nanomaterials

MDPI

Article

An Operational DNA Strand Displacement Encryption Approach

Enqiang Zhu [1], Xianhang Luo [1], Chanjuan Liu [2,*] and Congzhou Chen [3]

[1] Institute of Computing Science and Technology, Guangzhou University, Guangzhou 510006, China; zhuenqiang@gzhu.edu.cn (E.Z.); 2112006164@e.gzhu.edu.cn (X.L.)
[2] School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[3] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China; chencongzhou@pku.edu.cn
* Correspondence: chanjuanliu@dlut.edu.cn

Abstract: DeoxyriboNucleic Acid (DNA) encryption is a new encryption method that appeared along with the research of DNA nanotechnology in recent years. Due to the complexity of biology in DNA nanotechnology, DNA encryption brings in an additional difficulty in deciphering and, thus, can enhance information security. As a new approach in DNA nanotechnology, DNA strand displacement has particular advantages such as being enzyme free and self-assembly. However, the existing research on DNA-strand-displacement-based encryption has mostly stayed at a theoretical or simulation stage. To this end, this paper proposes a new DNA-strand-displacement-based encryption framework. This encryption framework involves three main strategies. The first strategy was a tri-phase conversion from plaintext to DNA sequences according to a Huffman-coding-based transformation rule, which enhances the concealment of the information. The second strategy was the development of DNA strand displacement molecular modules, which produce the initial key for information encryption. The third strategy was a cyclic-shift-based operation to extend the initial key long enough, and thus increase the deciphering difficulty. The results of simulation and biological experiments demonstrated the feasibility of our scheme for encryption. The approach was further validated in terms of the key sensitivity, key space, and statistic characteristic. Our encryption framework provides a potential way to realize DNA-strand-displacement-based encryption via biological experiments and promotes the research on DNA-strand-displacement-based encryption.

Keywords: DNA strand displacement reaction; DNA encryption; huffman coding

check for updates

Citation: Enqiang, Z.; Xianhang, L.; Chanjuan, L.; Congzhou, C. An Operational DNA Strand Displacement Encryption Approach. *Nanomaterials* **2022**, *12*, 877. https://doi.org/10.3390/nano12050877

Academic Editor: Sungha Park

Received: 14 January 2022
Accepted: 3 March 2022
Published: 6 March 2022

1. Introduction

Over the past few years, the world has seen a stunning transformation in how information is exchanged. Communication online (through various platforms) has gradually become an indispensable means for information exchange, and ensuring data security has become one of the most concerning problems.

Cryptography plays a pivotal role in protecting the security of data communication by transforming plaintexts into unrecognizable codes [1,2]. Conventional cryptography, which depends excessively on the high computational complexity of mathematical calculations, is facing increasing risks of encryption cracking as computing capabilities are rising. Therefore, new encryption methods have been increasingly studied. As a nanomaterial, DeoxyriboNucleic Acid (DNA) can store a large amount of information, and with the rapid development of nanotechnology, DNA nanotechnology has been widely studied for encryption. DNA encryption, as a novel technique of cryptography, was proposed by Gehani et al. [3]. In DNA encryption, data are protected by transforming them into digital DNA codes. Because of the exclusive advantages of DNA molecules, including their large scale of parallelism, high storage capacity, and low power consumption, it is widely believed that DNA encryption can work with huge data and can potentially increases information security [4].

There is a growing body of literature recognizing the importance of DNA encryption. To solve the storage problem of one-time pad, Gehani et al. [3] first designed a one-time-pad-based DNA encryption program. In 2012, Wang et al. [5] proposed a new one-time one-key encryption algorithm based on the ergodicity of the skew tent chaotic graph. In 2014, Mokhtar et al. [6] combined a chaotic system with DNA coding to design a one-time pad encryption scheme. In [7], Yang et al. proposed a one-time pad encryption device based on DNA self-assembly technology. Because the keys generated in one-time pad approaches are not reusable, it is difficult to produce enough keys for encryption. A common method to address this problem is code transformation (i.e., transforming (0,1)-sequences into DNA sequences). In 2012, Liu et al. [8] proposed an image encryption method by means of a novel confusion and diffusion method, in which a DNA complementary rule was designed to confuse the pixels. To enhance the degree of confusing the pixels, Rehman et al. [9] in 2014 proposed a new gray image block cipher, which dynamically selects a rule from newly designed DNA complementary rules to encode and decode each pixel in a block. In 2016, based on the combination of the dynamic S-box and chaotic systems, Liu et al. [10] proposed a new image encryption scheme and showed that the proposed algorithm can reduce the correlation coefficients of images in three directions. In 2018, Wu et al. [11] designed a new chaotic mapping, called 2D-HSM. Then, they proposed an image encryption scheme combining 2D-HSM with DNA approaches and demonstrated its excellent performance. In [12,13], the authors employed chaotic series generated by a chaotic system to randomly select the coding rules, by which the security of encryption can be improved significantly. More recently, Wang et al. [14] proposed an image encryption algorithm based on ladder scrambling and DNA coding, which has a lower correlation of images compared to previous algorithms. In addition, some studies have attempted to improve the security of DNA encryption by performing operations on DNA codes, such as Addition (ADD) [15,16], Subtraction (SUB) [15,16], Exclusive Or (XOR) [16–18], and Exclusive Nor (XNOR) [18].

DNA encryption has been extensively studied along with the research on DNA nanotechnology in recent years. Due to the biological complexity of DNA nanotechnology, DNA encryption brings in the additional difficulty of deciphering, and thus can enhance information security. As a new approach in dynamic DNA nanotechnology, DNA Strand Displacement Reaction (SDR) has particular advantages such as being enzyme free and self-assembly. SDR has attracted considerable attention in recent years and has been widely applied to build various molecular systems [19] (it should be noted that the materials (DNA single strands) required for DNA strand displacement experiments are first designed by researchers, then commissioned to manufacture, and finally assembled into DNA molecules (complex structure)). A DNA SDR can be described as a molecular dynamic process (Figure 1), where a single-stranded DNA molecule is combined with a double-stranded DNA molecule through short complementary single-stranded DNA domains (called toeholds; see td and td^*), and a new stable double-stranded DNA molecule will be formed and a new single-stranded DNA molecule released from the original double strand. Notice that this can only happen gradually. Previous research has demonstrated that by designing appropriate DNA SDR, one can approximately realize all chemical reactions with ideal forms [20,21]. For example, in [22], SDR-based DNA switching circuits were designed for digital computing; in [23], the authors developed a time-sensitive molecular circuit based on SDR, called the cross-inhibitor, which can execute mutual inhibition; in [24,25], DNA strand displacement for microRNA detection was investigated; in [26], the authors analyzed the morphological manipulation of DNA gel microbeads with biomolecular stimuli by using SDR; in [27], the authors proposed an SDR-based chemical reaction network to solve 0–1 integer programming problems.

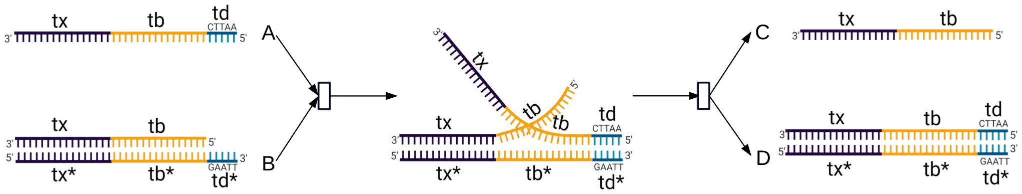


Figure 1. Principle of strand displacement reaction.

Designing encryption algorithms with the aid of DNA SDR has also been attempted. In [28], by using DNA SDR to extract secret keys, Zhang et al. proposed an image encryption algorithm on the basis of a chaos system. To obtain the keys with this approach, the DNA of the chains obtained by SDR must be sequenced. This may lead to decryption failures when current sequencing techniques are used. In [29], the authors designed six DNA SDR modules and combined them with the XOR operation to create a new encryption algorithm. Although the proposed algorithm may have a high capacity to resist statistical attacks, it relies heavily on real-time concentration detection. Therefore, it is still in a simulation stage and is difficult to realize via biological experiments because of the complicated design program.

During a DNA strand displacement experiment, it is difficult to monitor and detect the concentration of the target DNA strand in real time, and the changes in the design of the DNA sequence can easily lead to changes in the reaction rate. For these reasons, the study of DNA-strand-displacement-based encryption is still in the theoretical or simulation stage. To facilitate the implementation of DNA encryption via the biological experiment of DNA strand displacement, we introduced in this work a novel bio-experiment-based encryption framework. In this approach, three strategies were adopted, including a Huffman-coding-based transformation rule to confuse the plaintext, two SDR-based molecular modules to generate the initial key, and a cyclic-shift-based mechanism to extend and confuse the key. Note that most studies on DNA encryption techniques focus mainly on how to design complex rules to hide confidential information in DNA codes, without considering whether the designed scheme can be realized by biochemical experiments. Our approach enhances the feasibility of biochemical experiments and reveals two advantages. First, it improves the security of key transmission. To obtain the keys, one has to perform biochemical experiments, for which the results are sensitive to various conditions, such as temperature, time, and concentration. Therefore, our approach provides excellent protection against decoding. Second, it combines biochemical experiments with other techniques such as code transformation, which generates a new confusion and diffusion method to create a secure cipher, thus enhancing the cipher strength.

In order to verify the feasibility of the proposed approach, we first present an encryption example. Then, we refer to [29] for the analysis of its performance in encryption in terms of three aspects, viz., key sensitivity, key space, and statistic characteristics. Note that a good encryption method should be sensitive to the key, that is, when the key changes slightly, the encryption and decryption results will be sufficiently different. Meanwhile, a good encryption method should also have a large key space to resist brute force attacks. Besides, we also analyzed the statistical characteristic of our approach to demonstrate that it can cope with statistical attacks. Our encryption framework provides a potential way to realize DNA-strand-displacement-based encryption via biological experiments and promotes the research on DNA-strand-displacement-based encryption.

The remainder of the paper is organized as follows. Section 2 introduces the encryption framework and the process of the encryption algorithm. Section 3 presents the experimental validation of the feasibility of our approach by designing specific modular reactions. In Section 4, we analyze the performance of our approach in encryption security. The results imply that the proposed scheme is sensitive to the keys and possesses high resistance against statistical attacks. Finally, a summary of the main findings, along with some discussion and concluding remarks are provided in Section 5.

2. A Bio-Experiment-Based Encryption Approach

2.1. Encryption Framework

In view of the increasing need for dealing with large data and ensuring data security, we propose a novel bio-experiment-based DNA encryption method based on the DNA strand displacement technique. In this section, we first present the framework of our encryption method (Algorithm 1).

Algorithm 1: A new bio-experiment-based encryption framework.

```
Input: Plaintext P (an arbitrary string)
Output: Ciphertext C
begin
    Transform P into a DNA sequence $\mathbf{D_1}$;
    Design the DNA strand displacement molecular module, and obtain the initial key—a DNA
      sequence D;
    Design a shift rule, by which D is extended to a new DNA sequence $\mathbf{D_2}$ whose length is not less
      than that of $\mathbf{D_1}$;
    Perform DNA operation between $\mathbf{D_1}$ and $\mathbf{D_2}$, and transform the result into ciphertext C;
    return C;
end
```

The encryption starts with a plaintext input **P**, i.e., an arbitrary string, and transforms it into a DNA sequence $\mathbf{D_1}$ (Line 2), which will be taken as a substrate in the subsequent DNA computation. To generate the DNA sequence key **D** by biochemical experiments, some digital seeds are first obtained by recording the state changes (such as fluorescence color change or concentration change) during a designed experiment (Line 3). The next step is to extend **D** (Line 4) to a new DNA sequence $\mathbf{D_2}$ with a length at least that of $\mathbf{D_1}$ for later use in DNA computation. Finally, it produces the desired ciphertext (Line 5) by performing DNA computations (such as XOR and ADD) between $\mathbf{D_1}$ and $\mathbf{D_2}$, together with some transformation strategies.

2.2. Huffman Coding and Data Transformation

Huffman coding is an efficient method for compressing data without losing information. By using this technique, Ailenberg and Rotstein [30] proposed a simple, but efficient coding method for information storage in DNA and showed its potential ability in coding DNA. Inspired by this, we designed a Huffman-coding-based method, called *tri-phase transformation* (TPT), to confuse **P**.

TPT first transforms **P** into a DNA sequence $\mathbf{P_1}$ according to the rule listed in Supplementary Table S1; then, it transforms $\mathbf{P_1}$ into a (0,1)-sequence $\mathbf{P_2}$ via Huffman coding; finally, by using the rules listed in the first column in Supplementary Table S2, it transforms $\mathbf{P_2}$ into a new DNA sequence $\mathbf{D_1}$, which is an ingredient for subsequent DNA operations. Specifically, the process from $\mathbf{P_1}$ to $\mathbf{P_2}$ can be described as follows.

For each base $x \in \{\mathbf{A}, \mathbf{T}, \mathbf{G}, \mathbf{C}\}$, denote by $\omega(x)$ the weight of x, which is defined as the number of x that appear in $\mathbf{P_1}$. Then, construct a Huffman binary tree with four leaves in the following way: select two bases with the smallest weights as two leaves, denoted by x_1 and x_2, where $\omega(x_1) \leq \omega(x_2)$, and add a new vertex y_1 joining x_1 and x_2 such that x_1 and x_2 are the left and the right children of y_1, respectively; set $\omega(y_1) = \omega(x_1) + \omega(x_2)$ select two elements from $\{\mathbf{A}, \mathbf{G}, \mathbf{C}, \mathbf{T}, y_1\} \backslash \{x_1, x_2\}$ with the smallest weights, denoted by x_3 and x_4, where $\omega(x_3) \leq \omega(x_4)$, and add a new vertex y_2 joining x_3 and x_4 such that x_3 and x_4 are the left and right children of y_2, respectively; set $\omega(y_2) = \omega(x_3) + \omega(x_4)$, and add a new vertex y_3 jointing y_2 and the element in $\{\mathbf{A}, \mathbf{G}, \mathbf{C}, \mathbf{T}, y_1\} \backslash \{x_1, x_2, x_3, x_4\}$ such that the one with the smaller weight is the left child of y_3 and the other is the right child of y_3. Now, for each edge xy of the constructed tree such that y is a child of x, assign weight zero to it if y is the left child of x, and assign weight one to it if y is the right child of x. As a result, each base (a leaf) can be encoded into a (0,1)-sequence, which subsequently appears in the edges of the path from the root to the leaf, and $\mathbf{P_1}$ is encoded into a (0,1)-sequence Z. Observe that the length of Z may be an odd number. To transform Z into the DNA sequence $\mathbf{D_1}$ according to Supplementary Table S2, we have to modify it to have an even length. Our approach was as follows: if Z has an even length, add 00 to Z at the end of Z; otherwise, add 101 to Z. As an example, we considered a DNA sequence **TTCCAGCGGAC**, for which $\omega(\mathbf{A}) = 2, \omega(\mathbf{G}) = 3, \omega(\mathbf{C}) = 4$, and $\omega(\mathbf{T}) = 2$. By constructing a Huffman tree, **A** is encoded into 000, **G** is encoded into 01, **C** is encoded into 1, and **T** is encoded into 001. As a result, **TTCCAGCGGAC** is encoded into Z = 0010011100001101010001. Since Z has an even length, 00 is added at the end of Z and $\mathbf{D_1}$ = **ACGTAATGGAGA**.

As described in Algorithm 1, our approach depends on DNA operations to generate the final ciphertext. Two such operations, XOR and ADD, are used in our subsequently designed algorithm, where the rules of these two operations are shown in Supplementary Tables S5 and S6, respectively.

2.3. SDR Modules and Seed Encoding

Let us now turn to the design of initial keys, which first generate seeds in the form of “2-1” or “1-2” for the keys via the corresponding SDR modules. Two SDR modules are used to encode these two seeds based on the concentration change of the main species before and after the strand displacement reactions (the concentration change of the species should be normalized to the form p-q such that both p and q are integers, i.e., $1 - \frac{1}{2}$ should be replaced by 2:1).

2.3.1. Degradation Reaction Module

The principle of this module is presented in Figure 2, and its mechanism can be described by the reactions listed in Equation (1).

$$\begin{cases} \mathrm{A} + \mathrm{B} \xrightarrow{k_1} \mathrm{W}_1 + \mathrm{W}_2 \\ \mathrm{A} + \mathrm{D} \xrightarrow{k_2} \mathrm{E} + \mathrm{F} + \mathrm{W}_2 \\ \mathrm{G} + \mathrm{E} + \mathrm{F} \xrightarrow{k_3} \mathrm{A} + \mathrm{W}_3 + \mathrm{W}_4 \end{cases} \tag{1}$$

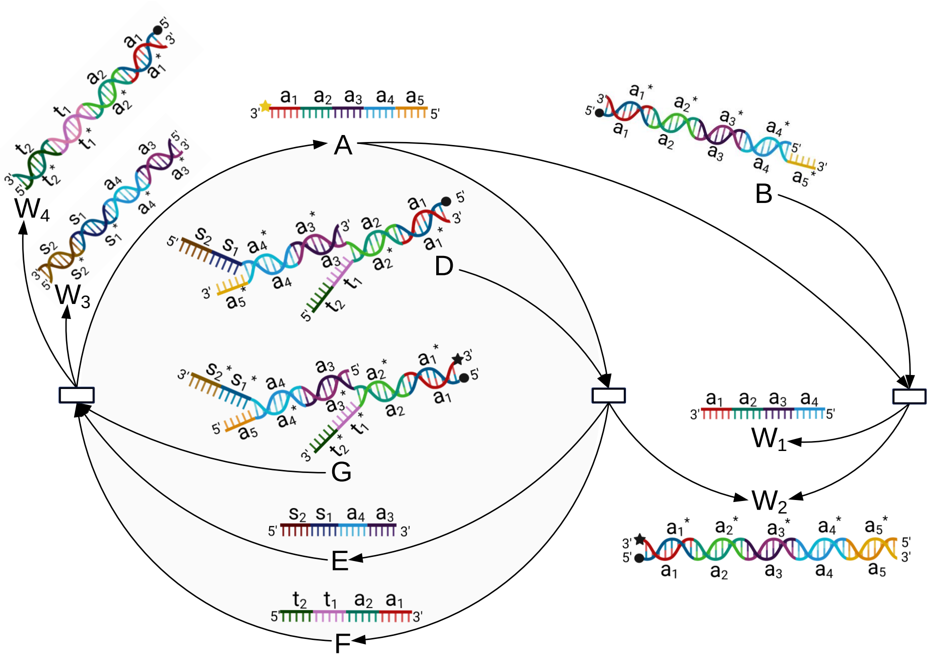


Figure 2. Schematic illustration of the degradation reaction module, by which the concentration of Species A is reduced to half of its original concentration. Thus, Species A can be used to encode the seed “2–1”.

The process of the reaction can be described as follows: this module mainly involves four initial species, including Single-stranded A and Complexes B, D, and G. We add the inputs A, B, D, and G into the biochemical reaction module simultaneously, and then a series of reactions is activated, after which the concentration of A is reduced to half of its original concentration, as shown in Equation (2). This is because A is consumed by both B and D and is generated by only one reaction (the third reaction listed in Equation (3). Specifically, the toehold a_5 of A binds to the domain a_5^* of B (and also D), and then, branch migration moves gradually to domain a_1, which releases single-stranded W_1 (and Single-stranded E and F) together with double-stranded W_2. Furthermore, the toehold s_2 of E (and t_2 of F) binds to the domain s_2^* (and t_2^*) of G, and then, branch migration moves gradually to the domain a_3 (and a_1), which releases the desired Single-stranded A and forms double strands W_3 and W_4. Observe that both A and G carry a dye at their 3′ end, and B, D, and G each carry a quencher at their 5′ end. Therefore, the beacon-labeled Strand A can be monitored in real time.

$$2A \xrightarrow{k_4} A \tag{2}$$

2.3.2. Catalysis Reaction Module

The principle of this module is presented in Figure 3, and its mechanism can be described by the reactions listed in Equations (3) and (4).

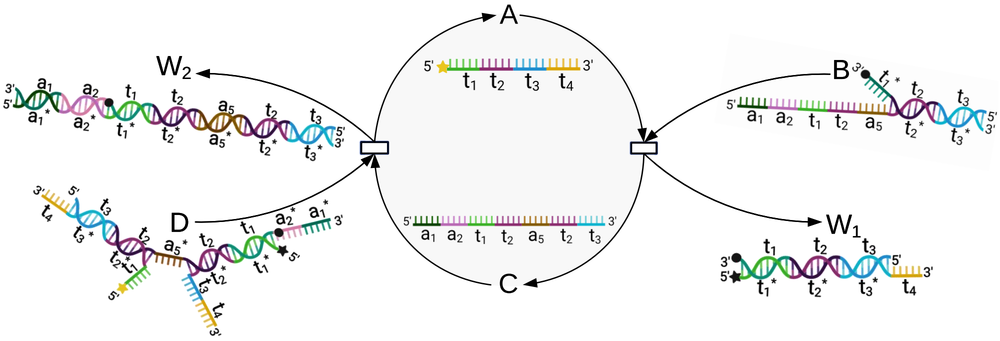

A
5′ 3′
t_1 t_2 t_3 t_4
W_2
3′ 5′
a_1 a_2 t_1 t_2 a_5 t_2 t_3
a_1^* a_2^* t_1^* t_2^* a_5^* t_2^* t_3^*
5′ 3′
B
3′ t_1^* t_2 t_3
5′
a_1 a_2 t_1 t_2 a_5
t_2^* t_3^*
5′ 3′
D
3′ 5′ t_4 t_3 t_3^* t_2 t_2^* t_1^* 5′
a_5^* t_2 t_3 t_2^* t_4 3′
t_1 t_1^* a_2^* a_1^* 3′ 5′
5′ 3′
a_1 a_2 t_1 t_2 a_5 t_2 t_3
C
W_1
t_1 t_2 t_3
3′ 5′ 5′ 3′
t_1^* t_2^* t_3^* t_4

Figure 3. Schematic illustration of the catalytic reaction module, by which the concentration of Species A is extended to twice its original concentration so that Species A can be used to encode the seed "1–2".

$$\begin{cases} A + B \xrightarrow{k_5} C + W_1 \\ C + D \xrightarrow{k_6} 2A + W_2 \end{cases} \tag{3}$$

$$A \xrightarrow{k_7} 2A \tag{4}$$

The process of the reaction can be described as follows: this module involves three main species, including Single-stranded A and Somplexes B and D, where A and D each carry a dye at their 5′ end, B carries a quencher at its 3′ end, and D carries a quencher at the end of t_1^* (close to its 3′ end).

The toehold t_1 of A binds to the domain t_1^* of B, and the branch migration moves gradually to domain t_3, which releases Single-stranded C together with double-stranded W_1. Then, toeholds a_1 and a_2 of C bind to the domains a_1^* and a_2^* of D, respectively, and the branch migration moves gradually to domain t_3, which releases double-stranded W_2 and two single-stranded A molecules. This implies that the concentration of A will be extended to twice its initial concentration.

2.4. Group Cyclic Shift

To extend the DNA-sequence-based initial key (Species A) so that it is sufficiently long, we introduce Algorithm 2 (to clearly describe these algorithms (Algorithms 2 and 3), we followed the way mentioned in [31–33]), hereafter referred to as groupCS, based on the group Cyclic Shift. For any sequence $S = s_1 s_2 \dots s_{n-1} s_n$, let $\mathcal{O}(S) = s_2 \dots s_{n-1} s_n s_1$, and $\mathcal{E}(S) = s_3 \dots s_{n-1} s_n s_1 s_2$. For two sequences S and S', we denote by $S + S'$ the resulting sequence obtained by connecting S' to S (at the end of S).

Algorithm 2: groupCS(**D**, ℓ_0, ℓ), a procedure that extends a DNA sequence **D** of length ℓ_0 to a new one of length at least ℓ.

```
Input: A DNA sequence $\mathbf{D}$ of length $\ell_0$ and an integer $\ell$
Output: A DNA sequence $\mathbf{D_2}$ of length at least $\ell$
begin
    Transform $\mathbf{D}$ into the (0,1)-sequence $S$ according to the first column in Supplementary Table S2.
    $n \longleftarrow 2\ell_0, k \longleftarrow 2\lceil \frac{\ell}{\ell_0} \rceil, \ell' \longleftarrow 0$;
    while $\ell' < \ell$ do
        for $r = 1$ to $k$ do
            if $r \equiv 1 \pmod 2$ then
                $Q \longleftarrow Q + \mathcal{O}(S), S \longleftarrow \mathcal{O}(S)$;
            end
            else
                $Q \longleftarrow Q + \mathcal{E}(S), S \longleftarrow \mathcal{E}(S)$;
            end
        end
        Divide $Q$ into $m = \lceil \frac{nk}{8} \rceil$ groups $Q_1, Q_2, \dots, Q_m$ from left to right such that each $Q_i$ contains
          eight elements, except possibly the last group $Q_m$.
        for $j = 1$ to $m$ do
            if $Q_j$ has length eight (say $Q_j = q_1 q_2 \dots q_8$) and $(4q_6 + 2q_7 + q_8) \notin \{2,3,5,7\}$ then
                if the middle two positions (the fourth and fifth positions) of $Q_j$ are 00 or 11, then
                    then label $Q_j$ by xor;
                end
                else
                    label $Q_j$ by add;
                end
                $\mathbf{D}' \longleftarrow$ the DNA sequence obtained by transforming $Q_j$ according to the rule listed
                  in the $(4q_1 + 2q_2 + q_3 + 1)$-th column of Supplementary Table S2;
                $\mathbf{D_2} \longleftarrow \mathbf{D_2} + \mathbf{D}'$;
            end
        end
        $\ell' \longleftarrow$ the length of $\mathbf{D_2}$;
        if $\ell' \geq \ell$ then
            break;
        end
        else
            $k \longleftarrow 2\lceil \frac{\ell - \ell'}{\ell_0} \rceil$;
            if the length of $Q_m$ is less than eight then
                $Q \longleftarrow Q_m$;
            end
            else
                $Q \longleftarrow \emptyset$;
            end
        end
    end
    return $\mathbf{D_2}$;
end
```

The algorithm first transforms the input DNA sequence **D** into a (0,1)-sequence S according to the first column in Supplementary Table S2 (Line 2). Note that each base corresponds to a (0,1)-sequence of length 2; therefore, S has length $n = 2\ell_0$. Then, a loop iteratively generates a DNA sequence $\mathbf{D_2}$ with length at least ℓ (Lines 4–27).

In each iteration, a new (0,1)-sequence Q is constructed by k rounds of cyclic shift based on the current S (Lines 5–9), where the value of k is initially set as $2\lceil \frac{\ell}{\ell_0} \rceil$ and gradually decreases (Lines 3 and 23). To transform S into a DNA sequence, the algorithm divides S into $m = \lceil \frac{nk}{8} \rceil$ groups (say $Q_1, Q_2, \dots, Q_m$) from left to right such that each Q_i contains eight elements, except possibly the last group Q_m, that is, when $nk \neq 0$ (mod eight), the last group contains less than eight elements (Line 10). Only a group of length eight (say $Q_j = q_1q_2\ldots q_8$) such that $(4q_6 + 2q_7 + q_8) \notin \{2, 3, 5, 7\}$ is transformed into the corresponding DNA sequence according to the rule listed in the $(4q_1 + 2q_2 + q_3 + 1)$-th column of Supplementary Table S2 (Lines 11–18). Next, if the length of $\mathbf{D_2}$ is at least ℓ, then the algorithm breaks out of the loop and returns $\mathbf{D_2}$; otherwise, k is reduced to $2\lceil \frac{\ell-\ell'}{\ell_0} \rceil$, Q is updated by Q_m or $\emptyset$, and the algorithm implements the next iteration (Lines 19–27). We refer to the DNA sequence $\mathbf{D_2}$ returned by groupCS as the *final key*. For examples of groupCS, refer to Supplementary Table S7.

2.5. The BioEN Algorithm

Based on the encryption framework and the above techniques, we developed a DNA-strand-displacement-based encryption algorithm (Algorithm 3), hereafter referred to as BioEN, which utilizes Huffman coding, DNA SDR, and cyclic shift. Note that the reverse process of BioEN is the corresponding decryption algorithm. This is illustrated by an example in Supplementary Table S8.

Algorithm 3: BioEN (a DNA-strand-displacement-based encryption algorithm).

```
Input: Plaintext P
Output: Ciphertext C
begin
    $\mathbf{D_1}$ ⟵ a DNA sequence transformed from the plaintext P by the tri-phase transformation
      (Section 2.2);
    /* Design the key (denoted by K) as the combination of seeds 2-1 and 1-2, where the SDR modules
      encoding these two seeds are the DR-module and the CR-module (see Figures 2 and 3). Denote by
      $|\mathbf{P}|$ the length of P. */
    if $|\mathbf{P}| \equiv 1 \pmod 2$ then
        K ⟵ 2-11-2;
    end
    else
        K ⟵ 1-22-1;
    end
    D ⟵ the DNA sequence key transformed from K according to the rules listed in Supplementary
      Table S1;
    $\mathbf{D_2}$ ⟵ groupCS(D, $\ell_0$, $\ell$);// $\ell_0$ is the length of D, and $\ell$ is the length of $\mathbf{D}_1$.
    $\mathbf{D_{2,1}}, \mathbf{D_{2,2}}, \ldots, \mathbf{D_{2,\frac{n}{4}}}$ ⟵ divide $\mathbf{D_2}$ into $\frac{n}{4}$ groups of DNA sequences of length 4 from left to right,
      where $n$ is the length of $\mathbf{D_2}$; /* Note that each group $\mathbf{D_{2,i}}$ ($i = 1, 2, \ldots, \frac{n}{4}$) corresponds
      to a (0,1)-sequence of length 8 (by the rule listed in the first column in Supplementary Table S2),
      which is labeled as xor or add by groupCS. */
    $\mathbf{D_{1,1}}, \mathbf{D_{1,2}}, \ldots, \mathbf{D_{1,\frac{m}{4}}}$ ⟵ divide $\mathbf{D_1}$ into $\lceil \frac{m}{4} \rceil$ groups of DNA sequences of length 4 from left
      to right, where $m$ is the length of $D_1$; // note that the last group may contain less than 4 bases.
    for $i$=1 to $\frac{m}{4}$ do
        if $\mathbf{D_{2i}}$ is labeled as xor then
            $\mathbf{D_3}$ ⟵ $\mathbf{D_{1i}}$ XOR $\mathbf{D_{2i}}$; // Conduct the XOR operation between $\mathbf{D_{1i}}$ and $\mathbf{D_{2i}}$.
        end
        else
            $\mathbf{D_3}$ ⟵ $\mathbf{D_{1i}}$ ADD $\mathbf{D_{2i}}$;//conduct the ADD operation between $\mathbf{D_{1i}}$ and $\mathbf{D_{2i}}$.
        end
        /* Note that if there are remainder bases (when $n > m$) in $\mathbf{D_2}$ that do not take part in the
          operations, omit them.*/
    end
    C ⟵ the ciphertext obtained by transforming $\mathbf{D_3}$ into ASCII code according to the first column
      in Supplementary Table S2;
    return C;
end
```

In light of the foregoing discussion, it is enough to explain how to transform $\mathbf{D_3}$ into the final ASCII code, i.e., the ciphertext $\mathbf{C}$ (Line 18). First, transform $\mathbf{D_3}$ into a (0,1)-sequence, denoted by S, according to the first column in Supplementary Table S2. Then, divide S into $k = \lceil \frac{t}{8} \rceil$ groups, from left to right, such that each group contains eight elements, except possibly the last group, where t is the length of S. Now, if the last group contains exactly eight elements, then add a new group consisting of eight zeros to S

(at the end of S); if the last group contains less than eight elements, then add enough ones at the end of the last group so that the length of it is extended to eight, and add a new group of length eight consisting zeros or ones such that its corresponding decimal number is equal to the number of ones added to the last group. As a result, a (0,1)-sequence of length $8(k+1)$ is obtained, which can be divided into $(k+1)$ groups, from left to right, such that each group contains eight elements. We refer to each of these groups as an *ASC-group*. Observe that the last ASC-group is used to identify how many ones are added, which serves for the decryption.

3. Validation of Feasibility

3.1. Experimental Setup

To show the feasibility of our approach in encryption, each experiment was set up with an experimental group and a control group. The concentration of the target DNA was expressed in the form of fluorescence intensity. The assembled DNA molecules were mixed according to the designed ratio, and the fluorescence intensity was monitored to obtain the final concentration of the target DNA strand.

All spectrofluorimetric measurements were performed using a real-time PCR system (QuantStudio 3 & 5 fluorescence quantitative PCR, Thermo Fisher Scientific, Waltham, MA, USA) equipped with a 96-well fluorescence plate reader. In the hold stage, the temperature was decreased by 1.6 °C to 4 °C/s and was then held for 10 s prior to the PCR stage. Then, the temperature was increased by 3 °C to 23 °C/s, and the fluorescence intensity was monitored every 10 s. The volume of each DNA sample was 20 μL.

3.2. Tools and Data

The sequences of all DNA strands in the experiment, listed in Supplementary Table S4, were designed by obtaining the original sequences using Nupack and then modifying the sequences by hand. The DNA oligonucleotides used were manufactured by Sangon Biotech (Shanghai, China). DNA oligonucleotides were purified by Sangon using high-performance liquid chromatography. Individual unlabeled DNA oligonucleotides were dissolved in 1 × TE buffer (nuclease free, pH 8.0, Sigma-Aldrich, St. Louis, MO, USA) and stored at −20 °C. Oligos labeled with dyes or quenchers were dissolved in deionized water (Milli-Q) and stored in deionized water at −20 °C. The DNA sample concentration was measured by *NanoPhotometer*® N120 (Implen Inc., Westlake Village, CA, USA). All reagents were of analytical grade without further purification.

The DNA oligonucleotides were mixed in Tris-EDTA buffer (1×Tris-EDTA: 40 mM Tris base, 20 mM acetic acid, 2 mM EDTA adjusted to pH 8.0) with 12.5 mM $MgCl_2$. All DNA complexes (listed in Supplementary Table S3) were mixed with an equal amount of corresponding single-stranded DNA to 10 μM. All samples were annealed in a polymerase chain reaction (PCR) thermal cycler. The temperature was set at 95 °C for 2 min initially and then decreased to 4 °C at a rate of −0.1 °C every 6 s. The hybridized molecules were stored at 4 °C for further use.

For simulation and dynamic analysis, we used Visual DSD [34]. The simulation duration was set to 600 s. The reactant concentration was at least 10 nM.

3.3. Experimentation Procedure

The initial key was obtained by biological experiments. Two DNA strand displacement modules were designed to obtain seeds 2–1 and 1–2. Before carrying out the biological experiments, simulation experiments were conducted as an auxiliary verification.

3.3.1. Simulation Experiment of the DR-Module

In the Degradation Reaction (DR)-module, there were Single-stranded A and auxiliary Complexes B, D and G. The initial concentration of A was $[A]_0$= 20 nM, and the initial concentration of each of B, D, and G was C_m = 10 nM. The (DSD) reaction rates $k_1 = k_2 = 7 \times 10^{-4}$/nM/s and $k_3 = 10^{-1}$/nM/s, where k_3 is the maximum reaction rate.

The rate constants of the corresponding DNA reactions were determined according to the rate constants of the formal chemical reactions, which were equal to the rate constants of the corresponding DNA strand reaction multiplied by the initial concentration of the auxiliary complexes strands. k_1, k_4, and C_m satisfy $k_4 = k_1 C_m$. The simulation process was performed for 600 s, and the concentration of A was reduced from 20 nM to 10 nM (see Figure 4a).

3.3.2. Biological Experiments of the DR-Module

To obtain the seed 2–1, we conducted two groups of biochemical reactions, named experiment group and control group, respectively, where the concentration of all species involved in the experiments (i.e., A, B, D, and G) was 10 µM, and the control group was just for reference. The experiment group included 4 µL of A, D, and G, respectively, and 6 µL of B, while the control group included 4 µL of A and 14 µL Tris-EDTA buffer (1× Tris-EDTA: 40 mM Tris base, 20 mM acetic acid, and 2 mM EDTA adjusted to pH 8.0). We put these two groups into the fluorescence quantitative PCR instrument and examined the fluorescence intensity change of A. Initially, they had the same concentration of A. When the reaction tended to be stable, the concentration of A in the experiment group was reduced by half, while the concentration of A in the control group was unchanged (see Figure 5a).

To show the key sensitivity of our approach (see Section 4.1.2), we conducted a contrast experiment, in which the experiment group included 5 µL of A, B, D, and G, respectively, while the control group included 5 µL of A and 15 µL Tris-EDTA buffer. The results are shown in Figure S1.

3.3.3. Simulation Experiment of the CR-Module

In the Catalysis Reaction (CR)-module, there are single-stranded A and auxiliary complexes B and D. The initial concentration of A is $[A]_0$= 10 nM and the initial concentration of each of B and D is C_m= 10 nM. The (DSD) reaction rates $k_5 = 9 \times 10^{-3}/\mathrm{nM/s}$ and $k_6 = 10^{-2}/\mathrm{nM/s}$, where k_6 is the maximum reaction rate. The rate constants of the formal chemical reactions is equal to the rate constants of the corresponding DNA strand reaction multiplied by the initial concentration of the auxiliary complexes strands. k_5, k_7 and C_m satisfy $k_7 = k_5 C_m$. The simulation process was performed for 600 s, and the concentration of A was increased from 10 nM to 20 nM (see Figure 4b).

3.3.4. Biological Experiments of CR-Module

To obtain the seed 1–2, we also conducted the two groups of experiments for the DR-module, in which the concentration of A, B, and D was 10 µM. The experiment group included 5 µL of A and D, respectively, and 6 µL of B, while the control group included 5 µL of A and 11 µL Tris-EDTA buffer. We put these two groups into the fluorescence quantitative PCR instrument and examined the fluorescence intensity change of A. Initially, they had the same concentration of A. When the reaction tended to be stable, the concentration of A in the experiment group was doubled, while the concentration of A in the control group was unchanged (see Figure 5b).

To show the key sensitivity of our approach (see Section 4.1.2), we conducted a contrast experiment, in which the experiment group included 7 µL of A, B, and D, respectively, while the control group included 7 µL of A and 14 µL Tris-EDTA buffer. The results are shown in Figure S2.

3.4. Experimental Results

The results are shown in Figures 4 and 5, respectively. As expected, the simulation and biological experiment produced consistent results. This provides a guarantee for the performance of these two SDR modules, which can be used to encode the seeds 2–1 and 1–2, respectively.

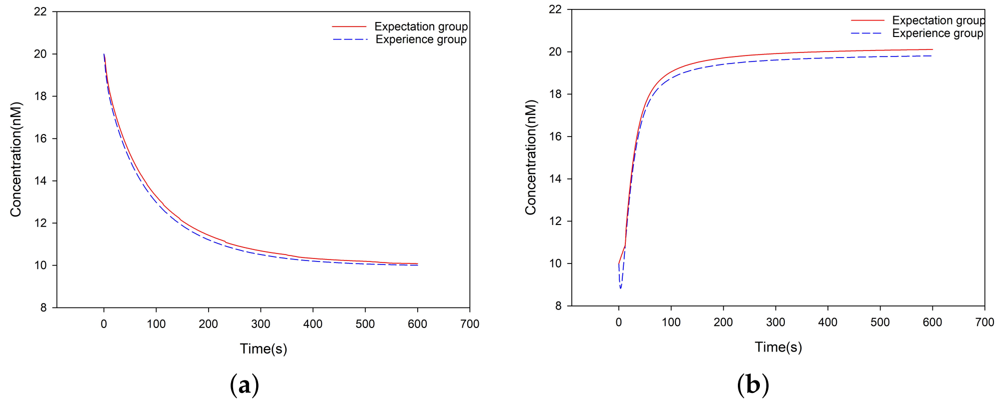


Figure 4. Simulation results. (**a**) The evolution of the concentration of Species A in the DR-module. The whole process takes 600 s, and the concentration of A is reduced from 20 nM to 10 nM. (**b**) The evolution of the concentration of Species A in the CR-module. The whole process takes 600 s, and the concentration of *A* is doubled.

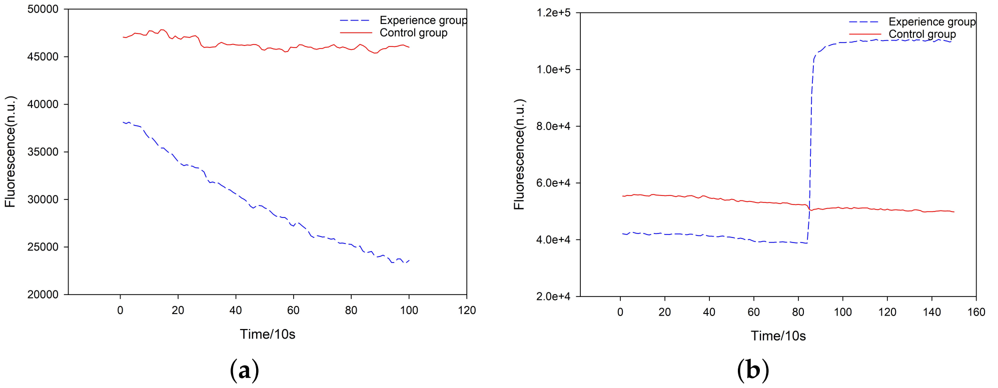


Figure 5. Biological experiment results. (**a**) The evolution of the concentration of Species A in the DR-module. (**b**) The evolution of the concentration of A in the CR-module. The concentration change of Species A in the experiment is the same as that in the simulation.

4. Security Analysis

In this section, we analyzed the security of our encryption algorithm.

4.1. Key Sensitivity

An excellent encryption scheme should be sensitive to the key, meaning a minor change to the key will cause major changes to the results of encryption and decryption. Because our key is highly associated with biological experiments and the experiments are very sensitive to the environment, the desired key can be generated only when all experimental conditions are set correctly. Any mistake will lead to a different result, which implies that the key is sensitive. In addition, the key extension mechanism (groupCS) introduces considerable confusion to the final key. To illustrate this, we designed three types of experiments. The plaintext we used was "*anewencryptionapproachusingdnabiotechnologyandhuffmancoding*", and the seed was 2-11-2.

4.1.1. Change One Base

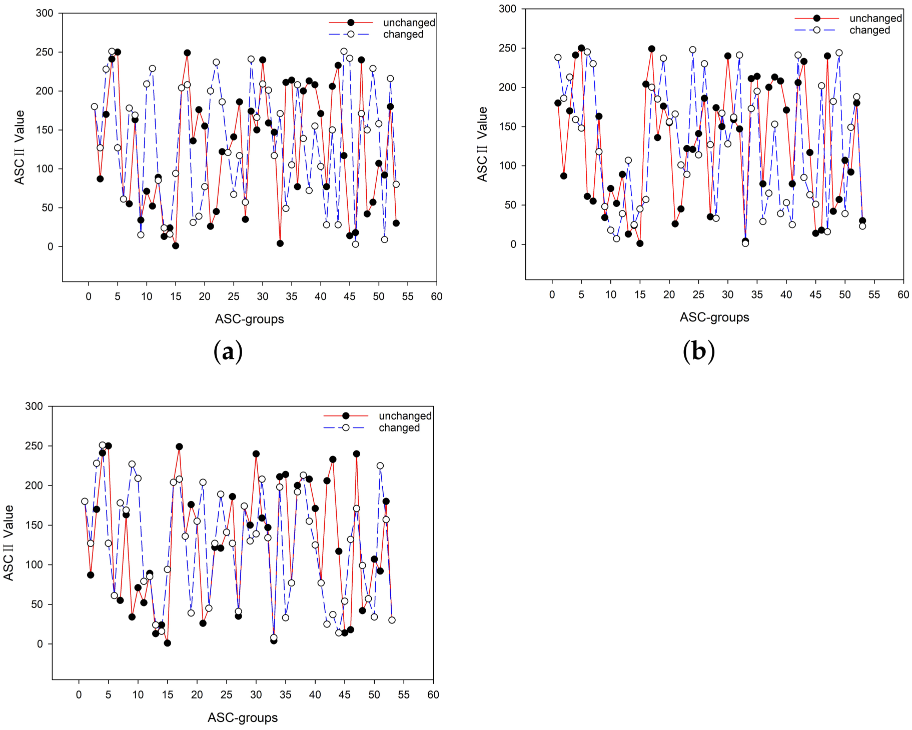
Referring to Supplementary Table S1, the seed 2–11–2 was transformed into the DNA sequence key D = **GCCCGCAAGCCGGCCGGCAAGCCC**. We wanted to investigate the difference of the encryption results (obtained by BioEN) when an arbitrary base in **D** is changed. In the experiment, we selected the fifth base **G** and changed it to **T**, i.e., the changed DNA sequence was D′= **GCCCTCAAGCCGGCCGGCAAGCCC**. Based on D and D′, the ciphertexts obtained by BioEN were completely different; see Figure 6a.

4.1.2. Change Experiment Conditions

Note that when conducting the biological experiment, for the DR-module, the concentration ratio of Species A, B, D, and G was 2:3:2:2; and for the CR-module, the concentration ratio of A, B, and D was 5:6:5. To show the key is sensitive, we conducted a new experiment by setting the concentration ratio of A, B, D, and G to 1:1:1:1 for the DR-module and the concentration ratio of A, B, and D to 1:1:1 for the CR-module (see Supplementary Figures S1 and S2 for the results of the experiment). Consequently, the concentration changes of Species A for the DR-module and CR-module were 8:5 and 3:5, respectively, by which the seed we obtained was 8–53–5. Thus, the ciphertexts obtained by BioEN, based on the seeds 2–11–2 and 8–53–5, respectively, were very different; see Figure 6b.

4.1.3. Change One Element in the Process of Extending the Key

To extend the seed 2-11-2, groupCS first transforms it into the DNA sequence **D**, which is further transformed to a (0,1)-sequence *S* by the rule listed in the first column of Supplementary Table S2. Then, based on *S*, a longer (0,1)-sequence *Q* is constructed according to the corresponding rules (Lines 6–9; groupCS). Note that here, we only considered the first iteration. We wanted to change an element of *Q* to test the effect on the final ciphertext. Thus, given the importance of each element's position in *Q* (Lines 10–14; groupCS), we changed the eighth element of *Q* from zero to one, and all other elements remained unchanged. Figure 6c shows that even such a slight modification led to a significant change in the final ciphertexts.



(**a**)

(**b**)

(**c**)

Figure 6. Key sensitivity tests, where the horizontal axis displays the specific ASC-groups, and the vertical axis presents the corresponding ASCII value of each ASC-group. The corresponding ASCII values exhibit great differences when factors related to the key are changed: (**a**) change one base; (**b**) change experimental conditions; (**c**) change one element in the process of extending the key.

4.2. Key Space Analysis

Note that the (0,1)-sequence S mentioned in Section 4.1.3 has length 48. Denote by $\mathcal{R}(S)$ the resulting (0,1)-sequence obtained from S by conducting the shift operation once, and let:

$$\mathcal{R}^i(S) = \underbrace{\mathcal{R}(\mathcal{R}(\mathcal{R}(\ldots\mathcal{R}}_{i\ \mathcal{R}s}(S))))$$

i.e., when $i \equiv 1(\text{mod } 2)$, $\mathcal{R}^i(S) = \mathcal{O}(\mathcal{R}^{i-1}(S))$; when $i \equiv 0(\text{mod } 2)$, $\mathcal{R}^i(S) = \mathcal{E}(\mathcal{R}^{i-1}(S))$, where $\mathcal{R}^0(S) = S$ and i is a positive integer. Since the length of S is finite, there may exist some positive integer r such that $\mathcal{R}^r(S) = S$ and $\mathcal{R}^{r+i}(S) = \mathcal{R}^i(S)$, where i is a nonnegative positive integer. We call the smallest r with this property the *rank* of S. Clearly, the final key is generated based on a (0,1)-sequence of length 48r, where r is the rank of S. We refer to the set of all distinct (0,1)-sequences of length 48r as the *key space* of the encryption algorithm BioEN. By a simple exhaustive analysis, we have the following proposition, which shows that the key space of BioEN is large enough to be secure. The detailed proof can be found in Supplementary Table S9.

Theorem 1. *The rank of S is 32, and the cardinality of the key space of BioEN is* 2^{1536}.

4.3. Statistic Characteristic

We investigated the ASCII values of the characters appearing in the plaintext and ciphertext. Compared to the range of the ASCII values, we saw that the ASCII value distribution of the plaintext was 95–125, whereas that of the ciphertext was 0–255; see Figure 7. Such a large difference in ASCII values provides a strong guarantee for protection against statistical attacks.

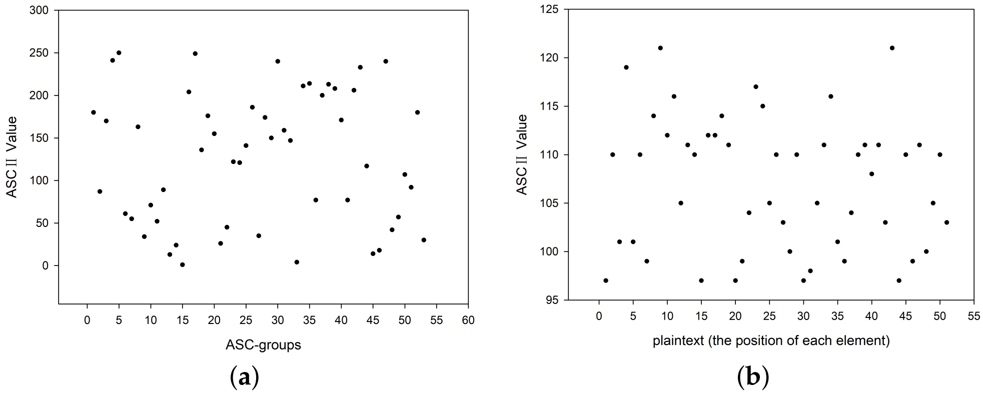


Figure 7. The ASCII value distribution of (**a**) the plaintext and (**b**) the ciphertext.

5. Conclusions

We proposed a bio-experiment-based DNA encryption framework for data security (i.e., Algorithm 1). Based on the proposed framework, we introduced an encryption algorithm (i.e., BioEN) by designing a Huffman-coding-based method tri-phase transformation to deal with the unprocessed plaintext, two DNA SDR modules to generate the initial key, and a cyclic-shift-based mechanism (i.e., groupCS) to extend the key. The proposed algorithm highlights the importance of biochemical experiments. To validate the feasibility of the proposed algorithm, we conducted both a DSD simulation and a biochemical experiment. Compared to the existing DNA strand replacement encryption algorithms, the proposed algorithm is heavily dependent on the experiments and generates pseudo-random sequences by tracing the concentration change of the target DNA strand. Further analysis of the security showed that our algorithm is key sensitive, has a large key space, and can effectively resist statistical attacks. Compared with the works in [28,29], our encryption approach has the advantage of performing encryption through DNA strand displacement experiments rather than staying in the theory or simulation stage, which is expected to push forward the research of DNA-strand-displacement-based encryption. Though designed for text encryption, our encryption framework may be also applicable to image encryption or other areas of encryption, which would be worth exploring in future work.

Supplementary Materials: The following supporting information can be downloaded at: www.mdpi.com/article/10.3390/nano12050877/s1, Table S1: DNA coding. Table S2: DNA encoding and decoding rules. Table S3: Synthetic DNA complexes. Table S4: DNA sequence design. Table S5: DNA XOR operation. Table S6: DNA ADD operation. Table S7: Example of groupCS. Table S8. An example of the algorithm BioEN. Table S9: Proof of the key space analysis. Figure S1: The fluorescence intensity changes when the concentration ratio of A,B,D, and G is 1:1:1:1 in the DR-module. Figure S2: The fluorescence intensity changes when the concentration ratio of A, B, and D is 1:1:1 in the CR-module.

Author Contributions: Conceptualization, E.Z. and X.L.; investigation, E.Z., X.L. and C.L.; methodology, E.Z., X.L. and C.C.; supervision, C.L.; writing—original draft, E.Z. and X.L.; writing—review and editing, E.Z. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grants (61872101, 62172072), in part by Natural Science Foundation of Guangdong Province of China under Grant 2021A1515011940, in part by Natural Science Foundation of Liaoning Province of China under Grant 2021-MS-114, in part by Young Elite Scientists Sponsorship Program by CAST under grant 2018QNRC001, and in part by the Fundamental Research Funds for the Central Universities under grant DUT21JC18.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to express their sincere gratitude to the anonymous Reviewers for their constructive comments, which improved our manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mali, K.; Chakraborty, S.; Roy, M. A study on statistical analysis and security evaluation parameters in image encryption. *Int. J. Sci. Res. Dev.* **2015**, *3*, 339–343.
2. Liang, Z.; Qin, Q.; Zhou, C.; Wang, N.; Xu, Y.; Zhou, W. Medical image encryption algorithm based on a new five-dimensional three-leaf chaotic system and genetic operation. *PLoS ONE* **2021**, *16*, e0260014. [CrossRef] [PubMed]
3. Gehani, A.; LaBean, T.; Reif, J. DNA-based cryptography. In *Aspects of Molecular Computing*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 167–188.
4. Roy, M.; Mali, K.; Chatterjee, S.; Chakraborty, S.; Debnath, R.; Sen, S. A study on the applications of the biomedical image encryption methods for secured computer aided diagnostics. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 881–886.
5. Wang, X.; Teng, L. A one-time one-key encryption algorithm based on the ergodicity of chaos. *Chin. Phys. B* **2012**, *21*, 020504. [CrossRef]
6. Mokhtar, M.A.; Gobran, S.N.; El-Badawy, E.S.A. Colored image encryption algorithm using DNA code and chaos theory. In Proceedings of the 2014 International Conference on Computer and Communication Engineering, Tianjin, China, 27–28 March 2014; pp. 12–15.
7. Yang, J.; Ma, J.; Liu, S.; Zhang, C. A molecular cryptography model based on structures of DNA self-assembly. *Chin. Sci. Bull.* **2014**, *59*, 1192–1198. [CrossRef]
8. Liu, H.; Wang, X. Image encryption using DNA complementary rule and chaotic maps. *Appl. Soft Comput.* **2012**, *12*, 1457–1466. [CrossRef]
9. Rehman, A.; Liao, X.; Kulsoom, A.; Abbas, S. Selective encryption for gray images based on chaos and DNA complementary rules. *Multimed. Tools Appl.* **2015**, *74*, 4655–4677. [CrossRef]
10. Liu, Y.; Wang, J.; Fan, J.; Gong, L. Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences. *Multimed. Tools Appl.* **2016**, *75*, 4363–4382. [CrossRef]

11. Wu, J.; Liao, X.; Yang, B. Image encryption using 2D Hénon-Sine map and DNA approach. *Signal Process.* **2018**, *153*, 11–23. [CrossRef]
12. Zhang, J.; Huo, D. Image encryption algorithm based on quantum chaotic map and DNA coding. *Multimed. Tools Appl.* **2019**, *78*, 15605–15621. [CrossRef]
13. Zheng, J.; Liu, L. Novel image encryption by combining dynamic DNA sequence encryption and the improved 2D logistic sine map. *IET Image Process.* **2020**, *14*, 2310–2320. [CrossRef]
14. Wang, X.; Zhang, M. A new image encryption algorithm based on ladder transformation and DNA coding. *Multimed. Tools Appl.* **2021**, *80*, 13339–13365. [CrossRef]
15. Wang, X.; Zhang, Y.; Zhao, Y. A novel image encryption scheme based on 2-D logistic map and DNA sequence operations. *Nonlinear Dyn.* **2015**, *82*, 1269–1280. [CrossRef]
16. Wei, D.; Jiang, M. A fast image encryption algorithm based on parallel compressive sensing and DNA sequence. *Optik* **2021**, *238*, 166748. [CrossRef]
17. Popli, M. DNA Cryptography: A Novel Approach for Data Security Using Genetic Algorithm. *Int. J. Adv. Res. Comput. Sci. Manag. Stud.* **2018**, *6*, 53–63.
18. Ravichandran, D.; Murthy, B.; Balasubramanian, V.; Fathima, S.; Amirtharajan, R. An efficient medical image encryption using hybrid DNA computing and chaos in transform domain. *Med. Biol. Eng. Comput.* **2021**, *59*, 589–605. [CrossRef]
19. Zhu, E.; Chen, C.; Rao, Y.; Xiong, W. Biochemical Logic Circuits Based on DNA Combinatorial Displacement. *IEEE Access* **2020**, *8*, 34096–34103. [CrossRef]
20. Wang, Y.; Li, Z.; Sun, J. Three-variable chaotic oscillatory system based on DNA strand displacement and its coupling combination synchronization. *IEEE Trans. Nanobioscience* **2020**, *19*, 434–445. [CrossRef]
21. Zou, C.; Wei, X.; Zhang, Q.; Liu, C.; Liu, Y. Solution of equations based on analog DNA strand displacement circuits. *IEEE Trans. Nanobioscience* **2019**, *18*, 191–204. [CrossRef]
22. Wang, F.; Lv, H.; Li, Q.; Li, J.; Zhang, X.; Shi, J.; Wang, L.; Fan, C. Implementing digital computing with DNA-based switching circuits. *Nat. Commun.* **2020**, *11*, 1–8. [CrossRef]
23. Liu, C.; Liu, Y.; Zhu, E.; Zhang, Q.; Wei, X.; Wang, B. Cross-Inhibitor: a time-sensitive molecular circuit based on DNA strand displacement. *Nucleic Acids Res.* **2020**, *48*, 10691–10701. [CrossRef]
24. Wang, R.; Wang, S.; Xu, X.; Jiang, W.; Zhang, N. MNAzyme probes mediated DNA logic platform for microRNAs logic detection and cancer cell identification. *Anal. Chim. Acta* **2021**, *1149*, 338213. [CrossRef] [PubMed]
25. Gao, Y.; Yu, H.; Tian, J.; Xiao, B. Nonenzymatic DNA-Based Fluorescence Biosensor Combining Carbon Dots and Graphene Oxide with Target-Induced DNA Strand Displacement for microRNA Detection. *Nanomaterials* **2021**, *11*, 2608. [CrossRef] [PubMed]
26. Okumura, S.; Nixon Hapsianto, B.; Lobato-Dauzier, N.; Ohno, Y.; Benner, S.; Torii, Y.; Tanabe, Y.; Takada, K.; Baccouche, A.; Shinohara, M.; et al. Morphological Manipulation of DNA Gel Microbeads with Biomolecular Stimuli. *Nanomaterials* **2021**, *11*, 293. [CrossRef] [PubMed]
27. Tang, Z.; Yin, Z.; Wang, L.; Cui, J.; Yang, J.; Wang, R. Solving 0–1 Integer Programming Problem Based on DNA Strand Displacement Reaction Network. *ACS Synth. Biol.* **2021**, *10*, 2318–2330. [CrossRef] [PubMed]
28. Zhang, Z.; Xiao, B.; Zheng, X.; Zhou, C. An image encryption algorithm based on chaos system and DNA strand displacement model. In Proceedings of the 2nd International Conference on Artificial Intelligence: Techniques and Applications, DEStech Transactions on Computer Science and Engineering, Settat, Morocco, 28–30 June 2017; pp. 102–107.
29. Zou, C.; Wei, X.; Zhang, Q.; Zhou, C.; Zhou, S. Encryption Algorithm Based on DNA Strand Displacement and DNA Sequence Operation. *IEEE Trans. Nanobioscience* **2021**, *20*, 223–234. [CrossRef] [PubMed]
30. Ailenberg, M.; Rotstein, O.D. An improved Huffman coding method for archiving text, images, and music characters in DNA. *Biotechniques* **2009**, *47*, 747–754. [CrossRef]
31. Jäntschi, L. Formulas, Algorithms and Examples for Binomial Distributed Data Confidence Interval Calculation: Excess Risk, Relative Risk and Odds Ratio. *Mathematics* **2021**, *9*, 2506. [CrossRef]
32. Zhu, E.; Jiang, F.; Liu, C.; Xu, J. Partition independent set and reduction based approach for partition coloring problem. *IEEE Trans. Cybern.* **2020**, *in press*. [CrossRef]
33. Bolboacă, S.D.; Roşca, D.D.; Jäntschi, L. Structure-Activity Relationships from Natural Evolution. *MATCH Commun. Math. Comput. Chem.* **2014**, *71*, 149–172.
34. Lakin, M.R.; Youssef, S.; Polo, F.; Emmott, S.; Phillips, A. Visual DSD: A design and analysis tool for DNA strand displacement systems. *Bioinformatics* **2011**, *27*, 3211–3213. [CrossRef]