Review Paper

# Security analysis of DNA based steganography techniques

Omnia Abdullah Al-Harbi[1] · Walaa Essa Alahmadi[1] · Asia Othman Aljahdali[1]

Received: 2 October 2019 / Accepted: 20 December 2019 / Published online: 9 January 2020


**Abstract**
This study investigates the most recent data hiding techniques based on DNA steganography, including the highly improved DNA-based steganography technique, the data hiding using double DNA sequences method, and the enhanced DNA-based steganography technique. The strengths and weaknesses of these techniques are discussed. Additionally, the security of these techniques is analyzed based on several security parameters that measure the quality of DNA steganography with respect to many factors, including, but not limited to, cracking probability, blindness, modification rate and expansion rate, and layers of security. The goal of the comparison between the investigated techniques is to highlight the advantages and disadvantages of the existing data hiding algorithms and to motivate future research in this field. Moreover, the paper evaluates the discussed techniques based on some parameters, including capacity, payload, and bit per nucleotide (bpn). The result shows that the enhanced DNA-based steganography technique hides 2 bpn, whereas the highly improved method can hide on average 1.46 bpn, which is higher than data hiding using double DNA sequences method can hide .The paper also presents suggestions for how each technique can be optimized to to achieve a higher security level for hiding data within DNA sequences.

**Keywords** Hiding data · Security · Steganography · DNA sequence

## 1 Introduction

Cryptography and steganography are usually interrelated and share the common aims and services of preserving the confidentiality, integrity, and availability of information, which are some of the most significant fields in computer security [1–3]. Cryptography and steganography are methods allowing information to be sent securely [4]. Cryptography is an historical science that began in Egypt around 1900 B.C. with hieroglyphic writing [5]. It uses encryption to scramble the secret information in such a way that only the sender and the intended receiver can reveal it [6].On the other hand, steganography began in ancient Greece around 440 B.C. [7, 8]. It hides the secret information in different carriers in which the visibility of private information is made unavailable to unauthorized users. This is done by concealing the sensitive information within cover mediums such as images, video, and DNA in such a way that it becomes difficult to detect [5, 9]. Several algorithms have been proposed in image steganography for hiding secret information inside an image. However, the embedding capacity of the image is low, so it cannot hide a large data inside it [10]. In order to overcome the deficit of the capacity, DNA steganography has been introduced. DNA steganography is a research direction of DNA cryptography, which started in 1999. This approach uses DNA sequences as carriers to enable secure transfer of the critical data [4, 11]. The principal idea is basically to encrypt and conceal messages in a large number of DNA strands to prevent adversaries from reading and deciphering the messages. This could be achieved only if the original sequences are preserved from adversaries [3, 11, 12]. Hiding data in DNA sequences is a new and evolving scientific field. This paper intends to discuss

✉ Asia Othman Aljahdali, aoaljahdali@uj.edu.sa; Omnia Abdullah Al-Harbi, oalharbi0095.stu@uj.edu.sa; Walaa Essa Alahmadi, walahmadi0031.stu@uj.edu.sa | [1]College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia.

three different techniques recently presented for hiding data in DNA sequences and to investigate their security based on different factors. The paper suggests ways of enhancing and improving each technique with respect to the security requirements. This paper is divided into sections, Sect. 2 defines a DNA sequence and its elements, while Sect. 3 discusses in detail three different techniques that are used to hide information within DNA sequences. Sect. 4 analyzes each technique, clarifies their strengths and weaknesses, and compares them. The last section discusses each technique and proposed ideas for further improvements.

## 2 Elements of DNA

In biology, a deoxyribonucleic acid (DNA) is a huge molecule that exists within the cells of all living organisms, containing the genetic information that allows the functioning, reproduction, and evolution of these organisms [13]. DNA has many small subunits called nucleotides. It is made up of four types of nucleotide bases: Adenine (A), thymine (T), guanine (G), and cytosine (C) [14, 15]. The two strands are held together by bonds between the bases; adenine binds to thymine, and cytosine binds to guanine [1, 14, 16]. Every three neighboring nucleotides make up a codon so we get $4^3 = 64$ different possible codon combinations. In living organisms, the arrangement of these combinations determines the structure and function of the resultant protein [17]. DNA encoding techniques are binary coding schemes for the purpose of DNA computation. The most popular binary mapping of digital coding is given in Table 1.

## 3 DNA data hiding techniques

Over the years, different algorithms have been proposed in hiding sensitive data within DNA sequences. In this section, we will investigate and discuss the strengths and weaknesses of the lately proposed DNA-based data hiding techniques. The analysis aims to help future research in designing more reliable and secure data hiding techniques.

**Table 1** DNA digital coding [11]

| DNA nucleotide | Decimal | Binary |
|---|---|---|
| A | 0 | 00 |
| C | 1 | 01 |
| G | 3 | 10 |
| T | 3 | 11 |

### 3.1 Highly improved DNA-based steganography techniques

Malathi in [18] modifies the insertion algorithm to decrease the cracking probability of the fake DNA sequence. The algorithm uses two different keys. The first key ($K_1$) is a number in the range of 0 to 255, which is used to XOR the last character in the message ($M$); the result will be XORed with the character preceding the last one in the M, and so on. Accordingly, the first key is used to encrypt the message. The second key ($K_2$) is randomly generated and is used to divide the DNA sequence into same-length segments. The resulting cipher characters are inserted as binary bits one by one at the beginning of each segment. Then, the binary sequence is converted into DNA bases using Table 1. The second key is preferred to be a small number so that the DNA sequence has a minimum length while hiding the secret message.

*The encryption process* The proposed algorithm [18] follows several steps to encrypt and hide messages inside a DNA sequence. The encryption process steps are as follows:

1. Split $M$ into characters, $M = m_1, m_2, m_3, \ldots, m_n$, and each character is converted into its 8-bit binary equivalent based upon the ASCII standard.
2. Randomly generate a number between 0 and 255 to form $K_1$, and then the key is converted into an 8-bit binary sequence.
3. The last character in $M$ is XORed with $K_1$.
4. The result is XORed with the character preceding the last one in $M$; the XORing is repeated until all the characters are converted and stored in $A$.
5. The binary sequence $A$ is converted into a protein sequence.
6. A sample DNA sequence $S$ is selected randomly and converted into a binary bit sequence using Table 1.
7. Generate a random number, which is preferred to be a small number $K_2$, and then divide the DNA sequence $S$ into segments; the segment length should be equal to $K_2$.
6. Add the first binary value of $A$ at the beginning of the first DNA binary segment, and insert the second binary value of $K_1$ into the second binary segment, and so on.
7. Concatenate all the binary sequences, and then convert it to produce a fake DNA sequence using Table 1.

An illustrative example is given in Fig. 1 showing the encryption processes.

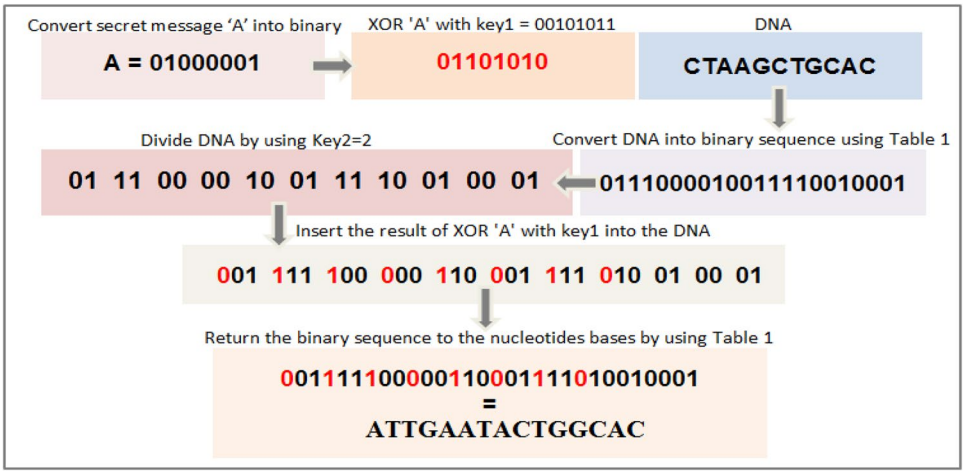


**Fig. 1** Example of the encryption process

*The decryption process* The receiver must have knowledge about $K_1$ and $K_2$ in order to decrypt the message. Additionally, he/she must receive the original DNA from the sender. The receiver performs the following steps:

1. Convert the received fake DNA sequence into a binary sequence using Table 1.
2. Divide the binary DNA sequence into segments; each segment's size will be equal to $K_2 + 1$.
3. Get the first bit from each segment and concatenate them to produce significant bits $B$.
4. XOR the first 8 binary bits of $B$ with $K_1$, and then XOR the second 8 bits in $B$ with the previous 8 bits of $B$, and so on.
5. Convert the binary bits of the DNA sequence into ASCII text value [18].

An illustrative example is given in Fig. 2 showing the decryption processes.

*Probability of cracking* The reference DNA size is about 163 million, thus, the probability of predicting the reference DNA sequence is $\frac{1}{1.63\times10^8}$. The probability of guessing the different binary coding (A,C,G,T) combinations is $\frac{1}{24}$. The probability of finding the message and reference DNA sequence is $\frac{1}{n-1}$, where $n$ is the number of bits in the fake DNA sequence. The message and DNA are segmented using a random key; thus, the probability of guessing the segmentation of the message is $\frac{1}{2^m-1}$, where $m$ is the number of bits in the secret message. The probability of guessing segmentation of DNA is $\frac{1}{2^{s-1}}$, where $s$ is the number of bits in the reference DNA sequence. The XOR operation is performed for encoding the data inside the DNA sequence, and the probability of the XOR combination is $\frac{1}{2^{8m}}$ [18].

Thus, the probability of finding the message hidden in the DNA sequence is:

$$\frac{1}{1.63\times10^8}\times\frac{1}{24}\times\frac{1}{(n-1)}\times\frac{1}{(2^m-1)}\times\frac{1}{2^{s-1}}\times\frac{1}{2^{8m}} \quad (1)$$

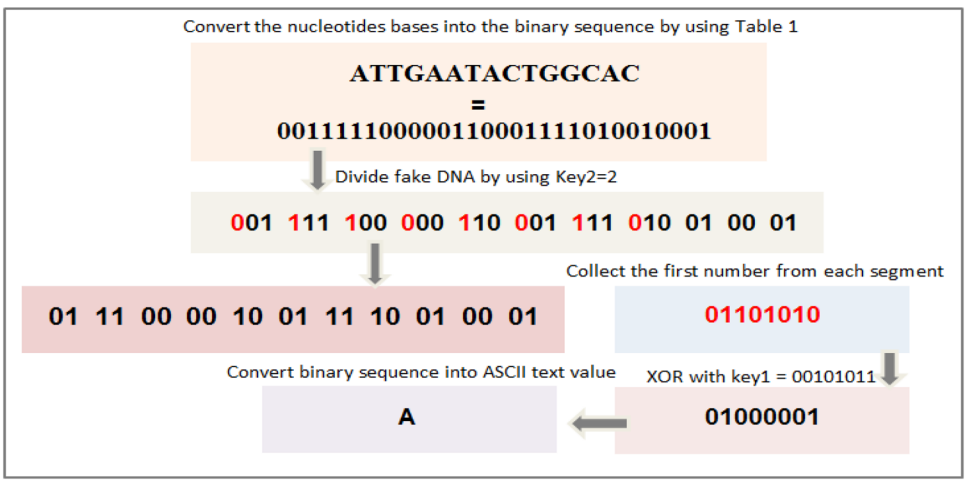


**Fig. 2** Example of the decryption process

### 3.2 Data hiding using double DNA sequences techniques

Ibrahim, Abdalkader, and Moussa [19] proposed an algorithm that uses a double DNA sequences technique. The main idea is to pick a random pair of DNA sequences from the DNA database (S, Ś), which is a combination of two DNA sequences. The proposed algorithm consists of two phases. In the first phase, the secret message *P* is encoded into the DNA sequence *DP*, in which each letter is replaced by three nucleotides. The first selected DNA sequence S is used for the encryption of *DP*. In the second phase, the other DNA sequence Ś is used to hide the encrypted secret data. The encryption and decryption processes are explained below.

*The encryption process* Two inputs are used in the encryption process: the secret message *P* and the DNA sequence pair (S, Ś ). The encryption steps are as follows:

1. Encode the secret message *P* into DNA sequence *DP* using Algorithm 1 to generate a total of 64 DNA codons. The NUM_FORMAT is a combination of three digits, and the DNA (NUM_FORMAT) transfers the resulting number to DNA codons (e.g, 122 converts into CGG).
2. Generate the encrypted message DṔ by performing the bitwise XOR of the secret message *DP*, and the reference sequence *S*, and then delete the extra unused nucleotide (DṔ = DP XOR S).
3. Append the ID of *S* at the beginning of the resulting DṔ to get IDsDṔ.
4. Read the second DNA sequence Ś and mark the second repeated characters.
5. Replace the second repeated characters with encrypted message characters DṔ.
6. Hide the characters of encrypted message DṔ in Ś using the replacement rules in Table 2. We refer to nucleotides in DṔ by {A(DṔ), C(DṔ), G(DṔ), T(DṔ)}. Table 2 is used to hide DṔ in Ś by replacing the second repeated letter in Ś with one of the four letters {A, C, G, T} according to the encrypted message.
7. Append the ID of Ś to the beginning of the resulting S̈ = (IDś (IDsDṔ)′) and send it to the receiver [19]. An illustrative example is given in Fig. 3 showing the encryption processes.

**Table 2** Hiding and Recovery

| A | | C | | G | | T | |
|---|---|---|---|---|---|---|---|
| Msg | sbs | Msg | sbs | Msg | sbs | Msg | sbs |
| A | A | A | C | A | G | A | T |
| C | C | C | A | C | T | C | G |
| G | G | G | T | G | A | G | C |
| T | T | T | G | T | C | T | A |

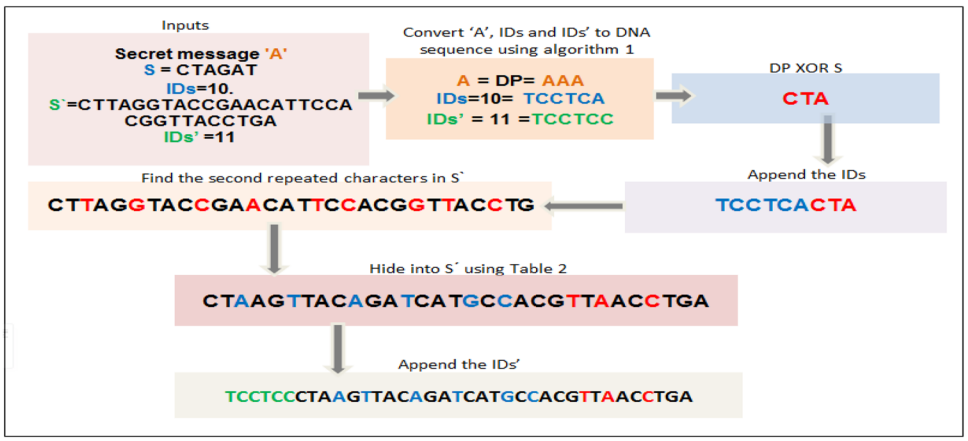


**Fig. 3** Example of the encryption process

**Algorithm** 1: Generating DNA Codons.

```
for i = 0 to 3 do
    for j = 0 to 3 do
        for k = 0 to 3 do
            NUM_FORMAT= ijk
            codon=DNA(NUM_FORMAT)
        end for
    end for
end for
```

*The decryption process* The Dncryption process's input is a faked DNA sequence S̈= ((IDś (IDsDṕ)′)) with a secret hidden message. The decryption process is as follows:

1. Extract the first bases that represent ID of Ś used by the sender to hide the data.
2. Find the second repeated nucleotide in Ś.
3. Extract the Dṕ sequence from S̈ using the replacement inverse rules in Table 2.
4. Extract the first bases form Dṕ that represent the ID of S.
5. Decrypt Dṕ as follows: use the commutative property of XOR Dṕ XOR S = (*DP* XOR S) XOR S = *DP* XOR (S XOR S) = *DP*.
6. Decode *DP* to letters, with each group of three nucleotides representing a letter from the English alphabet.
7. Get plaintext *P* [19].

An illustrative example is given in Fig. 4 showing the decryption processes.

*Probability of cracking* The reference DNA size is about 163 million, thus, the probability of predicting the reference DNA sequence is $\frac{1}{1.63\times10^8}$. The probability of guessing the second selection Ś is $\frac{1}{1.63\times10^8}$, where the reference DNA sequence Ś is used to hide the secret message. There are $24^4$ possible situations for hiding based on Table 2 in the hiding process; thus, the probability of an attacker making a successful guess is $\frac{1}{24^4}$. The total number of permutations between the letters of English alphabet (capital and small letters), the ten digital numbers, the two punctuation marks, and the 64 codons that are generated from Algorithm 1 is $p_{64}^{64}$ [19]. Thus, the probability of inferring the secret message is:

$$\left(\frac{1}{1.63\times10^8}\right)^2\times\frac{1}{24^4}\times\frac{1}{p_{64}^{64}} \tag{2}$$

### 3.3 Enhanced DNA-based steganography technique with a higher hiding capacity

Marwan, Shawish, and Nagaty [20] introduced this approach to simplify the current techniques and obtain a higher hiding capacity. This technique follows two phases. The first phase is the encryption phase, which is a modified version of the 5 × 5 Playfair cipher grid called the 4 × 4 Playfair cipher grid. The result of this phase is an encrypted message. The second phase is the hiding phase, which is a substitution process used for hiding the encrypted message. The result of this phase is a fake DNA sequence. The encryption and decryption processes are described below.

*The encryption process* There are four inputs for the encryption process: a message, a key, initial values of the 4 × 4 binary grid, and initial values of the 4 × 4 DNA grid. The 4 × 4 binary grid and DNA grid must be shared between the sender and the receiver before the encryption and decryption processes. The encryption process steps are as follows:

1. Generate 16 random English letters to create the 4 × 4 Playfair cipher grid using the given key input as a seed value; an example of Playfair cipher grid is given in Table 3.
2. Shuffle the initial values of the 4 × 4 binary grid and the 4 × 4 DNA grid using the key; an example of a shuffled 4 × 4 binary and DNA grid is given in Tables 4 and 5, respectively.

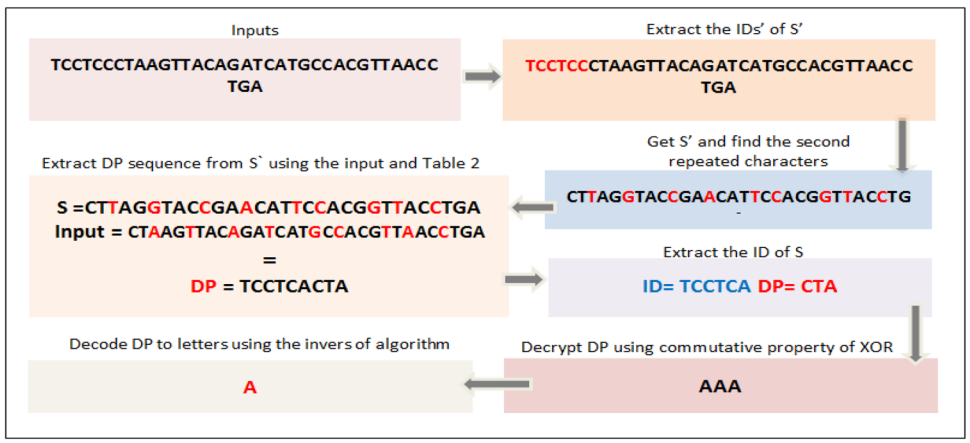


**Fig. 4** Example of the decryption process

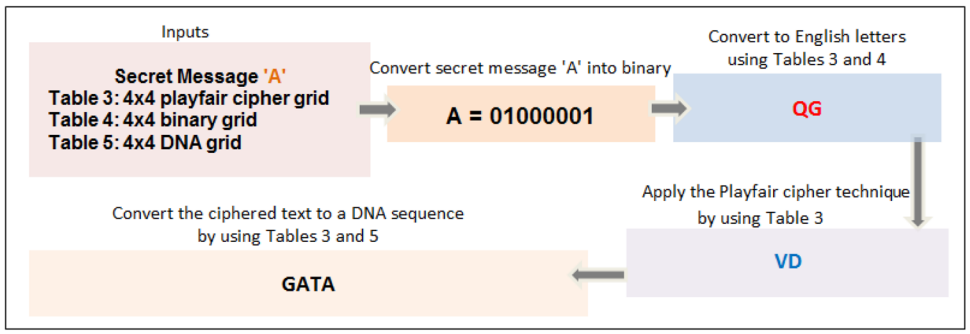


**Fig. 5** Example of the encryption process

**Table 3** Example of 16 randomly generated English letters

| | | | |
|---|---|---|---|
| H | C | M | U |
| D | G | Z | B |
| I | A | X | J |
| Q | V | W | F |

**Table 4** 4 × 4 Shuffled binary grid

| | | | |
|---|---|---|---|
| 0110 | 0001 | 1000 | 0000 |
| 1001 | 0101 | 1010 | 1110 |
| 0100 | 1111 | 1011 | 1101 |

**Table 5** 4 × 4 Shuffled DNA grid

| | | | |
|---|---|---|---|
| GT | CG | CA | TG |
| TA | TC | GG | AA |
| AT | TT | CT | AG |
| GC | GA | CC | AC |

3. Convert the input message into a binary sequence (B).
4. Find all 4-bit values of B in the 4 × 4 binary grid, and then map their positions to the corresponding positions in a 4 × 4 cipher grid and fetch the English letter. The result of this step is a sequence of English letters (E).
5. Apply the Playfair cipher technique to the sequence of English letters (E) to get the encrypted text (C).
6. For each letter of (C), map its position in a 4 × 4 Playfair cipher grid to its corresponding position in the 4 × 4 DNA grid and get the values. The outcome of this step is a DNA sequence (Enc).
7. Pick a reference DNA sequence from the database (DNA database).
8 Hide the encrypted DNA sequence in the chosen reference DNA sequence using the substitution process [20].

An illustrative example is given in Fig. 5 showing the encryption processes.

*The decryption process* There are two inputs for the decryption process: the encrypted DNA sequence and the key. The receiver will receive these inputs through a secure channel. The initial values of the 4 × 4 binary grid and DNA grid should be shared before the encryption and decryption processes.

1. Use the reverse of the substitution process to extract the hidden encrypted DNA sequence.
2. Shuffle the initial values of the 4 × 4 binary grid and 4 × 4 DNA grid using the key.
3. Generate 16 random English letters to create the 4 × 4 Playfair cipher grid using the received key as a seed value.
4. For each group of two letters of DNA sequence (Enc), map its position in a 4 × 4 DNA grid to its corresponding position in the 4 × 4 cipher grid and get the values. The outcome of this step is encrypted text (C).
5. Apply the inverse of the Playfair cipher technique to the encrypted text (C) to get a sequence of English letters (E).
6. For each English letter in (E), map its positions in a 4 × 4 cipher grid to its corresponding position in a 4 × 4 a binary grid and get the values. The outcome of this step is a binary sequence (B).
7. Convert the binary sequence (B) into the original message [20].

An illustrative example is given in Fig. 6 showing the decryption processes.

*Probability of cracking* In this technique, the attacker needs 4 types of information to decrypt a message, which are the binary representation, the reference DNA, the complementary rule, and the ciphering technique. Thus, the probability of getting the binary scheme $b$ is $\frac{1}{4!}$. Since we have 4 DNA bases, the number of possible binary schemes is 4!. The probability of guessing the reference DNA $r$ is $\frac{1}{1.6\times10^8}$. The probability of the complementary rule $c$ is $\frac{1}{16}$. Thus, probability of cracking the $k$ is

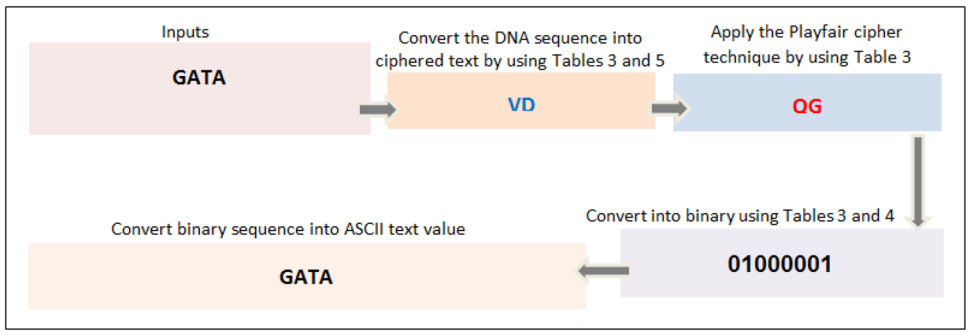


**Fig. 6** Example of the decryption process

$$\frac{1}{24 \times 1.6 \times 10^8 \times 16} \quad (3)$$

## 4 Security analysis

In this section, we will analyze the security of all the discussed algorithms in the previous section and investigate whether they fulfill the security requirements summarized by their factors.

The quality of a steganography DNA technique depends on many factors, including the following:

- Cracking probability of the algorithm; analyzing the probability to ensure that a minimum cracking probability leads to a secure steganography technique.
- Layer of security, which refers to the number of DNA sequences used in the data hiding technique; the algorithm can have a single or a double hiding layer. A double hiding layer technique is more secure than a single one [4].
- Blindness, which means that the algorithm does not require to send the original DNA sequence to the recipient. The security degree is maximized by minimizing the required data sent to the recipient.
- Modification rate and payload; a low modification rate and a payload equal to zero result in higher quality DNA steganography.
- Encrypting the secret data into cipher before embedding it into a DNA sequence is more secure than embedding the original data in a DNA sequence.
- Preserving the original functionality of the reference DNA in such a way that its function of producing proteins is not affected.
- The most essential factor in a data hiding techniques is the key that makes attacking data hiding system much more difficult.
- The capacity, defined as the amount of data hidden within the carrier. it is necessary that a data hiding technique maintains a sizable hiding capacity.
- Bits per nucleotides (bpn), it is the total number of bits hidden per character [21–25]

*Highly improved DNA based steganography techniques* The algorithm has many features distinguishing it from other algorithms used in hiding data in DNA sequences. One of the most important advantages is that the probability of knowing the data inside the DNA sequence by an adversary is very small. In addition, this algorithm encrypts the information before it is hidden into a DNA sequence, and the encryption process is based on a randomly chosen key ($K_1$). Also, another random key ($K_2$) is used in dividing DNA sequence into segments. The ciphertext bits are then placed between these segments, with the two keys kept secret, which increases the strength of this algorithm. In contrast, other data hiding algorithms send the message as an integrated series placed inside the DNA sequence, which is insecure. However, in this technique, after the encryption process of the message, the ciphertext is divided into binary bits, and then each bit is inserted between the segments of the DNA sequence. Thus, it is difficult to extract the ciphertext and distinguish it from a long series of DNA sequence. Although this technique has many advantages, it has also many defects. The algorithm changes the length of the DNA sequence, and a human genome consists of $3 \times 10^9$ pairs of nucleotides [11]. This is because the algorithm does not remove DNA stretches to replace them with the encrypted message, but it directly inserts the ciphertext, thereby increasing the length of the DNA sequence; thus, it would be easy for an adversary to figure out the fake DNA. Also, this technique does not take into account preserving the functionality of the DNA sequence.

*Data hiding using double DNA sequences techniques* The technique is highly secure, for several reasons: the secret data is encrypted before being embedded in the DNA sequence. Moreover, it is a double-layer technique that uses two different DNA sequences (S, Ś) to ensure the

security. This technique also possesses very high data hiding capacity and preserves the reference DNA's original function of producing proteins. The output of the encryption process is a fake DNA sequence with a modification rate of approximately 28.4%, which is low [19]. The expansion rate is equal to zero, which means that after embedding the secret data, the length of the reference DNA sequence is not expanded. In fact, a low modification rate and zero expansion rate ensures the security and results in a better quality of the fake DNA sequence. Furthermore, this technique is a blindness technique, meaning that there is no need to send the original DNA to the receiver, so the security degree is maximized. Finally, the probability of cracking is low. On the other hand, this technique has some weaknesses. The replacement rules should be sent to the receiver, and plain text must contain only capital letters, small letters, 0, … , 9, a period, and a dot; it cannot contain other punctuation marks. Also, the algorithm does not use any type of key.

*Enhanced DNA-based steganography technique with a higher hiding capacity* The security of this technique is based on several elements. First, the secret data is encrypted before being embedded into the DNA sequence. Moreover, the encryption and hiding processes of secret data are done by using the Playfair and substitution methods. Accordingly, the Playfair method provides a higher hiding capacity and stronger security, besides being a fast and simple method. The substitution method preserves the length of DNA sequence, so the payload is always zero. Furthermore, this technique uses a secret key, which grants a higher security level to the data hiding system. Finally, preserving the reference DNA's original function of producing proteins is a considerable asset of this technique. On the other hand, the technique is not a blindness technique. The sender and receiver must share some data before the encryption and decryption processes. The blindness feature is to maximize the security level by reducing as much as possible the required data that are transferred to the receiver.

A comparison between the investigated techniques is given to highlight the advantages and disadvantages of the existing data hiding algorithms and to provide motivation for future research in this field. Table 6 presents the strengths and weaknesses of each previously explained technique.

## 5 Experimental results

The techniques were tested in [18–20] using eight real DNA sequences from the NCBI database [Ref:https://www.ncbi.nlm.nih.gov/]. The experiment's goal was to evaluate the discussed techniques based on some parameters, including capacity, payload, and bpn.

As mentioned before, the capacity refers to the total length of the extended reference sequence after the secret message is hidden within it, which can be calculated by $|S| + \frac{|M|}{2}$ [18]. The payload is the remaining length of the new sequence after extracting out the reference DNA sequence, and can be calculated by $\frac{|M|}{2}$ [18]. The bpn is the number of bits hidden per character, which can be calculated by $bpn = \frac{|M|}{C}$ [19], where $|M|$ is the length of the secret message, $C$ is the capacity, and $|S|$ is the length of the reference DNA sequence.

We will show and compare the experimental result of the three techniques. Table 7 shows the performance of the data hiding using double DNA sequences, and the highly improved DNA-based steganography techniques for hiding a 20000-byte secret message in the DNA sequence regarding the capacity, payload, and bits per nucleotide (bpn).

**Table 6** Comparing the discussed techniques based on different quality factors

| Quality factors | Highly improved DNA based steganography | Data hiding using double DNA sequences | Enhanced DNA-based steganography |
|---|---|---|---|
| *Cracking probability* | Very low cracking probability | Low cracking probability | Low cracking probability |
| *Security layer* | Double layer | Double layer | Double layer |
| *Blindness* | Does not support blindness | Support Blindness | Does not support blindness |
| *Modification rate* | Low | Low | Low |
| *Payload* | Not equal to zero | Always equal to zero | Always equal to zero |
| *Expansion rate* | Other DNA length | Same DNA length | Same DNA length |
| *Encrypting the secret data* | Yes (XOR) | Yes | Yes |
| *Preserving DNA functionality* | Changing DNA functionality | Preserving DNA functionality | Preserving DNA functionality |
| *Using keys* | Uses two keys | Doesn't use a key | Use a key |
| *High capacity* | Yes | Yes | Yes |
| *Easy to apply* | Easy to implement | Not easy to implement | Easy to implement |
| *Number of used DNA sequences* | One | Two | One |

**Table 7** The capacity, payload, and bpn for each technique

| Sequence | No. of nucleotides | Highly improved DNA based steganography techniques | | | Data hiding using double DNA sequences techniques | | |
|---|---|---|---|---|---|---|---|
| | | Capacity | Payload | Bpn | Capacity | Payload | Bpn |
| AC153526 | 200,117 | 280,117 | 80,000 | 1.52 | 200,117 | 0 | 0.577 |
| AC166252 | 149,884 | 229,884 | 80,000 | 1.2 | 149,884 | 0 | 0.580 |
| AC167221 | 204,841 | 284,841 | 80,000 | 1 | 204,841 | 0 | 0.563 |
| AC168874 | 206,488 | 286,488 | 80,000 | 1.38 | 206,488 | 0 | 0.560 |
| AC168897 | 200,203 | 280,203 | 80,000 | 1.49 | 200,203 | 0 | 0.565 |
| AC168901 | 191,456 | 271,456 | 80,000 | 1.99 | 191,456 | 0 | 0.583 |
| AC168907 | 194,226 | 274,226 | 80,000 | 1.6 | 194,226 | 0 | 0.580 |
| AC168908 | 218,028 | 298,028 | 80,000 | 1.52 | 218,028 | 0 | 0.583 |
| Average bpn | | 1.46 | | | 0.574 | | |

The enhanced DNA-based steganography technique hides 2 bits per nucleotide; for example, a reference DNA sequence of 149,884 bp can hide a message up to 36.56 Kb [20], whereas the highly improved method can hide on average 1.46 which is higher than the 0.574 bpn that the data hiding using double DNA sequences method can hide on average.

The data hiding using double DNA sequences method preserves the length of the original DNA sequence (the payload is always 0), whereas the highly improved method and the enhanced DNA-based steganography method increase the length of the reference DNA sequence.

## 6 Discussion

The modified insertion technique [18] is used to hide secret messages in DNA sequence. It merges the message within protein sequences. However, to solve the problem of increasing DNA length, we recommend concatenating the bits that arise from XORing the messages with $K_1$, and then adding the encoded text to the DNA series. After that, delete the same number of bits from the original DNA sequence, but from the DNA series that exist after the fake DNA. Additionally, this technique does not support blindness, which requires sending the original DNA sequence to the recipient, which would minimize the security degree. To solve this problem, we suggest using a primer key ($K_s$), which determines the position of embedding the ciphertext into the fake DNA. Thus, the technique now supports blindness and does not need to send the original DNA to the recipient.

Data hiding using double DNA sequences [19] is considered a highly secure technique, but like many techniques, it has some weaknesses. To improve the security and efficiency of any technique, we should focus on its vulnerabilities. One of the weaknesses in this technique is that it does not use a key. To provide more security to this technique, an asymmetric encryption schema could be used and implemented by encrypting the message using any schema, and then starting the encryption process of data hiding using the double DNA sequences technique. This will increase the security degree and eliminate one of its vulnerabilities.

As mentioned before, the enhanced DNA-based steganography technique technique [20] has a weakness. The initial values of the $4 \times 4$ binary grid and DNA grid must be shared before the encryption and decryption processes, which means that this technique is not a blindness technique. To overcome this weakness, it is required to minimize the shared data. Furthermore, using a public key rather than a secret key would improve the algorithm's security.

## 7 Conclusion

Hiding data in DNA sequences is a new science and an evolving field. This paper investigates several DNA-based steganography techniques that have recently been proposed and analyzes each technique separately by specifying its advantages and disadvantages. A comparison between these techniques was carried out based on security and quality factors that are important for developing efficient and secure DNA-based data hiding techniques. The analysis shows that all the reviewed techniques meet most of the security and quality requirements. Accordingly, all three techniques have a low cracking probability, a double layer of security, a low modification rate, and a high hiding capacity. On the other hand, the highly improved DNA-based steganography technique does not achieve the blindness and expansion rate, and does not preserve the original functionality of the reference DNA sequence. Moreover, the data hiding using double DNA sequences technique does not use a key, which is

the most essential element in data hiding. Furthermore, the enhanced DNA-based steganography technique does not possess the blindness property. The aim of the comparison in this study is to help in designing efficient and secure DNA data hiding techniques; thus, the paper suggests ways of enhancing and improving the investigated techniques with respect to the security requirements.

## Compliance with ethical standards

**Conflicts of interest** The authors declare that they have no conflict of interest.

## References


1. Information Resources Management Association (2018) Cyber security and threats: concepts, methodologies, tools, and applications. IGI Global
2. Provos Niels, Honeyman Peter (2003) Hide and seek: an introduction to steganography. IEEE Secur Priv 1(3):32–44
3. Krishnan RB, Thandra PK, Sai Baba M (2017) An overview of text steganography. In: 2017 4th international conference on signal processing, communication and networking (ICSCN). IEEE
4. Sokół B, Yarmolik VN (2005) Cryptography and steganography: teaching experience. Enhanced methods in computer security, biometric and artificial intelligence systems. Springer, Boston, pp 83–92
5. Siper A, Farley R, Lombardo C (2005) The rise of steganography. In: Proceedings of student/faculty research day, CSIS, Pace University
6. Selvaraj D (2017) Development of a secure communication system based on steganography for mobile devices. p 3
7. Vinodhini RE, Malathi P, Gireesh Kumar T (2017) A survey on DNA and image steganography. 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE
8. Kahn David (1996) The history of steganography, international workshop on information hiding. Springer, Berlin
9. Petitcolas Fabien AP, Anderson Ross J, Kuhn Markus G (1999) Information hiding—a survey. Proc IEEE 877:1062–1078
10. Malathi P, Gireeshkumar T (2016) Relating the embedding efficiency of LSB steganography techniques in spatial and transform domains. Procedia Comput Sci 93:878–885
11. Clelland CT, Risca V, Bancroft C (1999) Hiding messages in DNA microdots. Nature 399:533–534
12. Sharma A (2016) Security and information hiding based on DNA Steganography. A Monthly J Comput Sci Inf Technol 5(3):827–832
13. Ginu A, Jeenu J, Vishnu P, Jerin D (2017) DNA based cryptography and steganography. GRD J Glob Res Dev J Eng 2:249–253
14. Kiss Gábor (2018) How to teach the history of cryptography and steganography. Educaţia Plus 20(2):13–23
15. Abbasy MR et al (2012) DNA base data hiding algorithm. Int J New Comput Archit Appl 2(1):183–192
16. Khalifa A (2013) LSBase: a key encapsulation scheme to improve hybrid crypto-systems using DNA steganography. In: 2013 8th international conference on computer engineering & systems (ICCES). IEEE
17. Petsko Gregory A, Ringe Dagmar (2004) Protein structure and function. New Science Press, Beijing
18. Pa Malathi, Ma Manoaj, Ra Manoj, Vaikunth R, Vinodhini R (2017) Highly improved DNA based steganography. Procedia Comput Sci 115:651–659
19. Ibrahim Fatma E, Abdalkader HM, Moussa MI. Enhancing the security of data hiding using double DNA sequences. In: Industry Academia collaboration conference (IAC)
20. Marwan S, Shawish A, Nagaty K (2015) An enhanced DNA-based steganography technique with a higher hiding capacity. Bioinformatics 1:150–157
21. S Sajisha K (2017) An encryption based on DNA cryptography and steganography. In: International conference on electronics, communication and aerospace technology(ICECA)
22. Jain S, Bhatnagar V (2014) Analogy of various DNA based security algorithms using cryptography and steganography. In: 2014 international conference on issues and challenges in intelligent computing techniques (ICICT). IEEE
23. Hamed G et al (2016) Comparative study for various DNA based steganography techniques with the essential conclusions about the future research. In: 2016 11th international conference on computer engineering & systems (ICCES). IEEE
24. Hamed G et al (2015) Hybrid technique for steganography-based on DNA with n-bits binary coding rule. In: 2015 7th International conference of soft computing and pattern recognition (SoCPaR). IEEE
25. Dilovan Z, Habibollah H, Subhi z (2017) Security issues in DNA based on data hiding: a review . Int J Appl Eng Res ISSN