A Review of Modern DNA-based Steganography Approaches

Omar Haitham Alhabeeb[1]
Department of Software Engineering, Mosul University
PhD. Candidate at Universiti Kebangsaan Malaysia
Mosul, Iraq

Fariza Fauzi[2], Rossilawati Sulaiman[3]
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
Selangor, Malaysia

***Abstract*—In the last two decades, the field of DNA-based steganography has emerged as a promising domain to provide security for sensitive information transmitted over an untrusted channel. DNA is strongly nominated by researchers in this field to exceed other data covering mediums like video, image, and text due to its structural characteristics. Features like enormous hiding capacity, high computational power, and the randomness of its building contents, all sustained to prove DNA supremacy. There are mainly three types of DNA-based algorithms. These are insertion, substitution, and complementary rule-based algorithms. In the last few years, a new generation of DNA-based steganography approaches has been proposed by researchers. These modern algorithms overpass the performance of the old ones either by exploiting a biological factor that exists in the DNA itself or by using a suitable technique available in another field of computer science like artificial intelligence, data structure, networking, etc. The main goal of this paper is to thoroughly analyze these modern DNA-based steganography approaches. This will be achieved by explaining their working mechanisms, stating their pros and cons, and proposing suggestions to improve these methods. Additionally, a biological background about DNA structure, the main security parameters, and classical concealing approaches will be illustrated to give a comprehensive picture of the field.**

***Keywords*—Information security in bioinformatics; deoxyribonucleic acid-based steganography; modern hiding approaches**

I. Introduction

Nowadays, the reliance on computer systems and the Internet has dramatically increased. The huge advancement in the technology of data storage and transmission has led to an increase in the information traffic between any two parties at an exponential rate. Many of these information are considered sensitive especially those belonging to the government, the army, or the big companies. It is quite risky to send such information over an untrusted channel [1]. The field of cybersecurity has the cumbersome task of protecting information from different types of threats and attacks. The attacks on shared information may cause it to be disrupted, corrupted, or stolen. Despite the availability of many information security techniques, cryptography and steganography seem to work perfectly together to achieve the mission of securely conveying information from source to destination. While cryptography aims to alter the message in such a way it becomes unreadable to a third party, steganography provides a concealing medium to the message [2]. In cryptography, a key is usually used to perform the task of data encryption and decryption. There are mainly two types of cryptosystems, these are symmetric key encryption and asymmetric key encryption cryptosystems. The encryption and decryption processes are applied in the symmetric key method via a secret key shared between the sender and the receiver. The length of this key is relatively short which aids in completing the decryption stage quickly. Symmetric key encryption can be categorized into stream cipher or block cipher. In a stream cipher, every character in the message will be individually encrypted. While in block cipher, many bits are assembled in a single unit. Then, this unit will be encrypted. Some of the famous and dominant encryption algorithms that use symmetric key encryption are Digital Encryption Standard (DES), AES, RC4, and IDEA [3]. In the case of asymmetric key encryption, two keys are used to handle the encryption and the decryption processes. A public key is used to encrypt the message by the sender and a private key is used to decrypt the message by the receiver. This will eliminate the risk of losing a pre-shared key as in the previous technique. Although this technique provides high-security measures, a third party must be trusted as a key manager. An instance of asymmetric key encryption is the RSA encryption algorithm [4]. In the field of steganography, mediums like text [5], image [6], audio [7], and video [8,9] are used by researchers as containers to hide a message inside them. Concurrently, technologies of data storage and transmission withstand rapid development and improvements. Also, there are a noticeable diversity of threats and attacking methods that appear every year. All the covering mediums mentioned above struggle to cope with the increasing size of information as well as meet the demanded security measures [10]. An urgent need has arisen for a concealing medium capable of holding a large amount of data without corrupting or degrading the quality of this medium. Consequently, deoxyribonucleic acid (DNA) is proposed as the ultimate concealing medium that avoids or mitigates the drawbacks of other mediums [11]. DNA's most important feature is the huge capacity it has. Around 215 petabytes of data can be stored in one gram of DNA. Another useful feature is the randomness of the building blocks forming the DNA. Besides that, low power is required when dealing with DNA computing which leads to fast execution. For all the reasons mentioned above, many DNA steganography algorithms have been suggested since the beginning of the twenty-first century. As shown in Fig. 1, three components are combined to get the fake DNA sequence. These are the covering medium (which is the DNA sequence in our case), the message, and the secret

key [12]. An additional component can be added to the formula by firstly encrypting the message before hiding it. This will decrease the penetration probability and creates a complete crypto-stego system.

Fig. 1. DNA Steganography Process.

Generally, there are three classical techniques used in DNA-based steganography to embed the hidden message in the cover medium. These are insertion, substitution, and complementary rules-based algorithms. Each one of these techniques has its way of implementation, benefits, and limitations. Even hybrid algorithms of these techniques have been proposed to improve upon them [60]. In the attempt of improving the performance of currently used DNA steganography algorithms, a new trend has evolved. This trend relies on the concept of utilizing one of the DNA biological features and/or merging a technique existing in one of the fields of computer science with a DNA-based steganography algorithm. Doing so showed promising results in terms of overcoming or at least alleviating the issues and gaps that existed in the field of DNA computing. But with the advent of new solutions, new issues have also arisen. This motivates the authors of this manuscript to address these issues and suggest different methods to solve them. Therefore, the main purpose of this paper is to highlight the strong points and drawbacks of recently proposed DNA-base steganography algorithms. It also aims to be a reference for any researcher who wants to develop his/her novel hiding technique. This is accomplished by achieving four objectives. The first objective is exploring the field of genetics, dissecting the structure of the DNA sequence, and informing the reader on the constantly used biological terms in this area without delving into unnecessary clinical concepts. Secondly is enumerating and elaborating the security parameters in this field. Then, classical techniques used to hide a secret message in a DNA sequence are briefly explained. Last but not least is conducting an in-depth analysis of modern approaches.

The rest of the paper is organized as follows: section two includes a biological background about DNA. Section three enumerates the parameters of security measures. Section four contains a concise description of classical DNA-based steganography algorithms. Most importantly section five presents a critical analysis of some modern DNA-based steganography algorithms. Finally, section six discusses the primary open issues of DNA-based steganography, suggests some solutions, and concludes the paper.

II. Biological Background

To work in the field of DNA computing, prior knowledge of biology especially in the field of genetics is required. DNA preserves the genetic information that denotes the physical shape, behavior, and functions of all living organisms. DNA is constructed from two lengthy strands twisted on each other. These two strands are attached via units called nucleotides to give DNA the shape of a helix ladder. Nucleotides are considered the building blocks of DNA and can be one of four main types. These types are adenine (A), guanine (G), cytosine (C), and thymine (T). The two strands are linked to each other by pairs of nucleotides. This linkage is not haphazardly formed. A is always paired with T via double hydrogen bonds, while C is always paired with G via triple hydrogen bonds [13]. Fig. 2 depicts the structure of the DNA.

Every three consecutive nucleotides represent a unit called a codon. Codons represent either one of the 20 possible amino acids or a stop signal. Since each nucleotide in the three locations of the codon has one of four possible values (A, C, G, and T), there are 43 or 64 different types of codons. Amino acids are represented with 61 types, and 3 dedicated for a stop signal. Every amino acid can be either represented with a single codon up to six types of codons. For example, tryptophan represented with (TGG), glutamic acid represented with (GAA, GAG), and threonine represented with (ACT, ACC, ACA, ACG). The case when an amino acid is represented with more than one codon is called ambiguity. Essential proteins for the human body are formulated from a long chain of amino acids [14]. A complete genetic code of DNA is presented in Table I. A distinctive biological feature in DNA is mutations. These mutations refer to changes that occur in the DNA sequence which modify the contents of one or more nucleotides. One of the reasons that cause this change is due to the errors in the DNA replication or recombination processes [13]. In general, there are three types of mutations:

1) Base substitution: occurs when the content of one or more bases is replaced with new content. The replacement process can be either transitional or transversional. Transitional replacement appears when a purine is replaced with another purine or when pyrimidine is replaced with another pyrimidine.

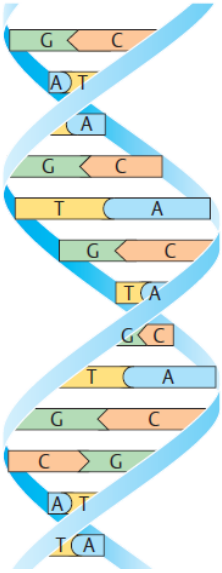

Fig. 2. DNA Double Helix.

TABLE I. THE GENETIC CODE [13]

First Position		Second position								Third position
		T		C		A		G		
		Code	Amino	Code	Amino	Code	Amino	Code	Amino	
T	T	TTT	phe	TCT	ser	TAT	tyr	TGT	cys	T
		TTC		TCC		TAC		TGC		C
		TTA	leu	TCA		TAA	STOP	TGA	STOP	A
		TTG		TCG		TAG	STOP	TGG	trp	G
	C	CTT		CCT	pro	CAT	his	CGT	arg	T
		CTC		CCC		CAC		CGC		C
		CTA		CCA		CAA	gln	CGA		A
		CTG		CCG		CAG		CGG		G
	A	ATT	ile	ACT	thr	AAT	asn	AGT	ser	T
		ATC		ACC		AAC		AGC		C
		ATA		ACA		AAA	lys	AGA	arg	A
		ATG	met	ACG		AAG		AGG		G
	G	GTT	val	GCT	ala	GAT	asp	GGT	gly	T
		GTC		GCC		GAC		GGC		C
		GTA		GCA		GAA	glu	GGA		A
		GTG		GCG		GAG		GGG		G

On the other hand, transversional replacement appears when a purine is replaced with a pyrimidine and vice versa. Base substitution mutations lead to different kinds of situations. For instance, when replacement occurs in the third location of a codon, there is a high probability that the amino acid is still the same. This kind of mutation is called a silent mutation. Another case arises when the replacement of bases converts an amino acid to a new one. This is called missence mutation and can be either conservative or non-conservative. In conservative mutation, the new amino acid has a similar structure and working mechanism compared to the old one. Consequently, the functionality of the protein is maintained regardless of the change in one of constructing amino acids. In non-conservative mutations, the new amino acid differs greatly compared with the old one. This will lead to altering the functionality of the whole protein. The last case of base substitution mutations appears when the replacement leads to one of the three-stop signals. This is called a nonsense mutation, which will terminate the sequence of amino acids building the protein and spoil its functionality.

2) Base deletion: occurs when one or more bases are omitted from the DNA sequence. Deleting one or more bases will cause a frameshift to the whole DNA sequence and make it non-functional. Deleting three or more bases may or may not affect the function of the protein.

3) Base insertion: occurs when one or more bases are added to the DNA sequence. Similar to the deletion case, this may have a negative effect on the biological function of the DNA sequence.

Studying and understanding the different types of mutations is essential to utilize them for data hiding purposes. Silent and conservative mutations types can be used to preserve functionality and produce a blind DNA-based steganography algorithm. Finally, researchers in the field of DNA computing need DNA sequences for testing the performance of their proposed algorithms and comparing results. Fortunately, there are over 163 million DNA sequences that can be freely downloaded from different websites. Two of the most useful websites are the National Center for Biotechnology Information (NCBI) and the European Bioinformatics Institute (EBI) [15, 16].

III. SECURITY ANALYSIS

This section illustrates some of the most important methods and crucial factors used in the development process of the DNA-based steganography algorithms. It also includes the parameters used to evaluate the performance of these algorithms.

1) Data encoding: to hide a secret message of any formatting type in a DNA sequence, elements of this message need to be encoded into genetic letters. One method to achieve this task is to use a lookup table to encode every character of the message to several genetic letters [17]. This technique suffers from two main defects. Firstly, the table is fairly static and fixed. Once it is revealed by an intruder, the whole steganography algorithm is compromised. The second issue is lookup tables used by many researchers to encode only alphabetic and numerical characters [18]. Special characters are ignored due to insufficient codons number. Another data encoding method is using a binary coding rule (BCR) table as shown in Table II.

In this case, the characters of the message are converted to binary form based on their ASCII code. Then, every two bits are assigned to a genetic letter [19]. An improved N-bits BCR has been suggested to increase characters encoding probability [20].

TABLE II. 2-BITS BINARY CODING RULE

DNA nucleotide	Binary form
A	00
C	01
G	10
T	11

2) Hiding capacity: denotes the total amount of data DNA sequence can tolerate. It is usually measured by the number of bits per nucleotide (BPN). A higher data capacity is obtained with a higher (BPN) value [23].

3) Security key: a very critical element in any crypto-stego system. Different factors have a direct effect on the used key or keys. For example, the purpose of using the key whether it is for encrypting or concealing the message, the type of the key whether it is a symmetric or asymmetric key, whether the key is randomly generated or created with a specific method, the size of the key, and the possibility of encrypting the key and merging it with the hidden data [22].

4) Data encryption: before concealing the secret message in the covering medium, it is preferred to encrypt it to add another layer of security to the algorithm. Many ciphering techniques have been used in the field of DNA-based steganography like DES, RSA, Vigenere, AES, and Playfair. Selecting and implementing the suitable encryption method is crucial since it has a direct effect on the other security parameters like the used key, hiding capacity, and cracking probability [21].

5) Hiding method: denotes the way characters of the secret message are embedded in the reference DNA sequence. Authors in [19] established three possible methods adopted by the majority of researchers afterward. Chunks of the secret message can be either inserted in different locations of the DNA sequence, added before the longest complimentary rule, or substituted with characters from the original sequence. These methods are vastly elaborated in the next section.

6) Double layer embedding: In many literature manuscripts, researchers suggest hiding the secret message in two mediums instead of one to offer higher security. Besides DNA sequence, a secret message can be concealed in an image [29], audio [30], or microdot [31].

7) Cracking probability: A measurement of the success probability of a brute force attack to break the proposed security algorithm. Some of the elements that have a direct effect on the cracking probability are the total number of possibly used DNA sequences, the number and the size of segments of the DNA reference sequence and the secret message, and the way of using a binary coding rule with a lookup table [2].

8) Payload: is the amount of extra data added to the DNA sequence due to the implementation of the DNA steganography algorithm. The best scenario occurs when the payload equals zero [24].

9) Modification rate: denotes the ratio of change in the fake DNA sequence compared to the original one. A high modification rate may spoil the protein's operations in the DNA sequence [24].

10)Random variable: In some of the developed steganography algorithms, researchers add one or more random variables in the data encryption and hiding processes. The reason behind that is making the task of cracking the algorithm even harder for an intruder. Some examples of random variables are using a random number of segments generation for insertion method [19, 25], random locations generation [26], random key generation [27], and random sequence generator [28].

11)Blindness: the algorithm is considered blind when there is no need to send the original DNA sequence used by the sender to the receiver. Generally, less amount of information used in the transmission process will improve the effectiveness of the applied method [32].

12)Preserving functionality: Codons are arranged in the DNA sequence in a specific order to form proteins. Every protein is generated to perform a unique biological function. One of the challenging goals in the field of DNA-based steganography is preserving this function after implementing the proposed data hiding algorithm [14].

IV. DNA-BASED STEGANOGRAPHY TECHNIQUES

From the beginning of the new millennium, many DNA-based steganography algorithms have been proposed. Each one of them follows a specific technique for data embedding in the DNA sequence. This section illustrates these classical approaches, describes their issues, and explains how these issues affect the development of recently proposed DNA-based steganography algorithms.

A. Insertion-Based Techniques

The Insertion-based technique is one of the first methods to hide a message in a DNA sequence. It is performed by inserting the data of the embedded message in one or more different locations of the DNA reference sequence. In [19], both the DNA sequence and the embedded message are divided into a random number of segments. Then, each segment of the message will be inserted before a segment of the medium. It is difficult for an intruder to deduce the number of segments or packets per message and the amount of data per packet [27]. An additional layer of protection can be added by encrypting the secret message with a solid encryption algorithm. For example, data encrypted with modified Playfair [25], AES-128 [31], RSA [33], or RC4 encryption algorithms [34] before data embedding stage. In [35], two images are concealed in another image without distortion based on grayscale and Least Significant Bit (LSB) insertion of the DNA form of the secret information. An improved insertion technique is proposed by [36] with the use of two keys. The first key is XORed with every 8 bits of the secret message, while the other key is responsible for dividing the DNA sequence into segments. The binary bits of the cipher are inserted at the beginning of each segment. This technique has a strong cracking probability and high security but the payload is

not zero. A powerful algorithm with complex data encryption and embedding techniques is presented in [32]. In the encryption phase, a Playfair cipher with a randomly shuffled 8*8 matrix is used to gain an additional layer of security via codons replacement and circular rotation of rows and columns processes. In the embedding phase, both the message and the cover DNA sequence are spliced into a random number of segments. Then, these segments are concatenated to generate the fake DNA sequence. The main drawback of the insertion-based technique is the increasing size of the DNA reference sequence. This is a clear indication that data has been added to the carrier medium and will attract the attention of attackers. Also, algorithms adopt insertion technique are usually not blind. On the positive side, insertion algorithms have a low cracking probability. Furthermore, one or multiple random variables are used in the encryption or embedding stages. This will make breaking the algorithm even harder for intruders.

B. Substitution-Based Techniques

In substitution-based technique, nucleotides from the DNA sequence are replaced with secret message characters. The replacement method can be applied using a lookup table [17] or at specific locations like the LSBs [37]. In [38], a histogram of frequently appeared values in the selected DNA reference is created. Then, the message is hidden in locations marks with zero. A novel technique based on Chebyshev chaotic maps is developed by [39]. The plain text is encoded into a DNA sequence and encrypted to another sequence via a Chebyshev map. The result is circularly shifted for a finite number of times and embedded through character by character substitution in a word document. Excessive random substitution of bases inside the DNA sequence will ruin the function of the protein. The approach in [40] took advantage of the ambiguity feature to solve this issue. Only the Least significant bases (LSBase) locations of each codon are used as hiding locations of the secret message. This technique is blind and minimized the modification rate by a third. Enhanced algorithms are offered in [20] and [41] based on a 4-bits coding rule with data encryption using Playfair and AES. Another substitution-based technique is suggested by [42]. A codon dictionary table is manually generated. Then, every 6 bits of the message will be replaced with a codon. These codons are embedded in different intervals of the DNA reference. Several elements were used in [43] to provide dual cover steganography techniques based on an image and DNA. These elements are a 2D logistic map, three secret keys, and encrypt a secret message using RC4 and hiding it in LSB locations. Although offering high-security measures with double hiding layers, this technique demands much data for the hiding and extraction processes. Another double-layered steganography technique was proposed in [30]. After encrypting the message using RC4 and hiding it in the DNA sequence, a randomized LSB replacement method is used to embed the result in an audio file. Preserving the functionality of the DNA sequence can be achieved by embedding the secret message in non-coding regions of the sequence. This was accomplished in [44] by first encrypting the message using XOR and pseudo-random bit generator (PRBG) sequence. Then, the encoded message is segmented into pieces and replaced with nucleotides in sectors from non-coding regions. Another advantage of this technique is its ability to detect and recover deleted or changed nucleotides affected by mutations. Also, it has a decent capacity ranged from 1.2 to 2 BPN. An enhanced 4*4 Playfair cipher is developed in [21] instead of the 5*5 usually used grid. After encrypting the plain text with it, the hiding technique developed in [45] is used to improve hiding capacity by 25%. A double-layered hiding algorithm based on a color image as a cover is presented in [29]. The secret data is divided into three parts using the XOR operator, and each one of them is concealed in the color of the cover image. This is a highly reliable approach though it is time-consuming during the extraction phase. Two DNA strands are used in [46] to increase hiding capacity. A substitution table is used for the embedding purpose and the message is sent to the receiver via a microdot. In [18], a combination of image and DNA-based steganography is applied. A secret image is embedded in another image – both with the same pixel size – by converting their pixel values to DNA triplet based on a lookup table. Then, these triplets are converted to binary form and XORed to get the stego image. This method offers a higher security level than other techniques like LSB but with a lower hiding capacity. The prominent benefit of the substitution-based technique is implementing the embedding process without any expansion in the size of the fake DNA sequence. On the other hand, a high modification rate applied to the DNA reference sequence will ruin its biological functionality. Besides that, by restricting the replacement locations to a limited number - as in the LSBs-based methods -, will highly decrease the data hiding capacity. Trying to increase hiding capacity may lead to lowering the performance of other parameters like modification rate or cracking probability [46].

C. Complementary Rule-based Techniques

In this method, a complementary rule is initially considered and the data is inserted before the longest complimentary substring in the reference DNA sequence. The complementary rule is set in such a way where $x \neq c(x) \neq c(c(x)) \neq c(c(c(x)))$ and x is one of the four possible genetic letters [47]. For instance, if the following complementary rule is applied: ((AC) (CG) (GT) (TA)), the complementary string of AATGC will be CCATG [19]. In [48], each pair of encrypted secret message will be assigned to a matching index in the DNA reference sequence. Then, a list of these indices will be sent to the receiver. The hiding process can be implemented using a random generator function based on the size of the secret message [26]. Another method encoded text to DNA form and hide the result in grids of an image based on LSB and MSB [49]. A similar technique is suggested by [50] where the input message is encoded to DNA form and embedded in a randomly chosen frame of a video. A mathematical approach of data encryption and hiding is proposed in [28]. Two secret keys are generated using Elliptic Curve Cryptography (ECC) and Gaussian Kernal Function (GKF). The secret message is encrypted using these two keys, followed by DES. Characters of the encrypted message are embedded in locations from the new DNA sequence. While this technique is the simplest compared to other techniques, it is not blind and increases the message size. In some cases when a random DNA sequence is used, it is required to send the complementary rule to the receiver [51]. Security mainly depends on the selected DNA reference [52]. Also, it is not

always feasible to find the DNA sequence of all the matching elements with the secret message.

D. Hybrid Techniques

Many endeavors aimed to combine the advantages of the three techniques in the previous sections and nullify or at least mitigate their weaknesses by proposing hybridization between them. For example, authors in [45] used a substitution method with a complementary rule. A multi-level secured method that adopted a similar approach has been suggested in [53]. Data are firstly using Playfair and concealed in a randomly generated DNA sequence. Then, the DNA sequence is embedded in an audio file. An imperceptible double-layered algorithm based on image and DNA is proposed in [54]. Using the only component of the cover image – like blue color – to hide the DNA message is the main weakness in this algorithm. Generally, these algorithms hide the message with zero expansion and payload. Still, like their predecessors, they paid no attention to preserving the biological features of the reference DNA sequence. Also, they are un-blind algorithms. Authors in [19] employed a data hiding algorithm based on a combination of insertion and complementary rule methods. This algorithm is un-blind, expands the size of the DNA sequence, and ignores its biological functionality. In [55] the secret message is firstly encrypted with an improved Playfair cipher. After that, the encoded message is concealed via a novel substitution method in a cover DNA sequence and the result is inserted in another sequence. An improved version of this method with enhanced Generic complementary base substitution (GCBS) is recently presented in [61]. Using a DNA-based XOR cipher with a randomly generated key formulated from the cover medium is suggested. This proves to offer better results in terms of lower cracking probability and higher BPN.

V. Analysis of Modern Approaches

Classical or traditional methods of implementing DNA steganography discussed in the previous section used the traditional protection methods in the field of data security to achieve the desired task. In the last few years, a new generation of algorithms has been evolved. The new trend aims either to exploit the biological attributes of DNA or borrow a suitable technique from another field of computer science. The reason behind that is offering better solutions and performance compared to the old generation methods. In this section, a critical analysis will be initiated to the recently suggested techniques by describing their working mechanism, stating their pros and cons, and points to the possible ways to improve upon them.

A. Exploiting Biological Features

One of the most interesting features of DNA is the condition called single nucleotide polymorphism (SNP). In this case, a particular nucleotide in the genome sequence differs between members of the same species. As shown in Fig. 3, the two DNA sequences have different content in a specific position.

Researchers keenly noticed that regions including SNPs have the potential to hide a secret message. Attackers cannot distinguish between changes that occur because of data hiding and changes that occur because of SNPs. In [56], the author proposed a DNA steganography algorithm based on hiding a secret message in SNPs locations. Firstly, SNPs positions in the selected DNA sequence are assigned. Then, every character in the secret message is converted to three letters based on the DNA encoding table. For example, the letter (H) is encrypted to (TAC) and stored in three consecutive SNPs positions as shown in Fig. 4.

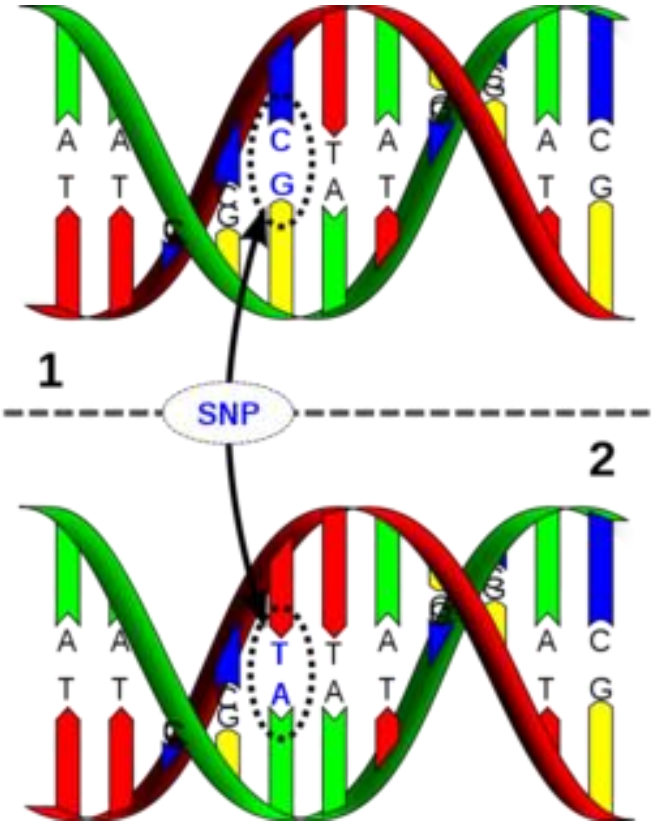

Fig. 3. Single Nucleotide Polymorphism [56].

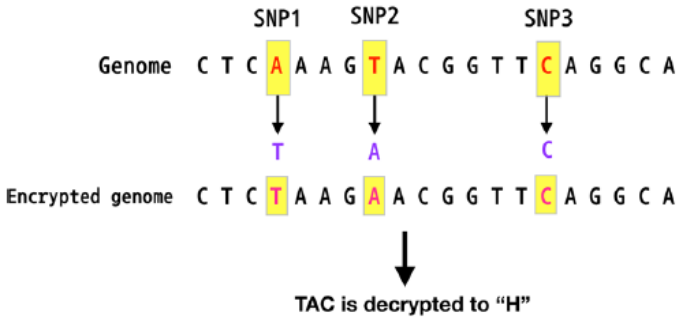

Fig. 4. Encrypting and Hiding a Character [56].

One of the limitations of the proposed algorithm noticed in the DNA encoding table, it has only alphabets and numbers. While SNPs offer good hiding locations in the encrypted message, they suffer from an obvious defect. The appearance probability of SNPs in a single DNA sequence is only 6%. Hence, the hiding capacity of the sequence is severely decreased. A suitable data compression technique can be used to increase the number of encrypted bits per location.

Another biological factor used in the process of data hiding is mutations. Although some types of mutations have negative effects like corrupting or terminating the protein, other types do not harm the protein. Silent and conservative mutations change nucleotides or amino acids in the protein and maintain its functionality. The approach in [23] took advantage of these types of mutations to develop a crypto-stego system with high hiding capacity and excellent security measures. This algorithm has two main stages on both sender and receiver sides. Firstly, the secret message is encrypted using an enhanced 8*8 Playfair cipher. This new encryption technique overcomes the weakness that existed in the 5*5 Playfair cipher where only alphabets are encrypted. Using Playfair cipher is justified since no extra information is required by the receiver in the decryption process besides lowering the cracking probability. In the second phase, LSBase is used to hide bits of the secret message. The challenge is to hide two bits per nucleotide, where every amino acid will handle four possible substitutions. Via silent mutation, this is possible in all amino acids that can be represented with four codons or more. For instance, Alanine has four codons: GCA, GCC, GCG, and CGT. The LSBase of these four codons can hide two bits – 00, 01, 10, and 11 – without issue. Unfortunately, not all amino acids can satisfy the rule of being represented with four codons or more. For example, Phenylalanine is an amino acid that can be represented with only two codons, TTT and TTC. These two codons can satisfy only half of the four possible values. To solve these issues, the authors suggested using the feature of conservative mutation. Here, an amino acid will replace another that shares a similar structure and functions. Amino acids that have less than four codons will be used interchangeably without disturbing the functionality of the protein. In the example mentioned above, Tyrosine will be used when Phenylalanine fails to satisfy the embedding requirements. One of the main contributions of the proposed algorithm is its ability to exploit concealing data in LSBase locations efficiently with a high hiding capacity of 2 BPN. Another important contribution is hiding the message in a non-sequential and non-order pattern along the DNA sequence. The cracking probability is low due to the various factors required to get the secret message. The intruder must not only know which DNA reference or BCR are used, but he/she must also know how the Playfair matrix is constructed, figure out the data hiding locations, and the adopted substitution rule. The drawback of the suggested algorithm is the need to send the random number of seeds used to generate BCR to the receiver. Also, using a fixed substitution table where the same unsatisfied amino acids always replace each other for every transition session is considered a weak point.

Maintaining the biological features of DNA after embedding the encrypted data in it is a really difficult task to achieve. Any modification or payload applied to the DNA sequence may lead to losing the functionality of this sequence. Although it seems leaving the DNA sequence intact is the only way to succeed in preserving its original signature, there are many other interesting methods to do that. In [57], the main objective is preserving the biological functionality of the DNA sequence after the embedding process. This is accomplished by exploiting a DNA feature, ambiguity, mentioned in section two. Every amino acid is composed of three nucleotides or a codon. Ambiguity arises when more than one codon represents an amino acid. As demonstrated in Table III, there are 20 different types of amino acids. Every amino acid can be represented with one to six codons. The table also includes the number of ambiguities for each amino acid and the number of required bits to represent them. Before initiating the data hiding operation, a preprocessing is deployed. Based on the content of Table III, the whole carrier DNA sequence is converted to amino acid codons. Every bit in the secret key is linked to its corresponding character of the DNA sequence in a specific way. Ones denote amino acids locations where hiding characters is possible, while zeros denote to do not care amino acid locations. In the data hiding process, the codon representing the secret character will substitute a codon in the carrier medium. This does not affect the DNA sequence functionality since both codons belong to the same amino acid. Consequently, the purpose of this work of applying the steganography process without disturbing the biological features of the medium is achieved. Also, the proposed algorithm provides low penetration probability versus different types of attacks like stego-only attack, known cover attack, and known message attack. The weaknesses of this algorithm are adding a new bit for hiding data in some cases which increases the payload. Besides that, the performance of this algorithm depends on the content of the secret message. Every amino acid used in the hiding process has its own effect on parameters like hiding capacity, BPN, and payload. This drawback can be alleviated by using only amino acids that guarantee the best performance.

B. The Field of Networking

For the mutation/modification detection technique, the author in [56] borrowed a concept used in the field of networking to detect errors in data transmission. A block checksum method is implemented by assigning a number to each nucleotide in the encrypted sequence where (A = 0 / T = 1 / G = 2 / C = 3). Then, the summation of nine consecutive nucleotides is divided on 4. The remainder of the division operation will be converted to the equivalent DNA value and attached to the end of the sequence as the tenth location. This operation is repeated for all components of the encrypted message. As shown in Fig. 5, three cases are presented.

TABLE III. NUMBER OF CODONS PER AMINO ACID [57]

Amino / Ambiguity			A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A 1	0	00 0	GCU	UAA	UGU	GAU	GAA	UUU	GGU	CAU	AUU	AAA	UUA	AUG	AAU	CCU	CAA	CGU	UCU	ACU	GUU	UGG	UAU
A 2	1	00 1	GCC	UAG	UGC	GAC	GAG	UUC	GGC	CAC	AUC	AAG	UUG		AAC	CCC	CAG	CGC	UCC	ACC	GUC		UAC
A 3	2	01 0	GCA	UGA					GGA		AUA		CUU			CCA		CGA	UCA	ACA	GUA		
A 4	3	01 1	GCG						GGG				CUC			CCG		CGG	UCG	ACG	GUG		
A 5	4	10 0											CUA					AGA	AGU				
A 6	5	10 1											CUG					AGG	AGC				
Number of ambiguities (X)			4	3	2	2	2	2	4	2	3	2	6	1	2	4	2	6	6	4	4	1	2
Number of bits (Y)			2	2	1	1	1	1	2	1	2	1	3	1	1	2	1	3	3	2	2	1	1

In the first case, no mutation has appeared in the encrypted sequence. This is confirmed by matching the last nucleotide with the result of the checksum method. In the second case, a mutation occurs in the first location of the sequence. This is immediately noticed since the last nucleotide in the sequence differs from the resulted sum check calculation. While the proposed mutation detection method works perfectly in the first and second cases, it sometimes flawed as presented in case three. Here, both that attached nucleotide and the result of the checksum are the same. Although the receiver thinks that the sequence has been correctly delivered, it is actually modified in different locations. The proposed mutation detection technique can work properly in the case of single location mutations. Alas, it may stumble in the case of other types of mutations cases with multiple locations frameshifting or swapped nucleotides. An enhanced modification detection technique is required to cover cases like transmission errors and deliberate change by an attacker. This can be accomplished by using two DNA sequences, one for the data hiding and the other for the modification detection. All SNPs locations of the first will be used solely for data hiding while locations in the second sequence are only used to hold data verification values. This will increase the number of locations dedicated for concealing purposes which improve hiding capacity and will make it easier to detect different kinds of changes in the carrier sequence.

C. The Field of Data Structure

Another field in computer science correlated with the field of DNA steganography to increase hiding capacity is data structure. In [58], a framework for hiding based on a balanced tree is suggested. This is accomplished by converting a randomly selected DNA sequence into a balanced tree where every node holds a single nucleotide. The height of the tree depends on the size of the message to be sent. Even levels have at most two children while odd levels have at most three. The traversing process is done by Depth First Search (DPS) approach. The encryption process starts by converting every character in the message to four nucleotides.

Then, characters in the leaf nodes will be replaced in reverse order with letters of the encrypted message. The last stage is obtaining the fake DNA sequence by assembling the letters based on the references of the nodes in the tree. For example, encrypting letters (ABC) will start by converting them to ASCII form to become (01000001, 01000010, and 01000011). Then, it is converted to (CAAC, CAAG, and CAAT). If the selected DNA sequence to hide the message inside is (ACGGTTCCAATGCCTAAGCTA), it is converted to a balanced tree as shown in Fig. 6.

Cases	Encrypted message	(A = 0 / T = 1 / G = 2 / C = 3)	
Case 1	AGT GCC TAT T	021 233 101	13 MOD 4 = 1 » T
Case 2	GGT GCC TAT T	221 233 101	15 MOD 4 = 3 » C
Case 3	GGT GCC TTG T	221 233 112	17 MOD 4 = 1 » T

Fig. 5. Examples of Mutation Detection Method.

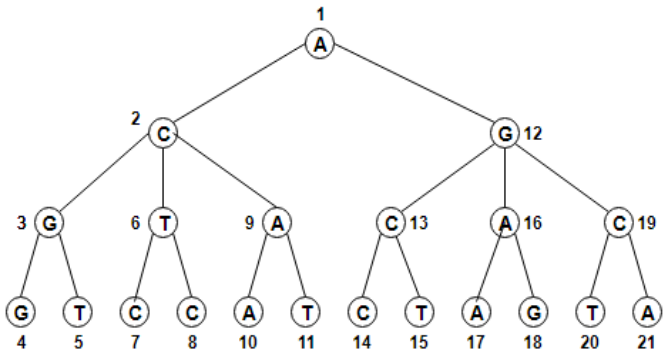

Fig. 6. Balanced tree for Random DNA Sequence [58].

After that, the encrypted message is integrated into the leaf nodes of the tree as shown in Fig. 7. The faked DNA sequence obtained from the balanced tree is (ACGCATATACAGCAGACACAC). The proposed framework offers high hiding capacity and accepts any type of input. Also, hiding spots are scattered all over the reference sequence in non-sequential manner. Nevertheless, the algorithm is pattern-dependent. Once the pattern is revealed, the secret message is compromised. An interesting idea to improve upon this work is using a link list for holding the contents of the DNA sequence instead of the balance tree. Linked lists have higher flexibility, complexity, and more fluent transition from an element to another than a tree.

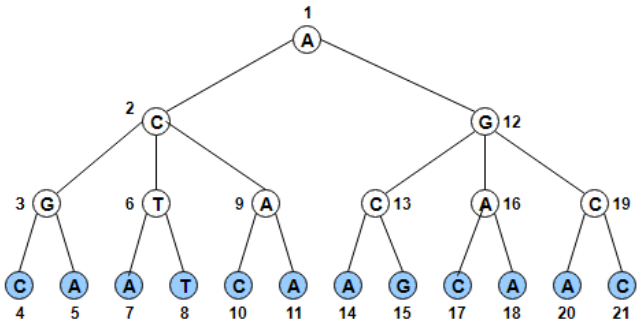

Fig. 7. Balanced Tree after Message Integration [58].

D. The Field of Artificial Intelligence

The field of artificial intelligence can also be used to develop a robust DNA steganography algorithm. The authors in [59] used backpropagation artificial neural network (ANN) to achieve high hiding capacity, low cracking probability, and zero payloads. In the embedding process, the inputs are the DNA reference with the secret message and the output is a set of weights from the neural network. At the beginning, every character in the secret message is converted to the equivalent ASCII binary value. Then, this value is converted to the equivalent DNA letter based on the DNA binary coding rule. Each letter in the DNA message will be assigned to a location in the DNA reference. ANN will be used to get a list of binary locations and the weights of the training phase will be stored in a set. This set and the chosen DNA reference are sent to the receiver. Fig. 8 represents a process diagram to illustrate the AI-based data encryption and data hiding processes.

At the receiver side, the extraction process is executed by implementing the steps of the embedding process in reverse order. Firstly, both the DNA sequence and the weights received from the sender are applied to the ANN. The result is a list of positions denoting the hiding locations. A set is constructed of DNA letters assembled from these locations. Based on the binary coding rule table, DNA letters are converted to binary form. The secret message is obtained by converting every 8-bits to a character using the ASCII code. Fig. 9 represents a process diagram to illustrate the data decryption and message extraction.

The proposed algorithm shows good results like 2 BPNs and zero payloads. It also preserves the biological functionality of the DNA sequence because no modification has been done to it. The weak points of this algorithm are the need to send the original DNA sequence to the receiver. Hence, it is not a blind algorithm. Also, the main defect of using the ANN is the high consumed time, especially in the training phase. This time can be reduced by using only a segment of the DNA sequence as the input to the ANN instead of using the whole sequence. Besides that, it is really interesting to investigate using swarm intelligence techniques like particle swarm optimization (PSO), bee colony, and ant colony to create a hiding locations pattern. This may outperform ANN in terms of complexity and time consumption.

To have a comprehensive look over the classical and modern techniques discussed in this paper, Table IV states the security parameters of these approaches. It can be concluded from this table that the substitution method is more suitable with small size secret messages. It is used when preserving features like blindness, zero payloads, and sequence functionality are crucial. Besides that, the substitution method usually uses only specific locations in the cover medium like LSBase for hiding purposes. This restriction leads to using only long DNA sequences as a cover medium. This issue can be solved by reducing the size of the hidden message with a proper lossless compression method. On the other hand, the insertion method is used when large chunks of data need to be exchanged. It ignores features like blindness and preserving functionality to offer no threshold boundary for the amount of embedded data. Finally, Table V summarizes the modern algorithms in terms of their purpose, advantages and disadvantages, and possible ways to improve them.

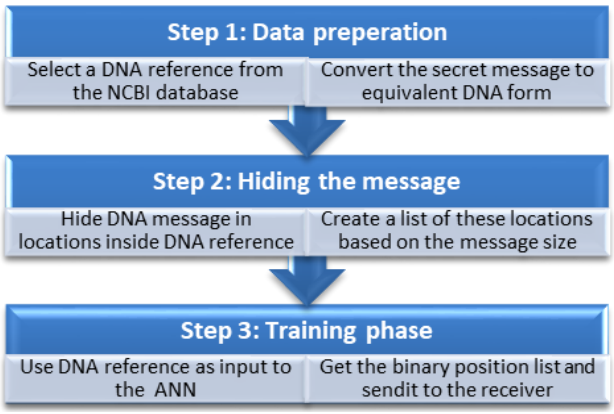

Fig. 8. Data Encryption and Embedding.

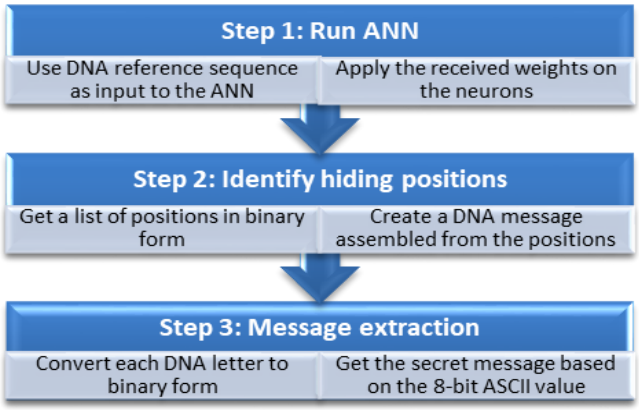

Fig. 9. Data Decryption and Extraction.

TABLE IV. SECURITY MEASUREMENTS OF RECENTLY PROPOSED METHODS

Author & Year	Method	Hiding Capacity	Key Type	Use Data Encryption	Number of Layers	Payload	Modification Rate	Use Random Variable	Blindness	Preserving Functionality
Khalifa et al. 2016 [55]	Insertion	High	Secret Key	Playfair	Single	Yes	High	Yes	Yes	No
Marwan et al. 2017 [46]	Substitution	High	Secret Key	Playfair or Vigenere	Single	No	High	Yes	No	Yes
Malathi et al. 2017 [36]	Insertion	High	Secret Key	No	Single	Yes	Low	Yes	No	No
Sajisha et al. 2017 [41]	Substitution	Low	Secret Key	AES	Single	No	Low	Yes	Yes	Yes
Vijayakumar et al. 2018 [18]	Substitution	Low	No	No	Double (Image)	---	---	No	No	---
Hamed et al. 2018 [23]	Substitution	High	No	Playfair	Single	No	Low	Yes	Yes	Yes
Saha et al. 2019 [58]	Substitution	High	No	No	Single	No	High	No	Yes	No
Sabry et al. 2019 [57]	Substitution	High	Secret Key	No	Single	Yes	High	Yes	Yes	Yes
Mohammed et al. 2019 [59]	Position based	High	Secret Key	No	Single	No	No	No	No	Yes
Dokuyn Na et al. 2020 [56]	Substitution	Low	No	No	Single	No	Low	No	Yes	Yes

TABLE V. SUMMARY OF THE REVIEWED MODERN APPROACHES

Author & Year	Fields	Aim of the paper	Pros	Cons	Suggestions
Hamed et al. 2018 [23]	Biology	Exploit the DNA conservative mutations to develop secured, high capacity, preserved algorithm.	Using advanced Playfair cipher. Very low cracking probability. Scattered hiding locations. High capacity. Preserve biological features.	Using fixed substitution table. BCR table is required to be send to the receiver every transmission session.	Use dynamic and randomly generated substitution table. Integrate the BCR table in the DNA sequence.
Saha etal. 2019 [58]	Data structure	Increase hiding capacity by embedding encrypted characters of the secret message in the leafs of a balanced tree.	Over 50% hiding capacity. Scattered hiding locations. Blind. Zero payload.	Pattern dependent technique. Do not Preserve biological features. High modification rate.	Use more than one nucleotide in leafs nodes will increase hiding capacity. Use a linked list instead of a tree may offer better results.
Mohammed etal. 2019 [59]	Artificial intelligence	Develop a robust DNA steganography algorithm by using ANN.	Preserve biological features. Zero payload. Low cracking probability.	Not blind. High execution time. Sequential hiding pattern.	Lower consumed time in training by reducing the range of ANN's inputs. Use swarm optimization techniques to create a hiding pattern.
Sabry etal. 2019 [57]	Biology	Maintaining DNA signature by exploiting ambiguity feature and hiding data in redundant codons of amino acids.	Preserve biological features. Very low penetration probability versus different types of attacks.	Adding new bit in some cases is required during embedding process. The performance of the algorithm depends on the contents of the secret message.	Hiding data in only locations related to amino acids guarantee best performance.
Na 2020 [56]	Biology Networking	Use SNPs in DNA as hiding locations. Developing a mutation detection method.	SNPs provide good noise cover. Single mutation detection. Zero payload. Blind algorithm. Preserve functionality.	Low hiding capacity. Deal only with alphabets and numbers. Block checksum fails in addition, deletion and swapping cases.	Compress the data before hiding process. Enhance the performance of the checksum technique by using a dedicated DNA sequence for it.

VI. Conclusion and Recommendation

DNA has proved it can be the optimal medium for data hiding and transmission. It gives promising solutions to issues that arise when other covering mediums are used. In the last twenty years, many DNA steganography algorithms have been proposed. Despite the wide range of ideas offered by steganography algorithms, there are many gaps and unresolved issues in this field. Some of them are:

- An improved data encoding method is required to convert the binary form of the secret message to genetic letters. The required number of genetic letters to hide the binary form of the secret message can be decreased using the lossless compression method. This will increase the availability of hiding locations and the value of BPN.
- One of the DNA-based cryptography techniques is using a dictionary or a lookup table to link characters of the secret message to a specific codon. Lookup tables have a limited capacity of only 64 locations if codons are used. This is sufficient to encode alphabets and numbers while special characters are ignored. Besides that, lookup tables are static, and using the same table in every data transition session is quite risky. If a third party revealed the contents of the lookup table, the whole steganography algorithm is compromised. One possible solution to this issue is using a dynamic lookup table randomly generated at each transition session. The sender will integrate the new table with the concealed secret message.
- There is a trade-off between providing high hiding capacity and preserving biological features of the DNA sequence. Using all nucleotides to hide data will ruin the functionality of the protein while exploiting biological characteristics like ambiguity or mutations will significantly decrease hiding capacity. Therefore, there is an obvious need for a technique that enables us to hide the maximum amount of data in LSBase locations in codons.
- The majority of developed DNA-based steganography algorithms used adjacent or sequential locations in the DNA sequence to hide characters of the secret message. Also, the hiding process is applied to these characters with the same order of their locations in the secret message. Alternatively, scattering hiding locations all over the sequence will make deducing the secret message from the covering medium even harder for a third party. One of the applied techniques in the literature to achieve such a task is using a balanced tree to distribute the message characters all over the sequence. This approach is fairly static and adopting another data structure such as the linked list will offer a more flexible transition from one character to another.
- One of the modern trends in the field of DNA-based steganography is adopting a predefined location-based hiding structure. Both the sender and the receiver agree on a specific method like a neural network with backpropagation and use its structure in the encryption and decryption phases. The selected method must meet several security and performance measures. For example, minimum deployment speed, variations of hiding locations at each run, and preserving the content of this secret message even if the hiding structure is predicted by an attacker. Identifying the optimal hiding structure method is still an open issue in this field.
- The transmission of the fake DNA sequence from the source to a destination over an untrusted channel may encounter different types of obstacles. Some of these obstacles are mutations, transition errors, and message modification by a third party. To authenticate the integrity of the obtained secret message, a message authentication technique is required. Previously proposed methods are limited to deal with unique cases like single mutations. A comprehensive solution to this issue is using a dedicated DNA sequence to guarantee the integrity of the fake DNA sequence. This will offer more locations for both data hiding and data checking.

All in all, the combination of exploiting biological attributes of the DNA with utilizing the different techniques borrowed from various fields of computer science proved to be fruitful. It offers promising results and solutions for the existing issues in the field of DNA-based steganography. These recently suggested novel approaches are expected to accelerate the movement towards consolidating the rank of DNA as one of the main concealing mediums. Also, they sustain using DNA-based steganography methods in real-life applications.

References

[1] S. Namasudra, “Cloud computing: A new era,” Journal of Fundamental and Applied Sciences, 10 (2), pp. 113–135, 2018.

[2] G. Hamed, M. Marey, S. El-Sayed, and F. Tolba, “DNA based steganography: Survey and analysis for parameters optimization,” In Intelligent Systems Reference Library, Vol. 96, 2016.

[3] R. Anusha, M. J. Dileep Kumar, V. S. Shetty, and N. Prajwal Hegde, “Symmetric Key Algorithm in Computer security: A Review,” Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020, pp. 765–769.

[4] P. Chaudhury, S. Dhang, M. Roy, S. Deb, J. Saha, A. Mallik, et al., “ACAFP: Asymmetric key based cryptographic algorithm using four prime numbers to secure message communication. A review on RSA algorithm,” 8th Industrial Automation and Electromechanical Engineering Conference, IEMECON, 2017, pp. 332–337.

[5] S. S. Baawi, M. R. Mokhtar, and R. Sulaiman, “New text steganography technique based on a set of two-letter words,” Journal of Theoretical and Applied Information Technology, 95 (22), pp. 6247–6255, 2017.

[6] M. A. Majeed, and R. Sulaiman, “An improved LSB image steganography technique using bit-inverse in 24 bit colour image,” Journal of Theoretical and Applied Information Technology, 80 (2), pp. 342–348, 2015.

[7] A. H. Ali, L. E. George, A. A. Zaidan, and M. R. Mokhtar, “High capacity, transparent and secure audio steganography model based on fractal coding and chaotic map in temporal domain,” Multimedia Tools and Applications, 77 (23), pp. 31487–31516, 2018.

[8] S. Kamil, M. A. Authors, S. N. H. S. Abdullah, and Z. Ahmad, “Lightweight and optimized multi-layer data hiding using video steganography paper” International Journal of Advanced Computer Science and Applications, 9 (12), pp. 256–262, 2018.

[9] S. Kamil, M. Ayob, S. N. H. Sheikh Abdullah, and Z. Ahmad, “Optimized Data Hiding in Complemented or Non-Complemented Form in Video Steganography,” Proceedings of the 2018 Cyber Resilience Conference, CRC, 2018, pp. 1–4.

[10] M. S. Subhedar, and V. H. Mankar, "Current status and key issues in image steganography: A survey," Computer Science Review, 13–14(C), pp. 95–113, 2014.

[11] C. T. Clelland, V. Risca, and C. Bancroft, "Hiding messages in DNA microdots" Nature, 399 (6736), pp. 533–534, 1999.

[12] D. A. Zebari, H. Haron, and S. R. M. Zeebaree, "Security issues in DNA based on data hiding: A review," International Journal of Applied Engineering Research, 12 (24), pp. 15363–15377, 2017.

[13] E. Passarge, "Color Atlas of Genetics," Fifth edition, Thieme, 2019.

[14] J. Momand, and A. McCurdy, "Concepts in bioinformatics and genetics," Oxford University Press, 2017.

[15] NCBI, [Online]. Available: https://www .ncbi.nlm.nih.gov /genbank/ statistics/.

[16] (European Bioinformatics Institute). Online website: https://www.ebi.ac.uk.

[17] J.S. Taur, H.Y. Lin, H.L. Lee, and C.W. Tao, "Data hiding in DNA sequences based on table lookup substitution," Int. J. Innov. Comput. Inf. Control 8, pp. 6585–6598, 2012.

[18] P. Vijayakumar, V. Vijayalakshmi, and R. Rajashree, "Increased level of security using DNA steganography," International Journal of Advanced Intelligence Paradigms, 10 (1–2), pp. 74–82, 2018.

[19] H.J. Shiu, K.L. Ng, J.F. Fang, R.C. Lee, and C.H. Huang, "Data hiding methods based upon DNA sequences,". Inf. Sci. 180, pp. 2196–2208, 2010.

[20] 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," Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition, SoCPaR 2015, pp. 95–102.

[21] S. Marwan, A. Shawish, and K. Nagaty, "DNA-based cryptographic methods for data hiding in DNA media," BioSystems, 150, pp. 110–118, 2016.

[22] W. Stallings, "Cryptography and network security principles and practice," 7th edition. Pearson. 2017.

[23] G. Hamed, M. Marey, S. E. S. Amin, and M. F. Tolba, "Hybrid, randomized and high capacity conservative mutations DNA-based steganography for large sized data," BioSystems, 167, pp. 47–61, 2018.

[24] O.A. Al-Harbi, W.E. Alahmadi, and A.O. Aljahdali, "Security analysis of DNA based steganography techniques" SN Applied Sciences, 2 (2), 2020.

[25] A. Atito, A. Khalifa, and S. Z. Rida, "DNA-Based Data Encryption and Hiding Using Playfair and Insertion Techniques" Journal of Communications and Computer Engineering, 2 (3), pp. 44, 2011.

[26] M. Torkaman, N. Kazazi, and A. Rouddini, "Innovative approach to improve hybrid cryptography by using DNA steganography," International Journal of New Computer Architectures and Their Applications, 2 (1), pp. 225–236, 2012.

[27] S. Manna, S. Roy, P. Roy, and S. K Bandyopadhyay, "Modified technique of insertion methods for data hiding using DNA sequences," 1st International Conference on Automation, Control, Energy and Systems - 2014, ACES 2014, pp. 1–5.

[28] E. I. Abd El-Latif, and M. I. Moussa, "Information hiding using artificial DNA sequences based on Gaussian kernel function," Journal of Information and Optimization Sciences, 40 (6), pp. 1181–1194, 2019.

[29] T. Tuncer, and E. Avci, "A reversible data hiding algorithm based on probabilistic DNA-XOR secret sharing scheme for color images," Displays, 41, pp. 1–8, 2016.

[30] R. M. Tank, , H. D. Vasava, and V. Agrawal, "DNA-based Audio Steganography," An International Open Free Access, Peer Reviewed Research Journal. Vol. 8, No. (1): pp. 43-48, 2015.

[31] H. Chaudhary, and V. Bhatnagar, "Hybrid approach for secure communication of data using chemical DNA," Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit, pp. 967–971.

[32] A. Khalifa, "A Blind DNA-Steganography Approach using Ciphering and Random Sequence Splicing," 10th International Conference on Information Science and Technology, ICIST 2020, pp. 86–90.

[33] M. Skariya, and M. Varghese, "Enhanced Double Layer Security using RSA over DNA based Data Encryption System," International Journal of Computer Science & Engineering Technology (IJCSET), 4 (06), pp. 746–750, 2013.

[34] P. Das, and N. Kar, "A DNA based image steganography using 2D chaotic map," 2014 International Conference on Electronics and Communication Systems, ICECS 2014.

[35] S. Chakraborty, and S. Kumar Bandyopadhyay, "Data Hiding by Image Steganography Appling DNA Sequence Arithmetic," International Journal of Advanced Information Science and Technology, 231944 (44), 2015.

[36] P. Malathi, , M. Manoaj, , R. Manoj, , V. Raghavan, , & R. E. Vinodhini, "Highly Improved DNA Based Steganography," Procedia Computer Science, 115, pp. 651–659, 2017.

[37] H. Mousa, K. Moustafa, W. Abdel-Wahed, and M. Hadhoud, "Data hiding based on contrast mapping using DNA medium," International Arab Journal of Information Technology, 8 (2), pp. 147–154, 2011.

[38] Y. H. Huang, C. C. Chang, and C. Y. Wu, "A DNA-based data hiding technique with low modification rates," Multimedia Tools and Applications, 70 (3), pp. 1439–1451, 2014.

[39] H. Liu, D. Lin, and A. Kadir, "A novel data hiding method based on deoxyribonucleic acid coding," Computers and Electrical Engineering, 39 (4), pp. 1164–1173, 2013.

[40] A. Khalifa, " LSBase: A key encapsulation scheme to improve hybrid crypto-systems using DNA steganography," Proceedings - 2013 8th International Conference on Computer Engineering and Systems, ICCES 2013, pp. 105–110.

[41] K. S. Sajisha, and S. Mathew, "An Encryption based on DNA cryptography and Steganography," International Conference on Electronics, Communication and Aerospace Technology ICECA 2017.

[42] R. Agrawal, M. Srivastava, and A. Sharma, "Data hiding using dictionary based substitution method in DNA sequences," 9th International Conference on Industrial and Information Systems, ICIIS 2014.

[43] P. Das, S. Deb, N. Kar, and B. Bhattacharya, "An improved DNA based dual cover steganography," Procedia Computer Science, 46 (Icict 2014), pp. 604–611, 2015.

[44] K. Santoso, S. Lee, W. Hwang, and K. Kwon, "Sector-based DNA information hiding method," Security Comm. Networks 2016, vol 9, pp. 4210–4226.

[45] A. Khalifa, and A. A. Elhadad, "High-Capacity DNA-based Steganography," The 8th international conference of informatics and systems. 2014.

[46] S. Marwan, A. Shawish, and K. Nagaty, "Utilizing DNA strands for secured data-hiding with high capacity," International Journal of Interactive Mobile Technologies, 11 (2), pp. 88–98, 2017.

[47] N. A. Zebari, , D. A. Zebari, , D. Q. Zeebaree, and J. N. Saeed, " Significant features for steganography techniques using deoxyribonucleic acid: a review," Indonesian Journal of Electrical Engineering and Computer Science, 21 (1), pp. 338–347, 2021.

[48] M. R. Abbasy, P. Nikfard, A. Ordi, and N. R. M. Torkaman, "DNA Base Data Hiding Algorithm," International Journal of New Computer Architectures & Their Applications, 2 (1), pp. 183–192, 2012.

[49] A. Majumdar, M. Sharma, and N. Kar, "An Improved Approach to Steganography using DNA Characteristics," 2016.

[50] Shweta, & S. Indora, "Cascaded DNA cryptography and steganography," International Conference on Green Computing and Internet of Things (ICGCIoT), 2015.

[51] K. Menaka, "Message encryption using DNA sequences," Proceedings - 2014 World Congress on Computing and Communication Technologies, WCCCT 2014, pp. 182–184.

[52] C. Guo, C. C. Chang, and Z. H. Wang, "A new data hiding scheme based on DNA sequence," International Journal of Innovative Computing, Information and Control, 8 (1A), pp. 139–149, 2012.

[53] C. M. Shyamasree, and S. Anees, "Highly secure DNA-based audio steganography," 2013 International Conference on Recent Trends in Information Technology, ICRTIT 2013, pp. 519–524.

[54] Manisha, Parvinder Bangar, and Mohit, “Double layered DNA based cryptography,” IJRET: International Journal of Research in Engineering and Technology, 2015.

[55] A. Khalifa, A. Elhadad, and S. Hamad, “Secure blind data hiding into pseudo dna sequences using playfair ciphering and generic complementary substitution,” Applied Mathematics and Information Sciences, 10 (4), pp. 1483–1492, 2016.

[56] D. Na, “DNA steganography: Hiding undetectable secret messages within the single nucleotide polymorphisms of a genome and detecting mutation-induced errors,” Microbial Cell Factories, 19 (1), pp. 1–9, 2020.

[57] M. Sabry, T. Nazmy, and M. E. Khalifa, “Steganography in DNA Sequence on the Level of Amino acids,” Proceedings - 2019 IEEE 9th International Conference on Intelligent Computing and Information Systems, ICICIS 2019, pp. 317–324.

[58] P. Saha, L. Y. Pinky, M. A. Islam, and P. Akter, "Higher Payload Capacity in DNA Steganography using Balanced Tree Data Structure," International Journal of Recent Technology and Engineering, 8 (4), pp. 6551–6556, 2019.

[59] M. H. Mohammed, B. H. Ali, and A. I. Taloba Mohamed, “Self-adaptive dna-based steganography using neural networks,” Information Sciences Letters, 8 (1), pp. 15–23, 2019.

[60] N. S. Terkawi, L. Berriche, A. A. Alamar, M. A. Ibrahim, and W. S. Alsaffar, “Comparative Study of Three DNA-based Information Hiding Methods,” International Journal of Computer Science and Security (IJCSS), Volume (15), Issue (2), 2021.

[61] A. Khalifa, “A secure steganographic channel using DNA sequence data and a bio-inspired XOR cipher,” Information (Switzerland), 12 (6), 2021.