic Complementary Base Substitution	1.5	√

Table 4: Robustness comparison between the proposed hiding approach and the methods proposed in [1]

Approach	P_{bf}
Insertion method	$\frac{1}{1.63\times10^8} \times \frac{1}{n-1} \times \frac{1}{2^m-1} \times \frac{1}{2^{S-1}} \times \frac{1}{24}$
Complementary method	$\frac{1}{1.63\times10^8} \times \frac{1}{24^2}$
Substitution method	$\frac{1}{(2^{\|S\|}-1)^2} \times \frac{1}{6}$
Generic Complementary Base Substitution**GCBS**	$\frac{1}{(2^{\|S\|}-1)^2} \times \frac{1}{6} \times \frac{1}{24}$

7 Conclusions

In this paper, a novel steganographic method was proposed. It exploits some basic properties of the Deoxyribonucleic Acid (DNA) and utilizes them as an encoding medium to secretly embed any kind of digital data. The proposed hiding method is implemented in two main levels: In level one, the secret message is hidden by a substitution method into some reference DNA sequence. In level two; an insertion technique is used to embed the modified DNA sequence into the original reference sequence. In this way, only the resultant stego-DNA needs to be sent through secure communication to the receiver. In this case, the secret message can be blindly retrieved without the need to separately communicate the reference sequence. Furthermore, in order to increase the security level of the proposed method, the secret message is encrypted using a DNA-based Playfair cipher before the hiding process actually starts.

The experimental results; using various reference DNA sequences, showed that the average processing time of the proposed scheme can be estimated to be 5.5 milliseconds per cover nucleotide. In addition, when compared with other similar hiding methods, the proposed technique provided an outstanding performance not only with respect to blindness, but also in capacity where it offers nearly double the hiding capacity of the best among them. Finally, the robustness of the suggested hiding method was investigated showing that it is almost impossible for an attacker to guess the hiding parameters in order to correctly extract the secret message.

References

[1] H. J. Shiu, K. L. Ng, J. F. Fang, R. C. T. Lee, and C. H. Huang, ”Data hiding methods based upon DNA sequences,” Inf. Sci., vol. 180, pp. 2196-2208, 2010.

[2] N. F. Johnson and S. Jajodia, ”Exploring steganography: Seeing the unseen,” Computer, vol. 31, pp. 26-34, 1998.

[3] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, ”Digital image steganography: Survey and analysis of current methods,” Signal Processing, vol. 90, pp. 727-752, 2010.

[4] D. Yan, R. Wang, X. Yu, and J. Zhu, ”Steganography for MP3 audio by exploiting the rule of window switching,” Computers & Security, vol. 31, pp. 704-716, 2012.

[5] K. Dasgupta, J. K. Mondal, and P. Dutta, ”Optimized Video Steganography Using Genetic Algorithm (GA),” Procedia Technology, vol. 10, pp. 131-137, 2013.

[6] R. Anderson, R. Needham, and A. Shamir, ”The Steganographic File System,” in Information Hiding. vol. 1525, ed: Springer Berlin Heidelberg, 1998, pp. 73-82.

[7] J. M. Steven and L. Stephen, ”Embedding covert channels into TCP/IP,” presented at the Proceedings of the 7th international conference on Information Hiding, Barcelona, Spain, 2005.

[8] P. Amat, W. Puech, S. Druon, and J. P. Pedeboy, ”Lossless 3D steganography based on MST and connectivity modification,” Signal Processing: Image Communication, vol. 25, pp. 400-412, 2010.

[9] J. Shu-Hong and G. Robert, ”Hiding data in DNA of living organisms,” Natural Science, vol. 01, pp. 181-184, 2009.

[10] M. R. N. Torkaman, N. S. Kazazi, and A. Rouddini, ”Innovative Approach to Improve Hybrid Cryptography by Using DNA Steganography,” International Journal of New Computer Architectures and their Applications (IJNCAA), vol. 1, pp. 224-235.

[11] S. Mona, H. Mohamed, N. Taymoor, and K. Mohamed Essam, ”A DNA and Amino Acids-Based Implementation of Playfair Cipher,” International Journal of Computer Science and Information Security, vol. 8, pp. 126-133, 2010.

[12] C. T. Clelland, V. Risca, and C. Bancroft, ”Hiding messages in DNA microdots,” Nature, vol. 399, pp. 533-534, 1999.

[13] I. R. Viviana, ”DNA-based steganography,” Cryptologia, vol. 25, pp. 37-49, 2001.

[14] M. Arita and Y. Ohashi, ”Secret Signatures Inside Genomic DNA,” Biotechnology Progress, vol. 20, pp. 1605-1607, 2004.

[15] D. Heider and A. Barnekow, ”DNA-based watermarks using the DNA-Crypt algorithm,” BMC Bioinformatics C7 - 176, vol. 8, pp. 1-10, 2007.

[16] D. Heider, M. Pyka, and A. Barnekow, ”DNA watermarks in non-coding regulatory sequences,” BMC Research Notes, vol. 2, p. 125, 2009.

[17] F. P. Petitcolas, B. Shimanovsky, J. Feng, and M. Potkonjak, ”Hiding Data in DNA,” in Information Hiding. vol. 2578, ed: Springer Berlin Heidelberg, 2003, pp. 373-386.

[18] A. Atito, A. Khalifa, and S. Rida, ”DNA-based data encryption and hiding using playfair and insertion techniques,” Journal of Communications and Computer Engineering, vol. 2, p. 44, 2012.

[19] A. Ghosh and M. Bansal, ”A glossary of DNA structures from A to Z,” Acta crystallographica. Section D, Biological crystallography, vol. 59, pp. 620-626, 2003.

[20] F. Crick, ”Central Dogma of Molecular Biology,” Nature, vol. 227, pp. 561-563, 1970.

[21] ”A Survey on different DNA Cryptographic Methods ” vol. 2, April 2013.

[22] Q. Gao, ”BioCryptography,” vol. 5, pp. 306-325, 2010.

[23] GEN. (2013, 11 March). genetic code. Available: http://www.britannica.com/EBchecked/topic/228838/genetic-code

[24] M. Giel-Pietraszuk, M. Hoffmann, S. Dolecka, J. Rychlewski, and J. Barciszewski, ”Palindromes in Proteins,” Journal of Protein Chemistry, vol. 22, pp. 109-113, 2003.

[25] C.-C. Chang, T.-C. Lu, Y.-F. Chang, and C.-T. Lee, ”Reversible data hiding schemes for deoxyribonucleic acid (DNA) medium,” International Journal of Innovative Computing, Information and Control (IJICIC), vol. 3, pp. 1145-1160, 2007.

Amal Khalifa Currently works as an assistant professor of Computer Science at College of Computer and Information Sciences, Princess Nora Bint Abdulrahman University, Riyadh KSA since 2013. She was working as an assistant professor of Scientific Computing at Faculty of Computer & Information Sciences, Ain Shams University, Egypt since 2009 (now on leave). She got her M.Sc. degree in the field of Information Hiding in Digital Images. In 2005 she was granted a 2 years research scholarship in University of Connecticut, USA. She earned her PhD degree in 2009 in the area of High performance Computing. Her main research interests are Steganography, computational biology, parallel computing, encryption and Security.

Ahmed Elhadad is working as an assistant professor at Computer Science department in the Faculty of Science - South Valley University, Egypt. He got his B.Sc, MSc and PhD degrees from the same department in 2007, 2010 and 2015 respectively. His research in MSc focused on the field of encryption and DNA Cryptography. In 2012, He was granted mobility scholarship to Instituto Superior Técnico (IST), Lisbon, Portugal. Now, his research focused on the area of Information Security.

Safwat Hamad currently works as an assistant professor of Scientific Computing at Faculty of Computer & Information Sciences, Ain Shams University, Egypt since 2009. He graduated in 2000 and worked as a teaching assistant for a number of undergraduate courses till 2004. He got his MSc degree in the field of Modelling Simulation and Visualization. He earned his PhD degree in 2008 in the area of High performance Computing. The PhD was under a joint supervision between Computer Science and Engineering Department at University of Connecticut, USA and the Faculty of Computer and Information Sciences at Ain Shams University, Egypt. His main research interests are Computer Graphics and Visualization, Image and Video Processing, High Performance Computing and Cryptography.