IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 August 2023, accepted 25 September 2023, date of publication 16 October 2023, date of current version 25 October 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3324875*

SURVEY

# A Survey on DNA-Based Cryptography and Steganography

**TASNUVA MAHJABIN[1], ALINA OLTEANU[2], (Member, IEEE), YANG XIAO[1], (Fellow, IEEE), WENLIN HAN[3], TIESHAN LI[4], (Senior Member, IEEE), AND WEI SUN[5], (Senior Member, IEEE)**

[1]Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487, USA
[2]Department of Biology, Chemistry, Mathematics and Computer Science, University of Montevallo, Montevallo, AL 35115, USA
[3]Department of Computer Science, California State University, Fullerton, CA 92831, USA
[4]School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[5]School of Electrical and Automation Engineering, Hefei University of Technology, Hefei, Anhui 230009, China

Corresponding author: Yang Xiao (yangxiao@ieee.org)

**ABSTRACT** This paper provides a comprehensive survey of different techniques for Deoxyribonucleic acid (DNA) -based cryptography and steganography. DNA-based cryptography is an emerging field that utilizes DNA molecules' massive parallelism and vast storage capacity to encode and decode information. The field has gained significant attention in recent years due to its potential advantages over traditional cryptographic methods, such as high storage capacity, low error rate, and resistance to environmental factors. In this paper, we review three types of DNA-based cryptography: natural DNA cryptography, pseudo-DNA cryptography, and DNA-based steganography. For each technique, we discuss its advantages and limitations, as well as future directions for research. Our goal is to contribute to a better understanding of the applications and limitations of using DNA for cryptographic purposes. We believe that our analysis will be useful for researchers working on developing new techniques for secure data transmission using DNA molecules.

**INDEX TERMS** DNA, DNA-computing, cryptography, steganography, security.

## I. INTRODUCTION

Deoxyribonucleic Acid (DNA) is the biological medium for storing and transmitting genetic material for all living things on earth. By extension, DNA's ever-evolving nature in providing biological information to empower cells to develop into a myriad of organisms lends itself to harnessing its use in computers. Its power is demonstrated through its impressive storage capacity - an ounce of DNA, the equivalent of the amount fitting on a penny, can store 30,000 terabytes of memory for up to 1 million years [1]. Further, after initial setup, DNA possesses the ability to replicate and assemble itself through the process of evolution without any intervention. These features of DNA showcase the inexpensive computing power and efficiency of DNA as a medium of transmission of digital information.

Accordingly, DNA Computing has been developed to utilize the natural abilities of DNA in computer science and mathematical applications revealing new and exciting computational possibilities. L.M. Adleman introduced the concept of DNA computing in 1994 [2]. In his work, Adleman presented DNA as a medium of data storage and parallel computation. To demonstrate the use of DNA as a possible computing medium, he coded an instance of the NP-complete Hamiltonian Path problem into DNA molecules. The work is remarkable because it solves the proposed instance of the Hamiltonian Path problem using only DNA and molecular operations. Since then, DNA computing has become a popular area of research for solving classical hard problems. DNA cryptography has developed alongside DNA Computing to become an important specialty in the field.

The field of cryptography has been concerned with data protection and secure communication dating back thousands of years to its roots in ancient Egypt [3]. To this day,

The associate editor coordinating the review of this manuscript and approving it for publication was Shu Xiao.

cryptography is still a popular and essential means of information security. As a result, researchers focus on newly discovered areas attempting to enhance information security in the digital space. DNA is a natural fit as a medium to expand and improve cryptography. C.T. Clelland, V. Risca, and C.T. Bancroft first explored the connection between DNA and secure communication in 1999. A steganography method was proposed to conceal secret messages in DNA strands [4]. Following this pioneering work, other researchers developed techniques to secure information using DNA, introducing new and exciting contributions to the field. These researchers are motivated by the promise of an ever-expanding use of DNA as a method to broaden understanding of data protection.

DNA as a means of securing information has been explored along three general methodologies. *First*, natural DNA cryptography applies cryptographic algorithms in a wet database of DNA strands (a solution of DNA strands in a test tube) or to synthesized DNA strands. DNA chemical processes are applied to DNA strands in both cases to generate encrypted data. One significant success in this field is the generation of truly random one-time-pads using DNA's massively parallel computing abilities. Furthermore, natural DNA cryptography methods provide a means to conduct cryptoanalysis. The earliest major crypto-analysis accomplishment using DNA-based cryptography was breaking the Data Encryption Standard (DES) using brute force operations [5], [6]. *Second*, Pseudo-DNA cryptography is similar to natural DNA cryptography, the critical difference being using theoretical models with no biological material. These theoretical modeling techniques are applied to binary data. Generally, the pseudo-DNA cryptography method starts with the message being translated into binary strings and then transformed into pseudo-DNA strands. Pseudo-DNA operations are applied to the pseudo-DNA strands to increase the security of existing algorithms. The resultant pseudo-DNA strands are then translated to binary strings and sent through the communication channel. *Lastly*, DNA-based steganography is used to conceal information. The word *steganography* originates from the Greek 'steganos,' meaning 'covered' and 'graphia' which stands for 'writing,' hence the term 'covered writing.' One of DNA's advantages in this respect is that messages hidden within DNA strands are difficult to detect and decode. With tens of millions of possibilities to sift through, the complexity and randomness of human DNA make DNA Steganography stand out in the field of information hiding [7], [8].

In recent years, there has been growing interest in using DNA as a medium for cryptographic purposes. DNA-based cryptography offers several advantages over traditional cryptographic methods, including high storage capacity, low error rate, and resistance to environmental factors. However, using DNA for cryptographic purposes poses several challenges: the high cost of synthesis and sequencing, the need for specialized equipment and expertise, and the potential for errors during encoding or decoding, among others. Researchers have developed different types of DNA-based cryptography techniques to address these challenges.

In this paper, we provide a comprehensive survey of different techniques for DNA-based cryptography and steganography. We review three types of DNA-based cryptography: natural DNA cryptography, pseudo-DNA cryptography, and DNA-based steganography. Natural DNA cryptography utilizes the inherent properties of DNA molecules to encode and decode information. Pseudo-DNA cryptography uses simulated DNA structures and traditional cryptographic techniques to enhance security and performance. Finally, DNA is primarily used as an information-hiding medium in DNA-based steganography. Each of these three types of DNA-based cryptography offers unique advantages and limitations. By reviewing these techniques in detail, we aim to provide a comprehensive overview of the current state-of-the-art in this field. We discuss the challenges associated with each technique and suggest potential future directions for research in these areas. We believe that our analysis will be useful for researchers working on developing new techniques for secure data transmission using DNA molecules.

The rest of the paper is organized as follows: Section II covers the basic structure of DNA, bio-molecular operations employed in cryptographic techniques, and coding techniques used to convert binary data to DNA strands and vice versa. Section III summarizes research on natural DNA cryptography, including encryption techniques and biomolecular operations applied to these methods. Section IV highlights work in pseudo-DNA cryptography, which involves utilizing DNA coding and simulated structures with traditional cryptographic techniques. Section V outlines DNA-based steganography techniques involving DNA as an information-hiding medium. Section VI analyzes and discusses the strengths and limitations of existing solutions in each field. A comparison is made between natural, pseudo, and DNA-based steganography. Section VI also discusses advancements in DNA-based cryptography and steganography, as well as limitations and challenges that must be addressed. The key findings from the survey are summarized in Section VII, along with final thoughts on the future of DNA-based cryptography and steganography.

## II. DNA STRUCTURE AND BACKGROUND

DNA is known as the blueprint of life. Nobel prize-winning geneticists Watson et al. discovered its structure and properties in 1953 [9]. To understand DNA-based cryptography, it is essential to review DNA's structure and the biological operations utilized in DNA cryptography, as set out in Watson and Crick's ground-breaking discovery and expanded upon by the scientific community. The following subsections cover the essential information regarding the DNA structure, its operations, and DNA coding, used to convert digital information into DNA form and vice versa. We relate this background knowledge in a manner that is simple and accessible to the general reader.

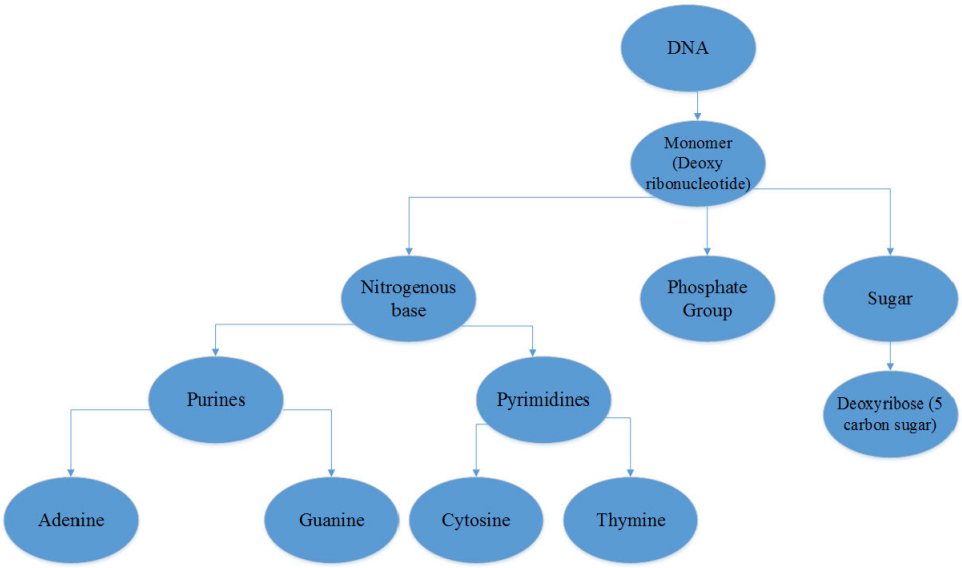


**FIGURE 1.** Building blocks of DNA molecules.

### A. DNA STRUCTURE

The basic building block of DNA is a molecule called a nucleotide [10]. The nucleotide comprises the following elements: a nitrogenous base, a phosphate group, and a sugar [10]. The sugar used to build the nucleotide is known as deoxyribose, hence DNA's 'deoxyribo' prefix. This sugar contains an atom of five carbons, numbered $1'$ to $5'$. In the DNA structure, the phosphate group attaches to the $5'$ carbon, whereas the nitrogenous base attaches to the $1'$ carbon. A hydroxyl group (OH) attaches to the $3'$ carbon of the sugar. DNA contains four different nucleotides (a type of molecule), each defined by a nitrogenous base: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). A and G belong to the nitrogenous bases called purines, while C and T fall into the pyrimidine category. Uracil (U), another pyramidine, is found in RNA (Ribonucleic Acid) and Cytosine. Fig. 1 shows DNA's constituent blocks.

Fig. 2 represents a basic DNA structure. Fig. 2 (a) shows a simple schematic structure of a deoxyribonucleotide. Here, B represents the nitrogenous base, attached to the $1'$ carbon, and P represents the phosphate group, attached to the $5'$ carbon of the sugar. The sugar is indicated by the golden thick line with the carbon hands depicted as circles. Different values of B (A, G, C, T) result in different nucleotides. A DNA strand is made up of multiple DNA nucleotides. The DNA strand's formation is completed in two phases. A strong covalent bond is generated between two different nucleotides in the first phase. This bond is known as the phosphodiester bond. One nucleotide's phosphate group joins with another nucleotide's hydroxyl group (OH). The resulting formation creates a single-stranded DNA, as shown in Fig. 2(b). There is an unattached $5'$ phosphate group on one end of the single-stranded DNA and a free $3'$ hydroxyl group on the other end. The unattached phosphate and hydroxyl groups are used to form a larger chain. The second phase occurs when a bond is formed between the nitrogenous bases: A, T, G, and C of two nucleotides. *In DNA, base A always pairs with base T and base G always pairs with base C*. This landmark result is known as Watson-Crick's complementarity principle [9] and led to the discovery of the famous spiraling staircase structure of the double-stranded DNA known as the 'double helix.' The double-stranded DNA, shown in Fig. 2 (c), can be written as:

$$5' - \text{ACG} - 3'$$
$$3' - \text{TGC} - 5'$$

This representation shows that a single-stranded DNA sequence pairs with another sequence in the opposite direction. The DNA structure shown in Fig. 2 (c) is a basic linear representation. As mentioned above, in the actual structure, two linear strands wound around each other to form the double-helical structure shown in Fig. 2 (d).

### B. TERMS AND OPERATIONS

Natural DNA cryptography utilizes biological materials in a laboratory setting, performing molecular operations on strands of natural or artificial DNA. Meanwhile, in pseudo-DNA cryptography, the biological operations from natural DNA cryptography are replaced by computer-simulated operations. This section introduces the terms and operations necessary to understand DNA-based cryptography.

#### 1) DNA SYNTHESIS

The process of creating artificial DNA strands is known as DNA synthesis [11]. In this process, base nucleotides are placed in a specialized DNA synthesis machine called a synthesizer. The synthesizer combines the base nucleotides according to instructions input into the synthesizer to

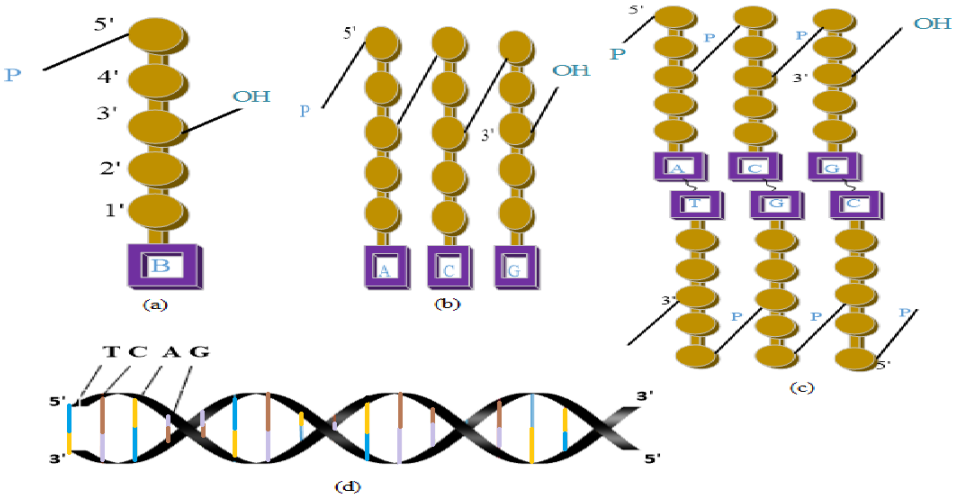


**FIGURE 2.** Structure of DNA: (a) Basic schematic structure (b) Single-stranded DNA, (c) Double-stranded DNA (d) DNA double helix [10].

generate millions of synthetic DNA sequences [12]. This vast range of distinct individual sequences provides the material used by researchers to conduct computations and experiments. Further, DNA computing can use the resulting strands as storage media. Based on the practical ease of creating varied sequences of synthetic DNA on a large scale, synthesized DNA strands are the key source material in DNA computing and DNA cryptography.

### 2) THE LENGTH OF DNA

In the process of DNA computing, it is necessary to measure the length of DNA molecules. The length of the DNA molecule indicates the total number of nucleotides or base pairs used to form it [10]. For example, to measure the length of a single-stranded DNA molecule, we count the total number of constituent nucleotides in it. Thus, if a strand has 20 nucleotides, its length is written as 20 mer, indicating that the DNA strand contains 20 monomers (molecules that can bind with other molecules). On the other hand, the length of a double-stranded DNA, where each base (A, T, G, C) pairs with its complementary base, consists of the total number of base pairs it contains. Thus, if a double-stranded DNA contains 20 base pairs, its length is 20 bp.

### 3) GEL ELECTROPHORESIS

Gel electrophoresis is a technique that separates and sorts DNA strands by size [10], [11]. Electrophoresis uses the movement of charged molecules in an electrically powered field. Since DNA molecules are negative in charge, they will migrate toward the positive pole in an electric field. DNA molecules contain equal charge per unit length. Therefore, in an aqueous solution, all molecules, independent of length, move with similar speed. A gel is introduced to vary the speed of different molecules. The gel increases the density of the aqueous solution, affecting the molecules' movement rate depending on their size. As such, smaller molecules move towards the positive pole faster than the larger molecules. The electric field is turned off when the first molecule reaches the positive side. The result is a sorted pattern of molecules in the gel according to their size. Since the molecules' length is almost proportional to their weight, through gel electrophoresis, DNA molecules can be sorted based on their lengths. Furthermore, the existence of molecules of a given length can be checked.

### 4) DENATURATION OR SEPARATION OF DNA STRANDS

Denaturation is the process of separating a double-stranded DNA into two single-stranded DNAs [11], [13]. As stated earlier, a single-stranded DNA gets attached to another single-stranded DNA using a weak hydrogen bond. This bond is much weaker than the phosphodiester bonds between the phosphate and hydroxyl groups. Thus, in denaturation, the solution is heated to a temperature of approximately $85^0 - 95^0$ C [13]. As a result, the hydrogen bond breaks, and single DNA strands are formed.

### 5) ANNEALING OR JOINING DNA STRANDS

The reverse of denaturation is renaturation or annealing. In this process, two single-stranded DNA sequences join to form a double-stranded DNA [10]. The method relies on cooling the sequences in a solution allowing the hydrogen bonds to fuse, forming the double helix. Cooling the solution containing the single strands is performed gradually to facilitate the strands with complementary bases to bind to each other.

### 6) ENZYMES

In DNA computing, enzymes play a key role in manipulating DNA. Enzymes are proteins that serve as biological catalysts for chemical reactions in living cells, working as accelerators of the process [10]. One example enzyme is DNA nuclease, which shortens DNA strands. Another is restriction

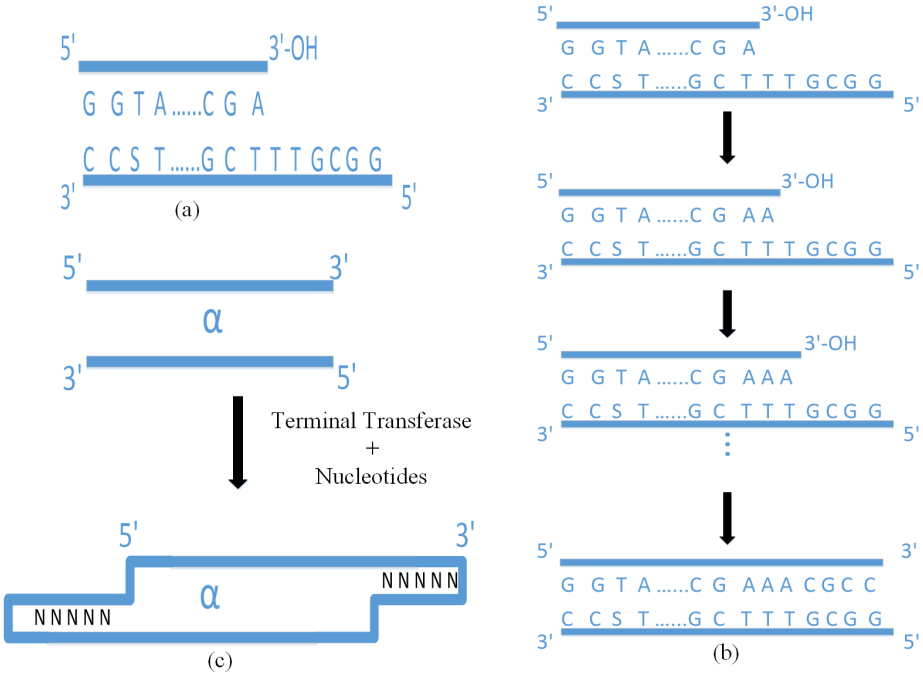


**FIGURE 3.** Extending DNA strands: (a) A DNA molecule with free OH at its 3′ ends (b) Extending double-stranded DNA with Polymerase (c) Extending DNA molecule $\alpha$ on both sides. Here NNNNN is the single-stranded DNA, added on both sides [10].

endonucleases, which recognize precise DNA sequences [11]. Yet another example of a class of enzymes utilized in DNA Computing is DNA polymerase. These enzymes do the important work of amplifying and extending DNA sequences [10], [11], [13].

### 7) EXTENDING DNA STRANDS

DNA strands can be extended in various ways [10], [11], [13], on one side or both sides. The polymerase enzyme extends a DNA strand by filling in an incomplete strand, only in the 5′ - 3′ direction. To perform this operation, the polymerase requires the existence of a free 3′ end. Fig. 3 describes this operation. In Fig. 3 (a), the upper strand contains a free 3′ end. The strand is extended by attaching nucleotides to the 3′ end, one at a time, as shown in Fig. 3 (b). The extension continues until all the nucleotides of the shorter strand pair with the nucleotides of the longer strand, following the Watson-Crick complementarity rule.

A specialized polymerase called Terminal Transferase extends a DNA strand on both sides. This polymerase attaches a single-stranded DNA as a tail to the 3′ end of a double-strand as shown in Fig. 3 (c). Here, $\alpha$ represents the double-stranded DNA molecule with the free 3′ ends for extension, and NNNNN represents the single-stranded DNA to be added on the 3′ ends on each side.

### 8) REDUCING THE LENGTH

Two types of enzymes are involved in shortening a DNA strand: exonucleases and endonucleases [10], [13]. Exonuclease can remove one nucleotide from any of a DNA strand's ends. For example, exonuclease III is used to degrade DNA in the $3' - 5'$ direction. Exonucleases can also be used to shorten double-stranded DNA. Fig. 4 shows this operation. In Fig. 4 (a), exonuclease Bal31 is used to shorten a double-stranded DNA. It is seen from the figure that, at each step, one nucleotide from each end is removed until the desired double-strand $\alpha$ is produced. On the other hand, endonuclease works by destroying phosphodiester bonds between nucleotides. It can cut a DNA molecule in different ways. For example, endonuclease S1 is used to cut only single-stranded DNA at any position, whereas endonuclease 'DNase I' can be used to cut both single and double-stranded DNA. Moreover, the well-known endonuclease called restriction endonuclease works more specifically. It is used to cut double-stranded DNA at specific sites known as restriction sites. For example, in Fig. 4 (b), ATCG (in orange color) is a restriction site. The action of the restriction enzyme Sau3AI is presented in the figure. Sau3AI produces a staggered cut at the restriction site. That is a double-stranded DNA with 'sticky ends' (the end of a DNA strand, which acts as a glue and attaches to other nucleotides) is produced, as shown in the figure. The sticky ends are then used to link the DNA molecule to other DNA molecules with complementary sticky ends.

### 9) LIGATION

Ligation is the process of joining two different DNA molecules [13]. In this process, two different double-stranded DNAs with complementary bases bond to each other with the help of ligase enzymes. Fig. 5 represents the process using a DNA ligase enzyme. Here, we see that the OH group of each strand attaches to the phosphate group (represented by P) of the other strand and forms one single long strand.

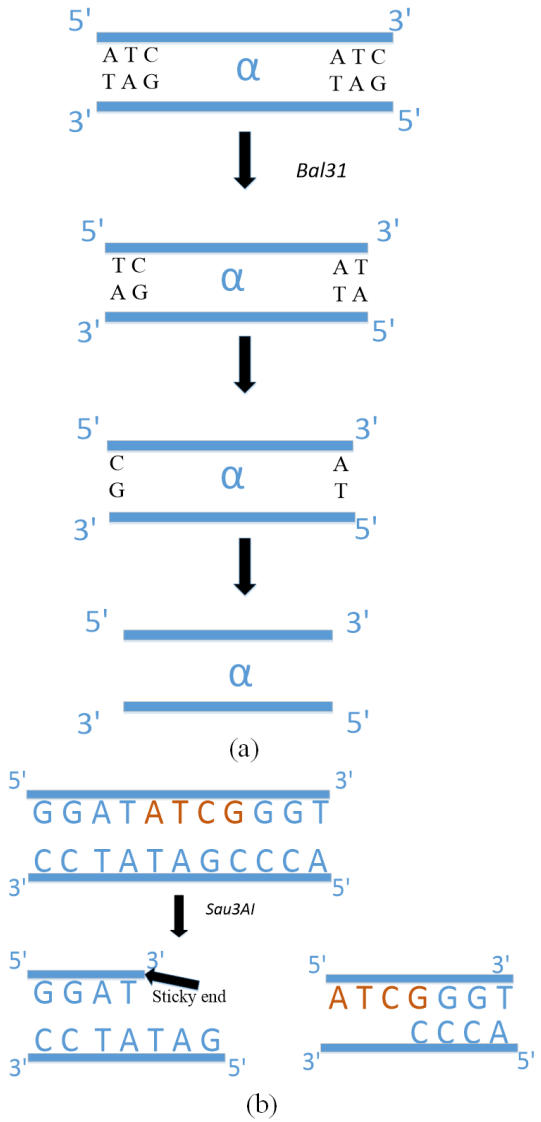


**FIGURE 4.** Reducing DNA length: (a) Shortening a double-stranded DNA with exonuclease Bal31 [10] (b) 'Cut' operation using a restriction enzyme [11].

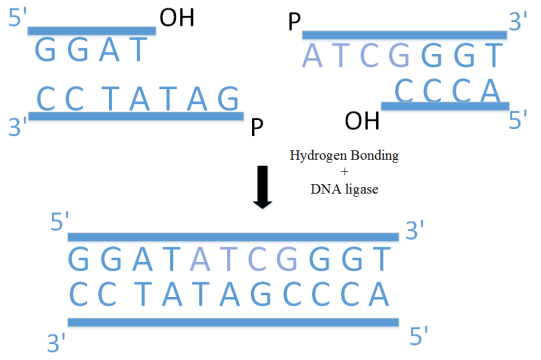


**FIGURE 5.** Ligation [10].

### 10) POLYMERASE CHAIN REACTION (PCR)

PCR is an extremely applicable and one of the most important operations in DNA computing [10], [13], [14]. Its main purpose is to amplify (produce millions of copies of) a certain sequence in a vast pool of mixed sequences. We summarize this important operation in Fig. 6. We consider amplifying a short DNA sequence $\alpha$, with borders $\beta$ and $\gamma$, as shown in Fig. 6 (a). The amplification process is performed by repeatedly executing a cycle consisting of three steps: denaturation, priming, and extension.

Initially, a solution is prepared, consisting of the target sequence $\alpha$, primer sequences $\overline{\beta}$ and $\overline{\gamma}$ (complementary to end sequences $\beta$ and $\gamma$), and other required enzymes and nucleotides. In the denaturation phase, the solution is heated close to boiling to break the double-stranded DNA into two single strands. As a result, strands $\alpha_1$ and $\alpha_2$ are created, as shown in Fig. 6 (b). Next, in the priming phase, the solution is cooled down to allow primers $\overline{\beta}$ and $\overline{\gamma}$ to bond to their complementary strands, such that $\overline{\beta}$ pairs with $\beta$ from $\alpha_1$ and $\overline{\gamma}$ pairs with $\gamma$ from $\alpha_2$, as depicted in Fig. 6 (c). Finally, in the extension phase, the primer sequences are expanded using polymerase to form a complete double-stranded DNA identical to $\alpha$. Therefore, $\alpha_1$ and $\alpha_2$ form two exact copies of $\alpha$, as shown in Fig. 6 (d). Each subsequent repetition produces two additional replicas of the initial sequence. Hence, in $n$ steps, $2^n$ copies of $\alpha$ will be produced.

The two primers above are called a *primer pair*. This simple yet efficient PCR process is used extensively in DNA cryptography and steganography, as evidenced in sections III, IV, and V below.

### 11) READING DNA SEQUENCES

To analyze DNA, we need to be able to read the order in which nucleotides A, T, C, and G are encoded withing a DNA strand. This process is known as DNA sequencing [10] and employs a multitude of biochemical operations such as primer extension with polymerase, denaturation, and gel electrophoresis.

### 12) THE CENTRAL DOGMA

The central dogma is a framework explaining the transfer of genetic information between three main classes of natural polymers: DNA, RNA (Ribonucleic Acid), and protein [15]. The Central Dogma identifies three general transfers of biological information through the following processes: DNA replication, transcription, and translation. Through DNA replication, DNA can be copied to itself. Transcription copies DNA information into messenger RNA (mRNA - a type of single-stranded RNA used in protein synthesis). Finally, the translation uses the information in mRNA as a template to synthesize proteins.

Let us look at transcription and translation more closely. DNA and RNA are both nucleic acids made up of nuclotides, while proteins are made up of amino acids. When DNA is transcribed to RNA, it's complement is paired to it. DNA basis A, G, T, and C are transferred to RNA basis U, C, A, and G, respectively. In short, the DNA alphabet is transcribed to the RNA alphabet. The encoding of proteins is done in groups of three RNA nucleotides, known as codons. Since there are $4^3 = 64$ possible combinations of nucleotide triplets and

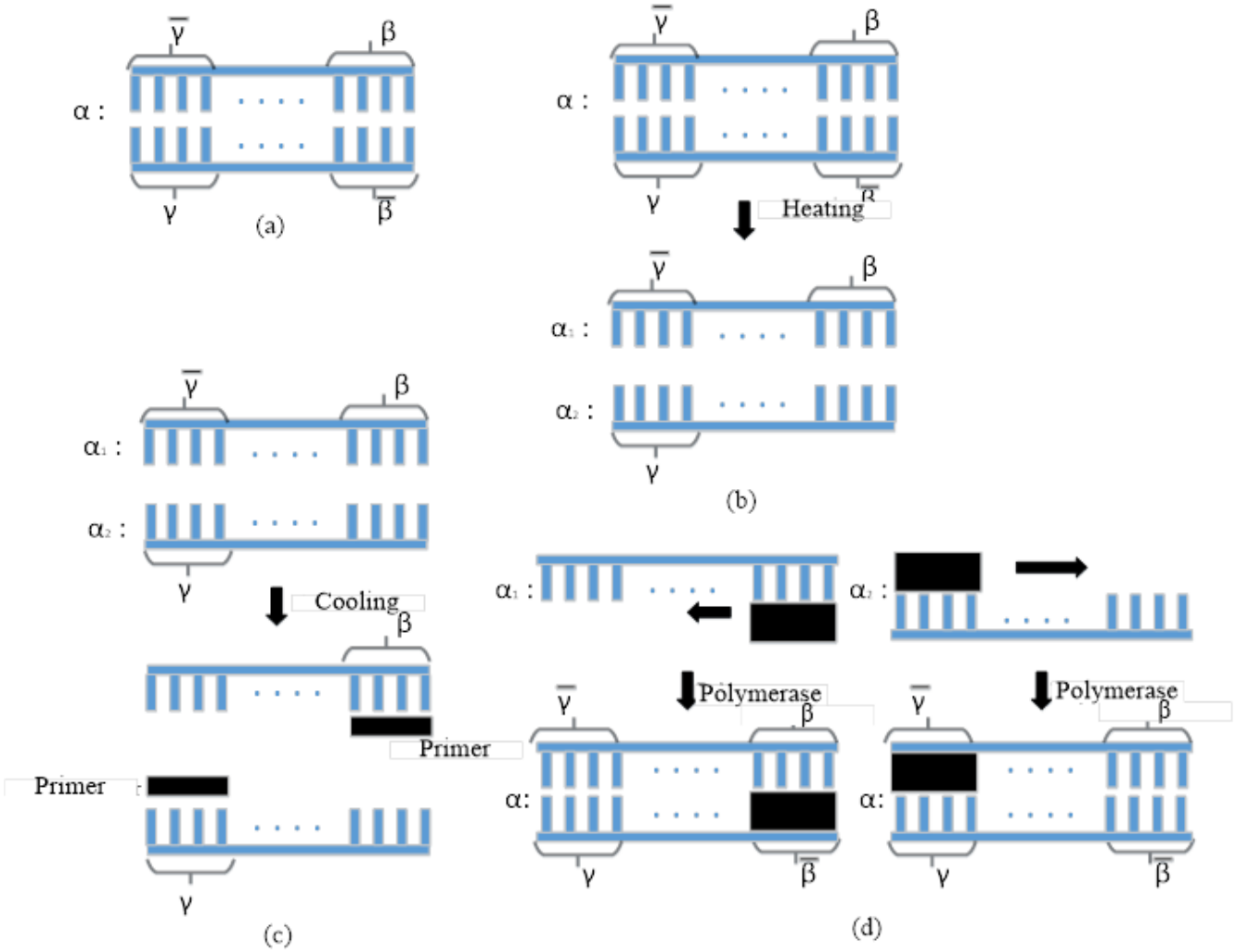


**FIGURE 6.** Polymerase Chain Reaction (PCR): (a) The sample DNA strand $\alpha$ with borders $\beta$ and $\gamma$ (b) Process of denaturation (c) Process of priming (d) Process of extension [10].

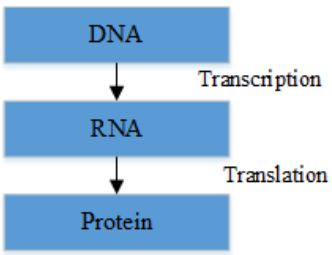


**FIGURE 7.** Basic idea of the Central Dogma.

only 20 amino acids, multiple codons may map to the same amino acid, as shown in Table 1. Fig. 7 summarizes how DNA information propagates to form a protein. The central dogma constitutes the basis of many DNA cryptographic schemes.

### 13) DNA MICROARRAY

DNA micro-arrays or DNA chips are an important tool in DNA analysis. The technology involves a huge 2-D or 3-D array with copies of a single-stranded DNA fragment in each slot, also called a DNA probe, attached to a solid surface with covalent or non-covalent bonds [11]. Millions of DNA probes can be assembled in less than one square inch of the area [17]. For example, DNA from two different cells of interest is added over the microarray of DNA probes. The added DNA from cell one will anneal (hybridize) to certain probes to a greater or lesser extent than the DNA from cell 2, highlighting the unique characteristics of each cell. Thousands of data points can be generated in one experiment to support such research.

### 14) SEPARATION BY HYBRIDIZATION

Another notable application of DNA micro-arrays is in DNA separation through hybridization. The separation method extracts single strands containing a specific short sequence X of DNA from a test tube of DNA sequences [11]. Many copies of X's complement are generated and bound to a micro-array to get started. The test tube is poured over the micro-array containing X's complementary strands. As a result, the

**TABLE 1. Codon to amino acid mapping [16].**

| | U | C | A | G | |
|---|---|---|---|---|---|
| U | UUU→Phe | UCU→Ser | UAU→Tyr | UGU→Cys | U |
| | UUC→Phe | UCC→Ser | UAC→Tyr | UGC→Cys | C |
| | UUA→Leu | UCA→Ser | UUA→STOP | UGA→STOP | A |
| | UUG→Leu | UCG→Ser | UAG→STOP | UGG→Trp | G |
| C | CUU→Leu | CCU→Pro | CAU→His | CGU→Arg | U |
| | CUC→Leu | CCC→Pro | CAC→His | CGC→Arg | C |
| | CUA→Leu | CCA→Pro | CUA→Gln | CGA→Arg | A |
| | CUG→Leu | CCG→Pro | CAG→Gln | CGG→Arg | G |
| A | AUU→Ile | ACU→Thr | AAU→Asn | CGU→Ser | U |
| | AUC→Ile | ACC→Thr | AAC→Asn | CGC→Ser | C |
| | AUA→Ile | ACA→Thr | AUA→Lys | CGA→Arg | A |
| | AUG→Met Start | ACG→Thr | AAG→Lys | CGG→Arg | G |
| G | GUU→Val | GCU→Ala | GAU→Asp | GGU→Gly | U |
| | GUC→Val | GCC→Ala | GAC→Asp | GGC→Gly | C |
| | GUA→Val | GCA→Ala | GUA→Glu | GGA→Gly | A |
| | GUG→Val | GCG→Ala | GAG→Glu | GGG→Gly | G |

**TABLE 2. Binary code for DNA and RNA [18].**

| Binary Sequence | DNA Nucleotide | RNA Nucleotide |
|---|---|---|
| 00 | T | U |
| 01 | G | G |
| 10 | C | C |
| 11 | A | A |

**TABLE 3. Valid combination of DNA digital code [19].**

| | | | |
|---|---|---|---|
| CTAG | CATG | GTAC | GATC |
| TCGA | TGCA | ACGT | AGCT |

strands containing X will anneal to the complementary strands. Finally, the array is washed, and all the strands that did not anneal are removed, leaving only the strands containing X.

### C. DNA DIGITAL CODING

DNA computing starts with determining how to represent binary information as DNA strands. This section introduces general DNA coding techniques commonly used in DNA-based cryptography.

A basic way to translate binary data into DNA strands is to use conversion Table 2 [18]. Since DNA is made up of four base nucleotides, we need four different combinations to represent the data in the DNA strands. Since in RNA, T is replaced by U, and the table also depicts the binary to RNA data conversion. The four base nucleotides can be combined in 24 different ways, assuming each occurs only once in a block of 4, but since the formation of double-stranded DNA follows Watson-Crick's complementarity, out of the 24 combinations, only 8 are valid representations, as shown in Table 3 [19]. Thus, for binary sequence 111001000110, the corresponding DNA sequence would be "ACGTGC."

**TABLE 4. Clelland's DNA coding [4].**

| Character | DNA Triplet | Character | DNA Triplet |
|---|---|---|---|
| A | CGA | U | CTG |
| B | CCA | V | CCT |
| C | GTT | W | CCG |
| D | TTG | X | CTA |
| E | GGC | Y | AAA |
| F | GGT | Z | CTT |
| G | TTT | 0 | ACT |
| H | CGC | 1 | ACC |
| I | ATG | 2 | TAG |
| J | AGT | 3 | GCA |
| K | AAG | 4 | GAG |
| L | TGC | 5 | AGA |
| M | TCC | 6 | TTA |
| N | TCT | 7 | ACA |
| O | GGA | 8 | AGG |
| P | GTG | 9 | GCG |
| Q | AAC | | ATA |
| R | TCA | , | TCG |
| S | ACG | . | GAT |
| T | TTC | : | SCT |

**TABLE 5. DNA coding based on complementarity [11].**

| Binary Bit | DNA Nucleotide |
|---|---|
| 0 | A |
| 1 | G |
| $\overline{1}$ | C |
| $\overline{0}$ | A |

On the other hand, in [4], Clelland introduced a different representation, shown in Table 4. A triplet of nucleotides represents each character (A-Z), number (0-9), and punctuation mark.

DNA mapping is required to allow DNA strands to be programmed and computed. The mapping can work with a universal Turing Machine. Amazingly, the natural feature of DNA complementarity provides such an opportunity in DNA computing [19]. Using DNA's complementarity, if A and G are represented as A:= 0 and G:= 1, we can then represent T and C as their complements, $\overline{0}$ and $\overline{1}$, respectively (Table 5, [11]). This representation leads to a binary alphabet $\{0, 1, \overline{0}, \overline{1}\}$ which is also the alphabet of a universal language, named the Twin Shuffle language (TS) [11], [20].

TS is defined as follows: for a binary word $x$ over $\{0, 1\}$, $\overline{x}$ represents a word over $\{\overline{0}, \overline{1}\}$ where each letter of $x$ is complemented in $\overline{x}$. Then, a word $y$, generated by shuffling $x$ and $\overline{x}$, will be in TS if shuffling does not change the order of the letters in $x$ and $\overline{x}$. For example, if $x = 010101$ then $\overline{x} = \overline{0}\,\overline{1}\,\overline{0}\,\overline{1}\,\overline{0}\,\overline{1}$, and the shuffling of $x$ and $\overline{x}$ will produce a random '$y$' as shown below:

$$y_1 = \overline{0}\,\overline{1}\,0\,1\,\overline{0}\,\overline{1}\,0\,1\,\overline{0}\,\overline{1}$$
$$y_2 = \overline{0}\,\overline{1}\,\overline{0}\,\overline{1}\,\overline{0}\,\overline{1}\,0\,1\,0\,1\,0\,1$$
$$y_3 = 0\,1\,\overline{0}\,\overline{1}\,1\,0\,\overline{0}\,\overline{1}\,1\,0\,\overline{0}\,\overline{1}$$

Here, $y_1$ and $y_2$ are in TS, but $y_3$ is not since the order of the letters in $x$ is changed in $y_3$.

To relate the Twin-Shuffle Language with DNA, we define the alphabet $\Sigma_{DNA} = \{A, G, C, T\}$ which can be represented in binary as $\{0, 1, \bar{1}, \bar{0}\}$. For example, the following is a double-stranded DNA:

AATCGCACTG
TTAGCGTGAC

And, using the alphabet above, it can be expressed in binary as:

$$0\,0\,\bar{0}\,\bar{1}\,1\,\bar{1}\,0\bar{1}\,\bar{0}\,1$$
$$\bar{0}\,\bar{0}\,1\,\bar{1}\,1\,\bar{0}\,1\,0\,\bar{1}$$

If we read this DNA strand using the conventional way of reading DNA strands (from left to right), it will produce the following binary string,

$$Y = 0\,0\,\bar{0}\,\bar{1}\,1\,\bar{1}\,0\,\bar{1}\,\bar{0}\,1\,\overline{000}\,1\,\bar{1}\,1\,\bar{0}\,1\,0\,\bar{1}.$$

This binary word belongs to TS, where the upper strand represents $x$, and the lower strand represents $\bar{x}$. This interconnection between DNA strands and TS shows that any DNA strand following this representation can work with a Turing Machine as TS is a Turing Machine acceptable language [11], [20].

Moreover, in [21] and [22], a new grammar $G_{rand} = (\Sigma, V, R, S)$ for random number generation is introduced using DNA coding. Here, $\Sigma := \{0, 1, s, e\}$, is the alphabet set and V: $= \{A\}$ is the variable set of the grammar. In the alphabet set, 0 and 1 represent bits, and $s$ and $e$ denote terminal (start and end) symbols. Rule $R$ is defined as:

$$S := sA$$
$$A \rightarrow 0A$$
$$A \rightarrow 1A$$
$$A \rightarrow e$$

These rules were constructed following 'rule molecules', presented in [23] and denoted as algomers. Algomers are short double-stranded DNA sequences with sticky ends, as shown in Fig. 8. In the figure, the double-stranded core sequences of the algomers represent terminals, and their sticky ends represent variables. Each of the algomers shown in Fig. 8 (a), (b), (c), and (d) represents a specific rule of grammar. The algomers in Fig. 8 (b) and (c), corresponding to $A \rightarrow 0A$ and $A \rightarrow 1A$, can concatenate in either direction, where $A'$ represents the complementary strand of $A$. The algomers in Fig. 8 (a) and (d) act as terminators and can concatenate only right or left, respectively. Here, $H$ and $B'$ work as restriction sites. Fig. 9 shows the word representations of $G_{rand}$, called logomers. The molecular representation of the algomers is shown in Fig. 9 (a), and the logomer representation is shown in (b). The logomer represents a word in the form $s\,\{0|1\}\,e$. As shown in the figure, any arbitrary number of bits can fit between the start and end brackets.

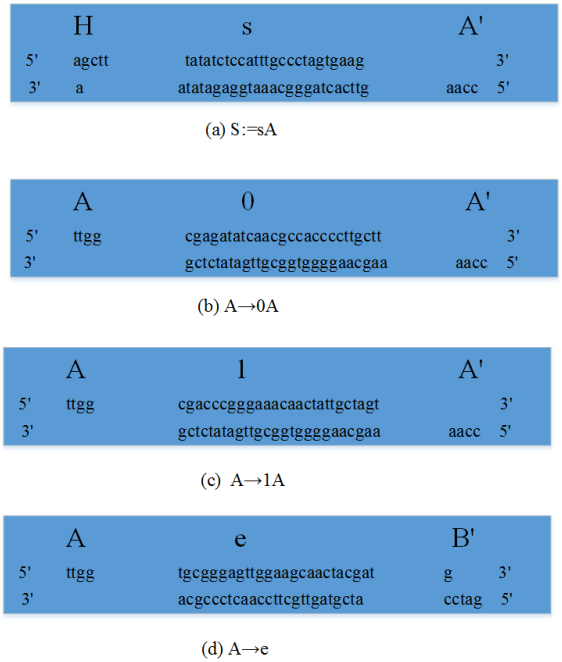


**FIGURE 8.** Algomers of $G_{rand}$ [21].

Another representation technique of DNA molecules uses self-assembling DNA tiles [24], [25], [26], [27]. Winfree first introduced the concept of self-assembling tiles [28]. This technique encodes each bit of a binary message using a single tile. The synthetic implementation of DNA tiles simulates the design of Wang tiles [29], which is Turing-universal in computation [30]. Thus, the self-assembling tile technique is also Turing-universal [28]. DNA tiles are formed when a DNA chain bonds with one chain to form one helix structure, and then the uncompleted end crosses over and bonds with another chain into another helix. Extending this process, multiple helix structures can intertwine to form various shapes, following a self-assembly algorithm [31]. Fig. 10 shows how the uncompleted sticky ends of helices in tile one and tile two bind with each other to form a larger structure.

Self-assembling DNA tiles have been used to model NP-complete problems with complex constraints, such as the timetable/graph coloring problem [26], [32]. In DNA-based-cryptography, the DNA tile model has been used in several works [26], [27], [31], [33], [34], [35].

## III. NATURAL DNA CRYPTOGRAPHY

Natural DNA cryptography is one of the most important applications of DNA computing. Researchers in this field apply cryptographic techniques to sequences of DNA. Most of these works are based on a one-time–pad (OTP) encryption scheme. In the cryptographic world, OTP provides perfect secrecy in principle through the randomness of the key, which should be as long as the plaintext and used only once [36]. However, generating a message-long, truly random key for each encrypted message is not feasible. In addition, storing a

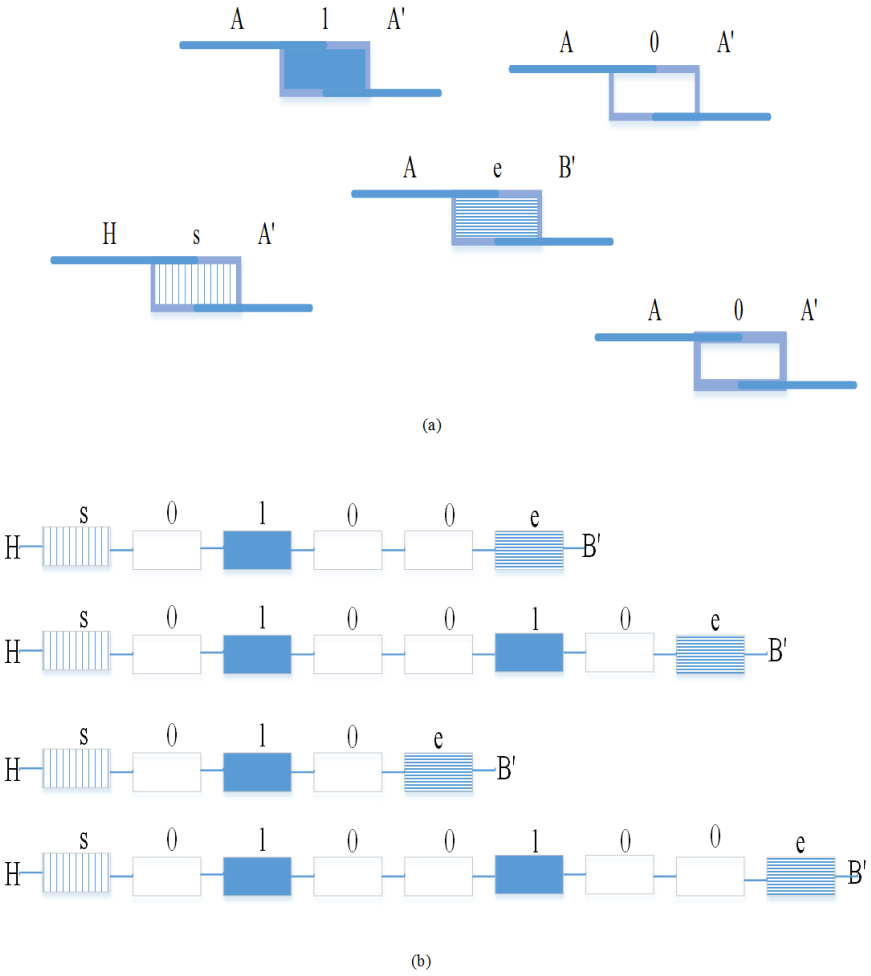


**FIGURE 9.** Molecular and Logomer Representation: (a) Molecular representation of algomers (b) logomers of $G_{rand}$ [21].

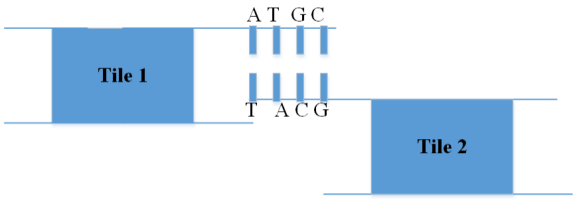


**FIGURE 10.** Self-Assembly of two different tiles [31].

vast amount of keys requires large amounts of space, which further hinders the use of OTP in cryptography. In DNA-based cryptography, researchers mitigate this problem using DNA's vast storage capacity to hold long, unique OTPs [37]. Biomolecular operations such as ligation, PCR, and gel electrophoresis are applied to generate the DNA sequences that make up the OTP and to support the substitution and XOR techniques used for encryption and decryption. Another notable advantage, the massive parallelism of DNA operations, makes it possible to produce millions of distinct short DNA sequences on a chip of $\sim$ 4 $\times$ 4 cm [37]. Furthermore, DNA's vast parallelism was successfully used in DNA-based cryptanalysis to break the Data Encryption Standard (DES) described in [5] and [6]. This section will outline the key research areas of natural DNA cryptography.

### A. METHODS OF NATURAL DNA CRYPTOGRAPHY

The first DNA-based cryptographic scheme was introduced in [37]. The authors proposed an OTP substitution-based method for DNA cryptography and exemplified it on a 2D micro-array.

The input plaintext message of length $n$ was partitioned into plaintext words of fixed lengths. First, a table of plaintext-ciphertext associations was constructed, mapping all possible plaintext strings (or words) of a given length to associated ciphertext strings. Each mapping in the table was unique, such that one plaintext word is associated with exactly one cipher word and vice versa. The OTP was then formed by constructing a large DNA strand of all the plaintext-cipher word pairs in the table. The OTP was therefore composed of a series of repeated units, each consisting of a plaintext word $B_i$, a cipher word $C_i$, and a stopper sequence. Fig. 11 (a) shows the structure of the OTP. The stopper sequence limits the growth of one unit beyond the plaintext word-cipher word

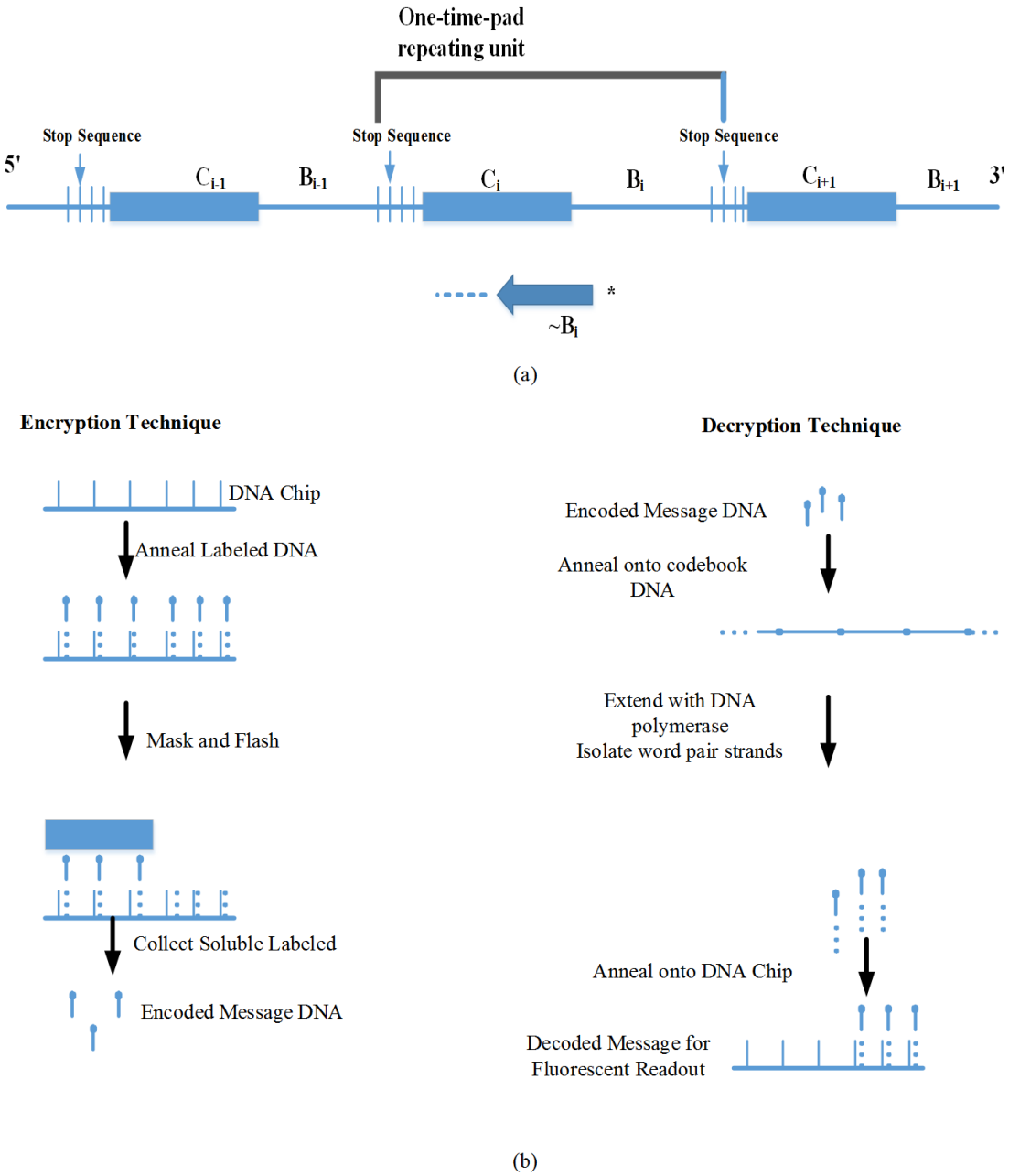


**FIGURE 11.** (a) One-time-pad codebook of DNA sequences, (b) Encryption and decryption [37].

pair. The number of such OTP units was denoted as $d$, where:

$$d = n/(L_1 + L_2 + L_3)$$
$$L_1 := \text{length of plaintext word}$$
$$L_2 := \text{length of cipher word}$$
$$L_3 := \text{length of stopper sequence}$$

The OTP was kept secret and shared in advance between the sender and the receiver. The encryption scheme, which applies to natural DNA and binary data coded as DNA, was demonstrated experimentally using chip-based DNA microarray technology and a 2D image as input/output. As a first step, the OTP was extended into a DNA strand using PCR and complemented plaintext words $\sim B_i$ as primers. The word pair DNA strand obtained this way from the OTP was fluorescently labeled and then bound to the DNA chip. Next, the plaintext message was applied to the chip causing the corresponding cipher words to separate and be collected. Fluorescent microscopy was used in this process. Finally, the collected ciphertext was transmitted to the receiver. The decryption process required an OTP and DNA chip identical to the ones used for encryption. To decrypt, first, a DNA word pair strand was constructed from the received ciphertext. Like its corresponding encryption step, the process used

PCR, this time with cipher words as primers and the OTP as a template. The reformed word pair strands were bound to the DNA chip, and the plaintext message was read using fluorescent microscopy. The encryption and decryption processes of the OTP substitution scheme are depicted in Fig. 11 (b).

In [38], Chen proposed a bio-molecular method for data encryption. The work encoded the plaintext message in a solution of DNA and then encrypted the encoded data using an OTP. The author used a carbon nanotube-based probe to transform data between DNA and conventional binary storage media. For encryption, the author used an OTP. However, bitwise modulo-2 addition of message words to OTPs, rather than substitution, was utilized. Fig 12 represents this method. Fig. 12 (a) shows the DNA representation of all possible non-negative binary bits to be added. Fig. 12 (b) illustrates the addition of binary bits 1 and 1. Here, the vertical dotted lines denote hybridization between complementary DNA elements and reiterated arrows are used to represent primer extensions. The resulting DNA strand is produced through primer extension. The binary addition algorithm proposed theoretically was also executed biochemically, yielding the expected results.

In [26], Chen and Xu implemented a cryptographic method using DNA tiles. Before this work, the application of bio-molecular computation in cryptography or steganography included a series of biochemical reactions which required continuous human monitoring. The authors outlined the difficulty of time-consuming laboratory procedures, which increased with input size. Therefore, they proposed a technique based on self-assembly DNA tiles that simultaneously achieved a secure OTP system's key uniqueness and randomness requirements. The computation with self-assembly tiles was first introduced by Winfree, as mentioned before in [28] and [39]. The tiles contained sticky ends to link to other DNA tiles. The technique facilitated further assembly and produced DNA tiling lattices. Using the self-assembling capability, the authors proposed a DNA XOR cryptosystem using random OTPs. The method included four different systems: an encryption system, a ciphertext-extracting system, a key-extracting system, and a decryption system. All the tile systems use O(1) input tiles and O(n) steps. First, the DNA tiles were generated. The tiles are shown in Fig. 13. Fig. 13 (a) shows the set of input tiles used to represent message bits 0 and 1 and two other tiles, $S$ and $E$, used to represent the start and end of a sequence. Each tile is identified by four sides named $<$ north, east, south, west $>$, and a value placed in the center. Fig. 13 (b) shows a sequence of tiles representing a message $m$ of $n$ input bits. The XOR tiles for encryption are shown in Fig. 13 (c). The tile input and output are on the south and north sides, respectively. The east and west sides of the tiles are used to link to other tiles. The value of the tile (a, b) represents the input bit and the ciphertext, respectively. Here, $a, b \in \{0, 1\}$ where, '$a$' represents the input bit and '$b$' represents the output bit of the operation.

Since the XOR tiles set the input bits to 0 or 1 with equal probability, the resulting encryption was truly random when the process repeated. To achieve equal probability, the 4 XOR tile types (0 to 0, 0 to 1, 1 to 0, and 1 to 1) depicted in Fig. 13 (d) were added to the reaction buffer in the same concentration. Encryption using tile system $E_{ab}$ produced the string "$m_i c_i$," stored as the tile value. By using the XOR tiles this way, the encryption results were randomized.

Next, in the ciphertext extracting system $C_{ab}$, the authors used the tiles shown in Fig. 13 (e). In the ciphertext extracting tile, the south and north sides represented the input and the copy of the input, respectively. The ciphertext bit was stored as the tile's value, indicated in the figure as '$b$.' As shown in Fig. 13 (f), for each input combination of bits $ab$, the tile extracted the second bit, $b$, which was stored as the tile's value. E.g., for input 01, the tile value was 1, while for input 10, the tile value was 0. Fig. 13 (f) shows the four extraction tiles of the ciphertext.

The key extracting process started next using the key extraction system $K_{ab}$. Fig. 13 (g) shows the key extracting tile, with the input on the south side, as before. Here, the output of the key computing function $k(a, b)$ was $y$. The input string $m_i c_i$ representing plaintext and ciphertext bits was denoted by $ab$. Thus $y = anXORb$ was the XOR result of the input bits, and it was stored as the tile value, as shown by the four key tiles in Fig. 13 (h). After creation, all the tiles were placed into a reaction buffer, and the self-assembling process was started. Human involvement was not required after this point. An example of encrypting message $=$ 1010011 with key-value 0010110 is shown in Fig. 13 (i).

The reporter strand, the strand resulting at the end of the encryption process, was identified using the ligase enzyme. The reporter strand contained the input message, the key, and the ciphertext. Finally, the reporter strand was extracted by melting the hydrogen bonds between the strands, using purification and PCR for amplification. Purification of the strand was done using gel electrophoresis. Once the key was extracted, it could be shared secretly with the receiver. To this end, the authors also proposed a key transfer protocol through ligation, immobilization, and amplification.

One drawback of the DNA XOR-based cryptosystem is the possible loss of synchronization between the message and the key or between the key and the ciphertext. In [27] and [35], the authors mitigate the error problem by introducing a tile-based multi-layer algorithm that increases the fault tolerance against mismatches. For example, in [35], blunt end tiles were used to calculate and produce reporter strands. The four types of key tiles and cipher tiles used in the process are shown in Fig. 14. The extraction procedures were repeated to produce a large sheet. The sheet can be observed under an atomic force microscope (AFM). According to [40], complementary DNA strands easily attach to the AFM cantilever. Thus, the authors did not use ligation to identify the reporter strand. Rather, they used PCR and assembled the tiles individually.

The proposed work has been analyzed against possible DNA cryptography errors. The authors overcame the

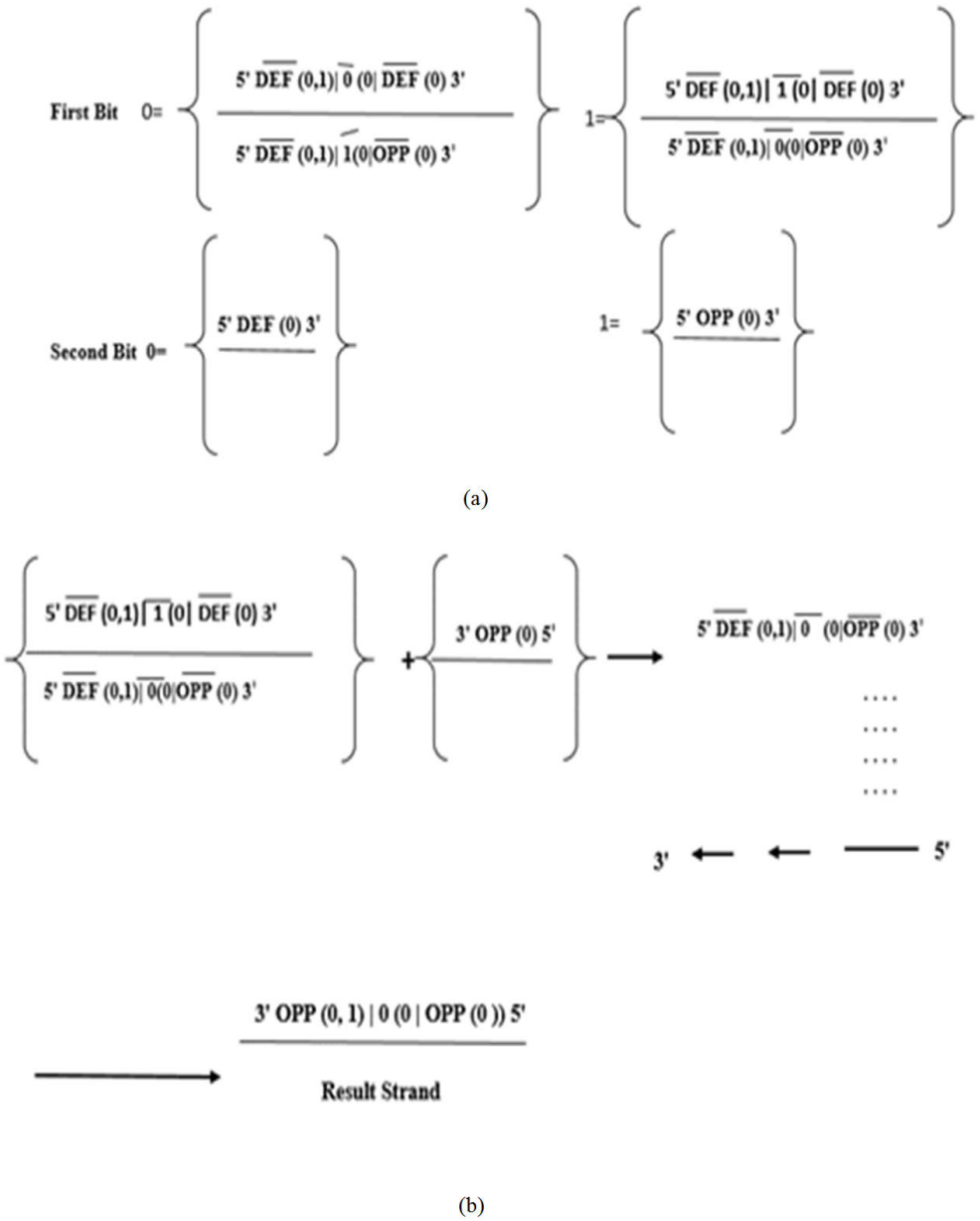


**FIGURE 12.** (a) DNA representation of all possible non-negative binary bits (b) Example of DNA-based addition of two binary numbers [38].

synchronization problem and presented an OTP cryptosystem with a significant error tolerance rate with the efficient use of the hybridization process.

Another implementation of self-assembling DNA tiles is in [41]. The work implemented the Elliptic Curve Diffie-Hellman Key Exchange protocol using self–assembling DNA tiles. The authors designed DNA tiles to implement scalar multiplication. After the required tiles are generated, self-assembly is ensured by mixing the tiles in a reaction buffer. PCR and gel electrophoresis are applied to read the result of the scalar multiplication. By extracting the result, the key exchange protocol is implemented over the elliptic curve.

In [19], Cui et al. proposed an encryption scheme with the help of DNA technology. Their proposed method used both traditional cryptographic techniques as well as DNA technology. The authors applied DNA synthesis and PCR amplification to the ciphertext to design a secure cryptographic method. Traditional cryptographic techniques, such as DES or RSA, produced the ciphertext. As we know, PCR requires a correct primer pair to amplify a message accurately, with the primer pair acting as a key in the encryption process. Thus, in the key generation phase, the encryption key was generated as a combination of the receiver's public key and the PCR primer pair. The decryption key also combined the receiver's private key and the primer pair. The primer pair

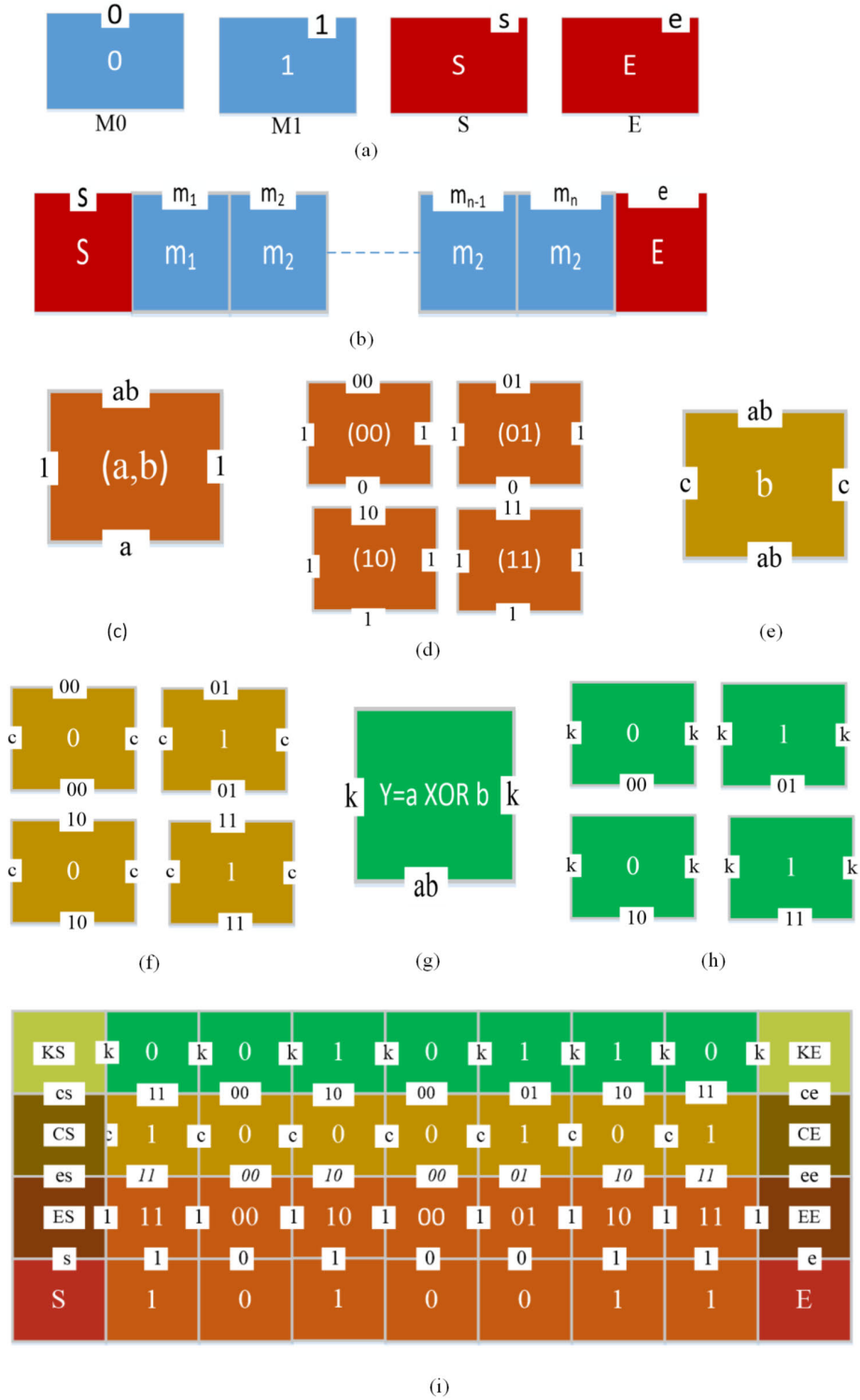


**FIGURE 13.** All the tile representations required for encryption: (a-b) input message m, represented as tiles, (c-d) the XOR tiles, (e-f) ciphertext tile, (g-h) key tile, (i) the whole encryption process [26].

was a combination of two primers, where one primer was generated by the sender and the other by the receiver. The sender created ciphertext C using the receiver's public key. This process was named Data Pretreatment. The ciphertext was then converted into DNA sequences using a general digital DNA coding technique. The resulting DNA sequences were synthesized and bounded by forward and reverse PCR primers to create the secret message. Finally, the secret message was mixed with dummy DNA sequences before transmission.

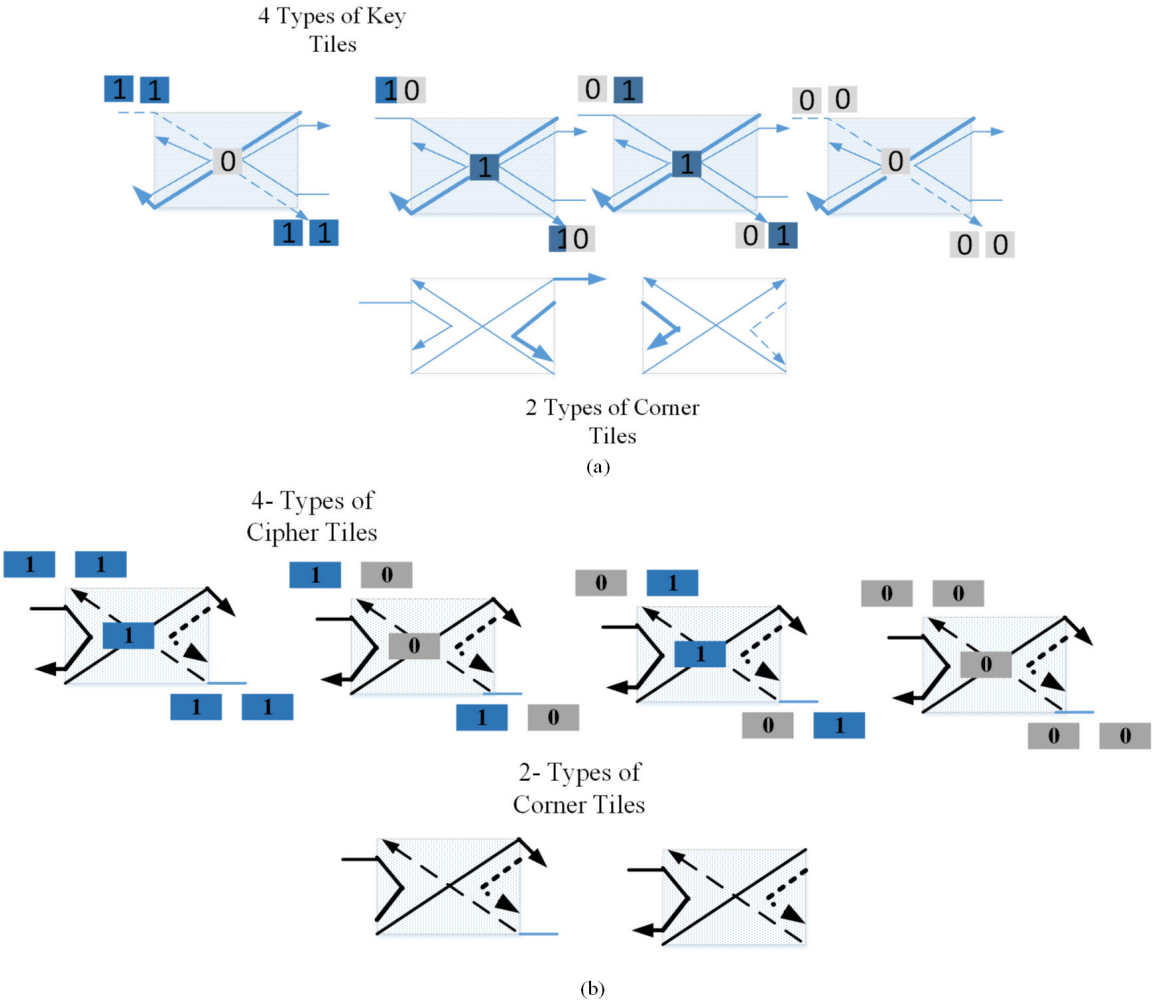


**FIGURE 14.** Four kinds of (a) key tiles (b) cipher tiles [27].

The receiver's decryption process involved PCR amplifying the secret DNA sequences using appropriate PCR primers and then converting the sequence into binary ciphertext C. The message was decrypted from the ciphertext using the receiver's private key. The whole process is depicted in Fig. 15 as a flowchart [19].

A DNA-aided key distribution technique for public key cryptography was proposed in [42]. At the sender, the secret message, consisting of the secret key, was hidden in a pool of dummy DNA sequences and placed between 2 primers for easy identification by the receiver. Both the sender and the receiver had prior knowledge of the primer pair, which played the role of a public key. At the destination, knowing the primer pair, the receiver restored the secret message using PCR amplification followed by sequencing. One drawback of this approach is that the data is not encrypted, and the scheme does not protect against brute force attacks [43].

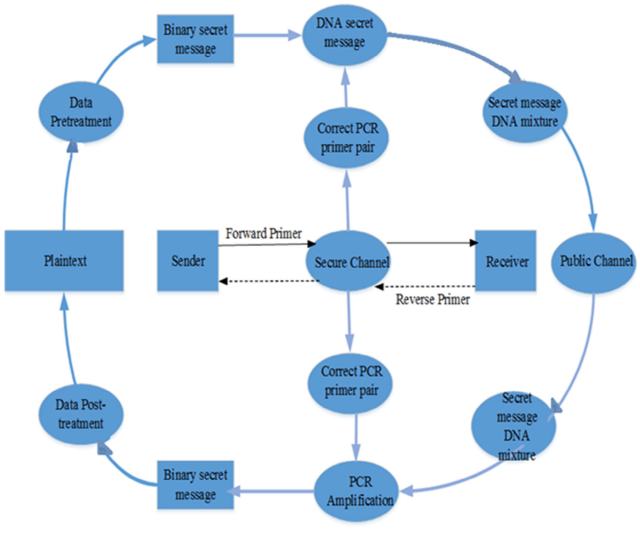


**FIGURE 15.** Encryption-decryption process [19].

### B. CRYPTANALYSIS WITH NATURAL DNA

DNA-based cryptosystem focuses on implementing encryption techniques and does cryptanalysis using DNA computations. For example, in 1995, Dan Boneh, Christopher Dunworth, and Richard J. Lipton applied molecular operations to break Data Encryption Standard (DES) [5]. Using

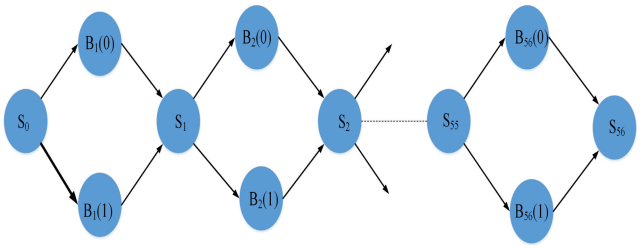


**FIGURE 16.** Initialization Graph [5].

a library of operations performed in a molecular computer, this system could accomplish DES key recovery in about four months of work [44], [45]. Also, using preprocessing techniques in a chosen-plaintext attack, the authors achieved subsequent DES key recovery in one day. Their method involved brute force attacks in DES. When the cryptanalyst obtains plaintext and its corresponding ciphertext, one can apply all possible key combinations to the plaintext and generate the matching ciphertext. The DES key is revealed and then used for future cryptanalysis. As the key size of DES is 56 bits long, there are $2^{56}$ different key patterns that could be applied to the plaintext [46]. The massive parallelism of DNA computing and the vast storage capacity of DNA strands help perform such brute-force attacks on a plaintext-ciphertext pair. The authors in [5] prepared a solution of DNA, denoted as $T_f$. The solution contained pairs of all possible ciphertext values resulting from the plaintext using all possible key combinations [47]. That is, the solution contained all possible key-ciphertext pairs. More formally, this was expressed in equation $g(k) = DES(M_0, k)$ where $k$ was the key, $M$ was the message, and $g(k)$ was a function that maps a 56-bit key to a 64-bit ciphertext. The goal was to find a key $k$, such that $g(k) = E$, where $E$ was the matching ciphertext, [48]. The inverse function of $g(k) = E$ needed to be calculated to do that. The construction graph of the initial solution of all possible $2^{56}$ DNA strands, representing all possible DES key combinations, is illustrated in Fig. 16. Each path from $S_0$ to $S_{56}$ indicates one DES key combination.

The researchers evaluated the XOR function in a DNA solution to perform DES operations on plaintext with key $k$. To evaluate the XOR of $i'$th and $j'$th bit of a string, the value of $x_i$ XOR $x_j$ is appended to the strands representing $x$. In order to do that, solution $T$ was separated into two different solutions, $T_0$ and $T_1$, representing $x_i$ XOR $x_j = 0$ and $x_i$ XOR $x_j = 1$ respectively. Following, $T_0$ and $T_1$ were tagged with the appropriate values. The authors also implemented a lookup table in DNA solution for S-box mapping operation. Thus, the work has evaluated Boolean gates on a molecular computer and broken DES in 916 steps.

Also, in [49], a theoretical model called Adleman proposes the sticker model. In this model, information representation uses physical strands of DNA in physical substrates. Here, as a method of separation, hybridization is used. But the exciting feature included here is its ability to act as a random

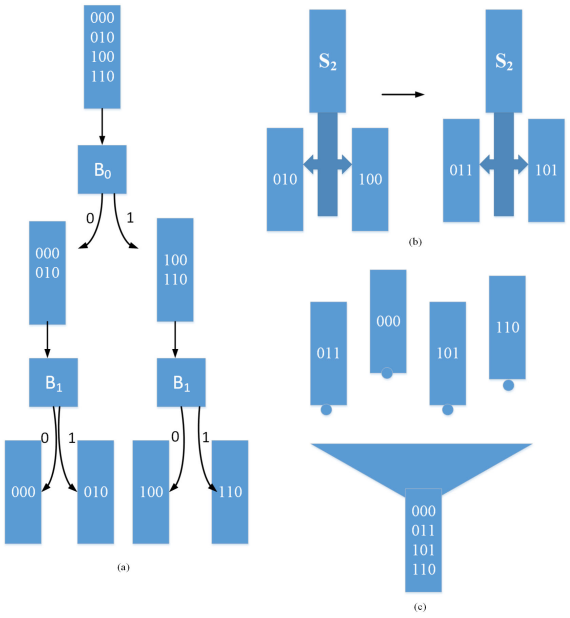


**FIGURE 17.** (a) Parallel Separate (b) Parallel Set (C) Parallel Combine [6].

access memory. It also works without enzymes; according to theory, the materials used in this model are reusable. Using the features of the sticker model, Adleman suggested an attack on DES on a logical level using approximately one gram of DNA [6]. In this work, the authors implemented a known plaintext-ciphertext attack. At first, $2^{56}$ unique ssDNA memory strands were created, each with a length comprising 11580 nucleotides. The researchers defined a memory strand as a region of 579 adjacent blocks: $B_0, B_1, B_2, \ldots\ldots B_{578}$, where 20 nucleotides and a sticker $S_i$ were assigned to each of the blocks. A total of 579 bits were stored in a memory strand since each block could store one bit. To implement the attack, all key combinations are generated and initialized $2^{56}$ memory strands with $2^{56}$ different combinations of the key. To work with the test tubes of physical DNA strands, their sticker model consists of robotics and a microprocessor to control the robotics. These robotics were designed to perform four parallel operations: separate, combine, set, and clear. At first, to initialize memory strands with key combinations, the memory strands were divided into two tubes, and extra stickers were added to each of the tubes to saturate the first 56 blocks. After that, the use of probing, mixing of tubes, heating, and cooling produced 63% of the keys, based on the Poisson distribution.

The authors also implemented basic DES, XOR, and S-BOX operations to create ciphertext from plaintext. Fig. 17 represents the XOR operation performed in their model by parallel separate, parallel set, and parallel combine. The XOR and S-BOX computation computed ciphertext for each key combination in 6655 steps. After generating all the ciphertext, selecting the appropriate ciphertext and its corresponding key required another 64 steps. Thus as a whole, 6719 steps were involved in breaking DES. As the whole work is

a theoretical proposal, its running time depends on the operations' performance.

Another theoretical cryptanalysis is found in [33]. In this work, the authors proposed a method to break the Number Theory Research Unit (NTRU) [50]. They implemented a convolution product on Wang tiles and applied the massive parallelism of DNA operations to emulate a cryptosystem-breaking device.

### C. BIOTECHNOLOGY FOR DNA MANIPULATION

The need for automation and inexpensive laboratory equipment has led to the development of several recent open-source and do-it-yourself (DIY) technologies for DNA manipulation. The paper [51] presents three examples of open-source scientific hardware (OSSH) for working with DNA: a gel scanner, a gel mold for horizontal polyacrylamide gel electrophoresis, and a homogenizer for generating DNA-coated particles. OSSH help produces artificial DNA structures which are used in DNA computing. These tools can be customized depending on the specific research and lab needs. Compared to commercial products, the savings were greater than 50%, up to 90%. The paper [52] proposes designs for high-throughput DNA computing circuits, system architecture, and a compiler, demonstrating their feasibility through simulation experiments to bridge the gap between the existing traditional computing community and DNA computing. The researchers in [53] propose a DNA cryptography technique that integrates DNA encoding and DNA operators into the Feistel network structure using DNA as a carrier. The encryption process is integrated into biotechnical hardware specifically designed with the help of a 3-D printer. Experimental results show that this is an efficient cryptographic method, with the time to perform a brute force attack of nearly 12 million years for only one block and the key space of $2^{80}$. The system has a capacity rate (plain text to ciphertext ratio) of 99% and information entropy values close to 2. Additionally, its implementation has been verified through in vitro experiments.

## IV. METHODS OF PSEUDO-DNA CRYPTOGRAPHY

Pseudo-DNA cryptography research uses pseudo-DNA properties and operations in cryptographic algorithms. These methods use pseudo or virtual DNA rather than natural DNA and simulate DNA operations in cryptographic techniques. For example, in [54], the researchers tried to solve the problem of a chaos-based encryption algorithm for images by combining DNA coding with chaos. In general, image encryption using chaos changes the gray value of a pixel, but it does not change the pixel's position. So, it is easy to attack encryption through the analysis of pixels. The algorithm proposed in this work follows the flowchart shown in Fig. 18.

Here, DNA encoding of binary data produces a DNA matrix to change the position of the pixels. This matrix is then divided into four equally-sized matrices. A permutation sequence generated from the chaotic sequence is applied to each of these submatrices to scramble its values. A DNA

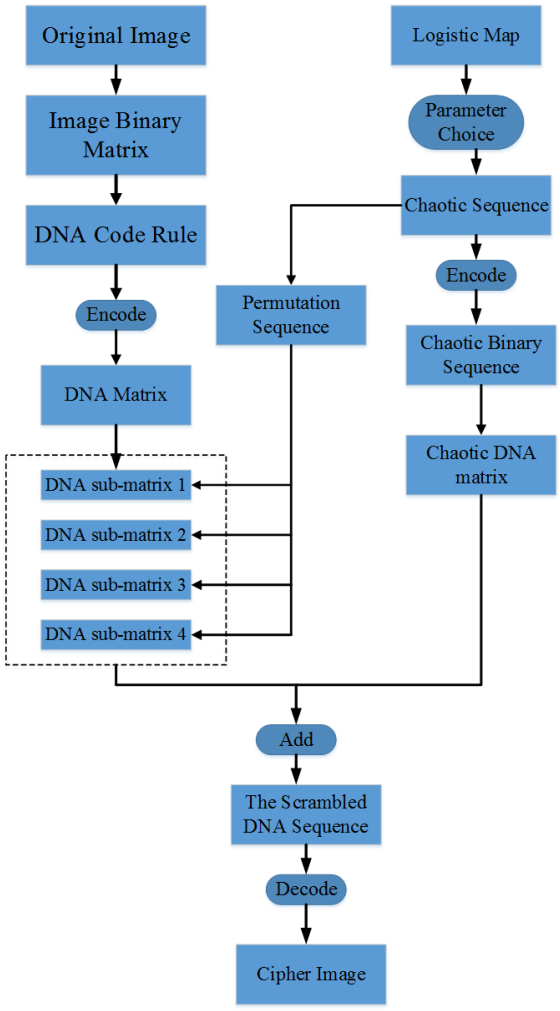


**FIGURE 18.** Encryption technique [54].

matrix of the size of the submatrix is also produced from the logistic map. The resulting matrices are added and decoded as an image file to obtain the encrypted image. The authors used addition and subtraction operations on DNA alphabets in the encryption process.

Pseudo-DNA cryptography utilizes synthetic DNA-like structures as cryptographic keys. These structures incorporate modified nucleotides or artificial sequences designed to mimic the behavior of natural DNA. Within pseudo-DNA cryptography, various coding schemes can be employed to encode information and enhance the security and functionality of the cryptographic system. Some common coding schemes used in pseudo-DNA cryptography include: 1) Reverse Coding: The DNA sequence is read reversely, where the last nucleotide becomes the first and vice versa. This reversal of the sequence provides an additional layer of complexity and can help increase the security of the encryption. 2) Reverse-Complement Coding: The DNA sequence is reversed and complemented. Each nucleotide is replaced by its complementary base (A with T, C with

G, and vice versa). Reverse-complement coding ensures that the resulting DNA sequence pairs with its original counterpart, facilitating accurate decoding. 3) GC-Content Constraints: This coding scheme imposes restrictions on the percentage of guanine (G) and cytosine (C) nucleotides in the synthetic DNA sequence. By controlling the GC-content, specific characteristics such as stability, melting temperature, or hybridization properties can be tailored to suit the desired cryptographic requirements. 4) Homopolymer-Run Length Freedom: Homopolymers are consecutive repetitions of the same nucleotide in a DNA sequence. This coding scheme aims to avoid long homopolymer runs to prevent difficulties in sequencing or decoding processes. Limiting the length of homopolymers can improve the reliability and efficiency of encoding and decoding. These coding schemes are just a few examples of the techniques employed in pseudo-DNA cryptography. Each scheme introduces specific constraints and considerations to enhance the security and functionality of the cryptographic system ([55], [56]).

A simple but secure encryption algorithm is proposed in [57]. In this work, the authors applied DNA computation to an existing symmetric encryption algorithm [58] to enrich its security. The algorithm is exemplified on a chromosome from the Canis Familiaris Genome [59]. First, each letter of the plaintext message is encoded as a DNA quadruple of nucleotides A, T, C, and G. An efficient searching algorithm is used to locate the positions of each quadruple in the Canis Familiaris Genome. A huge number of matches is found in the genome sequence for each quadruple. For example, a sequence representing the letter A appears in the genome sequence 150386 times. One of these locations is selected at random, and this location substitutes the plaintext letter. The process is repeated for each character of the plaintext. A sequence of pointers (representing the characters' locations in the canis genome) is produced and sent to the receiver over a public channel. This sequence is called the ciphered text. The Canis Familiaris Genome represents the key and is sent over a secure channel in advance. The reverse process is applied to recover the DNA sequence corresponding to the plaintext using the cipher's symmetry at the receiver. The process is outlined in Fig. 19. Using the vast randomness of DNA coding, similar techniques are found in [60] and [61].

In [62], the authors worked on a symmetric key encryption technique with a 128-bit message block and 128 or 256-bit key. The authors used the main features of AES and DES and tried to apply and demonstrate Shannon's principle of confusion and diffusion through a DNA module. The central dogma was simulated in this DNA module to achieve a strong substitution level. As mentioned earlier, in the transcription process of the central dogma, a DNA strand converts into an RNA strand. In the encryption algorithm, the principle of the transcription process is applied as a monoalphabetic substitution technique. The authors also applied a biological XOR (represented in Table 6) in the transcribed RNA strand. Again, based on a genetic code table of 256 amino acids, each element of the bio-XORed results was substituted with

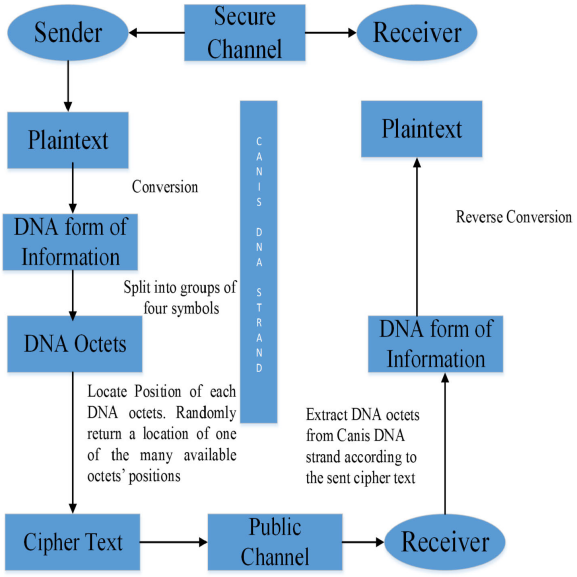


**FIGURE 19.** Proposed encryption and decryption method [57].

**TABLE 6.** The Bio-XOR operation [62].

| Bio-XOR | A | C | G | T |
|---|---|---|---|---|
| A | A | C | G | T |
| C | C | A | T | G |
| G | G | T | A | C |
| T | T | G | C | A |

an amino acid. This substitution is the translation process of the central dogma. The encryption technique is presented in Fig. 20. The pseudo-DNA module represents only one step of the algorithm that applies Shannon's principle through substitution and permutation.

Another work based on the central dogma is found in Kang Ning's work in [63]. Here, the author implemented a symmetric key encryption algorithm. The technique used two different substitution processes based on the central dogma. In this work, the splicing process is changed from its original form. In its initial form, the non-coding areas of a DNA strand (the introns) are identified by two codes representing the start and end of the non-coding zone. This makes it easy to identify introns in the sequence. This work used pattern codes (codes used to determine parts of the sequence that should be kept in a DNA sequence) as identifiers for the introns. Initially, the sender had the plaintext message, the starting code, and the pattern code. Here, the starting code and pattern code acted as a partial key. The sender used these keys to genetic code tablee key. The key generation also generated messenger RNA, which was then translated into protein based on the genetic code table. The genetic code table was also a key used to decipher the message. Thus the translated data, which appeared as a protein, was the ciphertext passed to the receiver. The whole process is represented in Fig. 21. According to the author, the encryption algorithm is robust in

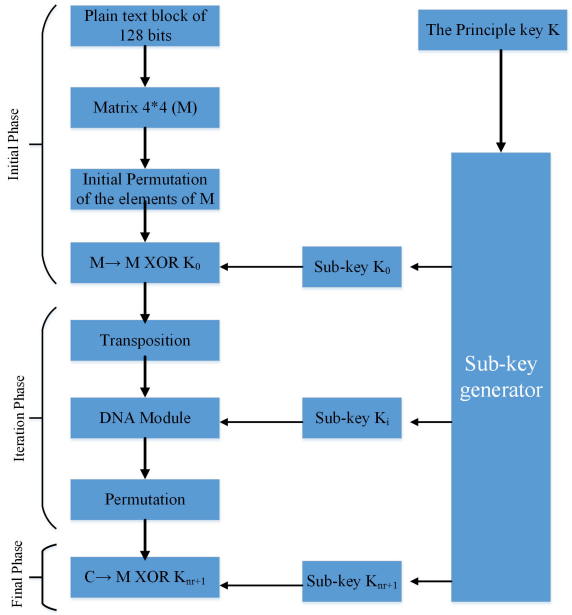


**FIGURE 20.** The encryption algorithm [62].

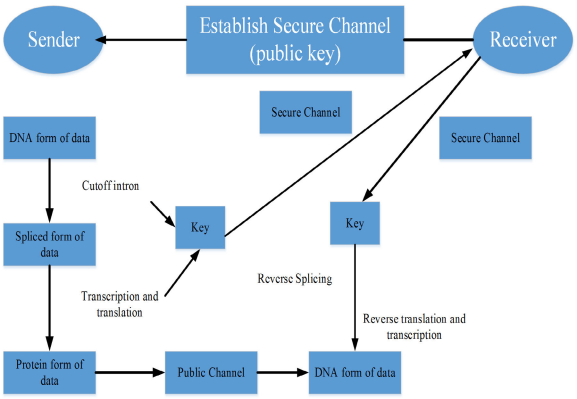


**FIGURE 21.** The pseudo-DNA cryptographic method [63].

the case of a brute force attack but presents certain challenges in the case of a statistical attack.

The asymmetric key cryptographic algorithm, RSA, combines DNA encoding of binary data in [64]. Before applying the RSA encryption technique, the plaintext message is preprocessed into a DNA strand according to Table 7, which is transferred to a number sequence based on Table 8. The RSA encryption algorithm is applied to the number sequence to generate the ciphertext.

Another pseudo-cryptographic approach is found in [65]. It includes a combination of mathematical and biological operations. A minicipher is generated from the original plaintext using matrix manipulation and other mathematical operations in the first part. Given the same plaintext and key, the method generates a different minicipher each time. The second part uses DNA operations and is depicted in Fig. 22. DNA coding and primer pairs are used for added security

**TABLE 7.** Plaintext encoding [64].

| | | | |
|---|---|---|---|
| A-CCA | B-GTT | C-TTG | D-GGT |
| E-TTT | F-TCG | G-CGC | H-ATG |
| I-AGT | J-CGA | K-GAA | L-CGT |
| M-CCT | N-TCT | O-CGG | P-ACA |
| Q-CAA | R-ACT | S-GCA | T-CTT |
| U-GTC | V-TCC | W-GCC | X-ATC |
| Y-AAA | Z-TCA | | |

**TABLE 8.** Mapping from nucleotide to number [64].

| | | | |
|---|---|---|---|
| A-01 | C-03 | G-07 | T-20 |

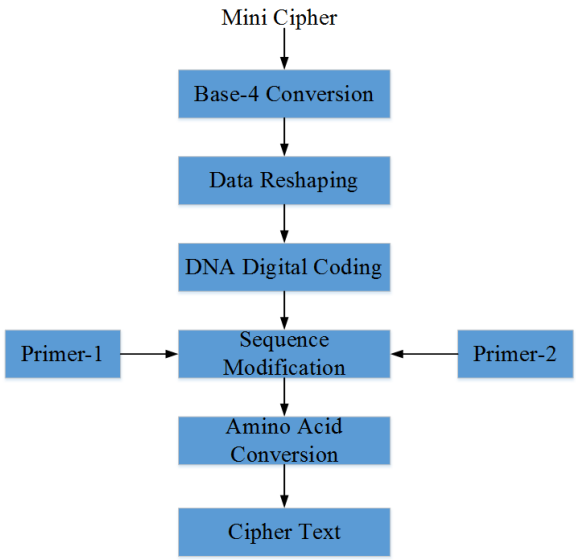


**FIGURE 22.** Ciphertext generation using DNA coding [65].

in this part. The minicipher is coded into DNA sequences, and primer pairs are used as keys to modify the sequences. As the last step, DNA sequences are converted into protein sequences constituting the resulting ciphertext. Decryption reverses the steps used in encryption, given that the scheme is symmetric.

A DNA cryptographic algorithm based on the Vigenere cipher is presented in [66]. At first, binary plaintext is converted into a DNA strand based on Table 9. This coding rule also considers a binary sequence of odd length. To apply the Vigenere cipher, the authors implemented a DNA Vigenere table presented in Table 10. The authors used NCBI (National Center for Biotechnology Information) Gene bank and variable mitochondria to generate a key of length 256 nucleotides.

Encryption uses this key and DNA Vigenere table to convert plaintext into ciphertext. The decryption process applies the reverse operation and generates a plaintext binary message using the DNA coding rule presented earlier. Thus, this cryptographic technique improved the complexity of the Vigenere cipher and enhanced its security.

**TABLE 9.** DNA coding table for [66].

| | |
|---|---|
| 00 | AA |
| 01 | T |
| 10 | C |
| 11 | GG |
| 0 | A |
| 1 | G |

**TABLE 10.** DNA Vigenere Table [66].

| | A | T | C | G |
|---|---|---|---|---|
| A | A | T | C | G |
| T | T | C | G | A |
| C | G | C | A | T |
| G | G | A | T | C |

In [67], an encryption algorithm based on the synthesized PCR amplification technique has been proposed. The whole encryption process is depicted in Fig. 23. First, the binary plaintext is divided into two halves. One half is used as the message, and the other as a key. Then, an XOR operation is performed between the two halves. The result is then converted into DNA strands. Pseudo-PCR amplification is applied to the resulting DNA strand using primer pairs as keys for extra security.

A DNA tile-based pseudo-encryption algorithm was implemented in [68]. Here, Matlab Bioinformatics Toolbox implemented an encryption technique based on DNA tiles and the XOR OTP system. A microcontroller-based system that can determine which bit to add to the tile was proposed to bind the plaintext message into DNA tiles. After the plaintext DNA tiles were generated, an XOR operation with OTPs was applied to create the encrypted message. The authors also proposed the use of restriction enzymes to extract the ciphertext.

Another encryption algorithm applied DNA tiles and their self-assembling property to encrypt image files [34]. Here, five different types of DNA tiles were designed to implement the algorithm. First, the gray values of the images were translated into DNA sequences. These sequences represent original image information in DNA form. After that, proposed DNA tiles were used on the DNA image files to produce new DNA sequences. The sequences were then translated into a gray-valued matrix generating the encrypted image. In [42], a symmetric key cryptographic algorithm was proposed in which DNA computations were used to create a strong cipher. The researchers applied DNA's way of encoding codons with genetic information in their algorithm to strengthen the ciphertext.

## V. DNA BASED STEGANOGRAPHY

Steganography is the science of information hiding. In steganography, the actual message is hidden in the media so that its existence cannot be identified [69]. Since the message is not encrypted but only embedded in an insusceptible manner, researchers sometimes disregard steganography as a means of secure communication, although its use dates back to ancient times. The main power of steganography is to disguise the existence of a message in a cover medium. Most of the works of steganography use images or other multimedia carriers such as audio or video as a medium for data hiding. However, the insertion of a secret message into these media is limited by the number of bits that can be re-purposed before distortions noticeable to the human eye or ear occur. DNA's huge data storage capacity makes the concealment of much larger secret messages possible using a small amount of space. This feature of DNA is the main driver behind DNA-based steganography.

The first DNA-based steganography approach was implemented in [4]. Here, two-layer steganography was applied using the human genome as a cover medium and microdot technology to hide even the cover medium. A microdot is a common form of stenogranography in which a regular-sized photograph is reduced to 1 millimeter in diameter [69]. The microdot can then be placed over a dot or under a postage stamp and sent undetected in letters. In [4], first, the message was encoded into DNA strands which were then disguised into the DNA of the human genome. Second, the DNA strands were stored in microdots. To amplify the message, PCR amplification was required using appropriate primer sequences. Thus, even if an adversary suspected the existence of a hidden message, the message could not be identified in the cover medium without knowledge of the specific primer sequences.

Reference [37] performed a security analysis of one DNA-based steganography technique and proposed methods to improve its security. In the steganography method referred to by the authors, the plaintext message, tagged by strands acting as secret keys, was concealed in a pool of random distracter DNA strands. The secret keys tagged the message, and DNA separation methods were used to find the hidden message. The authors showed how the plaintext could still be retrieved without knowledge of the secret keys. This was proven based on the relative entropy of the distracter stands and the plaintext strand. That is, randomly chosen distracters and the plaintext were likely to exhibit a difference in entropy. The authors discussed two approaches to address the issue. The first was to use distracter strands that mimicked the word distribution of the plaintext strands. The other approach was to compress the plaintext using lossless compression to reduce its relative entropy to the distracters.

In [70], two DNA-based steganography techniques were proposed. The first method created a collection of binary encoded DNA strands by mixing 'dummy' DNA strands with the message strand. These two types of strands were mixed in an equimolar amount. Dummy sequences can be randomly generated DNA strands, such as bacteriophage or herring sperm DNA. To ensure security, the dummy strands must have a similar format to the input message strand. A key sequence is tagged with the message sequence to identify the sequences in the decryption process. PCR and gel-electrophoresis are used to retrieve target sequences from

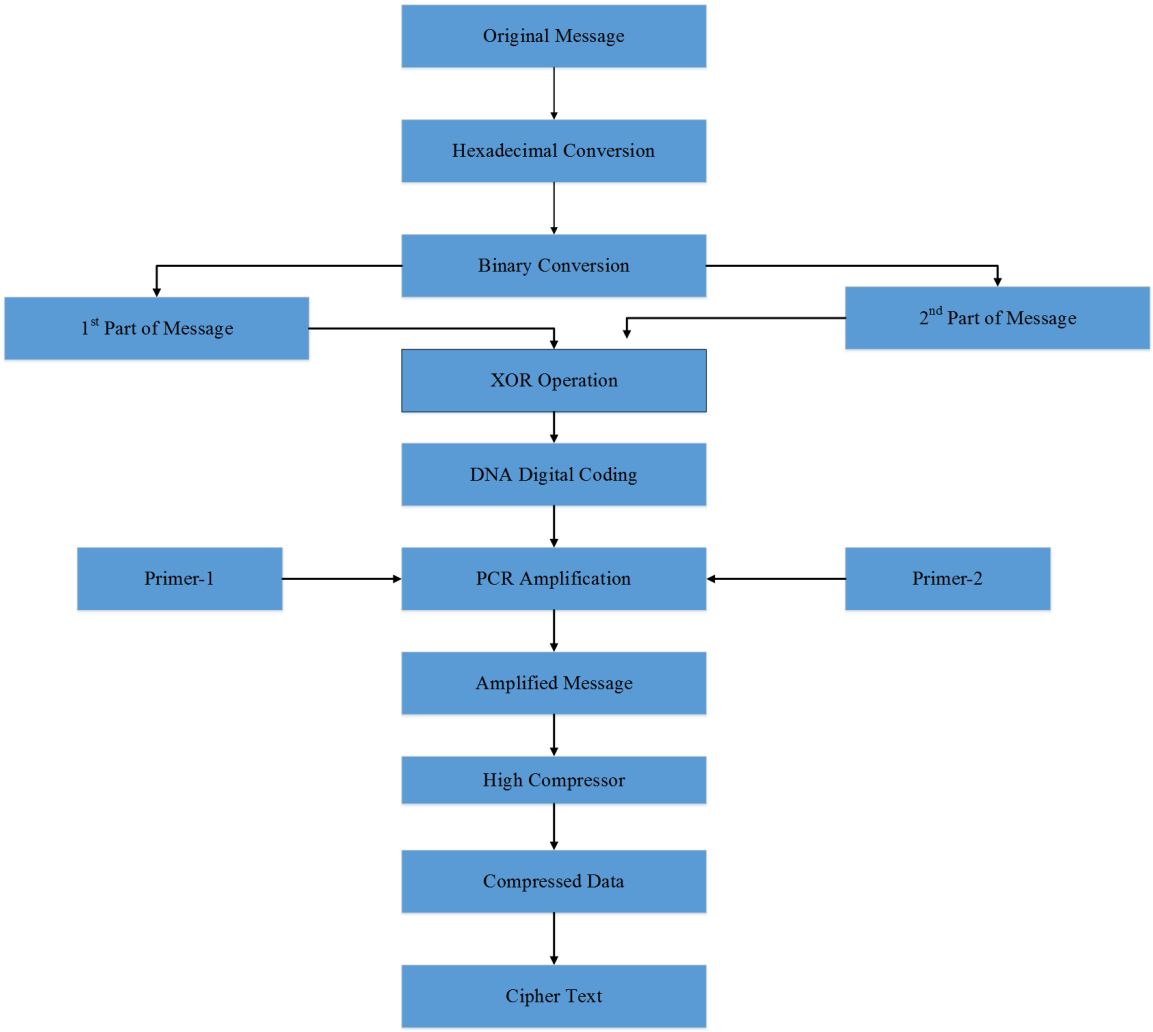


**FIGURE 23.** Bi-serial Encryption Technique [67].

**TABLE 11.** Distribution of amino acids over english alphabets [16].

| Letter | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ambiguity | 4 | 2 | 2 | 2 | 2 | 2 | 4 | 2 | 3 | 1 | 2 | 4 | 1 | 2 | 2 | 4 | 2 | 4 | 4 | 4 | 2 | 4 | 1 | 2 | 1 | 1 |
| A | GCU | UAA | UGU | GAU | GAA | UUU | GGU | CAU | AUU | UGA | AAA | CUU | AUG | AAU | UUA | CCU | CAA | CGU | UCU | ACU | AGA | GUU | UGG | AGU | UAU | UAC |
| C | GCC | UAG | UGC | GAC | GAG | UUC | GGC | CAC | AUC | | AAG | CUC | | AAC | UUG | CCC | CAG | CGC | UCC | ACC | AGG | GUC | | AGC | | |
| G | GCA | | | | | | GGA | | AUA | | | CUA | | | | CCA | | CGA | UCA | ACA | | GUA | | | | |
| U | GCG | | | | | | GGG | | | | | CUG | | | | CCG | | CGG | UCG | ACG | | GUG | | | | |
| Family | GC | UA | UG | GA | GA | UU | GG | CA | AU | UG | AA | CU | AU | AA | UU | CC | CA | CG | UC | AC | AG | GU | UG | AG | UA | UA |

the mixture. In the second approach, digital image processing techniques were used to decrypt the message. The encryption process was the same as in the first approach, but, in this case, the mixing was done between the message strand and a multitude of dummy strands that contained similar key sequences. Thus, the dummy pool was used as a key for decrypting the message in the readout process.

In [71], two different data hiding approaches were proposed for natural or live DNA and chemical DNAs. This theoretical method implements the notion of information hiding in DNA and RNA strands. In the first method, simple binary to DNA encoding techniques were used to insert data into the non-coding regions of a DNA strand. These non-coding regions include non-transcribed (DNA strands not transcribed into RNA in the transcription process) and non-translated (sequences of transcribed DNA that do not participate in protein formation) regions. The non-coding regions also include non-genetic DNAs formed as an output of DNA computing solutions and biotechnology. This straightforward technique seems useful for chemical DNA

watermarking, where the robustness of the scheme depends entirely on the data that is to be hidden. But it is not very strong for live DNA watermarking where the main goal is the protection of intellectual property. Thus the authors proposed a solution for such a situation using codon redundancy. As we know, three bases are used to represent an mRNA code. Hence, $4^3 = 64$ combinations for 20 amino acids. Multiple codons map to the same amino acid, as shown in Table 1. The authors used this redundancy in codons to hide information in live DNA strands. To determine which redundant codon to use to replace the actual codon, at first, the binary message was converted into a decimal number between 0 and 1. A mapping between each original amino acid and an alphabetically ordered redundant codon list was generated. A continuous subdivision into several redundant codon combinations regenerated the decimal number generated before. This process produced the intended mRNA sequence. Also, additional steps were introduced along with the existing method to enhance the strength of the watermark.

In [72], the authors proposed three different methods of information concealment. Their work used sequences from publicly available DNA databases containing around 163 million sequences [59], [73]. In all the proposed methods, select DNA sequences from the public databases were considered reference sequences. The reference sequence is known only to the sender and receiver and is used to hide the original message sequence. In the first method, that is, in the insertion method, bits of the plaintext message were inserted into the reference sequence following established rules. The authors defined their coding rule to convert DNA sequences into binary strands. The second method (Complementary Pair Method) works with the complementary rule of DNA computation. The authors defined their own complementary rule in this method and produced two complementary substring pairs. These substrings were inserted in the reference sequence and were used to determine the position in the reference sequence where the message bits needed to be inserted. In the third method (Substitution Method), the reference string was substituted by its complement based on the message bit and other rules. In each case, knowledge of the reference sequence is required to recover the message. In addition, it is unfeasible to identify the reference sequence from publicly available databases as it contains 163 million DNA sequences. The authors proved the robustness of their methods and analyzed and compared them based on different data-hiding parameters such as capacity, payload, and bpn. The substitution method proved superior to the others based on its compactness and efficiency. The methods were also compared with other existing methods regarding reversibility, robustness, and capacity.

Sabry et al. proposed three different data encoding algorithms based on the genetic code table in [16]. As their techniques are reversible, they can hide data in DNA strands. Their basic idea is to convert binary data into DNA strands and then into amino acids. Initially, a distribution table (Table 11) is created to associate English alphabet letters

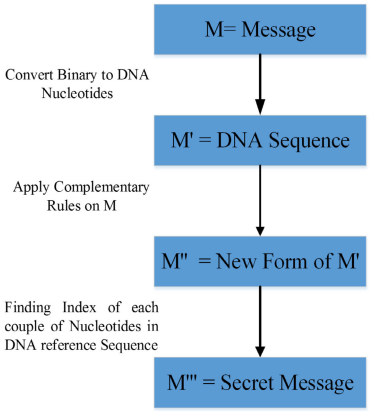


**FIGURE 24.** Steganography in [75].

with amino acids. Each column in the table represents one letter from the English alphabet, and each row represents an ambiguous number (represented as the nucleotides of RNA). Also, the corresponding amino acid family is shown at the bottom of the table for each letter. The table is then used to convert RNA into amino acids using three methods. In the first method, each group of two RNA characters (representing the family of the amino acid) is replaced with the corresponding letters of that family. In the second method (overlapping encoding), each RNA codon is substituted by one amino acid character (represented by column number) and one ambiguity number (represented by row number). The ambiguity number is appended at the end of the input. The third method differs from the second one in how ambiguous numbers are embedded. Instead of appending them at the end, the numbers are appended in DNA after each amino acid. The authors also converted this intermediary message into binary and back into DNA form to complete the encoding.

In [74], the authors implemented a steganography protocol to exchange the session key of public-key cryptographic techniques. A session key was hidden in DNA strand in this works. This implementation of a hidden secret key makes using an insecure communication channel possible to exchange the session key between sender and receiver. In the key hiding method, a reference sequence from a publicly available DNA database was selected and used. The authors defined a complementary rule and applied it to change the elements of the reference sequence into its complement based on the length and value of the message bits. The work also analyzed the method's robustness and compared it with two other data-hiding methods based on capacity and payload.

Another steganography method based on the basic features of DNA was proposed in [75]. The main process of this method is shown in Fig. 24. The authors also used reference strings in implementing their work. After converting the binary message into DNA strands, the complementary rule was applied to generate an intermediary strand. Based on the reference sequence, each nucleotide pair was replaced with a number representing the index of the nucleotide pair in the

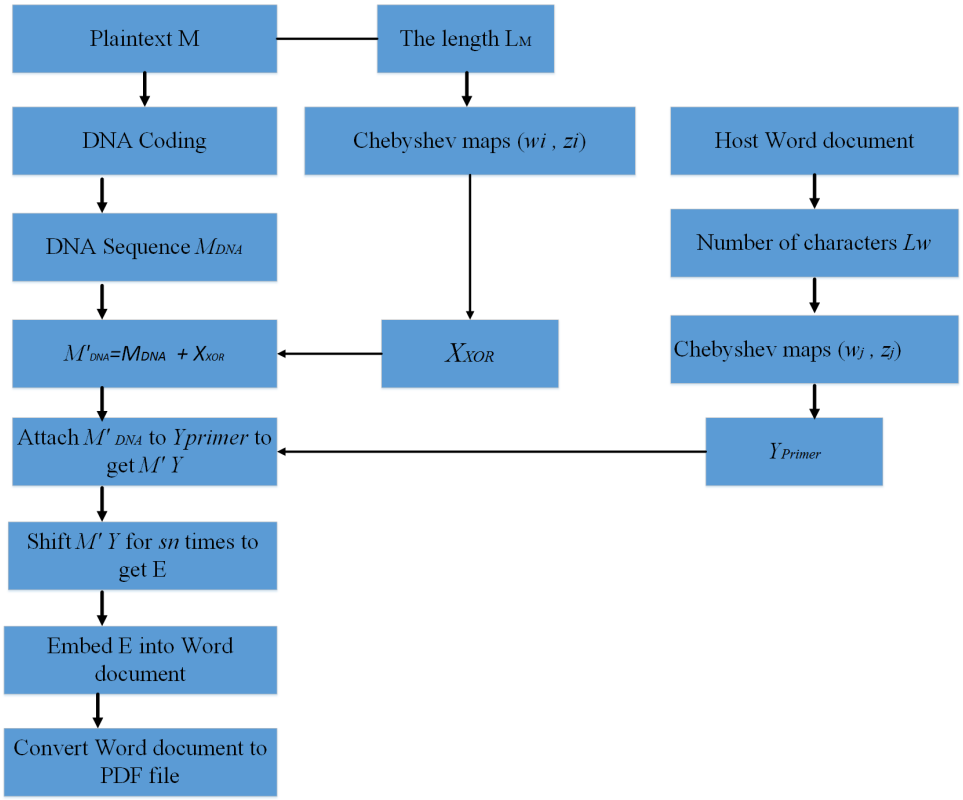


**FIGURE 25.** Flowchart for data encryption and hiding algorithm [76].

reference sequence. Thus, a secret message containing only the number was generated to be transferred to the receiver. The authors performed a robustness analysis of their method compared to other DNA-based information-hiding methods.

In [76], a text data hiding method was proposed utilizing DNA coding and operations to process plaintext messages before embedding them into a Word Document. The flowchart of the process is presented in Fig. 25. Here, Chebyshev mapping generated two random DNA sequences $X_{XOR}$ and $Y_{Primer}$. Sequence $X_{XOR}$ was used to encrypt DNA sequence $M_{DNA}$. $M_{DNA}$ was the encoded sequence generated from the plaintext using the DNA coding table presented in Table 3. The motivation behind this encryption was to resist chosen plaintext and known plaintext attacks. The other sequence, $Y_{Primer}$, was used as a primer sequence and attached to the XOR result. The resulting sequence was shifted multiple times and finally embedded into a Word Document. The authors carried out a performance and security analysis of their method. The analysis showed that DNA coding helped achieve higher capacity than other existing methods. Also, coding helped generate a larger keyspace, decreasing the vulnerability to brute force attacks.

## VI. DISCUSSION AND FUTURE DIRECTIONS

Although the security analysis of DNA-based cryptography is still a matter of research, our study shows that information security through natural DNA coding and computation is somewhat absolute as it works with a one-time-pad cryptosystem. The idea behind natural DNA cryptography is to overcome the challenges of digital cryptography and uncover the potential of DNA in computing. In digital cryptography, although it is known that an OTP ensures an unbreakable system, it is not possible to generate truly random OTPs [36]. Natural DNA cryptography solves this restriction of creating message-long, truly random OTPs by utilizing the vast parallelism and storage capacity of DNA computing [37]. Moreover, breaking a DNA-based algorithm requires sound knowledge of DNA coding and computation, which is not common in general. In addition, the main operation involved in those computations is PCR, which requires a correct pair of primers to extract the appropriate result. This provides another layer of security since knowing algorithms, techniques, or even the ciphertext is not enough to extract the correct message without specific primer pairs.

The history of natural DNA Cryptography is fairly recent. It is still in the stage of overcoming many limitations. The most significant limitation of this field is the lack of theories [77]. Traditional cryptography has a sound amount of theories and research. DNA-based cryptography mainly focuses on using DNA to implement existing theories from traditional cryptography. However, the biological operations involved in this field require a well-equipped laboratory, human intervention, and sound knowledge of DNA computing. Therefore, this research area is poorly understood and restricted to a

**TABLE 12. Summary of natural DNA cryptography methods.**

| Research Paper | Encryption Technique Used | Bio-molecular Operations Applied | Key Features |
|---|---|---|---|
| DNA-Based Cryptography [37] | One-time-pad based substitution technique and XOR operation | Primer annealing, Polymerase extending, Hybridization Ligation | Generated truly random OTPs |
| A DNA-based, Bio-Molecular Cryptography Design [38] | One-time-pad based substitution technique through the addition operation | Primer Extension Hybridization | Introduced modulo 2 binary additions in DNA strands |
| One-Time-Pads Encryption in the tile Assembly Model [26] | XOR operation with one time pads | Gel electrophoresis PCR Ligation | Introduced self-assembled DNA tiling in DNA-based Cryptography |
| Design of Truly Random One-Time-Pads in DNA XOR Cryptosystem [27] | One-time-pad based substitution technique through XOR operation | Hybridization PCR Atomic Force Micropscope (AFM) | Solved synchronization problem of hybridization through the use of AFM |
| Algorithm for Elliptic Curve Diffie-Hellman Key Exchange Based on DNA Tile Self-assembly [41] | | PCR Gel Electrophoresis | Implemented Elliptic Curve Diffie-Hellman key exchange protocol in DNA |
| An Encryption Scheme Using DNA Technology [19] | RSA, DES | DNA synthesis, PCR amplification | Implemented a hybrid cryptosystem which applied traditional digital cryptography as well as DNA computing |
| The public-key system using DNA as a one-way function for key distribution [42] | | Ligation, Cleavage, PCR | Implemented public key distribution technique using DNA computation |
| Broke DES Using a Molecular Computer [5] | Data Encryption Standard (DES) | Extraction, Tagging, PCR | Broken DES in 4 months, Subsequent breaking could be done in just one day |
| On Applying Molecular Computation To The Data Encryption Standard [6] | Data Encryption Standard (DES) | Separation, Combination, Set, PCR, Ligation | Introduced a new theoretical model known as the Sticker model to break DES |
| Algorithmic Self-Assembly of DNA Tiles and its Application to Cryptanalysis [33] | | | Applied convolution product in Wang tiles and emulated a system to break NTRU |

specific group of researchers. In addition, the methods and mechanisms are not generic enough to apply to any data or system. Compared to traditional cryptography operations, DNA-related operations are time-consuming [78]. Moreover, although the massive parallelism of DNA computing allows the execution of vast amounts of operations simultaneously, extracting the target data from this huge pool is difficult. Lastly, the operations involved in this field are dependent on environmental conditions. Therefore, any change in these conditions can generate error-prone results, compromising the entire experiment. Natural DNA steganography also suffers from the limitations of natural DNA Cryptography as it is dependent on biomolecular operations.

To overcome the limitation of natural DNA, researchers have started simulating DNA operations through computation in an alternate field called pseudo-DNA cryptography. The goal is to improve the security of the existing cryptographic algorithms using pseudo-DNA computations. DNA-inspired processes and operations are used to introduce more complexity in computations, enhancing the security of the resulting techniques. DNA coding, simulated principles of the central dogma, and virtual DNA tile self-assembly are employed in cryptosystems to improve security. Other times, DNA techniques have inspired substitution or permutation transformations used in specific components of the proposed cryptosystems.

While natural and pseudo-DNA cryptography deals with encrypting information, DNA-based steganography uses DNA as a medium for information hiding. One of the main challenges in steganography is ensuring that the hidden message is imperceptible. DNA shows success in this area through its high degree of randomness and its ability to conceal vast amounts of data.

In both pseudo-DNA cryptography and steganography, a lack of security analysis exists. Especially in the case of steganography techniques, it is important to establish standards so that different proposed methods can be compared with each other. In addition, an analysis of the appropriateness of DNA as a cover media required steganalysis techniques.

Tables 12 and 13 summarize the methods we discussed from Natural and Pseudo-DNA Cryptography. These tables describe the encryption techniques, the bio-molecular operations applied, and the key features of each method. DNA-based steganography research is summarized in Table 14. We compare the main features and differences of the three types of DNA-based cryptography and summarize the results in Table 15. Our survey of the main aspects of DNA

**TABLE 13. Summary of pseudo-DNA cryptography methods.**

| Research Paper | Encryption Technique Used | Pseudo Bio-molecular Operations Applied | Key Features |
|---|---|---|---|
| A Multilevel Image Encryption Algorithm Based on Chaos and DNA Coding [54] | Permutation and addition using logistic chaotic sequence | DNA coding | Improved resistance to statistical attacks |
| A DNA-based Implementation of YAEA Encryption Algorithm [57] | Substitution | DNA coding | Improved Resistance to frequency attack |
| An encryption algorithm inspired from DNA [62] | Symmetric Encryption Technique | Central Dogma of molecular biology | Improved confusion and diffusion |
| A pseudo-DNA cryptography Method [63] | Modified Symmetric Encryption Technique | Central Dogma of molecular biology | Improved robustness to brute force attacks |
| DNA computing-based cryptography [64] | Asymmetric Encryption Technique | DNA coding | Improved Complexity of the calculation |
| A DNA encryption technique based on matrix manipulation and secure key generation scheme [65] | Pseudo DNA encryption | DNA coding, sequence modification, Amino acid conversion | Improved Security of the ciphertext |
| A Method to Encrypt Information with DNA-Based Cryptography [66] | Vigenere Cipher | DNA coding | Improved complexity of Vigenere Cipher |
| Bi-serial DNA Encryption Algorithm (BDEA) [67] | XOR based substitution | PCR | Achieved Double layer security |
| DNA Cryptographic Algorithms [68] | Pseudo DNA encryption | Hybridization, Cleavage | Implemented DNA tile based encryption algorithm using Bio-informatics toolbox |
| An Image Encryption Algorithm Based on DNA Self-Assembly Technology [34] | Pseudo DNA encryption | DNA self-assembly | Implemented DNA tile-based image encryption algorithm |
| An improved symmetric key cryptographic algorithm with DNA-based strong cipher [42] | Symmetric key technique | Genetic codon table | Strengthened ciphertext |

**TABLE 14. Summary of DNA based steganography methods.**

| Research Paper | Data Hiding Process |
|---|---|
| Hiding Messages in DNA Microdots [4] | Data was concealed into human genomes and then confined into microdots |
| DNA-Based Cryptography [37] | Data was embedded into DNA strands and mixed with randomly generated distracter DNA strands to conceal its existence |
| Cryptography with DNA binary strands [70] | Encoded data in DNA strands was mixed with dummy strands, and graphical image decryption was applied to recover the message |
| Hiding data in DNA [71] | Data was embedded into non-coding regions of DNA strands as well as into non-genetic DNAs |
| Data hiding methods based upon DNA sequences [72] | Used reference DNA sequence from a publicly known database as a media of data hiding |
| Three Reversible Data Encoding Algorithms based on DNA and Amino Acids' Structure [16] | Implemented three reversible DNA coding techniques that could be used as a Steganography method |
| Improving Hybrid Cryptosystems with DNA Steganography [74] | Proposed a method to exchange session key by hiding it into reference DNA sequences |
| Data Hiding Method Based on DNA Basic Characteristics [75] | Embedded message was generated through basic DNA properties such as the complementary rule and the reference sequence |
| A novel data hiding method based on deoxyribonucleic acid coding [76] | Applied DNA coding and computation to process ciphertext |

Cryptography and Steganography provides researchers with a foundation for continued study in this field.

### A. FUTURE DIRECTIONS

The research in DNA-based Cryptography is still in its initial stage. The main motivation behind this research is to see the prospects of DNA in computation. By working with new techniques and implementing new methods, the researchers in this field are trying to implement existing hard problems of traditional cryptography on DNA strands. Because of the novelty of the techniques and methods, more opportunities remain to expand knowledge of DNA-based cryptography. This section identifies the main research directions in natural-DNA cryptography, pseudo-DNA cryptography, and steganography.

#### 1) IMPLEMENTATION OF THEORY AND STANDARDS

As previously discussed, DNA-based cryptography and steganography are not rich in theory and standards. Thus it is essential to develop theories and standards to advance research in these fields. For example, DNA is used as a cover medium in DNA-based steganography. Work is needed

**TABLE 15. Main differences between the three types of DNA-based cryptography.**

| Types of DNA-Based Cryptography | Key Features | Computational Complexity | Advantages | Disadvantages |
|---|---|---|---|---|
| Natural DNA Cryptography | Utilizes DNA sequences as cryptographic keys | Time: O(n), Space: O(1) | High level of security and confidentiality | Limited data storage capacity |
| Pseudo-DNA Cryptography | Synthetic DNA-like structures used as keys | Time: O(n log n), Space: O(n) | Scalable and adaptable to different applications | Less secure compared to natural DNA cryptography |
| DNA-Based Steganography | Conceals data within DNA sequences | Time: Varies depending on encoding/decoding | High data capacity and resistance to detection | Challenging extraction and decoding of hidden data |

to analyze DNA appropriateness as a cover medium by comparing this method's security to other cover media such as image, text, audio, etc. [79]. Checking the applicability of existing theories and standards of traditional steganography to DNA-based steganography is also a key element. Concerning security analysis, it is essential to establish standards for assessing the security of existing DNA-based methods. To this point, it is important to understand steganalysis techniques and test various attacks of traditional steganography on DNA-based steganography [80].

#### 2) DISCOVERING NEW OPERATIONS

DNA-based cryptography and steganography depend on biological operations. Therefore, finding new appropriate operations to implement a technique is still an open problem. So far, only a subset of bio-molecular operations have been applied in this field. Involving more operations will increase the scope of research in this field.

#### 3) DNA DATA STORAGE

DNA storage is an emerging technique utilizing DNA as a long-lasting biological data storage medium. Compared to traditional digital storage media (e.g., Hard Disk Drives), DNA storage has a tremendous storage capacity, increased stability, and reduced dependence on a power source. DNA offers tremendous density in terms of storage capacity: one gram of DNA can store 215 million gigabytes of data. In contrast, the average laptop hard drive (500 GB) can store only 1-2 millionth of that quantity. In addition, DNA is extremely stable and can be preserved intact for hundreds of thousands of years in cold conditions. In contrast, the preferred long-term storage digital media (such as gold CDs/DVDs) can only last up to a century. Its independence from hardware, formats, and from a power source make DNA storage less error-prone and significantly increases its permanence ([81], [82]).

Despite these unique properties, there are still significant challenges in making this technology mainstream. Major drawbacks are the high cost and error rates of synthesizing and sequencing, slow read and write speeds, and the lack of automation and standardization ([83], [84]). Recent advancements in DNA storage have shown promising results for the future of data storage. In 2019, researchers at the University of Washington and Microsoft successfully stored and retrieved a record-breaking 200 megabytes of data on DNA molecules [85]. This achievement was made possible by using a new coding scheme that improved the accuracy of reading and writing DNA sequences. In 2020, researchers at ETH Zurich developed a new method for storing digital information in DNA more efficiently [86]. This method used an error-correcting code to ensure that data could be accurately retrieved even if some DNA molecules were damaged or lost. These new developments in DNA storage demonstrate the potential for using DNA as a long-term storage medium.

The progress in DNA storage could also impact the fields of DNA-based cryptography and steganography. If more data can be stored on a single DNA molecule, this could increase the amount of information hidden using DNA-based steganography techniques. Additionally, improvements in reading and writing DNA sequences could make it easier to encode and decode encrypted messages using DNA molecules. The pace of advancements in DNA storage demonstrates that DNA can become a widespread medium for archiving data in the next ten years [87].

## VII. CONCLUSION

DNA-based cryptography is a growing field of research in both Biocomputing and the traditional cryptographic world. Throughout our survey, we have presented significant works in this field and related theories. We have highlighted three important research areas and identified their potential for future research. This survey work will assist researchers in assessing the current state-of-the-art of DNA-based cryptography. Finally, we can say that although the research on natural DNA has limitations, it has a great prospect in computing due to DNA's massively parallel computing power and storage capacity. Representing binary data as DNA nucleotides provides flexibility in working with current binary information. Researchers in DNA computing are developing ways to use these technologies to fully realize the potential inherent in DNA. As DNA presents a myriad of possibilities to expand and improve cryptography, DNA-based cryptography will continue to grow as a significant research area expanding the field of DNA Computing.

## REFERENCES

[1] C. Smith. (2015). *Breakthrough Data Storage Innovation: One DNA Molecule Can Store Tons of Data for 1M Years*. Accessed: Dec. 11, 2016. [Online]. Available: http://bgr.com/2015/08/18/dna-hard-drive-data-storage/

[2] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Nature*, vol. 266, no. 5187, p. 40, Nov. 1994.

[3] Wikipedia. (2016). *Biological Computing—Wikipedia, the Free Encyclopedia*. Accessed: Apr. 20, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Biological_computing

[4] C. T. Clelland, V. Risca, and C. Bancroft, "Hiding messages in DNA microdots," *Nature*, vol. 399, no. 6736, pp. 533–534, Jun. 1999.

[5] D. Boneh, C. Dimworth, and R. J. Lipton, "Breaking DBS using a molecular computer," *DNA Based Comput*, vol. 27, p. 37, 1996, doi: 10.1090/dimacs/027.

[6] L. M. Adleman, P. W. K. Rothemund, S. Roweis, and E. Winfree, "On applying molecular computation to the data encryption standard," *J. Comput. Biol.*, vol. 6, no. 1, pp. 53–63, Jan. 1999.

[7] M. Beck and R. Yampolskiy, "DNA as a medium for hiding data," *BMC Bioinf.*, vol. 13, no. S12, p. A23, Jul. 2012.

[8] A. Sinha, J. Mahapatro, and B. Bhabani, "Random binary sequences generation using heartbeats for cryptographic keys in WBSNs," *Int. J. Sensors Netw.*, vol. 38, no. 3, pp. 191–203, 2022.

[9] J. D. Watson and F. H. C. Crick, "Molecular structure of nucleic acids," *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.

[10] G. Paun, G. Rozenberg, and A. Salomaa, *DNA Computing: New Computing Paradigms*. Germany: Springer, 1998.

[11] M. Amos, G. Păun, G. Rozenberg, and A. Salomaa, "Topics in the theory of DNA computing," *Theor. Comput. Sci.*, vol. 287, no. 1, pp. 3–38, Sep. 2002.

[12] S. Palluk, D. H. Arlow, T. de Rond, S. Barthel, J. S. Kang, R. Bector, H. M. Baghdassarian, A. N. Truong, P. W. Kim, A. K. Singh, N. J. Hillson, and J. D. Keasling, "De novo DNA synthesis using polymerase-nucleotide conjugates," *Nature Biotechnol.*, vol. 36, no. 7, pp. 645–650, Aug. 2018.

[13] C. Calude and G. Paun, *Computing With Cells and Atoms: An Introduction to Quantum, DNA and Membrane Computing*. Boca Raton, FL, USA: CRC Press, 2000.

[14] N. Lewis and P. Weinberger, "DNA computing," MITRE Corp., McLean, VA, USA, Tech. Rep. JSR-95-116, 1995.

[15] F. Crick, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, pp. 561–563, Aug. 1970.

[16] M. Sabry, M. Hashem, and T. Nazmy, "Three reversible data encoding algorithms based on DNA and amino acids' structure," *Int. J. Comput. Appl.*, vol. 54, no. 8, pp. 24–30, Sep. 2012.

[17] T. Simonite. (2016). *Intel Puts the Brakes on Moore's Law*. Accessed: Nov. 15, 2016. [Online]. Available: https://www.technologyreview.com/s/601102/intel-puts-the-brakes-on-moores-law/

[18] D. Heider and A. Barnekow, "DNA-based watermarks using the DNA-crypt algorithm," *BMC Bioinf.*, vol. 8, no. 1, Dec. 2007, Art. no. 176.

[19] G. Cui, L. Qin, Y. Wang, and X. Zhang, "An encryption scheme using DNA technology," in *Proc. 3rd Int. Conf. Bio-Inspired Comput., Theories Appl.*, Sep. 2008, pp. 37–42.

[20] A. Salomaa, "Computability paradigms based on DNA complementarity," *WIT Trans. Eng. Sci.*, vol. 15, p. 14, Jan. 1970.

[21] H. Rauhe, G. Vopper, U. Feldkamp, W. Banzhaf, and J. C. Howard, "Digital DNA molecules," in *Proc. 6th DIMACS Workshop DNA Based Comput.*, 2000, pp. 13–17.

[22] U. Feldkamp, W. Banzhaf, and H. Rauhe, "A DNA sequence compiler," Presented at the 6th Int. Meeting DNA Based Comput., Leiden, The Netherlands, 2000.

[23] E. W. X. Yang and N. C. Seeman, "Universal computation via self-assembly of DNA: Some theory and experiments," *DNA Based Comput. Two*, vol. 44, p. 191, Aug. 1999.

[24] M. Borda, *Fundamentals in Information Theory and Coding*. Germany: Springer, 2011, p. 509.

[25] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, "Design and self-assembly of two-dimensional DNA crystals," *Nature*, vol. 394, no. 6693, pp. 539–544, Aug. 1998.

[26] Z. Chen and J. Xu, "One-time-pads encryption in the tile assembly model," in *Proc. 3rd Int. Conf. Bio-Inspired Comput., Theories Appl.*, Sep. 2008, pp. 23–30.

[27] M. Hirabayashi, H. Kojima, and K. Oiwa, "Design of true random one-time pads in DNA XOR cryptosystem," in *Natural Computing*. Tokyo, Japan: Springer, 2010, pp. 174–183.

[28] E. Winfree, "Algorithmic self-assembly of DNA," Ph.D. thesis, California Inst. Technol., Pasadena, CA, USA, 1998.

[29] H. Wang, "Proving theorems by pattern recognition I," *Commun. ACM*, vol. 3, no. 4, pp. 220–234, 1960.

[30] R. M. Robinson, "Undecidability and nonperiodicity for tilings of the plane," *Inventiones Mathematicae*, vol. 12, no. 3, pp. 177–209, 1971.

[31] M. Borda and O. Tornea, "DNA secret writing techniques," in *Proc. 8th Int. Conf. Commun.*, Jun. 2010, pp. 451–456.

[32] Z. Cheng, Z. Chen, Y. Huang, X. Zhang, and J. Xu, "Implementation of the timetable problem using self-assembly of DNA tiles," *Int. J. Comput. Commun. Control*, vol. 5, no. 4, pp. 490–505, 2010.

[33] O. Pelletier and A. Weimerskirch, "Algorithmic self-assembly of DNA tiles and its application to cryptanalysis," 2001, *arXiv:cs/0110009*.

[34] S. Zhou, Q. Zhang, and X. Wei, "An image encryption algorithm based on DNA self-assembly technology," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, vol. 2, Oct. 2010, pp. 315–319.

[35] M. Hirabayashi, H. Kojima, and K. Oiwa, "Effective algorithm to encrypt information based on self-assembly of DNA tiles," in *Proc. Nucleic Acids Symp. Ser.*, vol. 53. London, U.K.: Oxford Univ. Press, 2009, pp. 79–80.

[36] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Noida, India: Pearson, 2006.

[37] A. Gehani, T. LaBean, and J. Reif, "DNA-based cryptography," in *Aspects of Molecular Computing*. Germany: Springer, 2003, pp. 167–188.

[38] J. Chen, "A DNA-based, biomolecular cryptography design," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 3, May 2003, p. 822.

[39] E. Winfree, T. Eng, and G. Rozenberg, "String tile models for DNA computing by self-assembly," in *Proc. Int. Workshop DNA-Based Comput.* Germany: Springer, 2000, pp. 63–88.

[40] S. K. Kufer, E. M. Puchner, H. Gumpp, T. Liedl, and H. E. Gaub, "Single-molecule cut-and-paste surface assembly," *Science*, vol. 319, no. 5863, pp. 594–596, Feb. 2008.

[41] Z. Cheng, Y. Huang, and J. Xu, "Algorithm for elliptic curve Diffie–Hellman key exchange based on DNA tile self-assembly," in *Proc. 3rd Int. Conf. Bio-Inspired Comput., Theories Appl.*, Sep. 2008, pp. 31–36.

[42] K. Tanaka, A. Okamoto, and I. Saito, "Public-key system using DNA as a one-way function for key distribution," *Biosystems*, vol. 81, no. 1, pp. 25–29, Jul. 2005.

[43] S. Namasudra, S. Sharma, G. C. Deka, and P. Lorenz, "DNA computing and table based data accessing in the cloud environment," *J. Netw. Comput. Appl.*, vol. 172, Dec. 2020, Art. no. 102835.

[44] S. Shi and N. Xiao, "Security demonstration for the quantum noise-based physical layer using variable keys," *Int. J. Sensor Netw.*, vol. 41, no. 1, pp. 60–66, 2023.

[45] M. Yao and D. Chen, "Two-stage pricing strategy for personal cloud storage: Free trial and the cloud security risk," *Int. J. Sensor Netw.*, vol. 39, no. 1, pp. 56–66, 2022.

[46] X. Li, D. Zhu, J. Wu, H. Wang, L. Yang, and L. Song, "A quantum key injection scheme for mobile terminals based on commercial quantum key distribution," *Int. J. Sensor Netw.*, vol. 38, no. 2, pp. 132–141, 2022.

[47] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[48] A. Olteanu, Y. Xiao, and Y. Zhang, "Optimization between AES security and performance for IEEE 802.15.3 WPAN," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 6030–6037, Dec. 2009.

[49] S. Roweis, E. Winfree, R. Burgoyne, N. V. Chelyapov, M. F. Goodman, P. W. K. Rothemund, and L. M. Adleman, "A sticker-based model for DNA computation," *J. Comput. Biol.*, vol. 5, no. 4, pp. 615–629, Jan. 1998.

[50] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Proc. Int. Algorithmic Number Theory Symp.* Germany: Springer, 1998, pp. 267–288.

[51] T. R. Damase, D. Stephens, A. Spencer, and P. B. Allen, "Open source and DIY hardware for DNA nanotechnology labs," *J. Biol. Methods*, vol. 2, no. 3, p. e24, Aug. 2015.

[52] S. Yan and K.-C. Wong, "Future DNA computing device and accompanied tool stack: Towards high-throughput computation," *Future Gener. Comput. Syst.*, vol. 117, pp. 111–124, Apr. 2021.

[53] E. Şatir and O. Kendirli, "A symmetric DNA encryption process with a biotechnical hardware," *J. King Saud Univ., Sci.*, vol. 34, no. 3, Apr. 2022, Art. no. 101838.

[54] Q. Wang, Q. Zhang, and C. Zhou, "A multilevel image encryption algorithm based on chaos and DNA coding," in *Proc. 4th Int. Conf. Bio-Inspired Comput.*, Oct. 2009, pp. 1–5.

[55] S. T. Dougherty, A. Korban, S. Sahinkaya, and D. Ustun, "Construction of DNA codes from composite matrices and a bio-inspired optimization algorithm," *IEEE Trans. Inf. Theory*, vol. 69, no. 3, pp. 1588–1603, Mar. 2023.

[56] K. G. Benerjee, S. Deb, and M. K. Gupta, "On conflict free DNA codes," *Cryptogr. Commun.*, vol. 13, no. 1, pp. 143–171, Jan. 2021.
[57] S. T. Amin, M. Saeb, and S. El-Gindi, "A DNA-based implementation of YAEA encryption algorithm," in *Proc. Int. Conf. Climate Inform.*, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:9878880
[58] M. Saeb and A. Baith, "An encryption algorithm for data security," in *Recent Advances in Information Science and Technology*, N. E. Mastorakis, Ed. Singapore: World Scientific, 1998, pp. 350–354.
[59] NCBI. *Canis Lupus Familiaris (Dog)*. Accessed: Oct. 12, 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/genome?term=canis%20lupus%20familiaris
[60] O. Tornea, M. Borda, T. Hodorogea, and M. Vaida, "Encryption system with indexing DNA chromosomes cryptographic algorithm," in *Proc. IASTED Int. Conf.*, vol. 680, 2010, p. 99.
[61] Z. Yunpeng, Z. Yu, W. Zhong, and R. O. Sinnott, "Index-based symmetric DNA encryption algorithm," in *Proc. 4th Int. Congr. Image Signal Process.*, vol. 5, Oct. 2011, pp. 2290–2294.
[62] S. Sadeg, M. Gougache, N. Mansouri, and H. Drias, "An encryption algorithm inspired from DNA," in *Proc. Int. Conf. Mach. Web Intell.*, Oct. 2010, pp. 344–349.
[63] K. Ning, "A pseudo DNA cryptography method," 2009, *arXiv:0903.2693*.
[64] X. Wang and Q. Zhang, "DNA computing-based cryptography," in *Proc. 4th Int. Conf. Bio-Inspired Comput.*, Oct. 2009, pp. 1–3.
[65] T. Mandge and V. Choudhary, "A DNA encryption technique based on matrix manipulation and secure key generation scheme," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2013, pp. 47–52.
[66] M. Najaftorkaman and N. S. Kazazi, "A method to encrypt information with DNA-based cryptography," *Int. J. Cyber-Secur. Digit. Forensics*, vol. 4, no. 3, pp. 417–427, 2015.
[67] D. Prabhu and M. Adimoolam, "Bi-serial DNA encryption algorithm (BDEA)," 2011, *arXiv:1101.2577*.
[68] O. Tornea and M. Borda, "DNA cryptographic algorithms," in *Proc. Int. Conf. Adv. Med. Health Care Through Technol.* Germany: Springer, 2009, pp. 223–226.
[69] G. Kipper, *Investigator's Guide to Steganography*. Boca Raton, FL, USA: CRC Press, 2003.
[70] A. Leier, C. Richter, W. Banzhaf, and H. Rauhe, "Cryptography with DNA binary strands," *Biosystems*, vol. 57, no. 1, pp. 13–22, Jun. 2000.
[71] B. Shimanovsky, J. Feng, and M. Potkonjak, "Hiding data in DNA," in *Proc. Int. Workshop Inf. Hiding*. Germany: Springer, 2002, pp. 373–386.
[72] H. J. Shiu, K. L. Ng, J. F. Fang, R. C. T. Lee, and C. H. Huang, "Data hiding methods based upon DNA sequences," *Inf. Sci.*, vol. 180, no. 11, pp. 2196–2208, Jun. 2010.
[73] The European Bioinformatics Institute. *EBI*. Accessed: Nov. 12, 2016. [Online]. Available: http://www.ebi.ac.uk/
[74] M. R. N. Torkaman, P. Nikfard, N. S. Kazazi, M. R. Abbasy, and S. F. Tabatabaiee, "Improving hybrid cryptosystems with DNA steganography," in *Digital Enterprise and Information Systems*. Germany: Springer, 2011, pp. 42–52.
[75] M. R. Abbasy, A. A. Manaf, and M. Shahidan, "Data hiding method based on DNA basic characteristics," in *Digital Enterprise and Information Systems*. Germany: Springer, 2011, pp. 53–62.
[76] H. Liu, D. Lin, and A. Kadir, "A novel data hiding method based on deoxyribonucleic acid coding," *Comput. Electr. Eng.*, vol. 39, no. 4, pp. 1164–1173, May 2013.
[77] B. Anam, K. Sakib, M. Hossain, and K. Dahal, "Review on the advancements of DNA cryptography," 2010, *arXiv:1010.0186*.
[78] G. Cui, C. Li, H. Li, and X. Li, "DNA computing and its application to information security field," in *Proc. 5th Int. Conf. Natural Comput.*, vol. 6, Aug. 2009, pp. 148–152.
[79] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, no. 3.4, pp. 313–336, 1996.
[80] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.
[81] Potomac Institute for Policy Studies. (2018). *The Future of DNA Data Storage*. Accessed: Aug. 8, 2023. [Online]. Available: https://potomacinstitute.org/images/studies/Future_of_DNA_Data_Storage.pdf
[82] H. H. Lee, R. Kalhor, N. Goela, J. Bolot, and G. M. Church, "Terminator-free template-independent enzymatic DNA synthesis for digital information storage," *Nature Commun.*, vol. 10, no. 1, p. 2383, Jun. 2019.
[83] C. K. Lim, J. W. Yeoh, A. A. Kunartama, W. S. Yew, and C. L. Poh, "A biological camera that captures and stores images directly into DNA," *Nature Commun.*, vol. 14, no. 1, p. 3921, Jul. 2023.
[84] R. Appuswamy, K. L. Brigand, P. Barbry, M. Antonini, O. Madderson, P. S. Freemont, J. McDonald, and T. Heinis, "OligoArchive: Using DNA in the DBMS storage hierarchy," in *Proc. Conf. Innov. Data Syst. Res.*, 2019, p. 98.
[85] C. N. Takahashi, B. H. Nguyen, K. Strauss, and L. Ceze, "Demonstration of end-to-end automation of DNA data storage," *Sci. Rep.*, vol. 9, no. 1, p. 4998, Mar. 2019.
[86] J. Koch, S. Gantenbein, K. Masania, W. J. Stark, Y. Erlich, and R. N. Grass, "A DNA-of-things storage architecture to create materials with embedded memory," *Nature Biotechnol.*, vol. 38, no. 1, pp. 39–43, Jan. 2020.
[87] R. L. Hotz. *Scientists Store Data in Synthetic DNA Embedded in a Plastic Bunny*. Accessed: Sep. 12, 2019. [Online]. Available: https://www.wsj.com/articles/scientists-store-data-in-synthetic-dna-embedded-in-a-plastic-bunny-11575907200

**TASNUVA MAHJABIN** received the M.S. degree in computer science and engineering from the University of Dhaka, and the Ph.D. degree in computer science from The University of Alabama, Tuscaloosa, AL, USA, in 2020. Her research interests include cyber security, the IoT security, and cryptography.

**ALINA OLTEANU** (Member, IEEE) received the B.S. degree in computer science from the University of Bucharest, Romania, the M.S. degree in computer science from the University of Alabama, Tuscaloosa, the M.S. degree in applied mathematics from the Polytechnic University of Bucharest, and the Ph.D. degree in computer science from the University of Alabama, in 2009. She is an Assistant Professor of Computer Science and the coordinator of the Computer Science program at the University of Montevallo. Her research interests include wireless sensor networks, network optimization and security.

**YANG XIAO** (Fellow, IEEE) received the B.S. and M.S. degrees in computational mathematics from Jilin University, Changchun, China, in 1989 and 1991, respectively, and the M.S. and Ph.D. degrees in computer science and engineering from Wright State University, Dayton, OH, USA, in 2000 and 2001, respectively.

He is currently a Full Professor with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA. He directed over 20 doctoral dissertations and supervised over 20 M.S. theses/projects. He has published over 300 science citation index (SCI)-indexed journal articles (including over 60 IEEE/ACM TRANSACTIONS) and 300 engineering index (EI)-indexed refereed conference papers and book chapters related to these research areas. His research interests include cyber–physical systems, the Internet of Things, security, wireless networks, smart grids, and telemedicine. He was a Voting Member of the IEEE 802.11 Working Group, from 2001 to 2004, involving the IEEE 802.11 (Wi-Fi) standardization work. He is a fellow of IET and AAIA. He serves/served as a member of the Technical Program Committee for more than 300 conferences. He received

the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING EXCELLENT Editor Award 2022. He served as the Guest Editor over 35 times for different international journals, including the IEEE WIRELESS COMMUNICATIONS, in 2006 and 2021; IEEE NETWORK, in 2007; *Mobile Networks and Applications* (ACM/Springer), in 2008; IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, in 2021; IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, in 2021; *IEEE Communications Standards Magazine*, in 2021; and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, in 2022 and 2023. He also serves as the Editor-in-Chief for *Cyber-Physical Systems*, *International Journal of Sensor Networks*, and *International Journal of Security and Networks*. He has been serving/served as an Editorial Board Member or an Associate Editor for 20 international journals, including IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, since 2022; IEEE TRANSACTIONS ON CYBERNETICS, since 2020, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, from 2007 to 2009, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, from 2007 to 2014; and IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, from 2014 to 2015.

**WENLIN HAN** received the B.S. and M.S. degrees from the Department of Computer Science, Central China Normal University, China, and the M.S. and Ph.D. degrees from the Department of Computer Science, The University of Alabama. She is currently an Associate Professor with the Department of Computer Science, California State University, Fullerton, USA. Her research interests include cybersecurity, applied cryptography used in the IoT, CPS, blockchain, and other latest applications.

**TIESHAN LI** (Senior Member, IEEE) received the B.S. degree in ocean fisheries engineering from the Ocean University of China, Qingdao, China, in 1992, and the Ph.D. degree in vehicle operation engineering from Dalian Maritime University (DMU), Dalian, China, in 2005. From 2007 to 2015, he held various Postdoctoral/Senior Research Associate (SRA)/Visiting Scholar positions with Shanghai Jiao Tong University, City University of Hong Kong, and the University of Macao. He has been a Full Professor with DMU, since 2011, and has been a Chair Professor, since 2021. He is currently a Tenured Professor with the University of Electronic Science and Technology of China. His research interests include intelligent learning and control for nonlinear systems, multi-agent systems, and their applications to unmanned vehicles.

**WEI SUN** (Senior Member, IEEE) received the B.E. degree in automation, the M.S. degree in detection technology and automatic equipment, and the Ph.D. degree in electrical engineering from the Hefei University of Technology, Hefei, China, in 2004, 2007, and 2012, respectively. He is currently a Professor with the Hefei University of Technology. His research interests include wireless sensor networks, networked control systems, and smart grids.