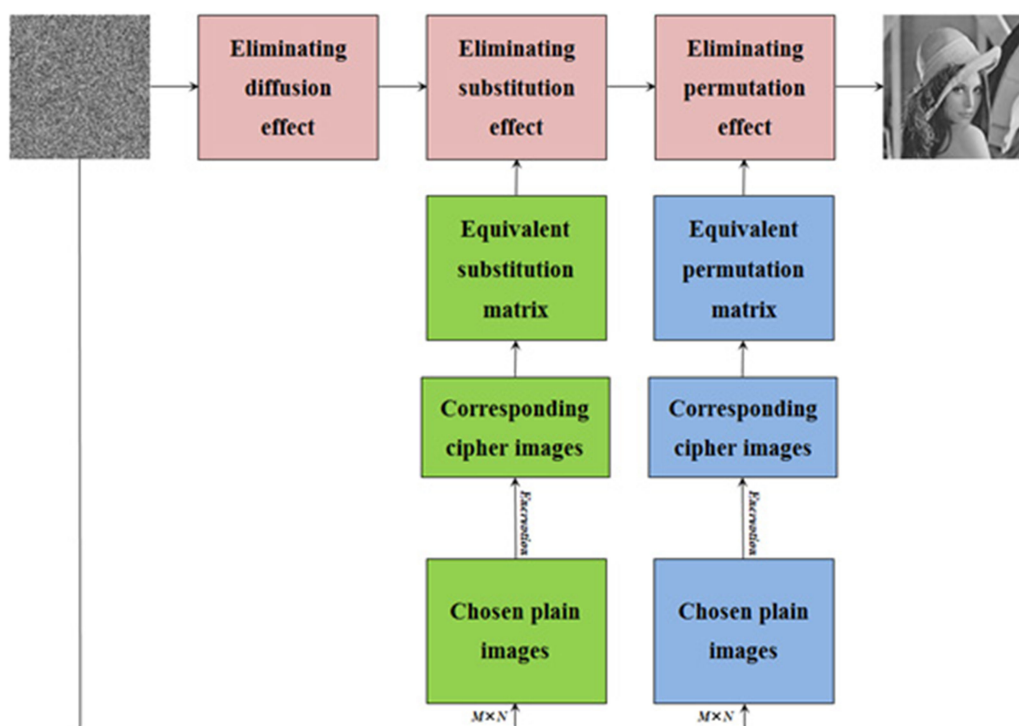


# Cryptanalysis and Improvement of the Hyper-Chaotic Image Encryption Scheme Based on DNA Encoding and Scrambling

Volume 10, Number 6, December 2018

Wei Feng  
Yi-Gang He



# Cryptanalysis and Improvement of the Hyper-Chaotic Image Encryption Scheme Based on DNA Encoding and Scrambling

Wei Feng <sup>1,2</sup> and Yi-Gang He <sup>1</sup>

<sup>1</sup>School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China

<sup>2</sup>School of Mathematics and Computer Science, Panzhihua University, Panzhihua 617000, China

DOI:10.1109/JPHOT.2018.2880590

1943-0655 © 2018 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission.

See [http://www.ieee.org/publications\\_standards/publications/rights/index.html](http://www.ieee.org/publications_standards/publications/rights/index.html) for more information.

Manuscript received September 8, 2018; revised October 30, 2018; accepted November 5, 2018. Date of publication November 9, 2018; date of current version November 29, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 51577046, in part by the State Key Program of National Natural Science Foundation of China under Grant 51637004, in part by the national key research and development plan “important scientific instruments and equipment development” under Grant 2016YFF0102200, and in part by the Anhui Provincial Science and Technology Foundation of China under Grant 1301022036. Corresponding author: Yi-Gang He (e-mail: yghe@hfut.edu.cn).

**Abstract:** In recent years, hybrid chaotic image encryption schemes combined with the DNA encryption technology or other technologies have become a research focus of many researchers. However, some researchers neglect security analyses of proposed schemes under the chosen plaintext attack, therefore leaving security vulnerabilities. Considering this situation, we carefully investigate a recently reported hyper-chaotic image encryption scheme using the DNA encryption technology. In this paper, we first point out some issues identified in the reported scheme and make some necessary improvements, and then cryptanalyze it and propose a chosen plaintext attack algorithm. With the proposed attack algorithm, theoretical analyses and test results show that it can completely recover plain images without knowing any secure key related information. Finally, some suggestions for improving the security and practicability of the reported hyper-chaotic image encryption scheme are presented.

**Index Terms:** Cryptanalysis, chaotic image encryption, chosen plaintext attack, DNA encoding.

## 1. Introduction

In today's information age, how to transmit image data more safely and efficiently has attracted more and more researchers' attentions [1]–[7]. As a result, a variety of image encryption technologies have emerged, such as optical image encryption technology [8], DNA image encryption technology [9], quantum image encryption technology [10] and chaotic image encryption technology [11], [12]. Each technology has its own advantages and disadvantages. For example, optical image encryption technology has the advantages of high parallelism and high speed, but also has the disadvantages of requiring special equipment, high cost and difficult implementation. Therefore, hybrid image encryption schemes combining the advantages of various encryption technologies have become a research trend, more and more hybrid encryption schemes have been proposed. One can enumerate some latest developments. Wang *et al.* present an image encryption scheme by

using a piecewise linear chaotic map to generate key images and using DNA rules to encrypt plain images [13]; Chen *et al.* present a solution for secure and efficient image encryption with the help of the self-adaptive permutation-diffusion and the DNA random encoding [14]; Zhang *et al.* use the Feistel network and the dynamic DNA encoding technology to propose a chaotic image encryption method using the permutation-diffusion-scrambling structure [15]; In [16], a novel image encryption scheme whose image pixels are diffused by the DNA approach and permuted by 2D-HSM, is proposed to protect image content; In [17], an original image is firstly diffused through Exclusive-OR with a key image transformed from constructed spatiotemporal chaotic sequences. And then DNA deletion and DNA insertion pseudo-operations are used to confuse the DNA encoded image.

However, the security of image encryption schemes is also worth attentions [18]–[20]. Because of the design defects, many hybrid image encryption schemes do not have the claimed security. In [21], Akhavan *et al.* investigate the security of a DNA based image encryption algorithm, and successfully recover plain images by the chosen plaintext attack; In [22], Su *et al.* identify two vulnerabilities of a chaotic image encryption scheme based on the DNA encoding and the information entropy, and crack the encryption scheme with the chosen plaintext attack; In [23], Dou *et al.* find a security weakness in an image encryption algorithm using the DNA technology and a chaotic logistic map. And according to this security weakness, the encryption algorithm is completely broken by a novel chosen plaintext attack scheme; In a hybrid optical image encryption scheme proposed by Jridi *et al.* [24], there is only a very simple plain image related parameter introduced in the confusion process, but not the more important diffusion process. An attacker can degrade the encryption algorithm into a permutation-only one by using the chosen plaintext attack, whereas permutation-only algorithms have been proved to be unsafe in many literatures [25]–[27].

As we all know, cryptanalysis plays a very important role in promoting the development of encryption technologies. For designs of encryption schemes in the future, pointing out and analyzing security, practicability and feasibility issues existing in latest encryption schemes have somewhat reference significance. Therefore, we investigate a recently reported hyper-chaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling [1] and point out some security, practicability and feasibility issues identified in it. After cryptanalyzing the reported scheme and proposing the chosen plaintext attack algorithm, we also put forward some suggestions to improve the security and practicability of it. For convenience sake, we call the encryption scheme under study as DS-HIES (DNA encoding and Scrambling based Hyper-chaotic Image Encryption Scheme) for short.

The rest of this paper is organized as follows. In Section 2, we briefly introduce DS-HIES, point out some security, practicability and feasibility issues identified in it and make some necessary improvements to the feasibility issues. In addition, we also highlighted the main security issues of DS-HIES and our core attacking principle at the end of this section. In Section 3, DS-HIES is cryptanalyzed and the chosen plaintext attack algorithm is proposed. In Section 4, the simulation tests are carried out for the plaintext sensitivity of DS-HIES and the feasibility of proposed attack algorithm. In Section 5, we present some suggestions to improve the security and practicability of DS-HIES. And in the last section, we conclude the paper.

## 2. DS-HIES and its Issues

In this section, we only give a brief introduction to DS-HIES. For full details of the encryption scheme, please refer to [1]. In the process of introducing the encryption scheme, we use the original notations as many as possible, but for the sake of discussion, we adjust unreasonable notations in the original paper. For example, in Eq. (11) of the original paper,  $a_1$  is used repeatedly to refer to a chaotic sequence and an integer sequence generated from this chaotic sequence, therefore we mark the integer sequence as  $a'_1$  for the sake of distinction. DS-HIES relies on the chaotic sequences generated by a 5D hyper-chaotic system [28] to complete the encryption. The 5D hyper-chaotic system is used twice in DS-HIES, the initial values used are entirely determined by the secure key, as is the numbers of state values that are discarded to avoid the transition effect. Specifically, in the first time, the hyper-chaotic system is used to generate chaotic sequences  $k_1$ ,

$k_2$  and  $k_3$ , and the lengths of these chaotic sequences are all  $M \times N$ . In the second time, chaotic sequences generated by the hyper-chaotic system are  $a_1$ ,  $a_2$ ,  $a_3$  and  $a_4$  with the lengths of  $4 \times M \times N$ . From the perspective of cryptography, DS-HIES actually consists of three processes, pixel-level permutation, pixel-level substitution and pixel-level forward diffusion. Next, we introduce them respectively.

### 2.1 Pixel-Level Permutation

In DS-HIES, the plain image with the size of  $M \times N$  is first permuted. That is, all plain image pixels  $P(i, j)$  are exchanged as follows.

$$P'(i, j) = P(i', j'), \quad P(i', j') = P(i, j) \quad (1)$$

In Eq. (1),  $P'$  is the permuted image,  $i = 1, 2, \dots, M, j = 1, 2, \dots, N$ .  $i'$  and  $j'$  are the integer sequences generated from chaotic sequences  $k_1$  and  $k_2$ .

$$\begin{cases} i' = i + \text{mod} \left( (abs(k_1(i)) - floor(abs(k_1(i)))) \times 10^{15}, M - i \right) \\ j' = j + \text{mod} \left( (abs(k_2(j)) - floor(abs(k_2(j)))) \times 10^{15}, N - j \right) \end{cases} \quad (2)$$

In Eq. (2),  $i = 1, 2, \dots, M, j = 1, 2, \dots, N$ ,  $abs(\bullet)$  returns the absolute value of the operand,  $floor(\bullet)$  returns the largest integer less than or equal to the operand.

### 2.2 Pixel-Level Substitution

DS-HIES performs a very complex pixel-level substitution operation on the permuted image. The pixel-level substitution operation is divided into three parts, intra-pixel circular shift, DNA Exclusive-OR and DNA replacement. Next, we introduce the three parts respectively.

The first one is the intra-pixel circular shift. The permuted image  $P'$  is stretched into the 1D sequence, which is then converted into the binary sequence. In addition, the chaotic sequence  $k_3$  is converted into the integer sequence  $k'_3$ .

$$k'_3(r) = \text{mod} \left( (abs(k_3(r)) - floor(abs(k_3(r)))) \times 10^{15}, 8 \right) \quad (3)$$

It is then similarly converted into the binary sequence. In Eq. (3),  $r = 1, 2, \dots, M \times N$ . Next, the scrambled sequence  $C$  is calculated as follows.

$$C(r) = circshift[P'(r), LSB(k'_3(r)), k'_3(r)] \quad (4)$$

In Eq. (4),  $r = 1, 2, \dots, M \times N$ .  $circshift(\bullet, \bullet, \bullet)$  circularly shifts the first binary sequence operand, the circular shift direction is determined by the second operand, the bit number of the circular shift is determined by the third operand.  $LSB(\bullet)$  returns the lowest bit of the binary sequence operand.

The second one is the DNA Exclusive-OR. First, the chaotic sequence  $a_3$  is converted into the integer sequence  $a'_3$ .

$$a'_3(r) = \text{mod} \left( (abs(a_3(r)) - floor(abs(a_3(r)))) \times 10^{15}, 256 \right) \quad (5)$$

In Eq. (5),  $r = 1, 2, \dots, 4 \times M \times N$ . And then  $C$  and  $a'_3$  are DNA encoded, that is, each two bits are encoded into a base, so the DNA sequences  $c$  and  $d$  are obtained. The specific encoding rule is 00 (1), 01 (2), 10 (3) and 11 (4) being encoded as A, C, G and T respectively. The DNA sequences  $c$  and  $d$  are DNA Exclusive-ORed to obtain the DNA sequence  $F$ .

$$F(i) = c'(i) \oplus d'(i) \quad (6)$$

In Eq. (6),  $i = 1, 2, \dots, 4 \times M \times N$ .

The last one is the DNA replacement, that is, the DNA sequence  $F$  is DNA replaced to obtain the DNA sequence  $F'$ . First, the chaotic sequences  $a_1$  and  $a_2$  are converted into integer sequences  $a'_1$

and  $a'_2$ .

$$a'_1(i) = \text{mod} \left( \left( \text{abs}(a_1(i)) - \text{floor}(\text{abs}(a_1(i))) \right) \times 10^{15}, 6 \right) + 1 \quad (7)$$

$$a'_2(i) = \text{mod} \left( \left( \text{abs}(a_2(i)) - \text{floor}(\text{abs}(a_2(i))) \right) \times 10^{15}, 4 \right) \quad (8)$$

In Eq. (7) and Eq. (8),  $i = 1, 2, \dots, 4 \times M \times N$ . The complementary rule of the DNA replacement is determined by  $a'_1$ , and the specific way of the DNA replacement is determined by  $a'_2$ .

$$F'(i) = E^{a'_2(i)}(F(i)) = \begin{cases} F(i), & \text{if } a'_2(i) = 0 \\ E(F(i)), & \text{if } a'_2(i) = 1 \\ E(E(F(i))), & \text{if } a'_2(i) = 2 \\ E(E(E(F(i)))) & \text{if } a'_2(i) = 3 \end{cases} \quad (9)$$

In Eq. (9),  $i = 1, 2, \dots, 4 \times M \times N$ ,  $E(\bullet)$  returns the base pair of the operand. Finally,  $F$  is decoded into the binary sequence  $G$ , which in turn is converted into the decimal sequence  $H$ .

### 2.3 Pixel-Level Forward Diffusion

First, the chaotic sequence  $a_4$  is converted into the integer sequence  $a'_4$ .

$$a'_4(i) = \text{mod} \left( \left( \text{abs}(a_4(i)) - \text{floor}(\text{abs}(a_4(i))) \right) \times 10^{15}, 256 \right) \quad (10)$$

In Eq. (10),  $i = 1, 2, \dots, 4 \times M \times N$ . And then the forward diffusion of  $H$  is carried out to obtain the cipher image  $R$ .

$$R(1) = a'_4(1) \oplus \text{mod}(a'_4(1) + H(1), 256) \oplus \text{mod} \left( \sum_{j=1}^6 x_j^0 \times 10^{15}, 256 \right) \quad (11)$$

$$R(r) = a'_4(r) \oplus \text{mod}(a'_4(r) + H(r), 256) \oplus R(r-1) \quad (12)$$

In Eq. (12),  $r = 2, \dots, M \times N$ .  $x_j^0 (j = 1, 2, \dots, 6)$  is the secure key. Since the decryption process of DS-HIES is the inverse one of the encryption process, we do not repeat it here.

### 2.4 Issues Identified in DS-HIES

In this subsection, we point out some security, practicability and feasibility issues found in DS-HIES. Furthermore, in order not to change the structure and cryptographic characteristics of DS-HIES, we only make some necessary improvements to the feasibility issues. Suggestions for improving the security and practicability of DS-HIES will be given in Section 5. Since the original paper only describes the encryption of grayscale images of 256 levels, we also discuss only the grayscale images of 256 levels in our paper.

*Issue 1:* The phase portraits of the 5D hyper-chaotic system provided by the original paper is not very clear, the characteristics of the hyper-chaotic system are not well demonstrated. Therefore, we redraw the following phase portraits of the 5D hyper-chaotic system.

*Issue 2:* When converting double precision floating point numbers into integers, DS-HIES always takes the following form.

$$v' = \text{mod} \left( \left( \text{abs}(v) - \text{floor}(\text{abs}(v)) \right) \times 10^{15}, w \right) \quad (13)$$

In Eq. (13),  $v$  is a double precision floating point number that needs to be converted,  $w$  is an integer used as the modulus of the modular operation,  $v'$  is the result of the conversion. Because the computer representation of double precision floating point numbers is in fact the binary floating point number representation of the IEEE 754 standard, the  $v'$  obtained is still a decimal rather

than an integer. Thus, when implementing DS-HIES on the Matlab platform, we make the following improvement to Eq. (13).

$$v' = \text{mod} \left( \text{floor}((\text{abs}(v) - \text{floor}(\text{abs}(v))) \times 10^{15}), w \right) \quad (14)$$

*Issue 3:* In Eq. (2), the calculation method of  $i'$  and  $j'$  is not reasonable. First of all, the issue pointed out in Issue 2 also exists in Eq. (2), we do not repeat it here. That is, we assume that Eq. (2) has been adjusted according to Eq. (14). Because  $i'$  and  $j'$  are calculated in a similar way, we just discuss the calculation method of  $i'$ . According to Eq. (2) and Eq. (14), when  $i = M$ ,  $i'$  is calculated as follows.

$$i' = M + \text{mod} \left( \text{floor}((\text{abs}(k_1(M) - \text{floor}(\text{abs}(k_1(M)))) \times 10^{15}), 0 \right) \quad (15)$$

Thus, the result obtained is unexpected.

$$i' = M + \text{floor}((\text{abs}(k_1(M) - \text{floor}(\text{abs}(k_1(M)))) \times 10^{15}) \quad (16)$$

In addition, when  $i \neq M$ , the result of the modular operation is an integer of  $[0, M - i - 1]$ , therefore the value range of  $i'$  is  $[i, M - 1]$ . It would be more reasonable if the value range of  $i'$  is  $[i, M]$ . Therefore, when implementing DS-HIES, we make the following improvement to Eq. (2).

$$\begin{cases} i' = i + \text{mod} \left( \text{floor}((\text{abs}(k_1(i) - \text{floor}(\text{abs}(k_1(i)))) \times 10^{15}), M - i + 1 \right) \\ j' = j + \text{mod} \left( \text{floor}((\text{abs}(k_2(j) - \text{floor}(\text{abs}(k_2(j)))) \times 10^{15}), N - j + 1 \right) \end{cases} \quad (17)$$

*Issue 4:* In DS-HIES, the uses of chaotic sequences are unreasonable. For the 5 chaotic sequences with the lengths of  $M \times N$  generated by the hyper-chaotic system in the first time, only 3 of them are used, namely,  $k_1$ ,  $k_2$  and  $k_3$ . Similarly, for the 5 chaotic sequences with the lengths of  $4 \times M \times N$  generated by the hyper-chaotic system in the second time, only 4 of them are used, namely,  $a_1$ ,  $a_2$ ,  $a_3$  and  $a_4$ . And not only that, use efficiencies of chaotic sequences  $k_1$ ,  $k_2$ ,  $a_1$ ,  $a_2$ ,  $a_3$  and  $a_4$  are very low. According to Eq. (2),  $k_1$  and  $k_2$  with the lengths of  $M \times N$  are used only in the ranges of  $k_1(1)$  to  $k_1(M)$  and  $k_2(1)$  to  $k_2(N)$ . In Eq. (7), the chaotic sequence  $a_1$  with the length of  $4 \times M \times N$  and effective precision of 15 decimal digits is used to generate the integer sequence  $a'_1$  with the size of  $4 \times M \times N$ , but the value range of  $a'_1$  is only  $[1, 6]$ . Therefore, the use efficiency of the chaotic sequence  $a_1$  is very low. In Eq. (8), the use efficiency of the chaotic sequence  $a_2$  is even lower, and  $a_2$  is only used to generate the integer sequence  $a'_2$  whose value range is  $[0, 3]$ .

*Issue 5:* Key streams are independent of plain images and depend entirely on the secure keys. Specifically, the key streams used in the encryption process of DS-HIES are converted from the chaotic sequences. However, the control parameters of the hyper-chaotic system are fixed, the initial values and the numbers of state values discarded are entirely dependent on the secure key. Therefore, the key streams of DS-HIES are independent of the plain image and depend entirely on the secure key. In this way, if the secure key remains unchanged, the key stream used in the encryption process by DS-HIES will not change. And in the sense of the chosen plaintext attack, although the security key is unknown, attackers can use the secure key and the secure key remains unchanged [21], so such design cannot resist the chosen plaintext attack.

*Issue 6:* The pixel-level substitution of DS-HIES is too complex and the time complexity of it is too high, which also results in the relatively poor encryption speed. However, although the substitution process of DS-HIES is very complex, we can still completely determine the equivalent substitution matrix by only 256 chosen plain images, and then eliminate the substitution effect by up to  $256 \times (M \times N)$  lookups. This point will be explained in detail in Section 3.1.

*Issue 7:* With regard to the confusion and diffusion requirements for strong encryption schemes proposed by Claude Shannon [18], the diffusion requirement is not well satisfied in DS-HIES. Only one forward diffusion is carried out on the intermediate cipher image pixel after the permutation process and the substitution process. Therefore, the plaintext sensitivity of the encryption scheme is very low, which we will test and analyze in Section 4.1. And not only that, the forward diffusion can also be eliminated by simple calculations. When the encryption process is complete, all  $R(r)$  in

Eq. (12) can be obtained directly through the cipher image  $R$ . Therefore, we can do the following simple calculations.

$$R'(r) = R(r) \oplus R(r - 1) \quad (18)$$

In Eq. (18),  $r = M \times N, \dots, 2$ . Substituting Eq. (12) into Eq. (18), we can get

$$R'(r) = a'_4(r) \oplus \text{mod}(a'_4(r) + H(r), 256) \quad (19)$$

According to Eq. (11), the first cipher image pixel is completely determined by the secure key and the first pixel of the substituted image, so there is no need to eliminate the forward diffusion effect. In this way, we obtain the cipher sequence  $R'(r)$  without the forward diffusion effect.

### 2.5 Main Security Issues and Core Attacking Principle

Before beginning the specific cryptanalysis, we want to first highlight the main security issues of DS-HIES and our core attacking principle. The main security issues of DS-HIES are that the key streams are independent of the plain images and the diffusion process can be eliminated by the simple calculations. Therefore, our core attacking principle is as follows: First, the diffusion process of DS-HIES can be eliminated by the simple calculations, then the equivalent substitution matrix and the equivalent permutation matrix can be obtained by the chosen plaintext attack. In this way, we can completely recover the plain image without knowing any secure key related information.

In addition, we also want to emphasize why DS-HIES adopts the DNA encoding, the DNA Exclusive-OR and the DNA replacement based on chaotic sequences, while it is still not secure. In fact, in the cryptographic sense, the ultimate effect of the DNA encoding, the DNA Exclusive-OR and the DNA replacement is a complex pixel value substitution. Specifically, the pixel values are substituted according to the chaotic sequences. However, the chaotic sequences adopted by DS-HIES depend entirely on the security key, while the security key is unknown but remains unchanged in the sense of the chosen plaintext attack. Therefore, the equivalent substitution matrix can be obtained by the chosen plaintext attack and then used to eliminate the substitution effect of the DNA encoding, the DNA Exclusive-OR and the DNA replacement.

## 3. Cryptanalysis and Proposed Attack Algorithm

In this section, we first cryptanalyze DS-HIES, and then propose the specific chosen plaintext attack algorithm based on the cryptanalysis.

### 3.1 Cryptanalysis

According to Eq. (18) and Eq. (19), without any prerequisites, DS-HIES can be simplified into a form only having the pixel-level permutation and the pixel-level substitution. In addition, for cipher images with the sizes of  $M \times N$ , this simplification only needs to perform  $M \times N - 1$  Exclusive-OR operations, thus it is feasible in calculation. For the sake of simplicity, we only discuss the simplified form of DS-HIES which have already eliminated the diffusion effect. In other words, for any cipher image obtained, we first eliminate the diffusion effect according to Eq. (18) and Eq. (19).

Since the simplified DS-HIES only permute and substitute the plain image, we consider eliminating the permutation effect by the chosen plain images of single value pixels. For the sake of simplicity, we assume that the size of cipher image to be attacked is  $2 \times 3$ , namely  $M = 2, N = 3$ . We first choose the plain image  $P_0$  of all zero-value pixels with the size of  $2 \times 3$ .

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Because all the pixel values of  $P_0$  are the same, it would not be affected by Eq. (1) or the permutation process of DS-HIES. Therefore, we can obtain the equivalent substitution matrix through the encryption results of the chosen plain images of single value pixels. And in the sense of the chosen

plaintext attack, although the security key is unknown, attackers can use the secure key [21], so we can encrypt the plain image  $P_0$  and get the corresponding 1D cipher sequence  $C_0$ .

$$C_0 = [c_{0,1}, c_{0,2}, c_{0,3}, c_{0,4}, c_{0,5}, c_{0,6}]$$

According to Issue 5 in Section 2.4, the substitution process of DS-HIES depends entirely on the secure key and is independent of the plain image. Hence, with the secure key unchanged,  $c_{0,1}$  is the substitution result when the first pixel  $P'(1, 1)$  of the permuted image  $P'$  is 0,  $c_{0,2}$  is the substitution result when the second pixel  $P'(1, 2)$  is 0,  $c_{0,3}$  is the substitution result when the third pixel  $P'(1, 3)$  is 0,  $c_{0,4}$  is the substitution result when the fourth pixel  $P'(2, 1)$  is 0,  $c_{0,5}$  is the substitution result when the fifth pixel  $P'(2, 2)$  is 0,  $c_{0,6}$  is the substitution result when the sixth pixel  $P'(2, 3)$  is 0. Next, we select the plain image  $P_1$  of all 1-value pixels with the size of  $2 \times 3$ .

$$P_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Repeat the above process, and we can get

$$C_1 = [c_{1,1}, c_{1,2}, c_{1,3}, c_{1,4}, c_{1,5}, c_{1,6}]$$

Similarly,  $c_{1,1}$  is the substitution result when the first pixel  $P'(1, 1)$  of  $P'$  is 1,  $c_{1,2}$  is the substitution result when the second pixel  $P'(1, 2)$  is 1,  $c_{1,3}$  is the substitution result when the third pixel  $P'(1, 3)$  is 1,  $c_{1,4}$  is the substitution result when the fourth pixel  $P'(2, 1)$  is 1,  $c_{1,5}$  is the substitution result when the fifth pixel  $P'(2, 2)$  is 1,  $c_{1,6}$  is the substitution result when the sixth pixel  $P'(2, 3)$  is 1. In general, we can obtain the substitution result  $C_i$  when all pixels of the permuted image  $P'$  are  $i$  through encrypting  $P_i$  ( $i = 0, \dots, 255$ )

$$P_i = \begin{bmatrix} i & i & i \\ i & i & i \end{bmatrix}$$

$$C_i = [c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}, c_{i,5}, c_{i,6}]$$

Therefore, we can construct the equivalent substitution matrix  $SM$  through  $C_0, C_1, C_2, \dots, C_{255}$ .

$$SM = \begin{bmatrix} c_{0,1} & c_{0,2} & c_{0,3} & c_{0,4} & c_{0,5} & c_{0,6} \\ c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} & c_{1,6} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{255,1} & c_{255,2} & c_{255,3} & c_{255,4} & c_{255,5} & c_{255,6} \end{bmatrix}$$

In this way, if the secure key remains unchanged, for any cipher image  $C$  of DS-HIES with the size of  $2 \times 3$ , we can all determine the value of  $P'(1, 1)$  by looking up  $C(1, 1)$  in column 1 of  $SM$ . That is, if  $C(1, 1) = c_{113,1}$ , then  $P'(1, 1) = 113$ . Similarly, we can determine  $P'(1, 2)$  by looking up  $C(1, 2)$  in column 2 of  $SM, \dots$ , and determine  $P'(2, 3)$  by looking up  $C(2, 3)$  in column 6 of  $SM$ . And in the sense of the chosen plaintext attack, the secure key just remains unchanged. Thus, we have obtained the permuted image  $P'$  with the size of  $2 \times 3$ .

Similarly, the above method can also be applied to the cipher image  $C$  of DS-HIES with the size of  $M \times N$ . Namely, by encrypting 256 chosen plain images  $P_i$  ( $i = 0, \dots, 255$ ) of single value pixels with the sizes of  $M \times N$ , we can also determine the 2D equivalent substitution matrix  $SM$  with size of  $256 \times (M \times N)$ . Therefore, for any cipher image  $C$  of DS-HIES with the size of  $M \times N$ , we can all determine its corresponding permuted image  $P'$  with the size of  $M \times N$  by conducting up to  $256 \times (M \times N)$  lookups in the equivalent substitution matrix  $SM$ . At this point, DS-HIES is degraded into a permutation-only encryption scheme.

Next, we can also get equivalent permutation matrix  $PM$  through the chosen plain images, thus completely recover the permuted image  $P'$  into the plain image  $P$ . First, the cipher image  $C_0$  of all zero-value pixels plain image  $P_0$  with the size of  $M \times N$  is used as the benchmark cipher image.



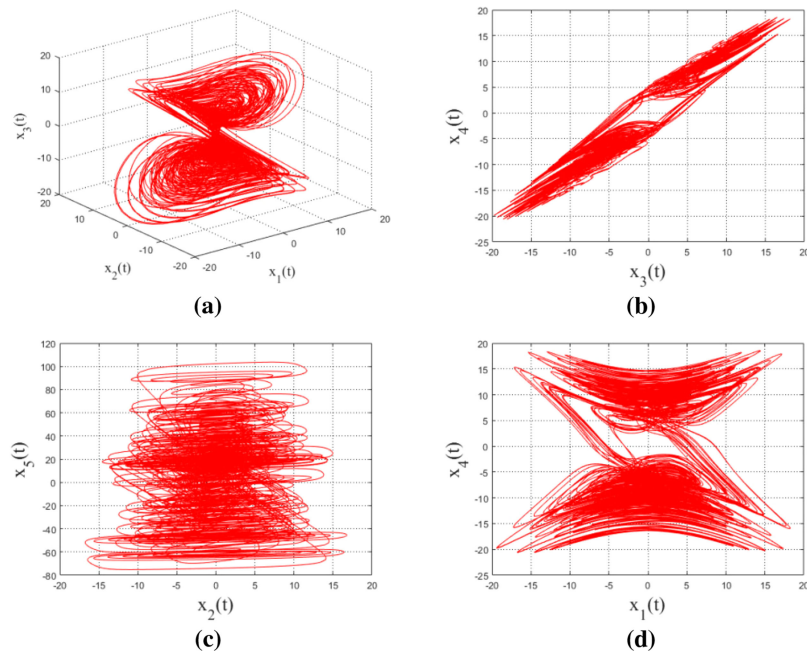


Fig. 1. Phase portraits of 5D hyper-chaotic system with parameters  $a = 30$ ,  $b = 10$ ,  $c = 15.7$ ,  $d = 5$ ,  $e = 2.5$ ,  $f = 4.45$  and  $g = 38.5$ ; (a)  $x_1-x_2-x_3$  phase portrait; (b)  $x_3-x_4$  phase portrait; (c)  $x_2-x_5$  phase portrait; (d)  $x_1-x_4$  phase portrait.

Next, the first 255 pixels of all zero-value pixels plain image  $P_0$  are replaced with  $1, 2, \dots, 255$  to form the chosen plain image  $CPI_1$ .  $CPI_1$  is then encrypted and the obtained cipher image  $CCPI_1$  and  $C_0$  are compared. Since the simplified DS-HIES has no diffusion effect, the positions where the cipher image pixel values have changed in  $CCPI_1$  are the corresponding positions of the first 255 pixels of the chosen plain image  $CPI_1$ . Finally, for each position of where the cipher image pixel values have changed, we look up the changed cipher image pixel value in the equivalent substitution matrix  $SM$ , thus determine the chosen plain image pixel value corresponding to this position. In this way, we can determine the positions of the first 255 pixels of the plain image  $P$  in the permuted image  $P'$  at one time. In general, we can completely determine the equivalent permutation matrix  $PM$  of DS-HIES by constructing up to  $\text{floor}(M \times N/255) + 2$  chosen plain images.

### 3.2 Proposed Chosen Plaintext Attack Algorithm

In Section 3.1, we have cryptanalyzed DS-HIES, and found that the forward diffusion effect of the encryption scheme can be eliminated by  $M \times N - 1$  Exclusive-OR operations for any cipher image  $C$  with the size of  $M \times N$ . And through 256 chosen plain images of single value pixels, the 2D equivalent substitution matrix  $SM$  with the size of  $256 \times (M \times N)$  can also be determined. In addition, through up to  $\text{floor}(M \times N/255) + 2$  chosen plain images, we can completely determine the equivalent permutation matrix  $PM$  of DS-HIES. Next, we give the main frame of the proposed attack algorithm. The flow chart of the proposed algorithm is shown in Fig. 2.

First, we present a simple algorithm for eliminating the forward diffusion effect that needs to be called in the proposed chosen plaintext attack algorithm.

Algorithm 1 is very simple, which is to loop from  $M \times N$  to 2 according to Eq. (18), and calculate the 1D cipher sequence without the forward diffusion effect through the Exclusive-OR operations. As we can see, the time complexity of Algorithm 1 is  $O(M \times N)$ . Next, we present the main frame of the proposed chosen plaintext attack algorithm in Algorithm 2.

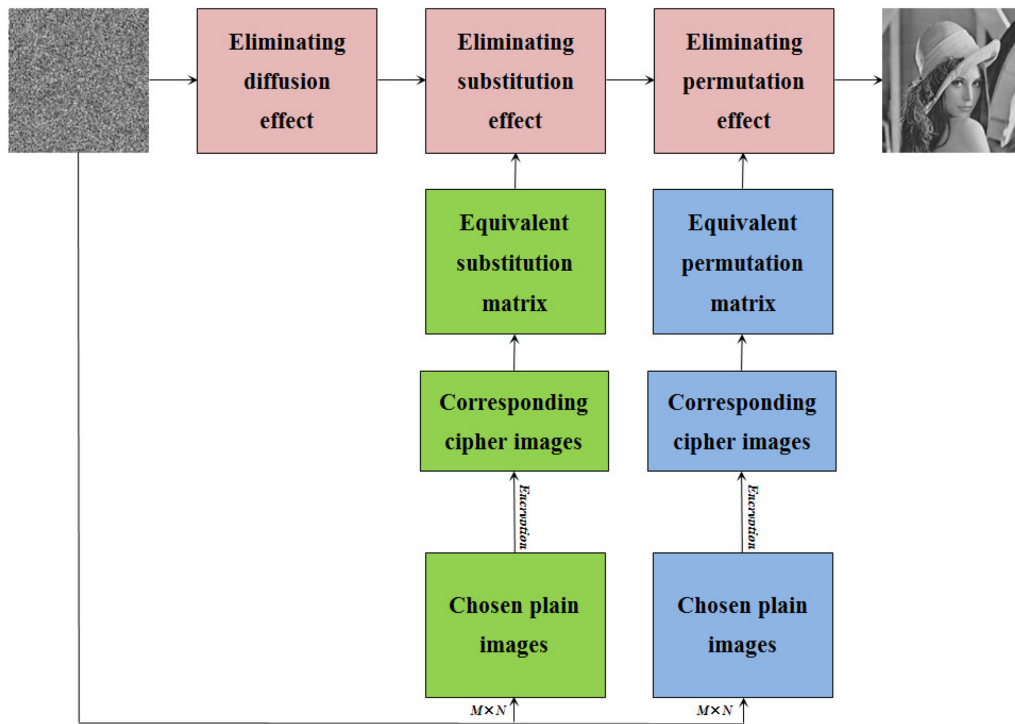


Fig. 2. Flow chart of proposed chosen plaintext attack algorithm.

---

**Algorithm 1:** Eliminating Diffusion Effect *DS\_HIES\_Elimination*.

---

**Input:** 1D sequence  $CI1D$  of the cipher image to be attacked.  
**Output:** 1D sequence  $CI1D$  of the cipher image without the forward diffusion effect.

- 1: for  $r = M*N:-1:2$
- 2:  $CI1D(r) = \text{bitxor}(CI1D(r), CI1D(r-1));$
- 3: end

---

In Algorithm 2, we first determine the height  $M$  and the width  $N$  of the target cipher image  $TCI$ , stretch  $TCI$  into the 1D vector  $TCI1D$ , and then eliminate the forward diffusion effect. Next, through the for loop iterating from 0 to 255, we construct the equivalent substitution matrix  $SM$ . Specifically, for each iteration of the for loop, we all generate one chosen plain image of the single value pixels, encrypt the plain image, stretch the cipher image into 1D vector, eliminate the forward diffusion effect, and add the substitution results of the specific pixel value obtained to the equivalent substitution matrix  $SM$ . After obtaining  $SM$ , we obtain the 1D permuted image  $PPI1D$  by looking up each cipher image pixel value of  $TCI1D$  separately in the column 1 to  $M \times N$  of  $SM$ . In the worst case, the number of lookups is  $256 \times (M \times N)$ , and the time complexity is  $O(M \times N)$ .

Next, according to the cryptanalysis in Section 3.1, we first calculate the number  $CPINum$  of chosen plain images needed to obtain the equivalent permutation matrix  $PM$ , and then construct a 3D matrix  $CPIM$  containing all chosen plain images. In the same way, we encrypt every plain image in the 3D matrix  $CPIM$ , stretch the cipher images into 1D vectors, and eliminate the forward diffusion effect of them. We use the processing result of all zero-value pixels plain image as the 1D benchmark cipher sequence  $BCI1D$ , and use the processing results of the chosen plain image 2 to the chosen plain image  $CPINum$  as the 1D comparison cipher sequences  $CCI1D$ . Then, we save all the comparison cipher sequences to the original cipher vector matrix  $OCVM$ , and save the difference value between each  $CCI1D$  and  $BCI1D$  to the change value matrix  $CVM$ . Next, the corresponding

**Algorithm 2:** Proposed Chosen Plaintext Attack Algorithm *DS\_HIES\_Attack*.

---

```

Input:    The target cipher image TCI to be attacked.
Output:  The recovered plain image RecoveredPlainImage.
1:           $[M,N] = \text{size}(TCI)$ ;
2:           $TCI1D = \text{reshape}(TCI, 1, M*N)$ ;
3:           $TCI1D = DS\_HIES\_Elimination(TCI1D)$ ;
4:          for  $PxVal = 0:255$ 
5:              Generate the chosen plain image CPI with the single pixel value PxVal.
6:              Construct the equivalent substitution matrix SM.
7:          end
8:          Obtain the 1D permuted image PPI1D according to the equivalent substitution
           matrix SM.
9:          The number CPINum of chosen plain images needed to obtain equivalent
           permutation matrix is
10:         calculated.
11:         Generate the 3D matrix CPIM containing all chosen plain images.
12:         for  $i = 1:CPINum$ 
13:              $CIM(:, :, i) = DS\_HIES\_Encryption(CPIM(:, :, i))$ ;
14:         end
15:          $BCI1D = \text{reshape}(CIM(:, :, 1), 1, M*N)$ ;
16:          $BCI1D = DS\_HIES\_Elimination(BCI1D)$ ;
17:         for  $j = 2:CPINum$ 
18:             save all the comparison cipher sequences to the original cipher vector matrix
           OCVM.
19:             save the difference value between each CCI1D and BCI1D to the change
           value matrix CVM.
20:         end
21:         for  $k = 2:CPINum$ 
22:             construct the equivalent permuted matrix PM.
23:         end
24:         for  $l = 1:M*N$ 
25:              $RPI1D(l) = PPI1D(PM(l))$ ;
26:         end
27:          $RecoveredPlainImage = (\text{reshape}(RPI1D, M, N))$ ;

```

---

position of each plain image pixel in the permuted image is determined according to *CVM* and *OCVM*, so as to construct the equivalent permuted matrix *PM*. Finally, we completely recover the 2D plain image *RecoveredPlainImage* through *PM* and the 1D permuted image *PPI1D*. In the process of obtaining the permutation matrix *PM*, compare operations needs to be performed at most  $(\text{floor}(M \times N/255) + 1) \times (M \times N)$  times, so the time complexity of the proposed chosen plaintext attack algorithm is  $O((M \times N)^2)$ .

#### 4. Simulation Results

In this section, many tests are carried out on the images of different content. These tests are based on following software and hardware environment: MATLAB R2017a (9.2.0.538062), Intel(R) Pentium(R) CPU G3260 @ 3.30 GHz, 8 GB RAM and 64-bit Windows 7 Ultimate. In this paper, we use the same control parameters  $a = 30$ ,  $b = 10$ ,  $c = 15.7$ ,  $d = 5$ ,  $e = 2.5$ ,  $f = 4.45$ ,  $g = 38.5$  of the hyper-chaotic system and the same secure key  $x_1^0 = 1.2356$ ,  $x_2^0 = 2.8905$ ,  $x_3^0 = 0.89648$ ,  $x_4^0 = 3.45797$ ,  $x_5^0 = 0.45723$ ,  $x_6^0 = 3.2579$  as the original paper.

TABLE 1  
Difference Images Between Changed Cipher Images and Original Cipher Image

Value changed position	Position after permutation	Cipher image	Difference image
(57,24)	(1,1)		
(64,255)	(64,256)		
(128,255)	(128,256)		
(192,255)	(192,256)		

#### 4.1 Plaintext Sensitivity of DS-HIES

















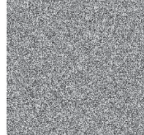

As we have mentioned in Issue 7 of Section 2.4, the DS-HIES only perform one forward diffusion on the intermediate cipher image pixel after the pixel-level permutation and the pixel-level substitution, so the plaintext sensitivity of the encryption scheme is very low. Next, we test the plaintext sensitivity of DS-HIES. The plain image used for tests is the grayscale image Lena with the size of  $256 \times 256$ , which is same as the original paper. We first encrypt the original plain image, then select one plain image pixel each time, add the value of this pixel by 100, and encrypt these 4 changed plain images respectively. Finally, we calculate the difference images between the changed cipher images and the original cipher image.

It can be seen from the test results that the substitution effect of DS-HIES is very good, but the permutation effect is not so good, which depends on limitations of the permutation design of DS-HIES or Eq. (2). This limited row, limited column and backward exchanging design cannot achieve good permutation effect. Although we have improved Eq. (2) in Eq. (17), our improvement is only to make sure DS-HIES can work properly. In addition, because DS-HIES only adopts the simple forward diffusion, when a pixel in the plain image is changed, the range affected by the change depends on the position of this pixel after the permutation process. When the permuted position is (1, 1), as shown in Table 1, the change can affect the maximum number of pixels which is 65535. However, when the changed pixel is in a relatively backward position of the permuted image, the change affects fewer pixels.

#### 4.2 Proposed Chosen Plaintext Attack Algorithm

In this subsection, we test the proposed chosen plaintext attack algorithm. The cipher images used in the tests are with the sizes of  $128 \times 128$  and  $256 \times 256$ , these cipher images are obtained by encrypting the grayscale images Lena, Peppers and Baboon with the sizes of  $128 \times 128$  and  $256 \times 256$ . Firstly, We use the Algorithm 2 to attack the cipher image of grayscale image Lena with the size

TABLE 2  
Specific Test Results of Proposed Chosen Plaintext Attack Algorithm

Plain image	Size	Cipher image	Time spent (seconds)				Recovered plain image
			Eliminate diffusion effect	Obtain substitution matrix	Obtain permutation matrix	Recover plain image	
	128×128		0.0212	105.0482	27.1749	0.0713	
	128×128		0.0216	0	0	0.0731	
	128×128		0.0221	0	0	0.0739	
	256×256		0.0863	381.4362	358.6542	0.3795	
	256×256		0.0861	0	0	0.3886	
	256×256		0.0867	0	0	0.3665	

of  $128 \times 128$ . Specifically, Algorithm 2 first eliminate the diffusion effect of the cipher image, then obtain the equivalent substitution matrix and the equivalent permutation matrix through the chosen plain images, and finally recover the plain image through the obtained equivalent substitution matrix and equivalent permutation matrix.

Next, since we have obtained the equivalent substitution matrix and equivalent permutation matrix of DS-HIES with the image size of  $128 \times 128$ , we only need to eliminate the diffusion effect when we attack the cipher images of Peppers and Baboon with the sizes of  $128 \times 128$ , and then recover the plain images directly based on the equivalent substitution matrix and equivalent permutation matrix.

The attack process of the cipher images with the sizes of  $256 \times 256$  is similar to the above process. The specific test results are shown in Table 2.

From the above test results, we can see that the chosen plaintext attack algorithm proposed in this paper can completely recover the plain images in relatively short time durations. And the time durations required to attack subsequent cipher images are extremely short. In addition, for cipher images of different sizes, the relationship between the time durations required to eliminate the diffusion effect and obtain the equivalent substitution matrix and the equivalent permutation matrix is consistent with our time complexity analysis in Section 3.2. Since the time complexities of eliminating the diffusion effect, obtaining the equivalent substitution matrix and obtaining the equivalent permutation matrix are  $O(M \times N)$ ,  $O(M \times N)$  and  $O((M \times N)^2)$  respectively. For the cipher images with the size of  $256 \times 256$ , the time durations required for the three processes should be 4 times, 4 times and 16 times as long as the time durations required for the cipher

images with the size of  $128 \times 128$ , and the actual test results are 4.07 times, 3.63 times and 15.56 times.

In conclusion, the test results show that the proposed algorithm is feasible and can completely recover the plain images without knowing any secure key related information.

## 5. Possible Improvements to DS-HIES

According to the analysis and the tests above, we know that DS-HIES has a strong substitution part, but its permutation part and diffusion part are relatively weak. Furthermore, another major weakness is that the encryption process is not plain image related, and the key stream remains unchanged when encrypting different plain images, so DS-HIES is unable to resist the chosen plaintext attack. Next, we propose some suggestions for improving the security and practicability of DS-HIES.

### 5.1 Generation of System Initial Values

When generating the initial values of the 5D hyper-chaotic system, considerations can be given to using the plain image related values, such as the hash value of the plain image or the statistical values of the plain image, rather than just using the secure key. This allows the key stream to be also related with the plain image, thus the chosen plaintext attack would not work, since the key stream would be changed with the content of the chosen plain images.

### 5.2 Uses of Chaotic Sequences

Because the generation of chaotic sequences of the 5D hyper-chaotic system is time-consuming, for improving the encryption speed of DS-HIES, the number of iterations of the 5D hyper-chaotic system should be reduced as many as possible. For example, the length of chaotic sequences  $k_1$ ,  $k_2$  and  $k_3$  used by DS-HIES is only  $M + N + M \times N$ , so  $k_1$ ,  $k_2$  and  $k_3$  can be generated by only iterating the 5D hyper-chaotic system  $\text{floor}((M + N + M \times N)/5) + 1$  times rather than  $M \times N$  times in the original paper. In addition, the use efficiencies of chaotic sequences are also need to be improved. For example, when it comes to  $a'_2$  with the length of  $4 \times M \times N$  in Eq. (8), generating it is only need to use a chaotic sequence with the length of  $M \times N$  at most, rather than  $a_2$  with the length of  $4 \times M \times N$  in the original paper.

### 5.3 Design of Encryption Process

The permutation process is always used to improve the key sensitivity and to meet the confusion requirement of cryptosystem design. Therefore, for the permutation of each pixel, the exchange within the entire subscript range  $[1, M \times N]$  should be implemented, instead of being limited to a single row or column.

The substitution process is also always used to improve the key sensitivity and to meet the confusion requirement of cryptosystem design. However, it is just one part of the encryption scheme, so the overly complex substitution process do not make much sense, but only slow down the encryption. The successful attack against DS-HIES in this paper just demonstrated this. For the intra-pixel circular shift, the DNA Exclusive-OR and the DNA replacement in the DS-HIES, Only one of them is need to be retained.

The diffusion process is always used to improve the plaintext sensitivity and to meet the diffusion requirement of cryptosystem design. Therefore, the effect of diffusion process should be sufficient and effective. To improve the sufficiency of the diffusion process of DS-HIES, adding a backward diffusion or iterating the encryption process can be considered. And for improving the effectiveness of the diffusion process of DS-HIES, Eq. (12) can be improved to avoid the situation that the diffusion effect can be eliminated by simple calculations.

## 6. Conclusion

In this paper, we give a brief introduction to DS-HIES and point out some security, practicability and feasibility issues identified in it. For the feasibility issues, without changing the structure and cryptographic characteristics of the original encryption scheme, we make some improvements to DS-HIES. Next, we cryptanalyze DS-HIES and propose the chosen plaintext attack algorithm. In addition, in order to verify the correctness of our cryptanalysis and the feasibility of the proposed chosen plaintext attack algorithm, we carry out the relevant tests. The test results show that some security issues we have pointed out do exist, the proposed chosen plaintext attack algorithm is also feasible and can completely recover the plain image without knowing any secure key related information. Finally, we put forward some suggestions on improving the security and practicability of DS-HIES. In the future, based on the work of this paper, we will design and implement a secure and practical hybrid chaotic image encryption scheme, so as to promote the development of hybrid chaotic image encryption.

## Acknowledgment

*Conflicts of Interest:* The authors declare that there is no conflict of interest regarding the publication of this paper.

---

## References

- [1] S. L. Sun, "A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling," *IEEE Photon. J.*, vol. 10, no. 2, Apr. 2018, Art. no. 7201714.
- [2] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos Solitons Fractals*, vol. 21, no. 3, pp. 749–761, 2004.
- [3] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image Vis. Comput.*, vol. 24, no. 9, pp. 926–934, 2006.
- [4] X. Y. Li *et al.*, "Multiple-image encryption based on compressive ghost imaging and coordinate sampling," *IEEE Photon. J.*, vol. 8, no. 4, pp. 1–11, Aug. 2016.
- [5] X. Wang, G. Zhou, C. Dai, and J. Chen, "Optical image encryption with divergent illumination and asymmetric keys," *IEEE Photon. J.*, vol. 9, no. 2, Apr. 2017, Art. no. 7801908.
- [6] W. Chen, "Optical multiple-image encryption using three-dimensional space," *IEEE Photon. J.*, vol. 8, no. 2, Apr. 2016, Art. no. 6900608.
- [7] J. Wang, Q. H. Wang, and Y. Hu, "Image encryption using compressive sensing and detour cylindrical diffraction," *IEEE Photon. J.*, vol. 10, no. 39, Jun. 2018, Art. no. 7801014.
- [8] P. Refregier and B. Javidi, "Optical image encryption based on input plane and Fourier plane random encoding," *Opt. Lett.*, vol. 20, no. 7, pp. 767–769, 1995.
- [9] M. Babaei, "A novel text and image encryption method based on chaos theory and DNA computing," *Natural Comput.*, vol. 12, no. 1, pp. 101–107, 2013.
- [10] A. A. A. El-Latif, L. Li, N. Wang, Q. Han, and X. Niu, "A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces," *Signal Process.*, vol. 93, no. 11, pp. 2986–3000, 2013.
- [11] J. Fridrich, "Image encryption based on chaotic maps," in *Proc. IEEE Int. Conf. Syst Man, Cybern.*, 1997, vol. 2, pp. 1105–1110.
- [12] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.
- [13] X. Y. Wang and C. M. Liu, "A novel and effective image encryption algorithm based on chaos and DNA encoding," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 1–17, 2016.
- [14] 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.
- [15] X. Zhang, Z. Zhou, and Y. Niu, "An image encryption method based on the feistel network and dynamic DNA encoding," *IEEE Photon. J.*, vol. 10, no. 4, Aug. 2018, Art. no. 3901014.
- [16] 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.
- [17] T. Hu, Y. Liu, L. H. Gong, S. F. Guo, and H. M. Yuan, "Chaotic image cryptosystem using DNA deletion and DNA insertion," *Signal Process.*, vol. 134, pp. 234–243, 2018.
- [18] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dyn.*, vol. 92, no. 2, pp. 305–313, 2018.
- [19] T. Li, Z. C. Miao, and Y. S. Shi, "Ciphertext-only attack on phase-shifting interferometry-based encryption," *IEEE Photon. J.*, vol. 9, no. 5, Oct. 2017, Art. no. 7803108.
- [20] T. Li and Y. S. Shi, "Security risk of diffractive-imaging-based optical cryptosystem," *Opt. Exp.*, vol. 23, no. 16, pp. 21384–21391, 2015.

- [21] 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.
- [22] 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. 1–13, 2016.
- [23] Y. Dou, X. Liu, H. Fan, and M. Li, "Cryptanalysis of a DNA and chaos based image encryption algorithm," *Int. J. Light Electron Opt.*, vol. 145, pp. 456–464, 2017.
- [24] M. Jridi and A. Alfalou, "Real-time and encryption efficiency improvements of simultaneous fusion, compression and encryption method based on chaotic generators," *Opt. Lasers Eng.*, vol. 102, pp. 59–69, 2018.
- [25] A. Jolfaei, X.W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Trans. Inform. Forensics Secur.*, vol. 11, no. 2, pp. 235–246, Feb. 2016.
- [26] S. Li, C. Li, G. Chen, N.G. Bourbakis, and K. T. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Process. Image Commun.*, vol. 23, no. 3, pp. 212–223, 2008.
- [27] C. Li and K. T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Process.*, vol. 91, no. 4, pp. 949–954, 2011.
- [28] B. Fan and L. Tang, "A new five-dimensional hyper-chaotic system and its application in DS-CDMA," in *Proc. 9th Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2012, pp. 2069–2073.