A Hybrid Encryption Technique based on DNA Cryptography and Steganography

Shahriar Hassan[1], Md. Asif Muztaba[1], Md. Shohrab Hossain[1] and Husnu S. Narman[2]
[1]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh
[2]Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, USA
Email: shahriar.hassan303@gmail.com, amuztaba18@gmail.com, mshohrabhossain@cse.buet.ac.bd, narman@marshall.edu

***Abstract*—The importance of data and its transmission rate are increasing as the world is moving towards online services every day. Thus, providing data security is becoming of utmost importance. This paper proposes a secure data encryption and hiding method based on DNA cryptography and steganography. Our approach uses DNA for encryption and data hiding processes due to its high capacity and simplicity in securing various kinds of data. Our proposed method has two phases. In the first phase, it encrypts the data using DNA bases along with Huffman coding. In the second phase, it hides the encrypted data into a DNA sequence using a substitution algorithm. Our proposed method is blind and preserves biological functionality. The result shows a decent cracking probability with comparatively better capacity. Our proposed method has eliminated most limitations identified in the related works. Our proposed hybrid technique can provide a double layer of security to sensitive data.**

***Index Terms*—Keywords: DNA Cryptography, DNA Steganography, Hybrid Encryption, Huffman Coding.**

I. Introduction

In this new era of information technology, the security and confidentiality of information are becoming crucial. The need for confidential information transmission is increasing (such as online transactions). Therefore, we need a strong encryption model. For this purpose, researchers aim to find out a more robust system of data encryption. Moreover, some information needs to be transferred by hiding the encrypted data in some medium, such as images, audio, video, etc., to avoid intruders' attention because of security concerns. Therefore, the hybridization of encryption and data hiding is getting more research attention.

Researchers are concentrating on Deoxyribonucleic Acid (DNA) to develop a more robust encryption model because of its advantages such as ultra-high storage density, ultra-low energy consumption, and the potential of ultra-large-scale parallel computing to realize the cryptographic functions of information encryption and authentication [1]. Furthermore, the DNA sequence is only comprised of four symbols, which can be used to encrypt any data. The DNA sequence is interesting for data hiding. There are around 163 Million DNA sequences available in the public database. Hence, using DNA sequence as a medium significantly lowers system cracking probability and makes the system robust.

Finding a specific algorithm to encrypt and hide the data in such a way that it does not get intruders' attention is challenging; because no extra information is sent (blind technique), and it should have a decent cracking probability.

Most previous works focused on developing only the encryption method [2], and others focused on developing only the data hiding method [3]–[8], thereby providing only a single layer of protection. The proposed hybrid methods [2], [4], [9] used DNA cryptography and steganography, which use the Playfair cipher method and decrease capacity. The works in [3], [9], [10] have expanded the reference DNA size, which will get the attention of intruders, and the works in [5], [6], [9], [10] have not preserved the biological functionality of the DNA.

The works using the Playfair cipher method generate ambiguous bits and transfer them in the reference sequence because the bits are like a key to deciphering the text. Hence that affects the capacity of the method. Our aim is to develop a method that improves the data hiding capacity while it is blind, which means no other information along with the reference sequence needs to be sent by preserving the biological functionality of DNA. Thus, our method addresses the limitations of the previous works.

Our *objective* of this work is to propose a robust method of data encryption by using DNA sequences so that data can be transmitted securely without getting the attention of the intruder. The *contributions* of this work are: (i) proposing a hybrid method for data encryption based on DNA cryptography and steganography, (ii) performing security analysis of our proposed hybrid model, and (iii) comparing the proposed model with existing works to analyze the efficiency.

In our proposed method, we used the DNA cryptography concept and Huffman coding for data encryption; and used DNA as a medium to hide encrypted data with the substitution method. Results show that our method has a decent system cracking probability and capacity. Moreover, the payload is zero for our method. The comparison result shows that our method has overcome the limitations of the previous methods. Our proposed approach will help secure data transmission, especially in banking, e-commerce, authentication, and server-client secure communication sector.

The rest of the paper is organized as follows. In Section II, we have explained some of the terminologies used in this paper and briefly discussed some existing works with their advantages and disadvantages. In Section III, the proposed approach is explained along with its strength and limitations. Section IV presents the security analysis of the proposed method. In section V, we have compared our method with

some related works and discussed the outcome. Section VI presents the implementation details and results. Finally, we conclude the paper in Section VII.

II. Background and Literature Review

A. Terminology

Security in data communication is required when message transfer between sender and receiver is needed to be confidential. *Cryptography* is the process of achieving confidentiality in message transfer. Cryptography can be thought of as a process of secret writing in order to protect data or messages from various intruders' attacks. Secret writing is achieved through the process of transforming a message called plaintext into cipher text using a cryptographic algorithm. Security is concerned with protecting messages or data while transmitting over networks. DNA stands for Deoxyribonucleic Acid. It contains biological information about every living being. A DNA sequence is the sequence of Nucleotides. Nucleotides are Adenine(A), Guanine(G), Cytosine(C) and Thymine(T). DNA cryptography refers to converting plain text into a sequence of nucleotides based on some specified rules. DNA can be used to hide data. Hiding data in some medium, like image, audio, video, etc., is known as steganography. Hiding data in a DNA sequence is known as DNA steganography.

B. Related Works based on DNA Steganography

Shiu et al. [3] proposed three methods of hiding messages based on DNA and considered them the main methods. The first method is the insertion method which inserts a message bit in random places of a DNA reference sequence. It obviously expands the real sequence. The second method is based on complementary rules, which detect the longest complementary pair in a DNA sequence. They insert message bits before them, which also increases DNA size. The third one is based on the substitution method, where some DNA nucleotides are substituted based on secret message bits. Guo et al. [4] proposed a substitution data hiding technique using motif finding in DNA sequence. Repeated nucleotides in a DNA sequence are known as motif. They [4] find those motifs and substitute them with other nucleotides based on message bits. Yunus et al. [5] also proposed a substitution method based on motif finding in a DNA sequence. This method does not expand the DNA length but is not blind. Also, high modification may be required if the number of the motif is high in a DNA sequence. Hamed et al. [6] also proposed a complementary rule-based steganography method. The complementary rule is the rule that specifies the strand of DNA directly opposite to a specified sequence. It does not expand the length of the sequence and is blind. However, it does not preserve the biological function of the DNA. Mousa et al. [7] proposed a hiding technique that preserves the biological functionality of the DNA sequence using the reverse mapping method. The method is based on the substitution method and does not expand the length of the sequence. Vijaykumar et al. [8] proposed a DNA steganography model for image encryption. This method first converts the 3*3 matrix of pixels of the cover image to nucleotides. Then, nucleotides are converted to binary. Then, the XOR operation is performed between the message and the cover image.

C. Related Works based on DNA Cryptography

Namdev et al. [2] proposed a method that does a significant modification of the old approach of using DNA and Amino Acids based approach with Playfair Cipher by using the same approach with a different encryption algorithm, i.e., a Foursquare cipher to the core of the ciphering process. In this study, a binary form of data, such as plaintext messages or images, is transformed into sequences of DNA nucleotides. Subsequently, these nucleotides pass through a Foursquare encryption process based on amino acid structure. The fundamental idea behind this encryption process is to enforce other conventional cryptographic algorithms that proved to be broken and to open the door for applying the DNA and Amino Acids concepts to more conventional cryptographic algorithms to enhance their security features.

D. Hybrid Methods based on DNA Cryptography and Steganography

Mitras et al. [10] proposed a hybrid method based on the RSA algorithm and DNA encryption. They mapped the message bits into a DNA sequence and Amino acid. Then they used the insertion method to hide the encrypted data into an actual DNA sequence. The method is not blind. Taur et al. [9] proposed a hybrid method that uses a Playfair cipher based on DNA and Amino acid followed by data hiding using the insertion method. They used the 5*5 Playfair cipher method. The method is blind and has a low cracking probability. Yadav et al. [11] proposed a hybrid technique that uses images to hide a message and DNA to encrypt a message. They first convert the message into DNA and from DNA to cipher text. Then, they take a cover image and manipulate the pixel values according to cipher text. The used algorithm for hiding in the image is a well-known algorithm named KIMLA.

E. Gap analysis

There have been few works [3], [9], [10] which hide data based on insertion method and expand the DNA sequence's size. Hence it might draw the attention of the intruder to the transmission. Few other works [4], [5], [7] used substitution methods for data hiding, and there was no expansion of DNA sequence. However, they did not use any encryption method. Others [3], [8] did not used any encryption before steganography. Again [2] did not use steganography. Thus, they provide a single layer of protection.

In some works, [8], [11] image has been used for data hiding. Image resolution is changed when it is used as a cover image, and it may get the attention of the intruder. Also, a long message cannot be hidden in images of low resolutions. On the other hand, DNA sequences can hide the long message, and also cracking probability of DNA steganography is very low since there are 163 million DNA sequences available in the public database [12].

In [6] biological functionality of the DNA sequence is not preserved. Again [2], [4], [9] used the Playfair cipher method for encryption, generating ambiguity and ambiguity bits that must be passed to the receiver for decryption. The ambiguous bits often decrease the capacity of the algorithm.

F. Novelty of Our Work

The *novelty* of our approach is that our approach provides double-layer security incorporating encryption and steganography techniques, and the technique is blind. It does not expand the DNA size so that it does not get the intruder's attention and preserves the sequence's biological functionality. It also decreases cracking probability and increases hiding capacity compared to the methods that use the Playfair cipher for data encryption. In short, it eliminates all the disadvantages mentioned above in the related works and incorporates advantages into it.

III. Proposed Approach

Our proposed method has two phases:

- In the first phase (data encryption phase), we encode the plain text message into ASCII binary and then encode it with only DNA bases using 2-bit binary encoding. Then we apply Huffman coding scheme to further encode the encoded message with a variable length code for the bases.
- In the second phase (data hiding phase), we hide the encoded message into an actual DNA sequence using the 3:1 LS Base method. Here we are using a modified 3:1 LS Base method to hide both data and key, making it quite impossible to break.

Thus, our first contribution is to encode the plain text message with DNA Cryptography and Huffman Coding scheme. Moreover, our second contribution is innovatively hiding the encoded message and key into actual DNA sequences. The whole process is shown in Fig. 1 and described in the following subsections.

A. Phase I: Data Encryption

The data encryption process starts with converting the plain text message P containing letters, numbers, and special characters into ASCII binary. Then we take each two binary digits from left to right and convert two bits into one DNA base according to the 2-bit binary encoding rule. Table I shows the digital DNA base coding. In this way, we convert the plain text P into DNA bases M. Next; we calculate the frequencies of the DNA bases in the encoded message. Moreover, based on the frequency, we apply the Huffman coding rule to get a variable length code for each base. After that, we convert the M into MBIN using that variable length code. The algorithms of this encryption method and the Huffman Coding scheme are given below:

Algorithm: Encryption Procedure

Step 1: Convert the Plain text message P into ASCII Binary PBIN.

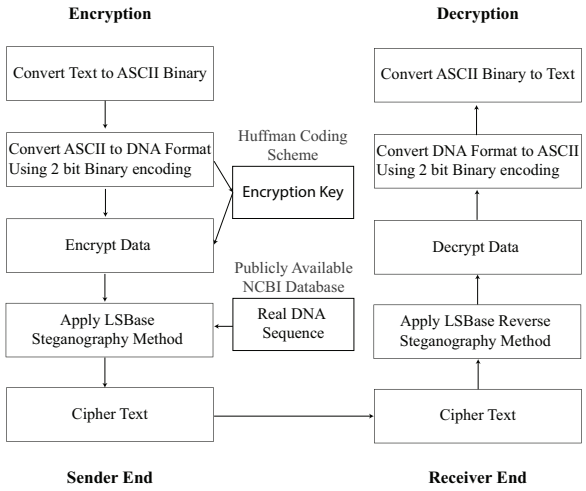

Fig. 1. Flowchart of the proposed method: encryption process is shown on the left side and decryption process on the right side.

TABLE I
DIGITAL DNA BASE CODING.

DNA Base	Binary Code
A	00
T	01
G	10
C	11

Step 2: Convert the PBIN into DNA Sequence M using 2-bit binary encoding.

Step 3: Derive the variable length Huffman code for each DNA Base, i.e., A, T, G, C.

Step 4: Convert the M into binary cipher text MBIN using variable length code from the Huffman scheme.

Algorithm: Huffman Coding Step 1: Obtain the frequency of each DNA base (A, T, C, G) from the DNA Encoded String M.

Step 2: Sort the bases in ascending order based on frequencies.

Step 3: Take two minimum frequencies and add them.

Step 4: Make the resultant frequency as root and the minimum frequencies as their left and right child.

Step 5: Repeat step 3-4 until a single tree is constructed.

Step 6: Starting from the root, label the left child with 0 and the right child with 1.

Step 7: Obtain binary code for A, T, G, C.

The process is explained using a flowchart in Fig. 2. Assume our text message is: "hello", which we want to send to our receiver securely. Hence, we want to encrypt it first with the above-mentioned way. Thus, we have P=hello. The ASCII binary of P, PBIN=01101000 01100101 01101100 01101100 01101111

We convert PBIN into M by substituting every two bits with its corresponding DNA base. Thus, we get M= TGGA TGTT TGCA TGCA TGCC. Next, we try to get the variable length code for A, T, G, and C with Huffman coding. Here the

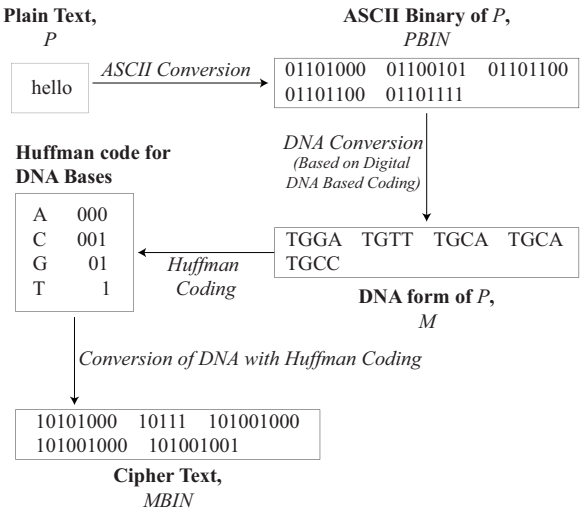

Fig. 2. Flowchart explaining the data encryption process with an example.

frequencies are A=3, T=7, G=6, and C=4. Hence, the sorted order of base is A - C - G - T. From the sorted order, we construct a tree-like Fig. 3. We get variable length code for A, T, G, and C, which is shown in Table II. Next, we convert M to MBIN according to Table II values. Therefore, we get MBIN=Cipher Text= 10101000 10111 101001000 101001000 101001001.

TABLE II
VARIABLE LENGTH CODE FOR DNA BASES.

DNA Base	Huffman Code
A	000
C	001
G	01
T	1

B. Phase II: Data Hiding

In this phase of our hybrid algorithm, we hide our cipher text which is the encrypted string of our plain text, into an actual DNA sequence. There are millions of natural DNA sequences available in the public database. We can get our DNA sequence from NCBI (National Center for Biotechnology Information) database. Then, we hide the cipher text into that actual DNA sequence using the 3:1 LS (Least Significant) base method. However, we have modified the method to increase capacity and security. The process is straightforward. First, from the left, we select each base from the actual DNA sequence placed into positions of multiple of 3, i.e., 3, 6,9, 12, 15, and so on. Moreover, we substitute them with another base based on the binary value of our cipher text from left to right. As 1 of the 3 bases in the actual DNA sequence contains cipher text, and that is the least significant among that 3. Hence it is called the 3:1 LS base method. If the base is a Purine base (A or G), then we substitute that with A to encode 0 and G to encode 1 from the cipher text. If the base is Pyrimidine base (T or C), then we substitute that with C

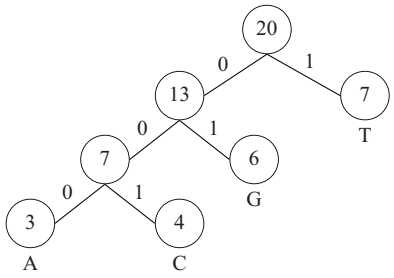

Fig. 3. Huffman Code generation for DNA bases based on frequency as described in Huffman Coding Algorithm.

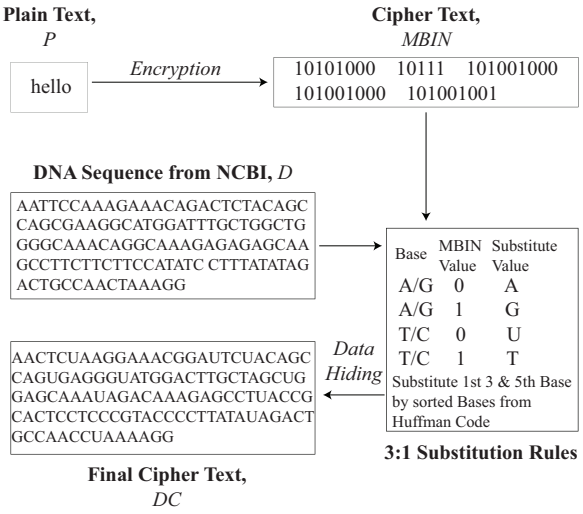

Fig. 4. Flowchart explaining the data hiding process with an example.

to encode 1 and U to encode 0 from cipher text. We encode this way until the length of the cipher text or the actual DNA sequence is reached. We hide our key (the variable length code from Huffman code) into the first 5 bases of the actual DNA sequence leaving the third base for cipher text encoding.

We get this opportunity because, in the case of the Huffman coding scheme, the variable length code is actually fixed, but it changes with the bases' frequencies. That means the variable length codes can only be 000, 001, 01, and 1 every time. However, which base represents which one depends on the frequency only. The least frequent base is encoded with 000, then 001, then 01, and the most frequent one in 1. Thus, if we send only the sorted order of that bases, the code can be obtained. We substitute the 1st, 2nd, 4th and 5th base with the sorted list of the bases based on the frequency. It also gives us another benefit in security which we will discuss in the security analysis section. Therefore, in this way, we get our cipher text with the key hidden in our actual DNA sequence. The process is explained using a flowchart in Fig. 4.

From the previous example, we saw that the length of the cipher text MBIN is 40 bits. To hide it in a DNA sequence, the length of that sequence needs to be at least 120bp. Let the DNA Sequence be D=AATTCCAAAGAAACAGACTCTACAGCCAGCGAAGG

CATGGATTTGCTGGCTGGGGCAAACAGGCAAAGAGA-
GAGCAAGCCTTCTTCTTCCATATC CTTTATATAGACT-
GCCAACTAAAGG. We check the 3rd base, which is T, and the 1st bit of our cipher text is 1. Thus, we substitute it with C. Then, we go to 6th base, which is C. The 2nd bit of cipher text is 0. Thus, we substitute it with U. And that way; it goes on. After all the cipher text gets hidden, we hide the key. The sorted list of the bases from the previous example was A - C - G - T. Hence, we substitute 1st base of the DNA sequence with A, 2nd one with C, 4th one with G and 5th one with T. The final DNA Sequence is
DC=AACTCUAAGGAAACGGAUTCUACAGCCAGUGA
GGGUATGGACTTGCTAGCUGGAGCAAAUA-
GACAAAGAGCCTUACCGCACTCCTCCCGTAC-
CCCTTATAUAGACTGCCAACCUAAAAGG

C. Data Extraction - Receiver Side

At the receiver end, the received message is just like an actual DNA sequence which contains a hidden encrypted message and a key to decrypt it. We need to do the opposite procedure to get back the actual message from the received message. First, we go through every 3 multiple bases and check it. If that is A or U, then the cipher bit was 0. If that is C or G, the cipher bit was 1. In that way, we first extract the cipher text. Now from the cipher text, we match which of the code is represented in places, i.e., 000, 001, 01, 1. Then, according to the Huffman coding scheme, we get the DNA encrypted message back. We get to know the Huffman representation of the bases from first 5 bases. Next, we convert the DNA encrypted message to binary using 2-bit binary encoding. That means we check each base of the DNA sequence and represent the digital code of that base. In this way, we get the ASCII representation of our actual message. Now just convert that to the character. That is the actual message we wanted to send securely. The whole process is shown in Fig. 1.

D. Strength of Our Approach

Our approach has several strong points described below:
1. This algorithm ensures three layers of protection against intruders.
a. Conversion of plain text to a DNA sequence.
b. Encoding it again with variable length coding.
c. Hiding it into an actual DNA sequence.
2. The process of hiding the key or Huffman code into a DNA sequence that we used made the fake DNA sequence unique, and difficult to find the actual DNA sequence from the database for the intruder.
3. On the receiver side decryption process is simpler and takes less effort, which will benefit this model in the server-client network as the client side machines are less powerful and hence less work for it in this model.
4. Though our model introduces data redundancy, it decreases cracking probability.

E. Limitations of Our Approach

Data redundancy is the main drawback of our approach. As we use the 3:1 LS base method, we need to take DNA of length 3 times longer than the cipher we got to hide. Still, the processing steps for hiding remain within the length of the cipher.

F. Cost Benefit Analysis

Though our model introduces redundancy, it makes the data sending highly secure. The machines today contain high processing power, and the internet connections are high speed. Hence, data redundancy is not the primary problem. From the security analysis below, we will see that the system cracking probability is very low; thus, it can be used to secure the transmission of highly secured data.

IV. Security Analysis

An intruder needs to know vital information to get the message back from the encrypted message we sent. They are: DNA reference, Encoding rule, and LSB substituted permutation. Analysis of the parameters is as follows.

A. DNA Reference Sequence

To decode the information, the intruder needs to guess the correct reference DNA so that he can analyze the changes in it to decode the message. This process is the toughest for our model as there are around 163 million DNA sequences in the public database. Again the first 6 bases of the sequence might be fully changed in our model. Therefore, the intruder needs to analyze the rest n-6 bases of a DNA sequence of length n to find the most related sequence. Therefore, the probability of making a correct guess of DNA reference is:

$$P(DNA_{Ref}) = \frac{1}{1.63 * 10^8 * (n-6)} \tag{1}$$

B. Binary Encoding Rule

Let us assume that the intruder knows the number of symbols used in the encoding process as it is a DNA sequence, so the number of symbols is 4. The Huffman code for the four symbols can be 000, 001, 01, and 1. Each of the four bases can get any of that code. The 2-bit binary encoding for DNA bases also creates 4 codes 00, 01, 10, and 11. Each of the bases can have any of these codes. Thus, the probability of guessing the right code each time P(BER) is:

$$P(BER) = \frac{1}{4! * 4!} \tag{2}$$

C. The Least Significant Base Substitution Rule

LS Base method is applied by substituting pyrimidine base by 'U' to encode the secret bit '0' or 'C' to encode '1'. However, it is also can encode '0' by C and '1' by U, and the same for the Purine base. Briefly, the '0' secret bit can be encoded by substituting the Pyrimidine base with 'U'or 'C'. If it is selected to be substituted by 'U', then 'C' will be used to substitute the Pyrimidine base to encode '1'. So the number

TABLE III
COMPARISON BETWEEN RELATED WORKS.

Comparison Criteria	P1: Enhanced Double Layer Security using RSA over DNA based Data Encryption [10]	P2: DNA Base Data Encryption and Hiding using Playfair and Insertion Techniques [9]	P3: Proposed Steganography Approach using DNA Properties [6]	P4: A New Data Hiding Scheme Based on DNA Sequence [5]	P5: The Proposed Method
Secret Text Type	Any Type of Data	Any Type of Data	Any Type of Data	Binary Data	Any Type of Data
Binary Coding Rule	2-Bit Binary Coding Rule	2-Bit Binary Coding Rule	2-Bit Binary Coding Rule	Binary Coding Rule Independent	2-Bit Binary Coding Rule
Encryption Type	Symmetric	Asymmetric	Not Applicable	Not Applicable	Symmetric
Encryption Algorithm	Encrypting secret data by mapping it to DNA and amino acids	5*5 Playfair cipher based on DNA and amino acids	No Encryption	No Encryption	DNA Based Huffman Coding Encryption
Data Hiding Algorithm	Insertion	Insertion	Complementary rules based hiding method, which is the rule that specifies the strand of DNA directly opposite a specified sequence	Substitution method using repeated nucleotides to hide the secret message bits	Substitution method using the least significant base of each codon in the DNA reference sequence
Blind/Not Blind	Not Blind	Blind	Not Blind	Not Blind	Blind
System Cracking Probability	$P(S) = 1/(1.63 * 10^8 * (n-1) * 24 * 2^(m-1) * 2^(s-1))$	$P(S) = 1/(1.63 * 10^8 * (n-1) * 24 * 2^(m-1) * 2^(s-1))$	$P(S) = 1/(1.63 * 10^8 * (n-1) * 24 * 24)$	$P(S) = 1/(1.63 * 10^8 * (n-1) * 24 * 6)$	$P(S) = 1/(1.63 * 10^8 * (n-6) * 4! * 4! * 4)$
Security Level	Double Layer	Double Layer	Single Layer	Single Layer	Double Layer
Modification Rate	High	High	Moderate	High	Low
Biological Functionality	Does not Preserve	Does not preserve	Does not preserve	Does not preserve	Preserves
Capacity	High	High	Moderate	Moderate	Moderate

of possibilities is 2*1 guesses, and the same will be done for the Purine base. Thus, the probability of making a successful guess for the substituted nucleotides N is:

$$P(N) = \frac{1}{4} \tag{3}$$

Using the proposed method, the probability of an attacker making a correct guess or the system cracking probability P(S) is:

$$P(S) = \frac{1}{1.63 * 10^8 * (n-6) * 4! * 4! * 4} \tag{4}$$

V. COMPARATIVE STUDY

In this section, we have compared our proposed model with some of the recent DNA-based steganography algorithms, and the result is shown in Table IV-B. For the comparison, we have chosen some crucial parameters [13], [14] as shown in Table IV-B. The first parameter of our consideration is the secret text type. That shows us if an algorithm hides all data formats comprising letters, symbols, or numbers. We can see that all the algorithms mentioned, excluding P4, support all types of data. P4 supports only binary data. The second parameter is the type of binary coding rule used in the conversion from the binary format of the message to DNA. All methods in the table use the 2-bit binary coding rule. The third parameter shows the type of encryption used in every algorithm that we mention in Table IV-B. Our proposed method uses symmetric key encryption. The fourth parameter shows if the method encrypts the secret data before hiding it or not. P1 encrypts the data by converting it to DNA then amino acids form. P2 encrypts the secret data using DNA and amino acids Playfair cipher. P3 and P4 hide the original format of the data without encryption; hence it increases the cracking probability and decreases processing overhead. Our proposed method uses Huffman coding scheme-based encryption followed by DNA encryption, providing extra protection against intruders.

The fifth parameter shows which data hiding algorithm is used. P1 and P2 use the insertion method to hide the secret message in the DNA sequence, increasing the DNA sequence's length. P3 hides the secret message using complementary rules. P4 and P5 hide the secret message by substituting DNA nucleotides based on the cipher text bits.

The sixth parameter shows us if the message can be retrieved without needing extra information other than the reference DNA sequence during data extraction. P2 and the proposed scheme P5 are blind algorithms. The seventh parameter is the cracking probability of each algorithm in the table. The eighth parameter shows the security level offered by each algorithm. Our proposed method provides a double layer of security as it encrypts the data before hiding it.

The ninth parameter shows us the modification rate. P1, P2, and P4 have high modification rates. The modification rate for P3 is moderate. Our proposed model has a low modification rate, as it only modifies the reference sequence for the length of the cipher text. The tenth parameter is the preservation of Biological functionality. It is also crucial to avoid intruders' attention. We can see that only our method preserves the biological functionality of reference DNA. This is because

we substitute Purine bases with Purine bases and pyrimidine bases with pyrimidine bases at the time of the steganographic process. The eleventh parameter shows the capacity, and we can see that only P1 and P2 have a high capacity, while our method also gives moderate capacity. Although we consider the method 3:1 LS base method, our method utilizes the maximum capacity that can be given in this method.

After considering all the aspects, we found that we have a decent cracking probability though it is not the best. P1 and P1, and P2 show the best cracking probability. However, they use the insertion method and hence increase the fake DNA sequence length and may get into the eye of the intruder. Also, P1 is not a blind method. Our method gives a double layer of security, making it better than P3 and P4. Again our method is the only one that preserves the biological functionality of the reference DNA sequence having a low modification rate. We can conclude that our proposed algorithm is decently strong compared to other algorithms represented here.

VI. Experimental Result

In this section, we have shown the performance of the proposed algorithm based on some of the predefined parameters that are used to evaluate the performance of an encryption algorithm in the literature. The proposed algorithm was tested on Intel(R) Core (TM) i5-8300H CPU @ 2.30 GHz personal computer with 8 GB RAM. The implementation is carried out with Jupyter Notebook version 6.1.4. We have experimented on a message kept in a file of size 5 kilobytes. The message contains letters, symbols, and numbers.

A. Used Dataset

The eight real DNA sequences in Table IV were used and they are publicly available from NCBI database [10]. In Table IV, the left-most column shows the locus of the DNA sequence, and the middle column shows the number of nucleotides in it. The right-most column shows the species definition for the locus.

TABLE IV
Specification of eight real DNA sequences used in our experiment.

Locus	Number of Nucleotides(bp)	Species Definition
AC166252	149,884	Mus musculus 6 BAC RP23-100G10
AC168901	191,456	Bos taurus clone CH240-1851
AC168907	194,226	Bos taurus clone CH240-19517
AC153526	200,117	Mus musculus 10 BAC RP23-383C2
AC168897	200,203	Bos taurus clone CH240-190B15
AC167221	204,481	Mus musculus 10 BAC RP23-3P24
AC168874	206,488	Bos taurus clone CH240-209N9
AC168908	218,028	Bos taurus clone CH240-195K23

B. Performance Metrics

We have used some parameters that are commonly used in evaluating the system's performance [2-11]. The first parameter is 'Capacity'. Capacity refers to the total length of the modified DNA sequence after hiding the encrypted message and key into it. The second parameter is the 'Payload'. Payload refers to the remaining length of the new DNA sequence after extracting the data from it. The third parameter is the 'bpn'. BPN stands for a bit per nucleotide, which is the number of bits hidden per nucleotide. It is the ratio of the total length of the message and key bit to the capacity in bits. The last two parameter shows the encryption and decryption time in seconds.

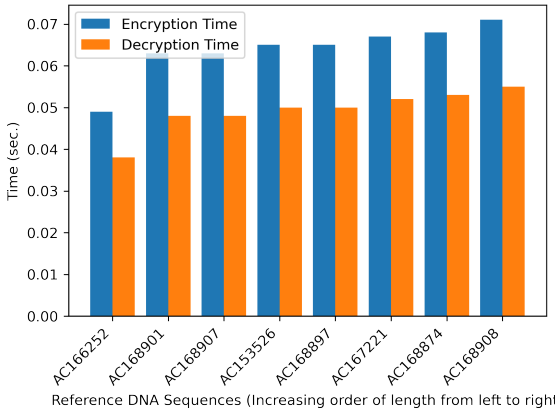

Fig. 5. The effect on encryption and decryption time based on the length of the reference DNA sequence. From left to right length of the DNA sequences increased. It shows that the encryption and decryption time increase as the length increases. Also, encryption time is more than decryption time.

C. Summary of Findings

Table V displays the experimental results in terms of capacity, payload, and bpn parameters to evaluate the system's performance. In the proposed algorithm, the capacity includes hiding the secret message and Huffman code (key) in the sequence. Payload is zero, meaning that the length of the fake DNA reference sequence is not expanded after hiding the message bits within it, which avoids drawing attention to it. This is achieved by hiding the secret data by substituting the nucleotides. Furthermore, bpn is within [2.5, 3.6], and the proposed scheme has a sufficient embedding capacity distributed on both the message and Huffman code(key), increasing the total number of nucleotides required for hiding the message bits only. Finally, the execution time to encrypt and hide 5KB data is calculated. Fig. 5 represents the relation found from Table V that the capacity and the execution time are affected by the length of the DNA sequence used, i.e. the DNA sequence's length is directly proportional to the execution time. As the DNA sequence's length increases, its hiding capacity increases, and consequently, the execution time and visa verse as shown in Fig. 5.

VII. Conclusion

In this paper, we have proposed a novel cryptographic technique combining DNA cryptography and steganography. The technique encrypts the data in its first stage and then hides the encrypted message into an actual DNA sequence. The encryption method uses DNA bases to encrypt the message,

TABLE V
EXPERIMENTAL RESULTS.

Locus	Capacity(bits)	Payload	bpn = (M+K)/C	Encryption Time(Sec)	Decryption Time (Sec)
AC166252	49965	0	3.6	0.049	0.038
AC168901	63822	0	2.8	0.063	0.048
AC168907	64746	0	2.8	0.063	0.048
AC153526	66709	0	2.7	0.065	0.050
AC168897	66738	0	2.7	0.065	0.050
AC167221	68284	0	2.6	0.067	0.052
AC168874	68833	0	2.6	0.068	0.053
AC168908	72680	0	2.5	0.071	0.055

followed by a variable length code generation and assignment for each DNA base using Huffman coding. The proposed method is blind as it does not need to send the actual reference DNA sequence with the fake one. Also, it does not expand the actual DNA sequence while keeping its biological functionality. From our security analysis and comparison with a number of promising methods from different literature, we found that our proposed method gives a decent level of security which is quite impossible to break without having full knowledge of the steps involved in particular encryption. The proposed method can be modified in our future work to increase its data hiding capabilities and security.

REFERENCES

[1] Y. Niu, K. Zhao, X. Zhang, and G. Cui, "Review on DNA cryptography," in *Bio-inspired Computing: Theories and Applications*. Singapore: Springer Singapore, 2020, pp. 134–148.

[2] S.Namdev and V. Gupta, "A DNA and amino-acids based implementation of four-square cipher," *Journal of Engineering Research and Applications*, vol. 6, pp. 90–96, January 2016.

[3] H. Shiu, K. Ng, J. Fang, R. lee, and C. Huang, "Data hiding methods based upon DNA sequences," *Journal of Information Sciences: an International Journal*, vol. 180, pp. 2196–2208, June 2010.

[4] C. Guo, C. Change, and Z. Wang, "A new data hiding scheme based on DNA sequence," *International Journal of Innovative Computing, Information and Control*, vol. 8, pp. 139–149, January 2014.

[5] Y. A. Yunus, S. Ab-Rahman, and J. Ibrahim, "Steganography: A review of information security research and development in muslim world," *American Journal of Engineering Research*, vol. 11, pp. 122–128, 2013.

[6] H. Ghada, M. Mohammed, E. S., and T. Fahmy, *DNA Based Steganography: Survey and Analysis for Parameters Optimization*. Springer International Publishing, 2016, pp. 47–89.

[7] H. Mousa, K. Moustafa, W. Abdel-Wahed, and M. Hadhoud, "Data hiding based on contrast mapping using DNA medium," *The International Arab Journal of Information Technology*, vol. 8, pp. 147–154, April 2011.

[8] P. Vijayakumar, V. Vijayalakshmi, and R. Rajashree, "Increased level of security using DNA steganography," *Int. J. Advanced Intelligence Paradigms*, vol. 10, pp. 74–82, January 2018.

[9] H. L. J. Taur, H. Lin and C. Tao, "Data hiding in DNA sequences based on table lookup substitution," *Journal of Innovative Computing, Information and Control*, vol. 8, pp. 6585–6598, October 2012.

[10] B. A. Mitras and A. K. Abo, "Proposed steganography approach using DNA properties," *International Journal of Information Technology and Business Management*, vol. 14, pp. 96–102, June 2013.

[11] V. Yadav and I. Gupta, "A hybrid approach to metamorphic cryptography using kimla and DNA concept," *Int. J. Computational Systems Engineering*, vol. 5, pp. 218–229, January 2019.

[12] R. E. Vinodhini, P. Malathi, and T. G. Kumar, "A survey on DNA and image steganography," in *4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 6-7 Jan, 2017, pp. 1–7.

[13] G. Hamed, M. Marey, S. A. El-Sayed, and M. F. Tolba, "Hybrid technique for steganography-based on DNA with n-bits binary coding rule," in *7th International Conference of Soft Computing and Pattern Recognition*, Fukuoka, Japan, November 2015.

[14] K. S. Sajisha and S. Mathew, "An encryption based on DNA cryptography and steganography," in *International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, April 2017, pp. 162–167.