On Covert Data Communication Channels Employing DNA Recombinant and Mutagenesis-based Steganographic Techniques

MAGDY SAEB 1, EMAN EL-ABD 2, MOHAMED E. EL-ZANATY 1
1. School of Engineering, Computer Department, Arab Academy for Science, Technology and Maritime Transport Alexandria, EGYPT
2. Molecular Biology Group, Medical Technology Center, Medical Research Institute, Alexandria University, EGYPT

Abstract:- The well-known formulation of the Prisoner’s Problem was an inspiration to the academic community to pay more attention to a large number of issues that had arisen in data hiding communication security. Most of the techniques that were implemented were based on modulating a media file to hide the information required to be transmitted. If the transmission is rather continuous in time, one may refer to the technique as a covert communication channel between two parties. Various methods of advanced steganalysis techniques, with varying degrees of success, were developed to detect the existence of such hidden messages in a media file. In this work, we present an alternative approach to hiding data in media files. To be more precise, we use two different DNA-based steganographic techniques where the message is embedded within a vector sequence using recombinant DNA technology or created using DNA mutagenesis technique. The proposed method has many potential unconventional applications not only in data hiding but also in massive data storage.

Key words:- DNA, Steganography, Data Communication security, Recombinant, Mutagenesis.

1 Introduction
Clandestine communication between two parties is always meant to be covert and inconspicuous. This can be achieved using cryptography wherein, information is encrypted or, in other words, scrambled so that, it cannot be comprehended by the intruder. However, encrypted messages are more noticeable than unencrypted ones. Steganographic, or data hiding techniques, would be the more appropriate solution. In contrast to cryptography which scrambles the message so it can not be understood, steganography hides the message so it can't be observed. Encryption can be combined to most of steganographic methods to add another key of security in case the existence of a hidden message is detected.
Different methods of steganographic techniques were used from ancient times to hide secret messages including: invisible inks, microdots, character shifting arrangements, digital signatures, and covert channels and spread spectrum communications. Due to the enormous use of computers and digital media, other approaches such as least significant bit insertion, masking and filtering algorithms, transformation, and software packaging were all introduced relatively recent. More recently, DNA-based steganographic techniques were used. These techniques depend on the high randomness of the DNA to hide any message without being noticed [1]. DNA has many characteristics which make it a perfect steganographic media. These characteristics have two significant facts; the DNA has tremendous information storage capacity. For example, only 1 gram of DNA contains as much information as 1 trillion CD’s [8]. In addition, any DNA sequence can be synthesized in any desirable length.
In the following section, we will summarize the methodology that had been adopted. Section 3 formalizes the proposed procedure in an algorithmic pseudo code; Details are shown in [Appendix D]. Section 4 provides the details of implementation, and finally section 5 gives a summary and conclusions. Appendices A, B, and C provides the details regarding the recombinant, mutagenesis and the dictionary utilized in this work respectively.
2 Methodology
In this work we propose two methods of hiding message into DNA:

- Using DNA Recombinant:
  In this method we ligate the message strand with the DNA strand after opening it in a specific position with the restriction enzymes, and then we use the ligase enzymes to recombinant the two strands into one strand.

- Using DNA Mutagenesis:
  In this method we use the mutation of the DNA to change or replace some nucleotides in a specific position with the message nucleotides. To accomplish this, we assume the following two assumptions; we use a limited vocabulary for the 1024 most commonly used words in the English language [5]. In addition, we use five DNA bases for each word to cover all the words without any duplication. A sample of this encoding is shown in table 1, shown below.

<table>
<thead>
<tr>
<th>Word</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello</td>
<td>CGTGC</td>
</tr>
<tr>
<td>world</td>
<td>TTCGT</td>
</tr>
</tbody>
</table>

For a five-character encoding, we obtain $4^5$ that are equal to 1024 which is equal to the 1024 used words. This 1024-word vocabulary is used to encode all of our messages. In this respect we apply two different methods, Recombinant DNA, and DNA Mutagenesis, as shown in sections 2.1, 2.2 respectively.

2.1 Using DNA Recombinant
In this method we use the recombinant technology to embed a message strand into another DNA strand using a DNA vector and restriction enzymes. The method is discussed at the sender and the receiver sides as shown next.

At the Sender End: 
Fig.1 explains the procedure of hiding the message into DNA strand using DNA Recombinant.

Fig.1: procedure of hiding message into DNA vector using DNA recombinant technology.

This procedure can be explained as follows:

A. Preparing the message:
The message will be designed simply by making a substitution to each word in it by its corresponding DNA bases, as shown in Fig.2

Plain text (original message) : Hello world
Cipher text (after substitution): CGTGC TTCGT
A header sequence will be added. As an example:
ACGTG CGTGC TTCGT
The restriction sites will be added on each end [2,3]:
GG*CCACGTGC CGTGC TTCGT A*AGCTT-3'

Fig.2: Preparing the message

HaeIII site Hello world HindIII site
• Red refers to the message words.
• Green refers to first restriction site.
• Purple refers to second restriction site.
• Black refers to the original DNA strand.

Then the message sequence will be synthesized by DNA synthesizer.

B. Preparing the DNA vector:
DNA vector will be digested by the same restriction enzymes, and the vector will be gel purified and dephosphorylated.
C. Embedding the message using the ligase enzyme:
The message is digested and ligated to the dephosphorylated vector using DNA ligase enzyme. The recombinant vector and other vectors which carry dummy messages will be sent to the receiver. [4]. Example:

If we want to embed (Hello world) into DNA, Fig.3 will explain the procedure:

- Get the mapped DNA message
  
  \[
  \text{CC ACGTG CGTGC TTCGT A}
  \]

- Cut the vector with any two enzymes (HaeIII, HindIII)
  
  TAGG \underline{CCGTAGCTGCTCCA} AGCTTGGA

- Replace the cut fragment with the message fragment:
  
  TAGG\underline{CCACGTGCGTGCTTCA}AGCTTGGA

- Make the recombinant between the message fragment and the vector.
- Send the final DNA into microdot.

Fig.3: Embedding the message using the ligase enzyme

Fig.4 demonstrates this process using plasmid vector [Appendix A].

At the Receiver End:
In order to retrieve the embedded message at the receiver end, the next few steps should be followed:

- Recombinant vectors will be linearized by the two restriction enzymes (Key) into the DNA [7] molecule. In case of bacterial messages, bacteria will be cultured and vectors will be extracted first.
- The DNA molecule will be denatured and sequenced using specific DNA primer carrying key elements (i.e. header sequence such as: ACGTG).
- The DNA message will be codified.

2.1.1 Security Vulnerability:
During the execution of the procedure, we have noticed that the words of the message must be sequential, and the utilized vector is short so it may be noticed by an attacker. Therefore, we suggest the following modifications.

2.1.2 Modifications:
We propose two imperative modifications:

- To avoid the sequential words we will use a vector for each word in the message, And a sequence number for each vector.
- To overcome the shortness of the vector we will mix these vectors with other vectors having the same length.

2.2 Using DNA Mutagenesis
In this method we use the mutation of the DNA nucleotides to change the nucleotides between two restriction sites of two selected restriction enzymes to the nucleotides of the message [Appendix C].

At the Sender End:
Fig.5 gives details the procedure of hiding a message into DNA strand using DNA Mutagenesis.

A. Prepare the message

B. Scan specific DNA sequence for specific two restriction sites

C. Use Mutagenesis technique to modify bases and create your DNA message

Fig.5: procedures of hiding message into specific DNA sequence using mutagenesis technique.
This procedure can be explained as follows:

**A. Prepare the message:**
The message will be designed in the same way except that it will not be synthesized.

**B. Scan specific DNA sequence for specific two restriction sites:**
A specific DNA sequence will be scanned for the occurrence of specific two restriction sites of two selected restriction enzymes.

**C. Use DNA Mutagenesis to modify bases and create your DNA message:**
The bases will be modified using DNA mutagenesis technique to create the message [6]. Fig.6 will explain an example,

- If we want to embed (Hello world) into DNA then the message will be: CGTGC TTCGT

- If the original DNA sequence is:
  
  \[
  \text{AGG*CCGTA*AGCTTAAGG*CCAAA*AGCTT}
  \]

  With \(\text{Hae} III (\text{GG**CC})\), and \(\text{Hind} III (\text{A*AGCTT})\) restriction sites.

- Use Mutagenesis to insert the bases encoding the message between the two restriction site sequences.

  \[
  \text{AGGCC GT AAGCTTAAGGCC AA AAGCTT}
  \]

  \[
  \text{AGGCCCGTCAAAGCTTAAAGGCCCTTGTAAGCTT}
  \]

  [It must be considered that only one word between each two restriction sites is allowed].

- Send the mutated DNA into a micro dot.

Fig.6: Hiding message into DNA strand using DNA Mutagenesis.

**At the Receiver End:**
The next two steps are followed in order to retrieve the embedded message at the receiver end:

- DNA will be sequenced using specific DNA primer carrying the restriction site and other key elements (i.e. header and sequence number encoding sequences) [7].
- The DNA message will be codified.

**2.2.1 Security Vulnerability:**
During the execution of the procedure, we have noticed that the message may contain one of the target substrings of the cutting enzyme. Moreover, there are too many fragments will be found after cutting with enzymes which is very hard to get the message fragments. Also, the selected fragments are disordered, and that the mutagenesis technique is time and effort consuming. It is also more expensive than DNA recombinant technology.

**2.2.2 Modifications:**
In order to overcome some or all of these vulnerabilities, we propose the following modifications:

- To avoid that the message may contain one of restriction enzymes target substring we must choose another enzyme.
- To avoid long search for the message fragments a header for each message fragment can be made.
- To avoid unordered fragments a sequence number for each message fragment can be made. Example:
  Choose ACGTG as header (The Key)
  And ACCTT as 1
  And TTCGC as 2
  And so on….
  Fig.7 will explain an example.

  \[
  \text{TAGGCAACGTGACCTTCCTGCAAGCTT}
  \]

  \[
  \text{AAGGCCACGTGATTGCTTCGTAAGCTT}
  \]

  Fig.7: Adding the header and the sequence number

  In the next section, we will formalize all of these procedures in an algorithmic pseudo code.

**3 Procedure of hiding a message onto DNA**

We propose a procedure, shown in [Appendix D] in a formal description, of hiding message into DNA file. In order to retrieve the message from the DNA file we use the previously demonstrated algorithm but in a converse order. The next section outlines the details of the implementation of the proposed algorithm.
4 Implementation
In this work we have developed a C# program with its associated the Database, as shown in [Appendix C], that can encrypt and decrypt utilizing a given DNA file. The process is summarized as follows:
- Defining the mapping table with limited vocabulary and their mapped DNA without any duplication.
- Defining all the restriction enzymes and their cutting substring (restriction sites).
- Searching in the DNA file for the number of occurrences for its substring (restriction site) for each enzyme.
- Input the message and the header for each fragment.
- Selecting any two restriction enzymes that will cut the DNA.
  [Note: The number of occurrences must exceed the number of words in the message.]
- Running the program and it will generate the final DNA file with the hidden message.
- To get the message back, the user is asked to input the fragment header and the two restriction enzymes. The program generates the message again from the DNA file.

In the following section, we will summarize this report and provide conclusions.

5 Summary and conclusion
The DNA is a good medium for data hiding because of its great length and high randomness, so we proposed a new steganographic technique based on the DNA, which is summarized as follows:

At sender end, we encrypt the message by:
- Substitute each word to its corresponding DNA mapped word.
- Choose any two restriction enzymes.
- Replace the nucleotides between the two selected restriction sites with the message fragments using (DNA Recombinant or DNA Mutagenesis) and hide it in microdot.

At receiver end, we decrypt the message by:
- Getting all fragments that contain the header and arrange them according to the sequence number.
- Making the backward substitution to find the plain message.

Based on the given discussion, we believe that this type of data hiding technique, or steganography, employing DNA strands has many potential applications not only as a security device but also in massive data storage applications.

6 References
Appendix A: DNA Recombinant

DNA Recombinant: Two or more DNA strands are incorporated into a single recombinant molecule.

Plasmid vectors are small circular molecules of double stranded DNA derived from natural plasmids that occur in bacterial cells. A piece of DNA can be inserted into a plasmid if both the circular plasmid and the source of DNA have recognition sites for the same restriction endonuclease. The plasmid and the foreign DNA are cut by this restriction endonuclease (EcoRI in this example) producing intermediates with sticky and complementary ends. Those two intermediates recombine by base-pairing and are linked by the action of DNA ligase. A new plasmid containing the foreign DNA as an insert is obtained.

The new plasmid can be introduced into bacterial cells that can produce many copies of the inserted DNA. This technique is called DNA cloning. There is another cloning vector-like: Lambda phage, cosmid and yeast artificial chromosome (YAC).

Appendix B: Database dictionary

Enzyme Table:

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>numeric(18,0),</td>
</tr>
<tr>
<td>Enzyme_Name</td>
<td>char(10),</td>
</tr>
<tr>
<td>Sub_String</td>
<td>char(10)</td>
</tr>
</tbody>
</table>

Save the entire Enzymes list and their target substrings.

Mapping Table:

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>numeric(18,0),</td>
</tr>
<tr>
<td>Alphabet</td>
<td>char(10),</td>
</tr>
<tr>
<td>Mapping</td>
<td>char(10)</td>
</tr>
</tbody>
</table>

Save all the 1024 words&their DNA mapping.

Appendix C: DNA Mutagenesis

Mutation: Is change of the DNA sequence within a gene or chromosome of an organism resulting in the creation of a new character or trait not found in the parental type. The process by which such a change occurs in a chromosome, either through an alteration in the nucleotide sequence of the DNA coding for a gene or through a change in the physical arrangement of a chromosome.

For example : Insertion and Deletion (InDel) Mutation, Fig.8 will explain the example.

Moose+0:

<table>
<thead>
<tr>
<th>ggt tct cta</th>
<th>tta gga gtt</th>
<th>tgc tta atc tta gaa atc</th>
</tr>
</thead>
<tbody>
<tr>
<td>G S L L G V C L I L E I</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moose+1:

<table>
<thead>
<tr>
<th>ggt tct cta</th>
<th>tta Tgg agt</th>
<th>ttg ctt aat cga aat c</th>
</tr>
</thead>
<tbody>
<tr>
<td>G S L L W S L L N L *</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moose-1:

<table>
<thead>
<tr>
<th>ggt tct cta</th>
<th>tta Vgag tt</th>
<th>get taa tct tag aac tc</th>
</tr>
</thead>
<tbody>
<tr>
<td>G S L L E F A *</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moose+2:

<table>
<thead>
<tr>
<th>ggt tct cta</th>
<th>tta TTg gag tt</th>
<th>get taa tct tag aac tc</th>
</tr>
</thead>
<tbody>
<tr>
<td>G S L L L E F A *</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moose-2:

<table>
<thead>
<tr>
<th>ggt tct cta</th>
<th>tta Vgag tt</th>
<th>get taa tct tag aac tc</th>
</tr>
</thead>
<tbody>
<tr>
<td>G S L L S L L N L *</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moose+3:

<table>
<thead>
<tr>
<th>ggt tct cta</th>
<th>tta TTA gga gtt</th>
<th>tgc tta atc tta gaa atc</th>
</tr>
</thead>
<tbody>
<tr>
<td>G S L L L G V C L I L E I</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moose-3:

<table>
<thead>
<tr>
<th>ggt tct cta</th>
<th>tta Vggt tgc tta atc tta gaa atc</th>
</tr>
</thead>
<tbody>
<tr>
<td>G S L L V C L I L E I</td>
<td></td>
</tr>
</tbody>
</table>

The first line shows the standard moose sequence, with 4th, 5th, and 6th triplets highlighted. The second and third lines show a single-nucleotide insertion (T) or deletion (V) at the fifth triplet. The fourth and fifth lines show a double-nucleotide insertion (TT) or deletion (V) at the fifth triplet. Lines 6 and 7 show a triplet insertion (TTA), or deletion (V) of the fifth triplet.

Fig.8: InDel Mutations.
Appendix D: Formal description of The proposed procedure

In the following pseudo code, we describe the proposed procedure in a formal approach.

### Algorithm of hiding message into DNA text file

Given a DNA text file D, List of enzymes and its target substrings (Enzymes_{K,2}), List of vocabulary and its DNA mapping (Mapping_{M,2}), EnzymesAndOccurrences_{K,2}]

**Input:** D, Enzymes_{K,2}, Mapping_{M,2}, Message

**NumberOfWords**

**Output:** NewD with the hidden message

**Algorithm Body:**

1. **NumberOfEnzyme := K**
2. for i = 1 to NumberOfEnzyme
   - // loop for all known enzymes to get the number of occurrences
     - EnzymesAndOccurrences_{K,0} := Enzymes_{K,0}
     - EnzymesAndOccurrences_{K,1} := GetNumberOfOccurrence(Enzymes_{K,1})
     - next i
   - // Function to get the number of occurrences of each target substring
     - int GetNumberOfOccurrence(TargetSubString)
       - begin function
         - S := Read From D (DNA file)
         - Count := 0
         - While (S Contains TargetSubString)
           - Count := Count + 1
           - S := S started from Position of TargetSubString
         - End while
         - return Count
       - End function
   - // Procedure of hiding words of the message into the DNA file using two restriction enzymes
     - S := Read from D (DNA file)
     - Target1 := The target substring of Enzyme1
     - Target2 := The target substring of Enzyme2
     - S_{before} := ""
     - for i := 1 to NumberOfWords
       - position1 := The position of Target1
       - position2 := The position of Target2
       - while (position2 > position1)
         - S_{before} := S_{before} + S to position1
         - S := S from position1 to End
         - position1 := The position of Target1
         - position2 := The position of Target2
       - End while
     - S := S from position2 to End
     - Msg := S before and DNAword and Target2
     - // Test if the message contains one of the target substrings
     - if (!TestMessage(Msg)) then do
       - Error Message
       - Break
     - else
       - Write to NewD (the new DNA File)
     - end do
   - next i

End Algorithm.