Exploiting Substitution Box for Cryptanalyzing Image Encryption Schemes With DNA Coding and Nonlinear Dynamics

Chengrui Zhang, Junxin Chen, *Senior Member, IEEE*, Dongming Chen, *Member, IEEE*, Wei Wang, *Member, IEEE*, Yushu Zhang, *Senior Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*—In recent years, a number of image encryption schemes based on DNA coding and nonlinear dynamics have been proposed. Generally, these DNA-based schemes first encode plaintext images into DNA sequences and then encrypt them with pseudorandom elements produced by chaotic systems or other nonlinear dynamics. Although ciphertexts can pass some security tests, many image encryption schemes are being shown to have intrinsic flaws and that they cannot guarantee a high level of security. In this article, we cryptanalyze a family of image encryption schemes for which the encryption kernel is DNA coding or its variant. The complex DNA operation can be simplified as a substitution box (S-box). The whole cryptosystem's security level is thus significantly decreased and is vulnerable to the chosen-plaintext attack. Applications of this concept to break five ciphers are theoretically presented and experimentally verified. In addition, some suggestions for resisting similar attacks are also given in this article.**

***Index Terms*—Chosen-plaintext attack, DNA coding, image encryption, S-box matrix.**

I. Introduction

BENEFITING from advancements in information technologies, recent years have witnessed the dramatic popularity of multimedia exchange over public networks. The secure transmission of massive multimedia data has attracted increasing attention [1], [2]. Encryption is an important method of information security, and various encryption methods have been developed. However, image data have many characteristics that text data do not have, such as high data redundancy, correlation of adjacent data, and large data volume. Although traditional encryption algorithms such as 3-DES or AES can encrypt texts converted from images, there is still a desire to design encryption algorithms specific to digital images.

Many image encryption schemes based on chaotic systems and other auxiliary technologies have been proposed. Due to their characteristics, such as unpredictability and sensitivity to initial conditions, chaotic systems provide a promising basis for designing image encryption schemes [3]. In addition to chaotic systems, techniques from some other fields have also been exploited to improve the security of encryption schemes. Hua et al. [4] designed a novel image encryption scheme by using Latin squares to achieve a high level of security, while Mohamed et al. [5] exploited quaternion multiplication for fast image encryption. Gan et al. [6] proposed an image encryption scheme based on three-dimensional Brownian motion. In addition, cellular automata [7], compressed sensing [8], [9], deep learning [10], [11], [12], steganography and watermarking [13], [14], [15], [16] have also been exploited for securing image transmission over public transmission infrastructures. As early as 1994, Adleman et al. [17] used DNA molecules to store data and to perform calculations. A DNA-based encryption scheme encodes pixel values as DNA sequences and then uses those sequences to calculate with other elements produced by chaotic systems [18], [19], [20]. This type of encryption scheme is the primary concern of this article.

A. Related Work

Recently, many image encryption schemes based on DNA coding have been proposed. Most of them use novel chaotic systems or other nonlinear dynamics to generate encryption elements. Wang et al. [21] proposed a DNA-based image encryption scheme that uses a novel one-dimensional hybrid chaotic map. Simulation results show that the scheme has good security performance. Elsaid et al. [22] proposed a robust cryptosystem based on hyperchaotic systems. The execution time of the proposed scheme makes it convenient for various applications.

Manuscript received 9 November 2022; revised 7 February 2023 and 29 March 2023; accepted 2 May 2023. Date of publication 16 May 2023; date of current version 18 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62171114, in part by the Key Technologies Research and Development Program of Liaoning Province in China under Grant 2021JH1/10400079, in part by the Fundamental Research Funds for the Central Universities under Grant DUT22RC(3)099, and in part by the Applied Basic Research Project of Liaoning Province under Grant 2023JH2/101300185. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Amit Kumar Singh. *(Corresponding author: Dongming Chen.)*

Chengrui Zhang and Dongming Chen are with the Software College, Northeastern University, Shenyang 110169, China (e-mail: 2010496@stu.neu.edu.cn; chendm@mail.neu.edu.cn).

Junxin Chen is with the School of Software, Dalian University of Technology, Dalian 116621, China (e-mail: junxinchen@ieee.org).

Wei Wang is with the School of Medical Technology, Beijing Institute of Technology, Beijing 100081, China, and also with the Department of Engineering, Shenzhen MSU-BIT University, Shenzhen 518172, China (e-mail: ehomewang@ieee.org).

Yushu Zhang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: yushu@nuaa.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

Digital Object Identifier 10.1109/TMM.2023.3276504

Many DNA-based schemes use plaintext-related information to improve security. Paul et al. [23] proposed a color image encryption scheme by using SHA-2. Multiple diffusion operations were conducted to improve the security level of the scheme. Zhang et al. [24] utilized the hamming distances to enhance the proposed scheme's robustness in resisting plaintext attacks. A few DNA-based schemes have been specifically designed for medical images. Wu et al. [25] proposed an encryption scheme that incorporates the awareness property of medical image content to increase the complexity of breaking the encryption. Experiments demonstrate that their scheme can resist various attacks in robustness and effectiveness when transmitting data in real scenarios.

The chosen-plaintext attack (CPA) is a scenario where attackers can obtain ciphertexts for arbitrary plaintexts [26]. Then, the attackers use these known plaintext-ciphertext pairs to derive the information to break cryptosystems. In the past, the chosen-plaintext attack played an important role in war. For example, US Navy cryptanalysts decoded ciphertext with a chosen-plaintext attack and won the battle [27]. In modern life, chosen-plaintext attacks are often used to break symmetric ciphers. Many symmetric image encryption schemes are broken with the chosen-plaintext attack [28], [29], [30]. In addition, the chosen-plaintext attack is also important in public key cryptography. For example, Gregory et al. [31] introduced a chosen-plaintext vulnerability for SSL/TLS. Therefore, the chosen-plaintext attack plays a vital role in cryptography's design and security evaluation.

Many image encryption schemes based on DNA coding have been found to be vulnerable to various attacks. The image encryption scheme proposed in [32], which is essentially shuffling-only encryption, was cracked by Akhavan et al. [33] with only two chosen-plaintext images. The encryption scheme [34], which has an invalid diffusion phase and fixed DNA coding rules, is cryptanalyzed by Wen et al. [35]. Some encryption schemes are vulnerable to plaintext attacks. The secret key or its equivalent version can be revealed easily. The secret key of a chaos-based image encryption scheme [36] combining DNA coding and entropy was obtained by Su et al. [37]. Panwar et al. [38] presented the equivalent version of the original cipher [39] and obtained the equivalent key with chosen-plaintext images. Additionally, many image ciphers based on DNA coding depend too much on substitution and have unreasonable structures. A scheme [40] was cryptanalyzed by Chen et al. [29] with substitution boxes, which were obtained with the chosen-plaintext attack.

In addition to DNA-based schemes, many other image ciphers have been demonstrated to be less secure than expected [41]. The famous chaos-based image encryption scheme [3] proposed by Fridrich was broken by Solak et al. [42]. Chen et al. [30] found the security flaws in Ye's scheme [43] and broke it with the chosen-plaintext attack. Munir et al. [44] cracked the scheme [45] with various attacks and gave many numerical examples to illustrate. Arora et al. [46] cryptanalyzed Deb's cipher [47] and suggested detailed improvements. More similar works have been performed, such as Zhou's scheme [48] and Wu's cipher [39], which were cryptanalyzed by Dhall et al. [49] and Panwar et al. [38], respectively. The permutation-diffusion architecture has been widely applied to image ciphers. However, Chen et al. [50] proposed the chosen-ciphertext attack to demonstrate that it is not secure enough. There are also a few permutation-only encryption schemes, such as [51]. They have been proven vulnerable to plaintext attacks [28].

Taking an overview of the cryptanalysis works mentioned above, the keywords describing our work include 'generalized cryptanalysis', 'DNA coding', 'equivalent process', and 'S-box matrix'. They are the main motivations of and innovations in this article.

B. Our Contributions

In this article, we evaluate the security of a family of image encryption schemes that are based on DNA coding. The 'a family of ciphers' means some encryption schemes with the same characteristics. These schemes can be grouped into one category. In this article, simply speaking, this 'family' means some DNA-based encryption schemes. Specifically, the encryption schemes of the family satisfy the following conditions. We show that the cipher may be vulnerable to the chosen-plaintext attack if it satisfies these conditions.

- The encryption elements depend only on the secret key. In other words, these elements are independent of the plaintext image.
- DNA coding and its derived operations, e.g., DNA addition, subtraction, and XOR, are utilized in the encryption process.
- The encryption process may contain multiple consecutive derived operations of DNA coding.
- There is no diffusion phase at the DNA level.

Five image encryption schemes [52], [53], [54], [55], [56] based on DNA encoding were broken with the chosen-plaintext attack. The main idea of these attacks is the same, i.e., regarding the DNA-coding encryption process as an S-box. The idea of S-box transformation may be applied to more cryptanalysis tasks of image ciphers based on DNA coding. Our contributions are summarized as follows.

- DNA coding and its derived operations are analyzed in detail.
- The weaknesses of DNA coding technology have been found. The transformed S-box process is used as the equivalent key.
- Five image encryption schemes are broken with CPA, which takes S-box transformation as the core. Both theoretical and experimental results validate our attacks.
- Some suggestions for improvements are given.

The previous cryptanalysis works [29], [57], [58] based on the S-box analyzed only specific encryption schemes. They did not provide an intuitive S-box conversion process or a complete encryption information storage structure. They did not give suggestions for improvement of the flawed encryption schemes either. We apply the S-box transformation method to more DNA-based schemes. The specially designed figures illustrate the S-box transformation and equivalent encryption processes. The S-box matrix is used to store the information. Many suggestions are given for improving the security of these flawed encryption schemes.

TABLE I
THE NOTATIONS

Notation	Style	Description
x	lower case	a variable
X	capital	a matrix or set, generally denotes an image except A, T, C, and G
$X(i,j)$	with location coordinates	a matrix element at row i and column j, a pixel value of an image
$X(i)$	with serial number	the ith element of the sequence

While our cryptanalysis can be regarded as an extension of related works in [57] and [29], our work further exploits the S-box for cryptanalyzing image encryption schemes with DNA coding and successfully breaks more ciphers. Zhang et al. [57] broke the S-box-only ciphers with the chosen-plaintext attack. In other words, the encryption scheme uses the real S-box to complete the encryption. However, in this article, DNA coding operations are regarded as equivalent S-box processes. In addition, the S-box of [57] has detailed information, while our approach utilizes only the S-box's input and output. The function of the S-box is used only to simplify the encryption process. Chen et al. [29] cryptanalyzed a DNA-based encryption scheme with CPA. They regarded some DNA coding encryption processes as S-boxes to simplify the cipher. However, they did not give a detailed structure to restore the S-box information. They considered one encryption scheme only. In our article, we propose the S-box matrix to restore and to utilize the encryption information. We further explore the S-box to cryptanalyze other encryption schemes. The common characteristics of flawed ciphers are summarized, and more ciphers are broken, which is a great development of [29]. In addition, the weaknesses of a family of DNA-based ciphers are found, and some suggestions for resisting similar attacks are given. Although many image encryption schemes are cracked in this article, these findings do not indicate that the DNA coding technology is unsuitable for image encryption. Referring to the weaknesses described in this article and utilizing the DNA coding techniques properly, DNA-based encryption schemes with a high level of security can be designed in future research.

C. Organization of the Article

The remainder of this article is organized as follows. The notation and detailed DNA coding operations are given in Section II. Section III presents the equivalent S-box transformation method in detail, while its cryptographic applications are described in Section IV. Some discussions are provided in Section V Finally, conclusions are drawn in the last section.

II. PRELIMINARY

A. Notations

Most of the notation used in this article is listed in Table I. Complementary descriptions are as follows:

TABLE II
DNA ENCODING AND DECODING RULES

	1	2	3	4	5	6	7	8
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
G	01	10	00	11	00	11	01	10
C	10	01	11	00	11	00	10	01

TABLE III
DNA ADDITION AND XOR RULES

Addition					XOR				
+	A	G	C	T	$\oplus$	A	G	C	T
A	A	G	C	T	A	A	G	C	T
G	G	C	T	A	G	G	A	T	A
C	C	T	A	G	C	C	T	A	G
T	T	A	G	C	T	T	A	G	A

- Generally, P and C denote the plaintext and ciphertext in an encryption scheme, respectively.
- If the size of the image is $M \times N$, then the pixels of the plaintext image can be represented as $P = \{P(i,j), 1 \le i \le M, 1 \le j \le N\}$ or $P = \{P(i), 1 \le i \le N \times N\}$.

B. Overview of DNA Coding Encryption

The pixel values of the image can be converted into nucleic acid bases, which include A (adenine), C (cytosine), G (guanine), and T (thymine). Various operations can also be defined between bases. Generally, operations related to DNA include DNA encoding and decoding, XOR, addition, subtraction, and complementation operations. An example of these operations is given in Fig. 1. The detailed description is as follows.

1) DNA Encoding and Decoding: A pixel whose value is between 0 and 255 can be encoded with 4 nucleic acid bases. First, the decimal number is transformed into an eight-digit binary number and then into four nucleic acid bases. Namely, a nucleic acid base can present one of four kinds: 00, 01, 10, or 11. There could be $4! = 24$ encoding or decoding rules, but due to the principle of complementary base pairing, that is, A is complementary with T and C with G, there are only 8 rules. The coding rules are listed in Table II. Using different DNA encoding and decoding rules does not produce the original decimal pixel value. We can use different coding rules only in the encryption process. Identical coding rules must be used in the decoding part. DNA coding is the fundamental operation of DNA-based ciphers. While there are 8 different encoding and decoding rules, there are only 8 outcomes rather than $8 \times 8 = 64$ outcomes. Interested readers can find more information in [59].

2) DNA XOR, Addition and Subtraction: These operations can be regarded as derivations of DNA encoding. There are 8 different XOR and addition rules corresponding to the 8 different encoding rules. Subtraction rules are derived from addition rules. One kind of these rules is illustrated in Table III. These calculations follow coding rule 1 in Table II, i.e., A(00), T(11), G(01), and C(10). Regardless of whether it is XOR, addition or subtraction, the calculation is essentially of a two-bit

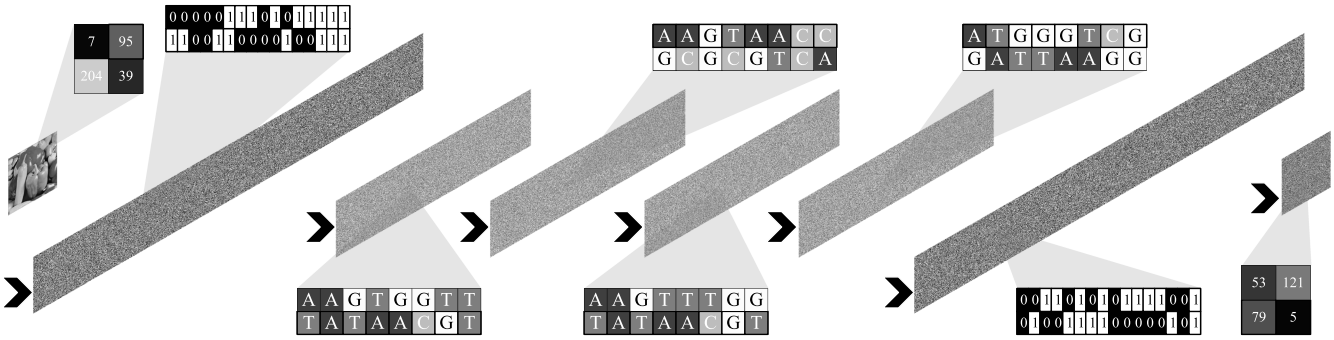

Fig. 1. Example of the DNA coding encryption process.

TABLE IV
DNA COMPLEMENT RULES

	1	2	3	4	5	6
A	T	T	C	G	C	G
T	G	C	G	A	A	C
G	C	A	A	C	T	T
C	A	G	T	T	G	A

binary 01 mask, but it is expressed in the form of nucleic acid bases. The following examples are given to illustrate this relationship. $A(00) + T(11) = T(11), T(11) + T(11) = C(10), A(00) \oplus T(11) = T(11), T(11) \oplus T(11) = A(00)$. However, the calculations in DNA format do not have continuous carry like them in bit level. For example, $0001 + 0011 = 0100$, $A(00)G(01) + A(00)T(11) = A(00)A(00) \neq G(01)A(00)$.

3) DNA Complementary: With this operation, one nucleic acid base can be converted into another. According to Watson and Crick's complementary rules, the DNA complementary operation must satisfy (1), where function $g(\cdot)$ returns the complementary base of the original base. To follow this law, only six rules are legal, as listed in Table IV.

$$\begin{cases} x \neq g(x) \neq g((x)) \neq g(((x))) \\ x = g((((x)))) \end{cases}. \tag{1}$$

C. Example of DNA Coding Encryption

To better explain the process of DNA coding encryption, a numerical example is employed for illustration. The corresponding encryption process is shown in Fig. 1.

The plaintext image is a matrix of size 2×2. It is randomly constructed as $\begin{pmatrix} 7 & 95 \\ 204 & 39 \end{pmatrix}$. First, it is converted to a binary matrix, expressed as $\begin{pmatrix} 00000111 & 01011111 \\ 11001100 & 00100111 \end{pmatrix}$. This matrix is then be converted into a DNA sequence by using DNA encoding rule 1. The result is indicated as $\begin{pmatrix} AAGT & GGTT \\ TATA & ACGT \end{pmatrix}$. The first encryption mask image is $\begin{pmatrix} AAAA & TTTT \\ CCCC & GGGG \end{pmatrix}$, which is used to perform DNA addition. The result is $\begin{pmatrix} AAGT & AACC \\ GCGC & GTCA \end{pmatrix}$. The second mask image is $\begin{pmatrix} AAAA & TTTT \\ CCCC & GGGG \end{pmatrix}$. It participates in the DNA XOR, and the result is $\begin{pmatrix} AAGT & TTGG \\ TATA & ACGT \end{pmatrix}$. The last mask image $\begin{pmatrix} 0000 & 1111 \\ 0101 & 1010 \end{pmatrix}$ is used for final DNA complementation. The result is $\begin{pmatrix} ATGG & GTCG \\ GATT & AAGG \end{pmatrix}$. The DNA sequence is decoded into a binary matrix with rule 1. The production is $\begin{pmatrix} 00110101 & 01111001 \\ 01001111 & 00000101 \end{pmatrix}$. Finally, the binary matrix is converted to decimal form to obtain the ciphertext as $\begin{pmatrix} 53 & 121 \\ 79 & 5 \end{pmatrix}$. So far, a series of encryption processes have been displayed. The plaintext image is $\begin{pmatrix} 7 & 95 \\ 204 & 39 \end{pmatrix}$, which is encrypted to $\begin{pmatrix} 53 & 121 \\ 79 & 5 \end{pmatrix}$.

III. MAIN RESULTS

A. S-Box

1) Introduction to S-Box: An S-box is a basic component of encryption algorithms. It always uses a fixed table to implement the substitution process. The S-box has been applied to well-known encryption algorithms such as the data encryption standard (DES) to obtain a good substitution effect. We discuss only the role of the S-box without involving its internal structure and specific implementation. Specifically, some consecutive steps in our encryption scheme are regarded as a whole with a similar function to the S-box. The equivalent key is the corresponding S-box matrix.

2) Properties of the S-Box: The S-box has many properties. We discuss only some of them that are helpful to cryptanalysis.

- Determinacy
  The mapping from the input to the output of the transformed S-box is determined. This property is given by
  $$f_S(a) = f_S(b) \text{ if and only if } a = b, \tag{2}$$
  where $f_S(\cdot)$ represents the function of the S-box transformation and a, b are the constants.
- Bijectivity
  The elements of the input set are mapped to the output according to a determined rule, which is a kind of surjective one-to-one mapping. The mapping is given by
  $$f_S : A \to B, \forall b \in B, \exists a \in A, b = f_S(a), \tag{3}$$
  where f_S is the S-box map and A, B are two sets.
- Reversibility
  The map must be reversible as part of the encryption process. The output through the S-box must be reversed

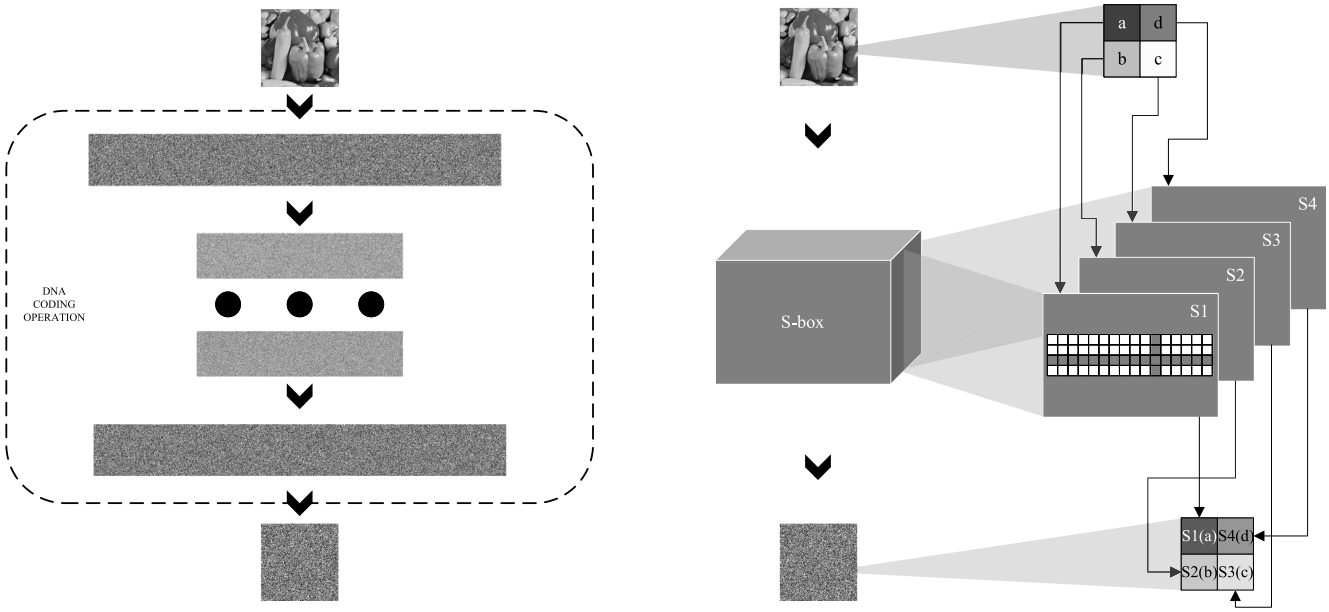

Fig. 2. Example using the S-box for simplifying DNA encryption.

through the S-box to become the original input. It is given by

$$f_S^{-1}(f_S(a)) = a, \tag{4}$$

where f_S^{-1} is the inverse map of f_S.

- Additivity
  The result of several S-box substitutions is the same as that of one S-box. This property is expressed as

$$f_{S_n}(\cdots f_{S_2}(f_{S_1}(a))) = f'_{S_1}(a), \tag{5}$$

where $f_{S_1}, f_{S_2}, \cdots$, and f_{S_n} may be different maps.

B. Simplifying DNA-Based Cryptosystems With S-Box

This process of converting to the S-box requires that the original encryption scheme meet the following conditions.

1) Plaintext-Independent Key: In other words, the key should be independent of the plaintext. If the secret key is related to the plaintext, such as the hash value of the plaintext or the Hamming distance involved in the encryption process, the encryption rules are uncertain. The converted S-box leads to uncertain rules.

2) No Diffusion at the DNA Level: In fact, the S-box equivalent transformation method is for each pixel. If the encryption can be transformed into the S-box process, there must be no diffusion phase at the DNA level. In other words, encrypted pixels do not affect each other.

3) No Encryption Element is Missing: Some cryptosystems add random elements during the encryption process. These encryption schemes sometimes recover only approximate but identifiable plaintext. Such a process cannot be regarded as an S-box process.

Some sequential operations can be regarded as the S-box in DNA-based encryption schemes. This transformation process is point-to-point. In other words, a series of DNA coding operations for each pixel in the image is the same as going through an S-box. This model is given by

$$f_{DNA\_DE}(f_{DNA}^n(f_{DNA\_EN}(a))) = f_S(a) \tag{6}$$

where f_{DNA_EN} and f_{DNA_DE} represent encoding and decoding pixels with DNA technology, respectively, and $f_{DNA} \in \{f_{DNA\_XOR}, f_{DNA\_A}, f_{DNA\_S}, f_{DNA\_C}\}$ is the general representation of DNA operation. $f_{DNA\_XOR}, f_{DNA\_A}, f_{DNA\_S}$, f_{DNA_C} denote DNA XOR, addition, subtraction and complementarity, respectively. The function $f_{DNA}^n(\cdot)$ represents that several DNA operations are performed. If the pixels in image A do not affect each other during the encryption process, then the conclusion is given by

$$f_{DNA\_DE}(f_{DNA}^n(f_{DNA\_EN}(A))) = f_S(A). \tag{7}$$

An example is shown in Fig. 2. Briefly, the DNA coding operation of plaintext into ciphertext is equivalent to passing through a large collection of S-boxes. From a local point of view, four pixels become corresponding pixels in the ciphertext by four specific S-boxes.

The S-box transformation method can simplify most DNA coding encryption processes. If a DNA-based encryption scheme only consists of DNA coding and its derive operations, it may be vulnerable to the chosen-plaintext attack. The S-box matrix is the corresponding equivalent key of this scheme. In reality, an encryption scheme often combines multiple technologies and uses complex chaotic systems to generate the encryption elements. The S-box transformation method is taken as the core. The specific attack process is used to attack successfully.

C. Cryptanalysis of the S-Box

The complete S-box transformation rules can be easily obtained by the chosen-plaintext attack. For each pixel, it is necessary to obtain a complete S-box. Specifically, a total of 256

Algorithm 1: Obtain the S-box Matrix

Input: The S-box mechanism cipher
Output: The S-box matrix
1: // step1: construct 256 chosen-plaintext images which are monochrome
2: **for** $i = 0$ to 255 **do**
3: $P_i = \text{zeros}(M \times N)$
4: **for** $h = 1$ to M **do**
5: **for** $w = 1$ to N **do**
6: $P_i(h, w) = i$
7: **end for**
8: **end for**
9: **end for**
10: // step2: obtain the corresponding plaintexts
11: **for** $i = 0$ to 255 **do**
12: $C_i = \text{encrypt}(P_i)$;
13: $T_{i+1} = \text{reshape}(C_i, 1, M \times N)$
14: **end for**
15: // step3: obtain the S-box matrix with the corresponding plaintexts
16: **for** $i = 1$ to 256 **do**
17: **for** $j = 1$ to $M \times N$ **do**
18: $S(i, j) = T_i(j)$
19: **end for**
20: **end for**

chosen plaintexts are constructed for a gray image cipher. The pixel values of each plaintext image are the same.

The detailed algorithm is shown in Algorithm 1. These S-box transformation rules obtained by our algorithm can be expressed in the following matrix form:

$$\begin{array}{c} 1 \\ 2 \\ \vdots \\ 256 \end{array} \begin{pmatrix} S_1(0) & S_2(0) & \cdots & S_{M\times N}(0) \\ S_1(1) & S_2(1) & \cdots & S_{M\times N}(1) \\ \vdots & \vdots & \ddots & \vdots \\ S_1(255) & S_2(255) & \cdots & S_{M\times N}(255) \end{pmatrix}. \quad (8)$$

The application of (8) is illustrated by giving an example. For the ith pixel with value x, the usage of the S-box matrix is given by

$$f_S(x) = S_i(x),$$
$$f_S^{-1}(f_S(x)) = f_S^{-1}(S_i(x)) = row - 1 = x, \quad (9)$$

in which row represents the number of rows where the pixel value x is located. In the S-box matrix, we use the number of rows to replace the range (0,255). Because the first number of rows is 1, not 0, we must subtract 1 from the final result to obtain the correct plaintext value.

Our attack's time and space complexity for the equivalent S-boxes are both $O(MN)$, where MN represents the size of the input images. A few image cryptanalysis approaches are very similar to ours, such as [58] and [29]. Although they broke different image ciphers, their core idea is the same as ours, i.e., the equivalent S-box transformation method. Therefore, the complexity of the attack for the S-boxes is equal to ours. Interested readers can refer to [60] and [61] to find more research on the cryptanalysis of the S-box in modern cryptography.

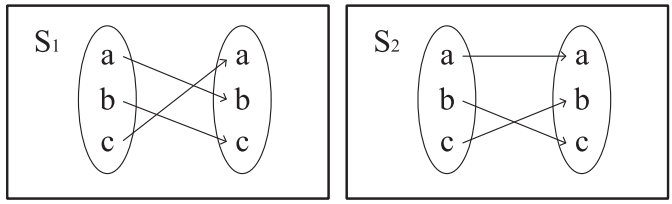

Fig. 3. Illustration of two S-box rules.

It is worth mentioning that it is not actually necessary to obtain all the S-box matrices. The pixels do not affect each other. As long as the uniformly distributed S-box matrices of pixels are obtained, then these correct pixels of an image allow us to identify the ciphertext image. This is because humans can actively make up incomplete places through association and other methods to obtain and to identify complete information.

D. A Simple Example

In this section, a simple example is given to illustrate the idea of the S-box transformation method. The pixel values include only a, b, and c. The plaintext image is (a, b). The corresponding ciphertext image is (b, c). Because both the plaintext and ciphertext have two pixels, we need two S-boxes to store information. They are denoted as S_1 and S_2.

Step 1: Three chosen plaintext images are constructed to obtain the S-box transformation rules. They are (a, a), (b, b), and (c, c). It is supposed that the corresponding ciphertexts are (b, a), (c, c), and (a, b).

Step 2: We have obtained the complete S-box transformation rules, which are given by $S_1(a) = b, S_1(b) = c, S_1(c) = a; S_2(a) = a, S_2(b) = c, S_2(c) = b$. Fig. 3 shows these rules intuitively.

Step 3: The S-box matrix is also used to present these rules. This matrix is given by

$$\begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{cc} S_1 & S_2 \end{array} \\ \begin{pmatrix} b & a \\ c & c \\ a & b \end{pmatrix} \end{array}.$$

Step 4: The ciphertext is (b, c). From the above matrix, we see that $S_1(a) = b$. Therefore, the first value of plaintext is a. Similarly, because $S_2(b) = c$, we can obtain the second value of plaintext b. Thus, we have obtained the complete plaintext image (a, b).

E. A Numerical Example

In this section, the S-box matrix is used to recover the plaintext of the example in Section II. The plaintext image and the ciphertext image are $\begin{pmatrix} 7 & 95 \\ 204 & 39 \end{pmatrix}$ and $\begin{pmatrix} 53 & 121 \\ 79 & 5 \end{pmatrix}$, respectively. The detailed chosen-plaintext attack process is as follows.

Algorithm 2: Recover Plaintext Image With S-box Matrix.

Input: The ciphertext C, the S-box matrix $SboxMatrix$
Output: The plaintext P
// step1: convert the 2D ciphertext matrix to 1D line by line and from left to right
$C = \text{reshape}(C, 1, M \times N)$
for $j = 1$ to $M \times N$ **do**
  for $i = 1$ to 256 **do**
    if $SboxMatrix(i, j) == C(i)$
    $P(j) = i - 1$
  end for
end for
// step2: convert the 1D plaintext to 2D matrix by line and from left to right
$P = \text{reshape}(P, M, N)$

Step 1: The S-box matrix is obtained by using Algorithm 1, which is given by

$$\begin{array}{c} 1 \\ \vdots \\ 8 \\ \vdots \\ 40 \\ \vdots \\ 96 \\ \vdots \\ 205 \\ \vdots \\ 256 \end{array} \begin{pmatrix} 51 & 204 & 195 & 60 \\ \vdots & \vdots & \vdots & \vdots \\ 53 & & & \\ \vdots & \vdots & \vdots & \vdots \\ & & & 5 \\ \vdots & \vdots & \vdots & \vdots \\ & 121 & & \\ \vdots & \vdots & \vdots & \vdots \\ & & 79 & \\ \vdots & \vdots & \vdots & \vdots \\ 221 & 153 & 125 & 105 \end{pmatrix}.$$

Step 2: With (9), the first plaintext pixel is recovered, i.e., $C(1) = 53 = S(8, 1)$, so $row = 8$, and the original pixel is $P(1) = row - 1 = 7$.

Step 3: The last pixel can also be recovered, i.e., $C(4) = 5 = S(40, 4)$, so $row = 40$, and the original pixel is $P(4) = row - 1 = 39$.

Step 4: The remaining plaintext pixels can be fully recovered by Algorithm 2.

IV. Applications for Cryptanalysis

In this section, several encryption schemes are analyzed to demonstrate the effectiveness of the S-box method. The S-box equivalent transformation is the core of the attack. The encryption schemes employed in this article are all cryptanalyzed with the proposed chosen-plaintext attack, but these schemes may also be vulnerable to other attack methods. We emphasize the universality of our S-box generalization concept.

A. Cryptanalysis of Amani's Cipher [52]

In this section, an encryption algorithm that uses DNA techniques is analyzed in detail. Additionally, a numerical example is given to illustrate the attack process.

1) Overview of Encryption Scheme: Amani's encryption scheme utilizes DNA techniques and hyperchaotic dynamics. The scheme originally encrypts color images, but the three components, i.e., R, G, and B, are performed separately. For simplicity, we discuss the case of only one channel. The encryption process can presumably include permutation and DNA diffusion. The permutation process uses the Arnold cat map, and diffusion uses DNA coding techniques. The detailed encryption process is as follows; interested readers are referred to Amani et al. [52] for more information.

Step 1: Pixels in the original image P are permutated to $S1$ by Arnold's chaotic cat map.

Step 2: The permutated image $S1$ is converted into a DNA sequence and recorded as $SD1$.

Step 3: Three mask images are obtained by iterating Chen's hyperchaotic system several times. Then, they are turned into DNA sequences. We discuss only the case of one component. The final mask image obtained in this step is denoted as $M1$.

Step 4: The mask image $M1$ is further transformed by

$$M2(i, j) = \text{mod}(\text{floor}(\text{abs}(M1(i, j) \times 10^{14})), 256),$$

where $M2$ represents the transformed mask image. Then, $M2$ were turned into DNA sequences by using DNA coding techniques. The final mask image is denoted as $MD2$.

Step 5: The DNA XOR is performed between $SD1$ and $MD2$ to yield a new DNA image, given by

$$SD2(i, j) = SD1(i, j) \oplus MD2(i, j),$$

where $SD2$ represents the new DNA image.

Step 6: The DNA image $SD2$ is transformed to the final encrypted image C by using one of the DNA decoding rules.

2) Cryptanalysis of the Encryption Scheme: As described in Section III, (7), DNA coding and its derived operations can be viewed as an S-box transformation process, i.e.,

$$f_{DNA\_DE}(f_{DNA\_XOR}(f_{DNA\_EN}(P))) = f_S(P).$$

Therefore, Chen's encryption scheme can be viewed as a combination of permutation and S-box transformation:

$$C = f_{\text{permute}}(f_S(P)),$$

where $f_{\text{permute}}(\cdot)$ denotes the function permutation. The original scheme and its equivalent process are shown in Fig. 4.

The permutation process changes only the position of the pixels. If the single-color image is chosen as the plaintext, the entire encryption process becomes an S-box-only cipher. Then, the complete S-box transformation rules can be obtained by using Algorithm 1. After obtaining the S-box transformation rules, the original encryption scheme becomes a permutation-only process. The permutation rules can be obtained according to the method in [28] and [62]. After obtaining all encrypted elements, the plaintext can be recovered successfully. The detailed chosen-plaintext attack process is as follows.

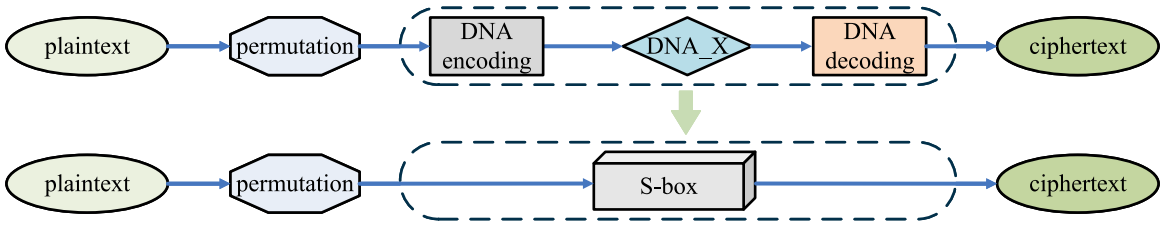

Fig. 4. Amani's encryption scheme and its equivalent process.

Step 1: Using Algorithm 1 to obtain S-box rules, a total of 256 chosen plaintexts are constructed, and the corresponding ciphertexts are obtained. These constructed plaintexts are as follows:

$$\begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \cdots \begin{pmatrix} 255 & \cdots & 255 & 255 \\ 255 & \cdots & 255 & 255 \\ \vdots & \ddots & \vdots & \vdots \\ 255 & \cdots & 255 & 255 \end{pmatrix}.$$

Step 2: After obtaining the S-box transformation rules, the original encryption scheme becomes a permutation-only cipher. There are many ways to crack this kind of encryption scheme. For simplicity, the following chosen plaintexts are constructed to obtain permutation rules. They are expressed as

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \cdots \begin{pmatrix} 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}.$$

There is only one value of pixels in each plaintext image that is different from others, and the permutated position of this pixel can be found quickly through the corresponding ciphertext. After enough repetitions, the complete permutation rules are obtained.

Step 3: After obtaining all encrypted elements, the plaintext can be recovered successfully.

First, the target ciphertext is transformed into a permutated image by using Algorithm 2 with the S-box matrix. Then, the permutation rules are used to recover the permutated image to a plaintext image.

3) A Numerical Example: To better illustrate the attack process, a numerical example of size 2×2 is given.

Without loss of generality, a random plaintext image is constructed as $\begin{pmatrix} 45 & 129 \\ 199 & 235 \end{pmatrix}$, which is encrypted to $\begin{pmatrix} 35 & 163 \\ 52 & 75 \end{pmatrix}$.

First, Algorithm 1 is used to obtain the S-box matrix, and the constructed plaintexts are

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \cdots \begin{pmatrix} 255 & 255 \\ 255 & 255 \end{pmatrix}.$$

The corresponding obtained ciphertexts are

$$\begin{pmatrix} 162 & 72 \\ 25 & 140 \end{pmatrix} \begin{pmatrix} 163 & 73 \\ 24 & 141 \end{pmatrix} \cdots \begin{pmatrix} 93 & 183 \\ 231 & 185 \end{pmatrix}.$$

The complete S-box matrix obtained is

$$\begin{array}{c} 1 \\ \vdots \\ 46 \\ \vdots \\ 130 \\ \vdots \\ 200 \\ \vdots \\ 236 \\ \vdots \\ 256 \end{array} \begin{pmatrix} 162 & 72 & 25 & 140 \\ \vdots & \vdots & \vdots & \vdots \\ & & 52 & \\ \vdots & \vdots & \vdots & \vdots \\ 35 & & & \\ \vdots & \vdots & \vdots & \vdots \\ & & & 75 \\ \vdots & \vdots & \vdots & \vdots \\ & 163 & & \\ \vdots & \vdots & \vdots & \vdots \\ 93 & 183 & 230 & 115 \end{pmatrix}.$$

Algorithm 2 is used with the S-box matrix to obtain the permutated image, which is $\begin{pmatrix} 129 & 235 \\ 45 & 199 \end{pmatrix}$.

Then, some simple plaintexts are constructed to obtain the permutation rules. These constructed plaintexts are expressed as $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$. The corresponding obtained ciphertexts are expressed as $\begin{pmatrix} 162 & 72 \\ 24 & 140 \end{pmatrix} \begin{pmatrix} 163 & 72 \\ 25 & 140 \end{pmatrix} \begin{pmatrix} 162 & 72 \\ 25 & 141 \end{pmatrix}$. Similarly, Algorithm 2 is used to obtain the permutated results of these chosen plaintexts. They are expressed as $\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. The permutation rule is obtained as $\begin{pmatrix} (2,1) & (1,1) \\ (2,2) & (1,2) \end{pmatrix}$, where each sequence number pair is the new position of the plaintext pixel in the ciphertext. At this point, all equivalent keys are obtained. The permutated result can be recovered to the original plaintext image through inverse permutation with the permutation matrix. Finally, the plaintext is obtained as $\begin{pmatrix} 45 & 129 \\ 199 & 235 \end{pmatrix}$.

4) Experimental Results: The experimental results of Amani's cipher are shown in Fig. 5. The numerical comparison demonstrates that the recovered image is exactly the same as the original plaintext image.

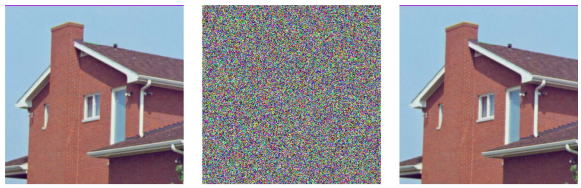

Fig. 5. Experimental results of Amani's cipher: (a) original plaintext image of size 256×256; (b) encrypted ciphertext image; (c) recovered image when using the chosen-plaintext attack.

B. Cryptanalysis of Parveiz's Cipher [53]

1) Overview of the Encryption Scheme: Parveiz's cipher combines DNA coding technology and three-dimensional chaotic maps. The 3D Arnold map and 3D logistic map are used to obtain encryption elements with sensitive parameters and initial values. This scheme encrypts color images. Because the three components do not affect each other, we discuss only the case of one component for simplicity.

The encryption scheme includes DNA diffusion, permutation, and random addition. It is briefly described as follows. If the readers want to know more details about the encryption scheme, please refer to Parveiz et al. [53].

Step 1: The original image P is converted to DNA sequence $SD1$.

Step 2: The DNA sequence $SD1$ is diffused to DNA sequence $SD2$ by DNA complementary operations.

Step 3: The mask image $M1$ is generated from the third dimension of the 3D Arnold map. Then, it is converted to the DNA sequence $MD1$.

Step 4: The DNA sequences $SD2$ and $MD1$ are subjected to DNA XOR to obtain the new sequences $SD3$.

Step 5: The DNA sequence $SD3$ is DNA decoding to its decimal form $S3$.

Step 6: Pixels in the original image $S3$ are permutated to $S4$.

Step 7: Another random sequence $M2$ is generated by the 3D logistic map.

Step 8: The mask image $M3$ is obtained with the sequence $M2$ by

$$M3(i,j) = \text{mod}(\text{round}(M2(i,j) \times 10^{15}), 256).$$

Step 9: The intermediate results $S3$ perform an addition operation with mask image $M3$ to obtain the final ciphertext image C.

2) Cryptanalysis of Encryption Scheme: The original encryption scheme contains DNA coding operations. Similarly, the process can be simplified with (7), i.e.,

$$f_{DNA\_DE}(f_{DNA\_X}(f_{DNA\_C})(f_{DNA\_EN}(P))) = f_S(P).$$

The whole process can be expressed as

$$C = f_{\text{add}}(f_{\text{permute}}(f_S(P))),$$

where $f_{\text{add}}(\cdot)$ denotes the process of addition with the random matrix. This operation is plaintext-independent, and each pixel acts individually, so it can be regarded as another S-box. Therefore, the encryption is

$$C = f_{S2}(f_{\text{permute}}(f_{S1}(P))).$$

The permutation operation changes only the location of pixels, and pixels do not affect each other during the equivalent S-box process. Therefore, the order of the two processes can be changed without affecting the result, i.e.,

$$C = f_{\text{permute}}(f_{S2}(f_{S1}(P))).$$

The S-box has the property of additivity. Combined with (5), the encryption process can be denoted as

$$C = f_{\text{permute}}(f_{S3}(P)).$$

Finally, the equivalent process, which includes permutation and the S-box operation of the original scheme, is obtained. The original and its equivalent processes are shown in Fig. 6.

After the equivalent transformation, the original encryption scheme becomes a combination of the S-box and permutation process. There is an obvious flaw in the encryption scheme; that is, a change of one point of the plaintext image affect only one point of the ciphertext. Based on this, some chosen plaintexts are constructed to obtain the equivalent key of the permutation process. Then, we obtain the equivalent S-box transformation rules. In this way, all the equivalent keys that are used to recover the plaintext image are obtained. The detailed steps are as follows.

Step 1: A total of $M \times N$ chosen plaintexts are constructed as

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$
$$\cdots \begin{pmatrix} 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Every two adjacent plaintexts differ by only one pixel. The ciphertexts corresponding to these two adjacent plaintexts also differ by only one pixel. The change in the position of this differentiated pixel is its permutation rule. By repeating this process, the complete permutation rules can be obtained.

Step 2: After obtaining the permutation rules, the original encryption scheme becomes an S-box cipher. Algorithm 1 is used to obtain the S-box rules. The 256 chosen-plaintexts are constructed, and the corresponding ciphertexts are used to obtain transformation rules. These constructed plaintexts are as follows.

$$\begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

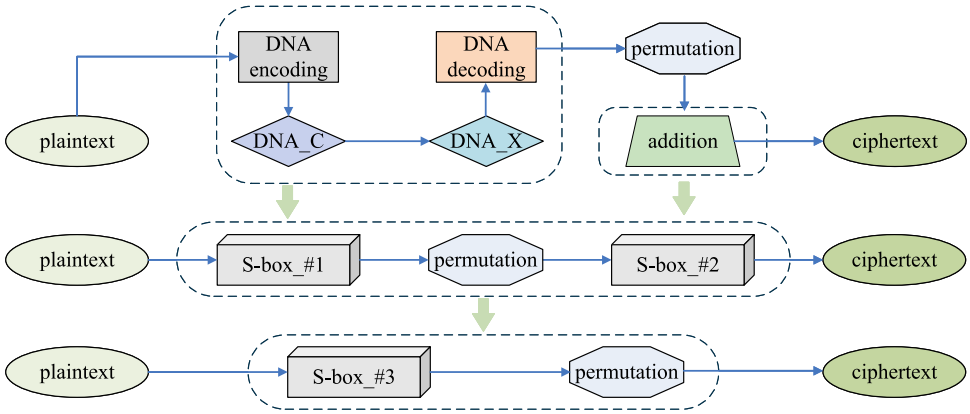

Fig. 6. Parveiz's encryption scheme and its equivalent process.

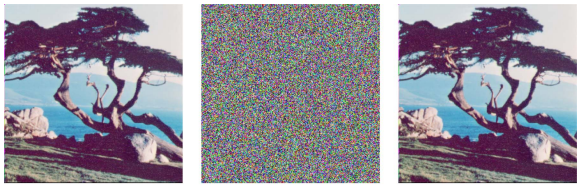

Fig. 7. Experimental results of Parveiz's cipher: (a) original plaintext image of size 256×256; (b) encrypted ciphertext image; (c) recovered image when using the chosen-plaintext attack.

$$\cdots \begin{pmatrix} 255 & \cdots & 255 & 255 \\ 255 & \cdots & 255 & 255 \\ \vdots & \ddots & \vdots & \vdots \\ 255 & \cdots & 255 & 255 \end{pmatrix}.$$

Step 3: After obtaining all encryption elements, the plaintext can be recovered successfully.
The target ciphertext is transformed into a permutated image by using Algorithm 2 with the S-box matrix. Then, the S-box rules are used to recover the permutated image to a plaintext image.

3) Experimental Results: The experimental results of Parveiz's cipher are shown in Fig. 7. The numerical comparison demonstrates that the recovered image is exactly the same as the original plaintext image.

C. Cryptanalysis of Azimi's Cipher [54]

1) Overview of Encryption Scheme: Azimi's cipher is a novel color image encryption based on DNA coding operations and pair-coupled chaotic maps. The original image is divided into three components (R, G, B), and then, complex transformations are performed.

The encryption scheme includes three components: mixing, DNA diffusion, and random addition. The complete encryption process is described below. More details can be found in Azimi et al. [54].

Step 1: The color image is separated into R, G, and B components. These three components are converted to DNA sequences, which are recorded with $SRD1$, $SGD1$, and $SBD1$.

Step 2: Three new components with $SRD2$, $SGD2$, and $SBD2$ are obtained with $SRD1$, $SGD1$, and $SBD1$ using DNA addition. The generation process is given by

$$\begin{cases} SRD2(i,j) = SRD1(i,j) + SGD1(i,j) \\ SGD2(i,j) = SGD1(i,j) + SBD1(i,j) \\ SBD2(i,j) = SGD2(i,j) + SBD1(i,j) \end{cases}.$$

Step 3: Three pairs of sequences $MR1$, $MG1$, and $MB1$ are generated by using pair-coupled chaotic maps. These sequences $MR1$, $MG1$, and $MB1$ are converted into binary sequences $MRB2$, $MGB2$, and $MBB2$ according to the threshold function, which is given by

$$g(x) = \begin{cases} 0, X(i) > Y(i) \\ 1, X(i) < Y(i) \end{cases}.$$

Step 4: Three new DNA images $SRD3$, $SGD3$, and $SBD3$ are obtained with $SRD2$, $SGD2$, $SBD2$ and $MRB2$, $MGB2$, $MBB2$ through DNA complementary operations.

Step 5: Three binary images $SRB3$, $SGB3$, and $SBB3$ are obtained with $SRD3$, $SGD3$, $SBD3$ by using the DNA decoding operation.

Step 6: Three mask binary images $MRB3$, $MGB3$, and $MBB3$ are generated with pair-coupled chaotic maps and the threshold function similarly.

Step 7: The images $SRB3$, $SGB3$, and $SBB3$ are XOR with images $MRB3$, $MGB3$, and $MBB3$ to obtain the final components $SRB4$, $SGB4$, and $SBB4$.

Step 8: The binary components $SRB4$, $SGB4$, and $SBB4$ are converted to decimal form $SR4$, $SG4$, and $SB4$. The final ciphertext image C is generated by combining the images $SR4$, $SG4$, and $SB4$.

2) Cryptanalysis of the Encryption Scheme: The original encryption scheme can be regarded as two processes. First, the plaintext image is separated into R, G, and B components. These

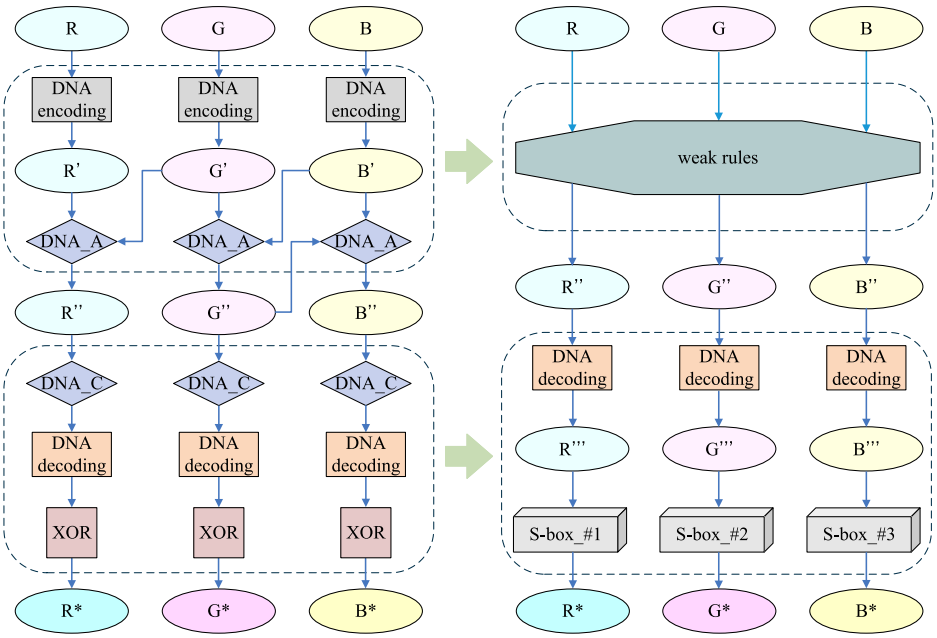

Fig. 8. Azimi's encryption scheme and its equivalent process.

components perform complex operations into new components, which are denoted as $SRD2$, $SGD2$, and $SBD2$ (denoted as R'', G'', and B'' in Fig. 8), respectively. Second, the three intermediate results $SRD2$, $SGD2$, and $SBD2$ perform DNA complementary and XOR operations to obtain the components of the final ciphertext. The first process seems complex, but only the DNA encoding rules are unknown. There are 8 encoding rules in total. For a small key space, a brute force attack can be used. If $SRD2$, $SGD2$, and $SBD2$ are known, the R, G, and B components of the original image are also known. For the second process, if DNA decoding operations are added, it can be simplified to an S-box process. Combined with (7), for one component, our transformation process can be expressed as

$$\begin{aligned} R^* &= f_{XOR}(f_{DNA\_DE'}(f_{DNA\_C}(R''))) \\ &= f_{XOR}(f_{DNA\_DE'}(f_{DNA\_C}(f_{DNA\_DE}(f_{DNA\_EN})(R'')))) \\ &= f_{XOR}(f_{DNA\_DE'}(f_{DNA\_C}(f_{DNA\_DE}(R''')))). \end{aligned}$$

XOR between elements is independent of plaintext, and the S-box is additive, so the equation can be transformed as

$$R^* = f_{S'}(f_{S''}(R''')) = f_S(R''').$$

The original and its equivalent processes are shown in Fig. 8. Some details that do not affect the conclusion are ignored in the above derivation.

Because the first process has weak rules that can be cracked by brute-force attack, the intermediate components R''', G''', B''' are constructed. These components are used to infer all possible plaintexts. After obtaining all possible S-boxes and corresponding component transformation rules, reasonable rules are chosen to recover the plaintext. The detailed steps are as follows.

Step 1: First, R''', G''', and B''' components are constructed. They are the same. They can be expressed as

$$\begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

$$\cdots \begin{pmatrix} 255 & \cdots & 255 & 255 \\ 255 & \cdots & 255 & 255 \\ \vdots & \ddots & \vdots & \vdots \\ 255 & \cdots & 255 & 255 \end{pmatrix}.$$

Step 2: All possible DNA encoding rules are used to infer the R, G, and B components of the plaintext, which is the inverse of the first process of encryption.

Step 3: All possible components are combined into color images as the constructed chosen-plaintexts, and the corresponding ciphertexts are obtained. These ciphertext images can be separated into R^*, G^*, and B^* components.

Step 4: By using Algorithm 1, all possible S-box rules between R''', G''', B''' and R^*, G^*, B^* are obtained.

Step 5: The ciphertext image is decrypted into multiple plaintexts by using all possible S-box transformation rules and corresponding DNA encoding rules. Each plaintext is recovered to R''', G''', and B''' components by using the corresponding S-box and Algorithm 2, and then recovered to plaintext through the corresponding DNA encoding rules.

Step 6: The plaintext is the most reasonable of all the recovered plaintexts.

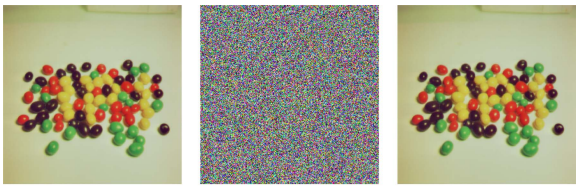

Fig. 9. Experimental results of Azimi's cipher: (a) original plaintext image of size 256×256; (b) encrypted ciphertext image; (c) recovered image using the chosen-plaintext attack.

3) Experimental results: The experimental results of Azimi's cipher are shown in Fig. 9. The numerical comparison demonstrates that the recovered image is exactly the same as the original plaintext image.

D. Cryptanalysis of Other Cryptosystems

1) Mondal's Cipher [55]: Mondal's encryption scheme is a typical permutation-diffusion architecture. In this scheme, the plaintext image is permutated at first by using a sequence of pseudorandom numbers and then diffused by DNA coding operations. The random sequences are generated by a pseudo random number generator (PRNG) based on the cross-coupled chaotic logistic map. To further improve the security, the results of DNA diffusion are XORed with each other to obtain the final ciphertext. If the reader wants to know more details about the encryption scheme, please refer to Mondal et al. [55].

The encryption process contains permutation and DNA diffusion. DNA diffusion can be regarded as an S-box transformation. Although the diffusion results are XORed to obtain the final ciphertext, there is no secret key involved in this process. The result before diffusion can be directly derived. The cryptanalysis of this cipher is similar to Amani's and is not repeated here.

2) Liu's Cipher [56]: Liu's encryption scheme combines DNA coding operations and several chaotic maps. It also includes permutation and diffusion, two main processes. The Arnold map is used to perform the permutation, and the mask images are generated with Lorenz and Rossler maps. The original image is separated into R, G, and B components to perform different processes. Both the number of permutation iterations and DNA coding rules are determined by the plaintext. If the reader wants to know more details about the encryption scheme, please refer to Liu et al. [56].

The R, G, and B components do not affect each other during the encryption process. Although they perform different DNA coding operations, each of them can be regarded as the process of permutation and S-box transformation. Both the number of permutation iterations and DNA coding rules are limited and small; therefore, brute-force attack methods can be used to obtain all possible encryption elements. The plaintext is the most reasonable of the recovered plaintexts.

V. Discussions

This section provides more discussion, including DNA-based schemes and other types of image ciphers. They are discussed from the perspective of both cryptanalysis and image encryption. Many suggestions are proposed for the future design of image encryption schemes. In addition, this section introduces the limitations of this article.

Although many DNA-based image schemes are broken with the chosen-plaintext attack, this does not mean that DNA coding techniques are inappropriate for image ciphers. Once a good diffusion process between pixels is added in the encryption process, it effectively resists the chosen-plaintext attack based on the equivalent S-box transformation. The permutation at the DNA level has better security performance than at the pixel level. In addition, DNA coding techniques have many other promising properties, including high speed, parallelism computation, and minimal storage. Therefore, the DNA coding operations are still applicable to the image cipher. Many DNA-based and other types of image ciphers are plaintext dependent. In other words, the information related to plaintext is used as part of the secret key. Although plaintext-dependent ciphers are highly resistant to plaintext attacks, we do not recommend them. This is because these schemes do not follow Shannon's theory [63]. They are difficult to implement in reality.

The permutation-diffusion architecture is popular in image encryption schemes. The permutation process relocates the pixels with unchanged values. The diffusion modifies the values and obtains the avalanche performance. This architecture may be repeated over many rounds to achieve a higher security level. For an image encryption scheme with a single permutation-diffusion round, if the values of the plaintext image are the same, the permutation is eliminated. If the diffusion contains only simple operations such as module addition and XOR, this encryption process may be vulnerable to plaintext attacks. Some schemes with many rounds of permutation-diffusion architecture seem to be more secure. However, the mapping from the differentials of the ciphertexts to those of the plaintexts is linear. These schemes may be vulnerable to chosen-ciphertext attacks. Interested readers can refer to Chen et al. [50] for more information. In fact, most encryption schemes with such architectures have not been broken. This is due to their complex diffusion process. Therefore, it is important to choose complex encryption operations in the design of ciphers.

Most image encryption schemes based on DNA coding technology can be simplified by using the S-box equivalent transformation method. Many other types of image encryption are vulnerable to various attacks. The following suggestions are given to help design future image encryption schemes.

- A secure image encryption scheme must have good performance in statistical tests. The performance indicators include information entropy, NPCR (number of pixel change rate), UACI (unified averaged changed intensity), and so on.
- More complex diffusion processes are needed, especially the interactions between elements that can cause avalanche effects. Designing a secure diffusion is not easy. An unreasonable process leaves many security risks [64].
- Multiple rounds of encryption are recommended. Multiple rounds of encryption help ciphers achieve a high level of security. Different keys can be used in different rounds, but

designers should make a compromise between the encryption rounds with implementation efficiency [65].
- It is highly suggested to add nonlinear substitution to the encryption schemes. After this nonlinear substitution encryption, the relationship between plaintext and ciphertext becomes more challenging to analyze.

The cryptanalysis of this article analyses only five DNA-based encryption schemes as examples to illustrate the equivalent S-box transformation methods. Researchers may find more schemes that are also vulnerable to our attack. There may be more similar flawed encryption schemes in the future. In addition to DNA coding, some other encryption operations can also be regarded as S-box processes. However, it seems they do not have similar characteristics. These processes are not summarized in this article. It is hoped that more encryption operations that can be regarded as S-boxes may be found in the future. More encryption schemes with shortcomings can be found and improved.

VI. Conclusion

In this article, the security of a family of DNA-based image encryption schemes has been evaluated. Encoding with the DNA sequence followed by encryption with a mask can be equivalently transformed into an S-box process. The S-box matrix is the equivalent key and can be easily obtained. These schemes have weaknesses and cannot increase security by repeating DNA coding operations. A total of five image ciphers were found to be vulnerable to the chosen-plaintext attack. Both theoretical analysis and experimental results confirm this. Some suggestions are made to help design future image encryption schemes. In addition, there exist other encryption operations that can be regarded as S-box processes. Future research should focus on attacking similar flawed schemes, while keeping the S-box matrix as the equivalent key.

Declaration of Conflicting Interest: The authors declare that they have no conflicts of interest.

References

[1] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 1055–1067, Nov./Dec. 2018.

[2] Z. Lv, Y. Han, A. K. Singh, G. Manogaran, and H. Lv, "Trustworthiness in industrial IoT systems based on artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1496–1504, Feb. 2021.

[3] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 06, pp. 1259–1284, 1998.

[4] Z. Hua, Z. Zhu, Y. Chen, and Y. Li, "Color image encryption using orthogonal latin squares and a new 2D chaotic system," *Nonlinear Dyn.*, vol. 104, no. 4, pp. 4505–4522, 2021.

[5] M. Boussif, N. Aloui, and A. Cherif, "Images encryption algorithm based on the quaternion multiplication and the XOR operation," *Multimedia Tools Appl.*, vol. 78, no. 24, pp. 35493–35510, 2019.

[6] Z. Gan, X. Chai, M. Zhang, and Y. Lu, "A double color image encryption scheme based on three-dimensional Brownian motion," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 27919–27953, 2018.

[7] P. K. Naskar, S. Bhattacharyya, D. Nandy, and A. Chaudhuri, "A robust image encryption scheme using chaotic tent map and cellular automata," *Nonlinear Dyn.*, vol. 100, no. 3, pp. 2877–2898, 2020.

[8] J. Chen, S. Sun, L.-B. Zhang, B. Yang, and W. Wang, "Compressed sensing framework for heart sound acquisition in internet of medical things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2000–2009, Mar. 2022.

[9] X. Chai, J. Fu, Z. Gan, Y. Lu, and Y. Zhang, "An image encryption scheme based on multi-objective optimization and block compressed sensing," *Nonlinear Dyn.*, vol. 108, no. 3, pp. 2671–2704, 2022.

[10] X. Chai, Y. Wang, X. Chen, Z. Gan, and Y. Zhang, "TPE-GAN: Thumbnail preserving encryption based on GAN with key," *IEEE Signal Process. Lett.*, vol. 29, pp. 972–976, 2022.

[11] X. Chai, Y. Wang, Z. Gan, X. Chen, and Y. Zhang, "Preserving privacy while revealing thumbnail for content-based encrypted image retrieval in the cloud," *Inf. Sci.*, vol. 604, pp. 115–141, 2022.

[12] S. Kumar et al., "Deep learning framework for recognition of cattle using muzzle point image pattern," *Measurement*, vol. 116, pp. 1–17, 2018.

[13] A. K. Singh, B. Kumar, S. K. Singh, S. Ghrera, and A. Mohan, "Multiple watermarking technique for securing online social network contents using back propagation neural network," *Future Gener. Comput. Syst.*, vol. 86, pp. 926–939, 2018.

[14] S. Thakur, A. K. Singh, S. P. Ghrera, and M. Elhoseny, "Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications," *Multimedia Tools Appl.*, vol. 78, pp. 3457–3470, 2019.

[15] G. Xie, J. Ren, S. Marshall, H. Zhao, and R. Li, "A novel gradient-guided post-processing method for adaptive image steganography," *Signal Process.*, vol. 203, 2023, Art. no. 108813.

[16] A. K. Singh, "Improved hybrid algorithm for robust and imperceptible multiple watermarking using digital images," *Multimedia Tools Appl.*, vol. 76, pp. 8881–8900, 2017.

[17] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, 1994.

[18] K. Jithin and S. Sankar, "Color image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set," *J. Inf. Secur. Appl.*, vol. 50, 2020, Art. no. 102428.

[19] M. B. Farah, R. Guesmi, A. Kachouri, and M. Samet, "A novel chaos based optical image encryption using fractional fourier transform and DNA sequence operation," *Opt. Laser Technol.*, vol. 121, 2020, Art. no. 105777.

[20] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, and B.-Q. Yang, "Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption," *Signal Process.*, vol. 142, pp. 340–353, 2018.

[21] Q. Wang, X. Zhang, and X. Zhao, "Image encryption algorithm based on improved Zigzag transformation and quaternary DNA coding," *J. Inf. Secur. Appl.*, vol. 70, 2022, Art. no. 103340.

[22] S. A. Elsaid, E. R. Alotaibi, and S. Alsaleh, "A robust hybrid cryptosystem based on DNA and hyperchaotic for images encryption," *Multimedia Tools Appl.*, vol. 82, pp. 1995–2019, 2022.

[23] L. Paul et al., "A novel colour image encryption scheme using dynamic DNA coding, chaotic maps, and SHA-2," *Multimedia Tools Appl.*, vol. 81, pp. 37873–37894, 2022.

[24] X. Zhang and R. Ye, "A novel RGB image encryption algorithm based on DNA sequences and chaos," *Multimedia Tools Appl.*, vol. 80, no. 6, pp. 8809–8833, 2021.

[25] Y. Wu, L. Zhang, S. Berretti, and S. Wan, "Medical image encryption by content-aware DNA computing for secure healthcare," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 2089–2098, Feb. 2023.

[26] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL, USA: CRC, 2020.

[27] S. Budiansky, *Battle of Wits: The Complete Story of Codebreaking in World War II*. New York, NY, USA: Simon Schuster, 2000.

[28] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 2, pp. 235–246, Feb. 2016.

[29] J. Chen, L. Chen, and Y. Zhou, "Cryptanalysis of a DNA-based image encryption scheme," *Inf. Sci.*, vol. 520, pp. 130–141, 2020.

[30] R. Chen, L. Liu, and Z. Zhang, "Cryptanalysis on a permutation–rewriting–diffusion (PRD) structure image encryption scheme," *Multimedia Tools Appl.*, vol. 82, pp. 4289–4317, 2022.

[31] G. V. Bard, "A challenging but feasible blockwise-adaptive chosen-plaintext attack on SSL," *Cryptol. ePrint Arch.*, 2006. [Online]. Available: https://eprint.iacr.org/2006/136

[32] M. Kumar, A. Iqbal, and P. Kumar, "A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie–Hellman cryptography," *Signal Process.*, vol. 125, pp. 187–202, 2016.

[33] A. Akhavan, A. Samsudin, and A. Akhshani, "Cryptanalysis of an image encryption algorithm based on DNA encoding," *Opt. Laser Technol.*, vol. 95, pp. 94–99, 2017.
[34] C. Song and Y. Qiao, "A novel image encryption algorithm based on DNA encoding and spatiotemporal chaos," *Entropy*, vol. 17, no. 10, pp. 6954–6968, 2015.
[35] H. Wen, S. Yu, and J. Lü, "Breaking an image encryption algorithm based on DNA encoding and spatiotemporal chaos," *Entropy*, vol. 21, no. 3, 2019, Art. no. 246.
[36] P. Zhen, G. Zhao, L. Min, and X. Jin, "Chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools Appl.*, vol. 75, no. 11, pp. 6303–6319, 2016.
[37] X. Su, W. Li, and H. Hu, "Cryptanalysis of a chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools Appl.*, vol. 76, no. 12, pp. 14021–14033, 2017.
[38] K. Panwar, R. K. Purwar, and A. Jain, "Cryptanalysis and improvement of a color image encryption scheme based on DNA sequences and multiple 1D chaotic maps," *Int. J. Bifurcation Chaos*, vol. 29, no. 08, 2019, Art. no. 1950103.
[39] X. Wu, H. Kan, and J. Kurths, "A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps," *Appl. Soft Comput.*, vol. 37, pp. 24–39, 2015.
[40] J. Wu, X. Liao, and B. Yang, "Image encryption using 2D Hénon-Sine map and DNA approach," *Signal Process.*, vol. 153, pp. 11–23, 2018.
[41] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 9, pp. 2137–2150, Sep. 2018.
[42] E. Solak, C. Cokal, O. T. Yildiz, and T. Biyikoğlu, "Cryptanalysis of Fridrich's chaotic image encryption," *Int. J. Bifurcation Chaos*, vol. 20, no. 05, pp. 1405–1413, 2010.
[43] G. Ye, C. Pan, X. Huang, and Q. Mei, "An efficient pixel-level chaotic image encryption algorithm," *Nonlinear Dyn.*, vol. 94, pp. 745–756, 2018.
[44] N. Munir, M. Khan, A. A. K. H. Ismail, and I. Hussain, "Cryptanalysis and improvement of novel image encryption technique using hybrid method of discrete dynamical chaotic maps and Brownian motion," *Multimedia Tools Appl.*, vol. 81, no. 5, pp. 6571–6584, 2022.
[45] M. Khan, F. Masood, A. Alghafis, M. Amin, and S. I. B. Naqvi, "A novel image encryption technique using hybrid method of discrete dynamical chaotic maps and Brownian motion," *PLoS One*, vol. 14, no. 12, 2019, Art. no. e0225031.
[46] A. Arora and R. K. Sharma, "Cryptanalysis and enhancement of image encryption scheme based on word-oriented feed back shift register," *Multimedia Tools Appl.*, vol. 81, no. 12, pp. 16679–16705, 2022.
[47] S. Deb, B. Biswas, and B. Bhuyan, "Secure image encryption scheme using high efficiency word-oriented feedback shift register over finite field," *Multimedia Tools Appl.*, vol. 78, pp. 34901–34925, 2019.
[48] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, pp. 172–182, 2014.
[49] S. Dhall, S. K. Pal, and K. Sharma, "Cryptanalysis of image encryption scheme based on a new 1D chaotic system," *Signal Process.*, vol. 146, pp. 22–32, 2018.
[50] J. Chen, L. Chen, and Y. Zhou, "Universal chosen-ciphertext attack for a family of image encryption schemes," *IEEE Trans. Multimedia*, vol. 23, pp. 2372–2385, 2021.
[51] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognit. Lett.*, vol. 31, no. 5, pp. 347–354, 2010.
[52] H. R. Amani and M. Yaghoobi, "A new approach in adaptive encryption algorithm for color images based on DNA sequence operation and hyper-chaotic system," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 21537–21556, 2019.
[53] P. N. Lone et al., "Image encryption using DNA coding and three-dimensional chaotic systems," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 5669–5693, 2022.
[54] Z. Azimi and S. Ahadpour, "Color image encryption based on DNA encoding and pair coupled chaotic maps," *Multimedia Tools Appl.*, vol. 79, no. 3, pp. 1727–1744, 2020.
[55] B. Mondal and T. Mandal, "A light weight secure image encryption scheme based on chaos & DNA computing," *J. King Saud Univ.- Comput. Inf. Sci.*, vol. 29, no. 4, pp. 499–504, 2017.
[56] Q. Liu and L. Liu, "Color image encryption algorithm based on DNA coding and double chaos system," *IEEE Access*, vol. 8, pp. 83596–83610, 2020.
[57] Y. Zhang and D. Xiao, "Cryptanalysis of S-box-only chaotic image ciphers against chosen plaintext attack," *Nonlinear Dyn.*, vol. 72, no. 4, pp. 751–756, 2013.
[58] W. Feng and Y.-G. He, "Cryptanalysis and improvement of the hyper-chaotic image encryption scheme based on DNA encoding and scrambling," *IEEE Photon. J.*, vol. 10, no. 6, Dec. 2018, Art. no. 7909215.
[59] Y. Zhang, D. Xiao, W. Wen, and K.-W. Wong, "On the security of symmetric ciphers based on DNA coding," *Inf. Sci.*, vol. 289, pp. 254–261, 2014.
[60] H. M. Heys, "A tutorial on linear and differential cryptanalysis," *Cryptologia*, vol. 26, no. 3, pp. 189–221, 2002.
[61] M.-J. O. Saarinen, "Cryptographic analysis of all 4 × 4-bit S-boxes," in *Proc. 18th Int. Workshop Sel. Areas Cryptogr.*, 2011, pp. 118–133.
[62] L. Y. Zhang et al., "Improved known-plaintext attack to permutation-only multimedia ciphers," *Inf. Sci.*, vol. 430, pp. 228–239, 2018.
[63] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.
[64] J. Chen, L. Y. Zhang, and Y. Zhou, "Re-evaluation of the security of a family of image diffusion mechanisms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4747–4758, Dec. 2021.
[65] Z. Hua, S. Yi, and Y. Zhou, "Medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Process.*, vol. 144, pp. 134–144, 2018.

Chengrui Zhang received the bachelor's degree in software engineering from Dalian Jiaotong University, Dalian, China, in 2020. He is currently working toward the Ph.D. degree in software engineering with Software College, Northeastern University, Shenyang, China. His research interests include image encryption, cryptanalysis, and forgery detection.

Junxin Chen (Senior Member, IEEE) received the B.Sc. degree in communications engineering and the M.Sc. and Ph.D. degrees in communications and information system from Northeastern University, Shenyang, China, in 2007, 2009, and 2016 respectively. He is currently a Professor with the School of Software, Dalian University of Technology, Dalian, China. From 2019 to 2020, he was a Postdoc Research Fellow with the Department of Computer and Information Science, under the UM Macau Talent Programme (Class A), University of Macau, Macau, China. He was an Assistant Professor and Associate Professor with the College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China. He has authored or coauthored more than 80 scientific papers in international peer-reviewed journals and conferences, such as, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE INTERNET OF THINGS JOURNAL. He has an H-index of 29, and a total of 2400 citations. His research interests include internet of medical things, artificial intelligence, and information security. He is the topic Editor of *Electronics*, the Leading Guest Editor of IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, Guest Editor of *Signal Processing: Image Communication*. He is a regular Reviewer of extensive top-tier field journals. He was the recipient of more than ten awards from Mainland China and Macau. He has received about 1.5 million funding from government and industry as PI.

Dongming Chen (Member, IEEE) received the Ph.D. degree in computer system architecture from Northeastern University, Shenyang, China, in 2006. He is currently a Professor with Software College, Northeastern University, Shenyang, China. He is a Member of IEEE Computer Society, Senior member of China Computer Federation, and Senior member of China Institute of Communications. His research interests include deep reinforcement learning, information security, and Big Data analysis.

Wei Wang (Member, IEEE) received the B.Sc. degree in electronic information science and technology from Shenyang University, Shenyang, China, in 2012, and the Ph.D. degree in software engineering from the Dalian University of Technology, Dalian, China, in 2018. He is currently a Professor with the Department of Engineering, Shezhen MSU-BIT University, Shenzhen, China, and also with the School of Medical Technology, Beijing Institute of Technology, Beijing, China. He has authored or coauthored more than 100 scientific papers in international journals and conferences. His research interests include artificial internet of things, computational social science, and affective computing.

Yushu Zhang (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from the College of Computer Science, Chongqing University, Chongqing, China, in December 2014. He held various research positions with Southwest University, Chongqing, City University of Hong Kong, Hong Kong, University of Macau, Macau, China, and Deakin University, Melbourne, VIC, Australia. He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

He has authored or coauthored more than 100 refereed journal articles and conference papers in his research areas, which include multimedia security, blockchain, and artificial intelligence. He is currently an Associate Editor for *Information Sciences and Signal Processing*.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. Dr. Zhou is currently a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security. Dr. Zhou is a Fellow of SPIE (the Society of Photo-Optical Instrumentation Engineers) and was recognized as one of Highly Cited Researchers in 2020 and 2021. He is an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.