Cryptanalysis of a New Chaotic Image Encryption Technique Based on Multiple Discrete Dynamical Maps

Haiju Fan 1,2,3,4, Chenjiu Zhang 1,2,3,4,*, Heng Lu 1,2,3,4, Ming Li 1,2,3,4 and Yanfang Liu 1,2,3,4

1 College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China; 121064@htu.edu.cn (H.F.); untoldsea@163.com (H.L.); liming@htu.edu.cn (M.L.); 121101@htu.edu.cn (Y.L.)
2 Engineering Lab of Intelligence Business & Internet of Things, Xinxiang 453007, China
3 Big Data Engineering Lab of Teaching Resources & Assessment of Education Quality, Xinxiang 453007, China
4 Key Laboratory of Artificial Intelligence and Personalized Learning in Education of Henan Province, Xinxiang 453007, China
* Correspondence: zcjmail2021@163.com

Abstract: Recently, a new chaotic image encryption technique was proposed based on multiple discrete dynamical maps. The authors claim that the scheme can provide excellent privacy for traditional digital images. However, in order to minimize the computational cost, the encryption scheme adopts one-round encryption and a traditional permutation–diffusion structure. Through cryptanalysis, there is no strong correlation between the key and the plain image, which leads to the collapse of cryptosystem. Based on this, two methods of chosen-plaintext attacks are proposed in this paper. The two methods require 3 pairs and 258 pairs of plain and cipher images, respectively, to break the original encryption system. The simulation results show the effectiveness of the two schemes.

Keywords: image encryption; multiple dynamic maps; permutation–diffusion structure; cryptanalysis; chosen-plaintext attack

1. Introduction

In this era of big data, information security has always been a sensitive topic [1]. Image files as one of the most typical multimedia information will inevitably pass through unstable and unsafe channels, and probably resulting in information leakage. Criminals intercept important information by certain technical means to change and transmit it at will, causing a series of information security problems [2]. Over the years, a large number of researchers have proposed a variety of image encryption ideas and strived to solve the security of image transmission. Among them, chaos-based methods are more suitable for image encryption than DES [3] and AES [4] because of the characteristics of pseudo-randomness and sensitivity. Therefore, chaos-based image encryption schemes have sprung up owing to the researchers who are investing a lot of time and effort in this field [5–12].

So far, many chaos-based image encryption schemes have been based on the permutation–diffusion structure proposed by Fridrich [13]. However, some encryption schemes [14–18] are insecure, which have been cracked by some researchers with known/chosen-ciphertext attacks or known/chosen-plaintext attacks [19–23]. Hu et al. in [17] found security vulnerabilities in the scheme [24] and carried out cryptanalysis and improvement. In the improved scheme, different secret keys were proposed to eliminate vulnerabilities in two rounds of diffusion. However, this scheme was still cracked by Li et al. in [25], who found a linear relationship of diffusion and used chosen-plaintext attacks. These broken encryption schemes are problematic because the key is fixed—it cannot be adapted to different plain images. At this point, the plain/cipher images can be constructed. The generated plain/cipher is then used to infer the key. To overcome this shortcoming, Hsiao et al. in [26] proposed a color image encryption scheme that uses the sum of plain image pixels to enhance the secret key space. The sum of its pixels (Mod 256) is set as the basic parameter of chaotic sequence. Since the sum of
its pixels (Mod 256) has 256 possibilities, it can be exhaustively listed and calculated for correlation analysis. Having found the most relevant image among 256 plain/cipher images, this scheme was cracked by Fan et al. in \[27\] with a chosen-plaintext attack. However, the broken encryption scheme exposed some features of the plain image due to the low-level key space and square image limitations. The attacker uses the number of key matrices and the range of pixel values in encryption to successfully crack the cryptosystem through correlation analysis. In \[28\], an image encryption algorithm based on improved Henon mapping is proposed to encrypt images through shift transformation and diffusion. This scheme was effectively cracked by Zhou et al. in \[29\] using 257 plaintext images, and if boundary pixels were not taken into account, two plaintext images were required to achieve effectively attack. Mondal et al. in \[30\] proposed a new 2D chaotic mapping. The encryption scheme first scrambles the pixel position and then changes the pixel position by diffusion. Li et al. in \[31\] found three security vulnerabilities in this scheme, and successfully cracked them by using a chosen-plaintext attack. Although these schemes involve multiple rounds of encryption with a high level of time complexity, they are all based on the permutation–diffusion structure, and the key is independent of the plain image. Therefore, the attacker can first eliminate the diffusion effect of the cipher image and observe the different characteristics leaked in the encryption stage, and then verify possible solutions to break the stages of multiple rounds of permutation and diffusion.

Recently, Majid in \[32\] proposed a new chaotic image encryption technique based on multiple discrete dynamical maps. It uses three chaotic maps to generate random sequences, and in order to save time, it only uses one round of encryption to achieve confusion and diffusion. The scheme can be divided into two parts: pixel permutation and pixel diffusion. This scheme involves one round of encryption based on a permutation–diffusion structure, which has been analyzed and studied by many researchers \[33,34\]. We conducted cryptanalysis on the encryption scheme and then proposed related attack methods to crack it. In this paper, we have used two methods of chosen-plaintext attack, both of which can destroy diffusion and permutation transformation, and successfully attacked the original encryption scheme. The rest of the paper is organized as follows: Section 2 summarizes Majid’s scheme \[30\]; Section 3 proposes two kinds of chosen-plaintext attacks and carries out simulation experiments; the final section simply summarizes the whole paper.

2. Overview of Majid’s Encryption Scheme

Majid et al. designed an original image encryption scheme, which used one-round encryption to provide better confidentiality. This encryption scheme is divided into five main steps: image sub-blocking, permutation within blocks, permutation between blocks, the first diffusion, and the second diffusion.

2.1. Chaotic Maps of the Original Cipher Scheme

A two-dimensional Henon chaotic map has good chaotic behavior, and its mathematical expression is presented as follows:

\[
\begin{align*}
x_{n+1} &= 1 - ax_n + y_n, \\
y_{n+1} &= bx_n,
\end{align*}
\]

where \(x_0\) and \(y_0\) are the initial parameters of the chaotic map, and \(x_0 = 1.61001, y_0 = 2.9996, a = 1.7085, b = 0.32032\).

A circle chaotic map has good chaotic behavior, and its mathematical expression is as follows:

\[
\theta_{n+1} = \text{mod}(\theta_n + \Omega - \frac{K}{2\pi} \sin(2\pi(\theta_n))), 1),
\]

where \(\theta_0 = 0.4, \Omega = 0.4, K\) is a constant.

A Duffing chaotic map also has good chaotic behavior. Its mathematical expression is as follows:
\[x_{n+1} = y_n, \quad (4)\]
\[y_{n+1} = -bx_n + ay_n - y_n^3, \quad (5)\]
where \(x_0 = -1.5, y_0 = 1.5, a = 2.738, b = 0.1534.\)

2.2. Majid’s Encryption Scheme

The specific steps are as follows:

**Step 1** Input a 256 \(\times\) 256 \(\times\) 3 plain image, divide it into three channels, and then separate the single-channel image into 32 \(\times\) 32 blocks.

**Step 2** Generate a chaotic sequence using a two-dimensional Henon chaotic map and perform pixel permutation within blocks of each channel in order.

**Step 3** Perform random permutation between blocks of each channel orderly.

**Step 4** Use a circle chaotic map to execute the first diffusion for each channel.

**Step 5** Use a Duffing chaotic map to execute the second diffusion for each channel.

3. Cryptanalysis

This section mainly introduces the preparation work and two schemes to eliminate the diffusion effect and obtain equivalent displacement mapping. The feasibility of the two schemes is verified by simulation experiments.

3.1. Preparations

**Proposition 1.** For the encryption scheme of the permutation–diffusion structure, using the all-zero image can break the diffusion transform to obtain the equivalent diffusion matrix.

**Proof.** Assuming that the plain image of 256 \(\times\) 256 \(\times\) 3 is \(P_i\), the permutation transformation expression is defined as:

\[S = F_i(P_i), \quad (6)\]

where \(F_i(.)\) is the permutation transform of the image, \(S\) is the permutation-only image of plain image \(P_i\). During the permutation transformation, the pixel value \(P_i(i, j)\) in the plain image only corresponds to the pixel value \(S(i, j)\) in the permutation-only image of row \(i\) and column \(j\).

The expression of image diffusion transformation is as follows:

\[D = S \oplus D_i, \quad (7)\]

where \(\oplus\) is the XOR operation, \(D_i\) is the diffusion matrix, and \(D\) is the image after diffusion transformation. Finally, the cipher image \(C_i\) corresponding to the plain image \(P_i\) is obtained as shown in Equation (8).

\[C_i = F_i(P_i) \oplus D_i = S \oplus D_i. \quad (8)\]

The permutation phase will not change the original pixel value of the image but simply changes the pixel position of the previous image. A plain image of all zero pixels is still with all zero pixels after permutation. Then, one will obtain the cipher image \(C_0\) from the all-zero image \(P_0\) after the diffusion step.

\[C_0 = F_i(P_0) \oplus D_i = P_0 \oplus D_i = D_i. \quad (9)\]

It can be seen that after a round of permutation and diffusion of all-zero image \(P_0\), the cipher image \(C_0\) obtained at this time is exactly equal to diffusion matrix \(D_i\). Then, the permutation-only image \(S\) can be obtained by XOR operation in Equation (10).

\[C_i \oplus C_0 = S \oplus D_i \oplus D_i = S. \quad (10)\]
Proposition 2. Two diffusions on a permutation-only image $S$ can be linearly transformed into one diffusion.

Proof.

$$C_i(i, j) = S(i, j) \oplus D_1(i, j) \oplus D_2(i, j),$$

where $D_1$ is the first diffusion matrix and $D_2$ is the second diffusion matrix.

At this time, according to the reversible operation of XOR, the XOR result of the next two matrices is calculated firstly as follows:

$$D_i(i, j) = D_1(i, j) \oplus D_2(i, j).$$

Finally, the pixel value of $D_i(i, j)$ is obtained and XOR operation is performed with the pixel value of $S(i, j)$.

$$C_i(i, j) = S(i, j) \oplus D_1(i, j).$$

Then,

$$C_i = S \oplus D_i.$$  \hspace{1cm} \Box

Proposition 3. Suppose a plain image is denoted by $P_1$, and $P_2$ is obtained by changing $n$ pixels of image $P_1$. Then, $P_1$ and $P_2$ are encrypted to obtain the cipher images $C_1$ and $C_2$. So, $C_1$ and $C_2$ will have $n$ different pixel values. The pixel value of each position in the plain image is one-to-one mapped to the pixel value of a certain position in the cipher image, that is, there is a mapping relationship between them.

Proof. The encryption process for plain image $P_1$ is as follows:

$$\begin{cases}
S_1(i', j') = F_2(F_1(P_1(i, j))) \\
C_1(i', j') = S_1(i', j') \oplus K_1(i', j') \oplus K_2(i', j')
\end{cases}$$

where $F_1$ is the first intra-block pixel displacement and $F_2$ is the second inter-block displacement. Image $S_1$ represents the permutation-only image after two permutations, $K_1$ is the diffusion XOR matrix for the first time, and $K_2$ is the diffusion XOR matrix for the second time. Position $(i', j')$ is the transposition of Position $(i, j)$.

The encryption process for plain image $P_2$ is as follows:

$$\begin{cases}
S_2(i', j') = F_2(F_1(P_2(i, j))) \\
C_2(i', j') = S_2(i', j') \oplus K_1(i', j') \oplus K_2(i', j')
\end{cases}$$

According to proposition 2, image $S_1$ and $S_2$ after two diffusions can be considered equivalent to those being processed by one linearly transformed diffusion.

$$\begin{cases}
S_1(i', j') = F_2(F_1(P_1(i, j))) \\
C_1(i', j') = S_1(i', j') \oplus K(i', j')
\end{cases}$$

$$\begin{cases}
S_2(i', j') = F_2(F_1(P_2(i, j))) \\
C_2(i', j') = S_2(i', j') \oplus K(i', j')
\end{cases}$$

where $K(i', j')$ is the XOR result of $K_1(i', j')$ and $K_2(i', j')$.

Since the permutation rules for $P_1$ and $P_2$ are the same, the permutation positions for $P_1$ and $P_2$ are the same. Now, assuming that the number of different pixel values $n = 1$, there is only one different pixel value at $(i'', j'')$ between $P_1$ and $P_2$. Therefore, in XOR, all pixels at the same positions will have the same values, except the one at $(i'', j'')$.

$$\begin{cases}
C_1(i'', j'') = S_1(i'', j'') \oplus K(i'', j'') \\
C_2(i'', j'') = S_2(i'', j'') \oplus K(i'', j'')
\end{cases}$$
Then,
\[ C_1(i'', j'') \neq C_2(i'', j''). \] (20)

Change the pixel values of \( P_1(1,1) \) and \( P_1(1,2) \) to obtain image \( P' \).

\[
\begin{align*}
P'(1, 1) &= \text{mod}(P_1(1, 1) + 1256) \\
P'(1, 2) &= \text{mod}(P_1(1, 2) - 1256) 
\end{align*}
\] (21)

Change the pixel values of \( P_1(1,1) \) and \( P_1(1,3) \) to obtain image \( P'' \).

\[
\begin{align*}
P''(1, 1) &= \text{mod}(P_1(1, 1) + 1256) \\
P''(1, 3) &= \text{mod}(P_1(1, 3) - 1256) 
\end{align*}
\] (22)

The image \( P' \) is encrypted to obtain the cipher image \( C_{51} \), and the image \( P'' \) is encrypted to obtain the cipher image \( C_{52} \). There are two different pixels between \( C_1 \) and \( C_{51} \).

\[ C_{56} = C_1 \oplus C_{51}. \] (23)

In this case, \( C_{56} \) has two non-zero values, the positions of the two non-zero values are represented by \( (X_{11}, Y_{11}) \) and \( (X_{12}, Y_{12}) \), where \( X \) represents the row and \( Y \) represents the column. These two non-zero values are the encrypted positions of \( P'(1,1) \) and \( P'(1,2) \).

There are two different pixels between \( C_1 \) and \( C_{52} \).

\[ C_{57} = C_1 \oplus C_{52}. \] (24)

In this case, \( C_{57} \) has two non-zero values, the positions of the two non-zero values are represented by \( (X_{13}, Y_{13}) \) and \( (X_{14}, Y_{14}) \). These two non-zero values are the encrypted positions of \( P''(1,1) \) and \( P''(1,3) \).

So, \( (X_{11}, Y_{11}) \) is the same position as \( (X_{13}, Y_{13}) \), and \( (X_{12}, Y_{12}) \) is different from \( (X_{14}, Y_{14}) \). The comparison between \( C_{51} \) and \( C_{52} \) shows that there is only one pixel difference. □

3.2. Elimination of Diffusion Effect

3.2.1. Method 1

In the original encryption scheme, a 256 × 256 × 3 plain image \( P \) was used, which was divided into three channel images: \( P_R = \{P_R(i, j)\}_{i=1, j=1}^{256,256} \), \( P_G = \{P_G(i, j)\}_{i=1, j=1}^{256,256} \), and \( P_B = \{P_B(i, j)\}_{i=1, j=1}^{256,256} \). Three channels are encrypted to obtain \( C_R = \{C_R(i, j)\}_{i=1, j=1}^{256,256} \), \( C_G = \{C_G(i, j)\}_{i=1, j=1}^{256,256} \), and \( C_B = \{C_B(i, j)\}_{i=1, j=1}^{256,256} \). The color cipher image \( C \) is obtained by merging the three channels. The original encryption scheme is expressed as follows:

\[
\begin{align*}
C_R(i, j) &= F_2(F_1(P_R(i, j))) \oplus K_1(i, j) \oplus K_2(i, j) \\
C_G(i, j) &= F_2(F_1(P_G(i, j))) \oplus K_1(i, j) \oplus K_3(i, j) \\
C_B(i, j) &= F_2(F_1(P_B(i, j))) \oplus K_1(i, j) \oplus K_4(i, j)
\end{align*}
\] (25)

where \( F_1 \) is the first intra-block pixel permutation and \( F_2 \) is the second inter-block permutation. In the original encryption scheme, the same permutation matrices \( F_1 \) and \( F_2 \) were used for the three channels. Matrix \( K_1 \) is the same matrix used for the first diffusion of the three channels, while \( K_2, K_3, \) and \( K_4 \) are the same for the second diffusion of the three channels.

Since the original scheme used different diffusion matrices for \( P_R, P_G \) and \( P_B \), it is necessary to consider the matrices used in the two diffusions. According to proposition 2, two diffusion matrices are transformed into one diffusion matrix for XOR operation.

\[
\begin{align*}
C_R(i, j) &= F_2(F_1(P_R(i, j))) \oplus K_{12}(i, j) \\
C_G(i, j) &= F_2(F_1(P_G(i, j))) \oplus K_{13}(i, j) \\
C_B(i, j) &= F_2(F_1(P_B(i, j))) \oplus K_{14}(i, j)
\end{align*}
\] (26)
where $K_{12}$ is the XOR matrix of $K_1$ and $K_2$, $K_{13}$ is the XOR matrix of $K_1$ and $K_3$, and $K_{14}$ is the XOR matrix of $K_1$ and $K_4$.

As shown in Figure 1, the position of the plain image is still one-to-one mapped to a certain position of the second permutation-only image. So, the two permutation matrices $F_1$ and $F_2$ can be linearly transformed into a permutation matrix $F$, as shown in the following formula:

$$
\begin{align*}
\begin{cases}
C_R(i,j) = F(P_R(i,j)) \oplus K_{12}(i,j) \\
C_G(i,j) = F(P_G(i,j)) \oplus K_{13}(i,j) \\
C_B(i,j) = F(P_B(i,j)) \oplus K_{14}(i,j)
\end{cases}
\end{align*}
$$

(27)

| Index | 1   | 2   | 3   | ...... | 512 | 513 | 514 | ...... | 65535 | 65536 |
|-------|-----|-----|-----|--------|-----|-----|-----|--------|--------|--------|
| 1D plain image | 158 | 160 | 155 | ...... | 48  | 255 | 45  | ...... | 100    | 102    |
| First permutation-only image | 45  | 158 | ...... | 100   | 255 | 155 | 102 | ...... | 160    | 48     |
| Second permutation-only image | 100 | 255 | 48  | ...... | 45  | 102 | 155 | ...... | 160    | 158    |

Figure 1. An example of second permutation for a 1D plain image.

Then, we can construct an all-zero image $P_0$ of size $256 \times 256 \times 3$ and use the all-zero matrix to obtain the diffusion matrix. The constructed all-zero image $P_0$ is shown as follows:

$$
P_0 = \begin{bmatrix}
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0
\end{bmatrix}_{256 \times 256 \times 3}
$$

(28)

We divide the image $P_0$ into three channels $P_0^R = \{P_0^R(i,j)\}_{i=1,j=1}^{256,256}$ and $P_0^B = \{P_0^B(i,j)\}_{i=1,j=1}^{256,256}$ and $P_0^G = \{P_0^G(i,j)\}_{i=1,j=1}^{256,256}$. After encryption, the cipher images of three channel are

$$
\begin{align*}
\begin{cases}
C_0^R(i,j) = F(P_0^R(i,j)) \oplus K_{12}(i,j) \\
C_0^G(i,j) = F(P_0^G(i,j)) \oplus K_{13}(i,j) \\
C_0^B(i,j) = F(P_0^B(i,j)) \oplus K_{14}(i,j)
\end{cases}
\end{align*}
$$

(29)

An all-zero matrix is still an all-zero zero matrix after permutation. As follows, Equation (29) can be transformed into Equation (30).

$$
\begin{align*}
\begin{cases}
C_0^R(i,j) = P_0^R(i,j) \oplus K_{12}(i,j) \\
C_0^G(i,j) = P_0^G(i,j) \oplus K_{13}(i,j) \\
C_0^B(i,j) = P_0^B(i,j) \oplus K_{14}(i,j)
\end{cases}
\end{align*}
$$

(30)
According to proposition 1, three diffusion matrices $K_{12}, K_{13}$ and $K_{14}$ can be obtained.

$$\begin{cases}
C^R_{0}(i,j) = K_{12}(i,j) \\
C^G_{0}(i,j) = K_{13}(i,j) \\
C^B_{0}(i,j) = K_{14}(i,j)
\end{cases} \quad (31)$$

The process of obtaining three diffusion matrices is shown in Figure 2. The all-zero image $P_0$ is encrypted to obtain the merged cipher image $C_0$. The three channel images $C^R_0$, $C^G_0$ and $C^B_0$ of cipher image $C_0$ can be transformed into three diffusion matrices $K_{12}$, $K_{13}$ and $K_{14}$.

![Figure 2](image2.png)

**Figure 2.** The process of obtaining three diffusion matrices.

In method 1, the obtained diffusion matrix is used to eliminate the diffusion effect, and the main steps are described as follows:

**Step 1** Construct an all-zero image $P_0$ with $256 \times 256 \times 3$, and the influence of two permutations is eliminated by the all-zero matrix.

**Step 2** Three equivalent diffusion matrices, $K_{12}$, $K_{13}$, and $K_{14}$, are obtained.

**Step 3** Then, permutation-only images can be obtained through Equation (32). In the cipher image $C$, the red channel image $C^R$ XOR diffusion matrix $K_{12}$ is permutation-only image $S_R$, the green channel image $C^G$ XOR diffusion matrix $K_{13}$ is permutation-only image $S_G$, and the red channel image $C^B$ XOR diffusion matrix $K_{14}$ is permutation-only image $S_B$. Three channel images, $S_R$, $S_G$, and $S_B$, are merged to obtain color permutation-only image $S$.

$$\begin{cases}
S_R(i,j) = C^R(i,j) \oplus K_{12}(i,j) \\
S_G(i,j) = C^G(i,j) \oplus K_{13}(i,j) \\
S_B(i,j) = C^B(i,j) \oplus K_{14}(i,j)
\end{cases} \quad (32)$$

Figure 3 shows the experiments of method 1. Figure 3(a1) is a plain image and Figure 3(a2) is the cipher image of Figure 3(a1). Figure 3(a3) is a permutation-only image of Figure 3(a2). Figure 3(b1–b3) are permutation-only images from the three channels of Figure 3(a2).

![Figure 3](image3.png)

**Figure 3.** Experiments of method 1: (a1) a plain image; (a2) the cipher image of (a1); (a3) permutation-only image of (a2); (b1–b3) permutation-only images from the three channels of (a2).
3.2.2. Method 2

In method 2, We can construct 256 plain image \( \{ P_x \}_{x=0}^{255} \). The plain images \( \{ P_x \}_{x=0}^{255} \) are encrypted to obtain the cipher images \( \{ C_x \}_{x=0}^{255} \). The three channel images of the cipher image \( \{ C_x \}_{x=0}^{255} \) are \( \{ C_{Rx} \}_{x=0}^{255}, \{ C_{Gx} \}_{x=0}^{255} \) and \( \{ C_{Bx} \}_{x=0}^{255} \). The plain image \( \{ P_x \}_{x=0}^{255} \) is constructed as shown in Equation (33).

\[
P_x = \begin{bmatrix} x & x & \cdots & x \\ x & x & \cdots & x \\ \vdots & \vdots & \ddots & \vdots \\ x & x & x & x \end{bmatrix}_{256 \times 256 \times 3},
\]

(33)

where \( x = 0, 1, 2, \ldots, 255 \). The \( x \) value in the plain image \( \{ P_x \}_{x=0}^{255} \) changes from 0 to 255.

Thus, 256 different plain images \( \{ P_x \}_{x=0}^{255} \) are encrypted to obtain 256 different cipher images \( \{ C_x \}_{x=0}^{255} \). If all the \( (i, j) \)th pixels of plain images \( \{ P_x \}_{x=0}^{255} \) are used to form a vector \( X = \{ x \}_{x=0}^{255} \), each of the elements of \( X \) are different. According to proposition 3, the encrypted cipher image of the plain image always has a unique corresponding position, and each element is mapped one-to-one. So, a given pixel in the target cipher image needs to be decrypted, then the pixel value of the cipher image can be compared with the pixel value at the same position in 256 different cipher images \( \{ C_x \}_{x=0}^{255} \). For a given cipher image \( C \), the three channel images are \( C_R, C_G \) and \( C_B \). Through Equations (34) and (35), the permutation-only images \( S_R = \{ S_R(i, j) \}_{i=1, j=1}^{256,256}, S_G = \{ S_G(i, j) \}_{i=1, j=1}^{256,256} \) and \( S_B = \{ S_B(i, j) \}_{i=1, j=1}^{256,256} \) of three channels are retrieved.

\[
\begin{align*}
C_R(i, j) &= C_{Rx}(i, j) \\
C_G(i, j) &= C_{Gx}(i, j) \\
C_B(i, j) &= C_{Bx}(i, j)
\end{align*}
\]

(34)

\[
\begin{align*}
S_R(i, j) &= x \\
S_G(i, j) &= x \\
S_B(i, j) &= x
\end{align*}
\]

(35)

Figure 4 shows the process of eliminating the diffusion effect, where the color permutation-only image \( S \) of color cipher image \( C \) is successfully obtained. Here, \( S \) is the combined image of three channel images \( S_R, S_G, \) and \( S_B \). Figure 5 shows the experiment of method 2, where Figure 5(a1) is a plain image, Figure 5(a2) is the cipher image of Figure 5(a1), Figure 5(a3) is the permutation-only image of Figure 5(a2), and Figure 5(b1–b3) are permutation-only images from the three channels of Figure 5(a2).

**Figure 4.** Process of eliminating diffusion effect in method 2.
3.3. Obtain Equivalent Permutation Mapping

In the previous section, the effects of diffusion can be eliminated by means of methods 1 and 2. In this way, we can obtain a permutation-only image of all the cipher images.

In this section, the substitution matrices $F_1$ and $F_2$ used in $P_R$, $P_G$, and $P_B$ in the original encryption scheme are the same. Two permutation matrices, $F_1$ and $F_2$, can be linearly transformed into one permutation matrix, $F$, so that the permutation matrix $F$ of each of the three channels is universal.

Cryptanalysis is used in [34] to obtain equivalent permutation matrices on the basis of a permutation-only image. Therefore, we can construct the chosen-plaintext image to obtain the corresponding equivalent permutation mapping $F'$, so that we can eliminate the influence of two permutations and the plain image $P$ can be restored.

Based on Lemma 1 in [35], equivalent permutation mapping can be revealed by $n$ pairs of plain/cipher images.

$$n \geq \lceil \log_g (MN) \rceil,$$

where $g$ is the gray level of the image and $M$ and $N$ are the rows and columns of the plain image. In the original encryption scheme, $256 \times 256 \times 3$ plain image is used, which is divided into three $256 \times 256$ channels for permutation transformation.

The size of each channel is $256 \times 256$, which is a total of 65,536 pixel values. Therefore, $g = 256$, $M = 256$, $N = 256$, $n$ pairs of plain/cipher images can be obtained from Equation (37).

$$2 = \lceil \log_{256} (256 \times 256) \rceil.$$

By calculating $n = 2$, the corresponding equivalent permutation map $F'$ can be solved by using two pairs of plain/cipher images.

The three channel images $P_R$, $P_G$, and $P_B$ are stretched into column vectors, which can be expanded to a two-digit representation in base 256.

$$A = \begin{bmatrix}
(0)(0) & (0)(1) & (0)(2) & \cdots & (0)(255) \\
(1)(0) & (1)(1) & (1)(2) & \cdots & (1)(255) \\
: & : & : & \ddots & : \\
(254)(0) & (254)(1) & (254)(2) & \cdots & (254)(255) \\
(255)(0) & (255)(1) & (255)(2) & \cdots & (255)(255)
\end{bmatrix}_{256 \times 256 \times 3},$$

Figure 5. Experiments of method 1: (a1) a plain image; (a2) the cipher image of (a1); (a3) permutation-only image of (a2); (b1–b3) are permutation-only images from the three channels of (a2).
Entropy is divided into two bitplane images, \( A \) and \( A_\text{perm} \). The chosen-plaintext image \( A \) is divided into two bitplane images, \( A_1 \) and \( A_2 \). Then, \( A_1 \) and \( A_2 \) are used to construct 2 pairs of plain/cipher images.

\[
A_1 = \begin{bmatrix}
0 & 1 & 2 & \cdots & 255 \\
0 & 1 & 2 & \cdots & 255 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & 2 & \cdots & 255 \\
0 & 1 & 2 & \cdots & 255
\end{bmatrix}_{256 \times 256 \times 3} \tag{39}
\]

\[
A_2 = \begin{bmatrix}
0 & 0 & 0 & \cdots & 0 \\
1 & 1 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
254 & 254 & 254 & \cdots & 254 \\
255 & 255 & 255 & \cdots & 255
\end{bmatrix}_{256 \times 256 \times 3} \tag{40}
\]

Two cipher images \( C_{A1} \) and \( C_{A2} \) are obtained by encrypting two plain images, \( A_1 \) and \( A_2 \). Plain image \( A_1 \) is divided into three channel images: \( A_{11} = \{ A_{11}(i,j) \}_{i=1,j=1}^{256,256} \) and \( A_{12} = \{ A_{12}(i,j) \}_{i=1,j=1}^{256,256} \). Plain image \( A_2 \) is divided into three channel images: \( A_{21} = \{ A_{21}(i,j) \}_{i=1,j=1}^{256,256} \) and \( A_{22} = \{ A_{22}(i,j) \}_{i=1,j=1}^{256,256} \). The cipher image \( C_{A1} \) is divided into three channel images: \( C_{r1} = \{ C_{r1}(i,j) \}_{i=1,j=1}^{256,256} \), \( C_{g1} = \{ C_{g1}(i,j) \}_{i=1,j=1}^{256,256} \) and \( C_{b1} = \{ C_{b1}(i,j) \}_{i=1,j=1}^{256,256} \). The cipher image \( C_{A2} \) is divided into three channel images: \( C_{r2} = \{ C_{r2}(i,j) \}_{i=1,j=1}^{256,256} \), \( C_{g2} = \{ C_{g2}(i,j) \}_{i=1,j=1}^{256,256} \) and \( C_{b2} = \{ C_{b2}(i,j) \}_{i=1,j=1}^{256,256} \). After the Equation (41) transformation, the red channel images \( A_{r1} \) and \( A_{r2} \) are transformed into matrices \( P_{r1} = \{ P_{r1}(i,j) \}_{i=1,j=1}^{256,256} \). Transforming the green channels \( A_{g1} \) and \( A_{g2} \) into matrices \( P_{g1} = \{ P_{g1}(i,j) \}_{i=1,j=1}^{256,256} \), the blue channel images \( A_{b1} \) and \( A_{b2} \) are transformed into a matrix \( P_{b1} = \{ P_{b1}(i,j) \}_{i=1,j=1}^{256,256} \).

\[
\begin{align*}
P_{r1} &= 256 \times A_{r2} + A_{r1} \\
P_{g1} &= 256 \times A_{g2} + A_{g1} \\
P_{b1} &= 256 \times A_{b2} + A_{b1}
\end{align*} \tag{41}
\]

By method 1 or 2, the permutation-only images \( S_{r1} = \{ S_{r1}(i,j) \}_{i=1,j=1}^{256,256} \) and \( S_{b1} = \{ S_{b1}(i,j) \}_{i=1,j=1}^{256,256} \) of \( C_{r1} \), \( C_{g1} \) and \( C_{b1} \) are obtained, and permutation-only images \( S_{r2} = \{ S_{r2}(i,j) \}_{i=1,j=1}^{256,256} \) and \( S_{b2} = \{ S_{b2}(i,j) \}_{i=1,j=1}^{256,256} \) of \( C_{r2} \), \( C_{g2} \) and \( C_{b2} \) are obtained.

\[
\begin{align*}
S_r &= 256 \times S_{r2} + S_{r1} \\
S_g &= 256 \times S_{g2} + S_{g1} \\
S_b &= 256 \times S_{b2} + S_{b1}
\end{align*} \tag{42}
\]

Then, it can be observed that all elements in image \( S_r \) are the permutation order of image \( P_{r1} \), all elements in image \( S_g \) are the permutation order of image \( P_{g1} \), and all elements in image \( S_b \) are the permutation order of image \( P_{b1} \). We know that the permutation matrices of the three channels are the same, and any of the three channels are stretched row by row to form a vector \( F' = \{ F'(i) \}_{i=1}^{256,256} \). Therefore, \( F' \) is an equivalent permutation mapping.

Figure 6 shows the flow to obtain this equivalent permutation mapping \( F' \). The main steps of equivalent permutation mapping are described as follows:
Step 1 Two plain images $A_1$ and $A_2$ were encrypted to obtain two cipher images: $C_{A1}$ and $C_{A2}$. The cipher image $C_{A1}$ is divided into three channel images: $C_{r1}$, $C_{g1}$, and $C_{b1}$. The cipher image $C_{A2}$ is divided into three channel images: $C_{r2}$, $C_{g2}$, and $C_{b2}$.

Step 2 By using method 1 or 2, the permutation-only images $S_{r1}$, $S_{g1}$, and $S_{b1}$ of $C_{r1}$, $C_{g1}$, and $C_{b1}$ are obtained first, and after that, the permutation-only images $S_{r2}$, $S_{g2}$, and $S_{b2}$ of $C_{r2}$, $C_{g2}$, and $C_{b2}$ are obtained.

Method 1

(i) Three equivalent diffusion matrices, $K_{12}$, $K_{13}$, and $K_{14}$, are known. The red channel image $C_{r1}$ is XORed with the diffusion matrix $K_{12}$ of the cipher image $C_{A1}$ to obtain the permutation-only image $S_{r1}$. The green channel image $C_{g1}$ is XORed with the diffusion matrix $K_{13}$ to obtain the permutation-only image $S_{g1}$, and the blue channel image $C_{b1}$ is XORed with the diffusion matrix $K_{14}$ to obtain the permutation-only image $S_{b1}$.

\[
\begin{align*}
S_{r1}(i,j) &= C_{r1}(i,j) \oplus K_{12}(i,j) \\
S_{g1}(i,j) &= C_{g1}(i,j) \oplus K_{13}(i,j) \\
S_{b1}(i,j) &= C_{b1}(i,j) \oplus K_{14}(i,j)
\end{align*}
\] \tag{43}

(ii) The red channel image $C_{r2}$ is XORed with the diffusion matrix $K_{12}$ of the cipher image $C_{A2}$ to obtain the permutation-only image $S_{r2}$. The green channel image $C_{g2}$ is XORed with the diffusion matrix $K_{13}$ to obtain the permutation-only image $S_{g2}$, and the blue channel image $C_{b2}$ is XORed with the diffusion matrix $K_{14}$ to obtain the permutation-only image $S_{b2}$.

\[
\begin{align*}
S_{r2}(i,j) &= C_{r2}(i,j) \oplus K_{12}(i,j) \\
S_{g2}(i,j) &= C_{g2}(i,j) \oplus K_{13}(i,j) \\
S_{b2}(i,j) &= C_{b2}(i,j) \oplus K_{14}(i,j)
\end{align*}
\] \tag{44}
Method 2
(i) The cipher image $C_{A_1}$ is divided into three channel images: $C_{r1}$, $C_{g1}$, and $C_{b1}$. The permutation-only images $S_{r1}$, $S_{g1}$, and $S_{b1}$ of the three channel images are retrieved by Equations (45) and (46).

$$
\begin{align*}
C_{r1}(i,j) &= Cr(i,j) \\
C_{g1}(i,j) &= Cg(i,j) \\
C_{b1}(i,j) &=Cb(i,j)
\end{align*}
\tag{45}
$$

$$
\begin{align*}
S_{r1}(i,j) &= x \\
S_{g1}(i,j) &= x \\
S_{b1}(i,j) &= x
\end{align*}
\tag{46}
$$

(ii) The cipher image $C_{A_2}$ is divided into three channel images: $C_{r2}$, $C_{g2}$ and $C_{b2}$. The permutation-only images $S_{r2}$, $S_{g2}$ and $S_{b2}$ of the three channel images are retrieved by Equations (47) and (48).

$$
\begin{align*}
C_{r2}(i,j) &= Cr(i,j) \\
C_{g2}(i,j) &= Cg(i,j) \\
C_{b2}(i,j) &=Cb(i,j)
\end{align*}
\tag{47}
$$

$$
\begin{align*}
S_{r2}(i,j) &= x \\
S_{g2}(i,j) &= x \\
S_{b2}(i,j) &= x
\end{align*}
\tag{48}
$$

Step 3 Image $S_r$, $S_g$ and $S_b$ are obtained from Equation (42). Any of the three channel images are stretched row by row to form $F' = \{F'(i)\}_{i=1}^{56}$. Therefore, $F'$ is an equivalent permutation mapping.

Figure 7 shows the permutation-only images of the constructed plain images $A_1$ and $A_2$, where the first column is the plain images $A_1$ and $A_2$, the second column is the encrypted images of the first column, and the third, fourth, and fifth columns are permutation-only images of the three channels retrieved from the second column. Since the permutation maps of the three channels in the original encryption scheme are the same, the permutation-only images of the three channels are the same.
3.4. Summary of the Attack Strategy

In this section, we provide a detailed overview of the attack process in methods 1 and 2.

3.4.1. Detailed Process of Method 1

Step 1 Based on propositions 1 and 2, the three equivalent diffusion matrices $K_{12}$, $K_{13}$ and $K_{14}$ are obtained from the all-zero image (shown in Figure 2) as described in Section 3.2.1. The diffusion effect is eliminated through the known matrices $K_{12}$, $K_{13}$ and $K_{14}$. Then the permutation-only images of the three channel images $C_R$, $C_G$ and $C_B$ are obtained through the three matrices as shown in the sub-steps (shown in Figure 3).

Sub-Steps The red channel image $C_R$ XORing equivalent diffusion matrix $K_{12}$ is the permutation-only image $S_{R}$, the green channel image $C_G$ XORing diffusion matrix $K_{13}$ is the permutation-only image $S_{G}$, and the red channel image $C_B$ XORing diffusion matrix $K_{14}$ is the permutation-only image $S_{B}$.

Step 2 The equivalent permutation mapping $F'$ can be obtained from 2 pairs of plain/cipher images (shown in Figure 6) in Section 3.3. The detailed sub-steps are as follows:

Sub-Step 1 Two plain images $A_1$ and $A_2$ were encrypted to obtain two cipher images $C_{A_1}$ and $C_{A_2}$. The cipher image $C_{A_1}$ is divided into three channel images $C_{r1}$, $C_{g1}$ and $C_{b1}$. The cipher image $C_{A_2}$ is divided into three channel images $C_{r2}$, $C_{g2}$ and $C_{b2}$.

Sub-Step 2 Based on Section 3.2.1, the permutation-only images $S_{r1}$, $S_{g1}$ and $S_{b1}$ of $C_{r1}$, $C_{g1}$ and $C_{b1}$ were obtained, and the permutation-only images $S_{r2}$, $S_{g2}$ and $S_{b2}$ of $C_{r2}$, $C_{g2}$ and $C_{b2}$ were obtained (shown in Figure 7).

Sub-Step 3 $S_r$, $S_g$ and $S_b$ are obtained from Equation (42). All the three channels are stretched row by row to form an equivalent permutation mapping $F'$.

Step 3 Invoke Algorithm 1 to obtain deciphered images $P_R$, $P_G$ and $P_B$. Merge the three channel images for obtaining image $P$.

Algorithm 1 Obtain the decrypted image $P$.

Decrypting $P = \text{De}_\text{Img}(S_r, S_g, S_b, F')$

1: $w = 0$
2: for $i$ from 1 to 256
3: for $j$ from 1 to 256
4: $[r_1, c_1] \leftarrow \text{find}(F' = w)$
5: $d_{r}(i, j) \leftarrow S_{r}(r_1, c_1)$
6: $d_{g}(i, j) \leftarrow S_{g}(r_1, c_1)$
7: $d_{b}(i, j) \leftarrow S_{b}(r_1, c_1)$
8: $w + 1 \leftarrow w$
9: end
10: end
11: $P \leftarrow d_r, d_g, d_b$ //Merge three channels

3.4.2. Detailed Process of Method 2

Step 1 Based on proposition 3, one can construct 256 different plain images $\{P_x\}_{x=0}^{255}$ and encrypt them to obtain 256 different cipher images $\{C_x\}_{x=0}^{255}$ (shown in Figure 4) in Section 3.2.2. For the three channel images $C_R$, $C_G$ and $C_B$ of the given cipher image $C$, Equations (34) and (35) are used to eliminate the diffusion effect. The permutation-only images $S_{R}$, $S_{G}$ and $S_{B}$ of the three channels are retrieved (shown in Figure 5).

Step 2 The equivalent permutation mapping $F'$ can be obtained from 2 pairs of plain/cipher images (shown in Figure 6) as described in Section 3.3. The detailed sub-steps are the same as the three sub-steps of step 2 in Section 3.4.1.
Step 3 Invoke Algorithm 1 to obtain deciphered images $P_R$, $P_G$ and $P_B$. Then, image $P$ is obtained by merging the three channel images.

3.5. Simulation Experiment

3.5.1. Computational Complexity Analysis

In our cryptanalysis, method 1 used three chosen plain images to attack the original encryption scheme, one all-zero image to obtain the diffusion matrix and two chosen plain images to obtain the equivalent permutation mapping. For a cipher image with a size of $256 \times 256 \times 3$, the data complexity is $O(3 \times 256 \times 256 \times 3)$, which is close to $O(2^{23})$. Method 2 used 258 chosen plain images to attack the original encryption scheme. A total of 256 chosen plain images were used to eliminate the diffusion effect, and 2 chosen plain images were used to obtain the equivalent permutation mapping. Similarly, for a cipher image with a size of $256 \times 256 \times 3$, the data complexity is $O(258 \times 256 \times 256 \times 3)$, which is close to $O(2^{26})$. More remarkably, method 1 is faster, but method 2 can break the diffusion phase of Majid’s encryption scheme more subtly.

3.5.2. Experimental Results

The simulation experiment was performed on a personal computer (Intel Core i5-10210U 2.11 GHz CPU, 16 GB). All experiments were carried out using MATLAB R2020a. In the simulation experiments, we used the same initial values as the Majid’s scheme. The two attack methods are identical in the steps of obtaining equivalent permutation mapping $F'$ but different in the steps of eliminating diffusion effect. We implemented these two attack methods using some different experimental images for accuracy. It can be found from Table 1 that the two methods will cause time differences due to different computational complexity, and different images will also cause time differences in different attack processes. However, the two solutions we proposed are sufficient in terms of real-time requirements.

Table 1. Execution time (seconds).

| Image   | Image Size | Encryption | Deciphering |
|---------|------------|------------|-------------|
| Lena    | 256 × 256 × 3 | 1.191755   | 9.057486 | 235.661488 |
| Pepper  | 256 × 256 × 3 | 1.083255   | 8.445166 | 233.262743 |
| Sailboat| 256 × 256 × 3 | 1.058173   | 8.273460 | 233.330141 |

The simulation results of the two attack methods in this paper are shown in Figure 8. Figure 8(a1,b1,c1) are the images of “Lena”, “Sailboat”, and “Pepper”. They are encrypted by the original cryptosystem as plain images, and the encrypted images are shown in Figure 8(a2,b2,c2). Then, we can execute two attack methods, respectively. The encrypted image can be restored to permutation-only images first, and then the decrypted image, as shown in column 3, column 4, and column 5 in Figure 8. Figure 8(a6,b6,c6) show the combined images of the three channels, that is, the recovered decrypted images. After comparison, we find that the decrypted image is consistent with the plain image, which confirms the feasibility of our attack methods.
In this paper, we cryptanalyzed a new chaotic image encryption technique based on multiple discrete dynamic maps, which adopts one-round encryption and permutation–diffusion structure. (1) The permutation–diffusion structure of one-round execution is found to be insecure because the cipher image generated by the all-zero image is the same as the diffusion matrix of the XOR phase. (2) The original encryption scheme’s secret key is irrelevant to the plain image, leading to the successful attack of the two proposed methods. The simulation results show that both of our methods are feasible.

In view of the original encryption scheme, we put forward some reasonable suggestions. (1) There should be strong correlation between the key and the plain image. (2) Multiple rounds of encryption should be adopted since it is difficult to avoid security problems with only one round of encryption. (3) Add nonlinear substitution phase to the encryption scheme as appropriate. (4) For the diffusion stage, some pixels in the plain image should be fused to complicate the diffusion, so as to avoid obtaining the diffusion matrix directly.

**Author Contributions:** Conceptualization, H.F., M.L. and Y.L.; methodology, M.L. and H.L.; software, C.Z. and H.L.; validation, H.F., C.Z. and Y.L.; formal analysis, H.F. and M.L.; investigation, H.F., C.Z. and H.L.; resources, H.F., C.Z. and H.L.; data curation, C.Z.; writing—original draft preparation, C.Z. and H.F.; writing—review and editing, C.Z., H.F. and M.L.; visualization, H.L.; supervision, H.F., M.L. and Y.L.; project administration, H.F., M.L.; funding acquisition, M.L. and H.F.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Science and Technology Research Project of Henan Province (Grant No. 212102210413).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All results and data obtained can be found in open access publications.

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

1. Zhang, E.; Li, M.; Yiu, S.; Du, J.; Zhu, J.; Jin, G. Fair hierarchical secret sharing scheme based on smart contract. Inf. Sci. 2021, 546, 166–176. [CrossRef]

2. Zhang, E.; Li, H.; Huang, Y.; Hong, S.; Zhao, L.; Ji, C. Practical multi-party private collaborative k-means clustering. Neurocomputing 2022, 467, 256–265. [CrossRef]

3. Daudigny, R.; Ledig, H.; Muller, F.; Valette, F. SCARE of the DES—(Side channel analysis for reverse engineering of the data encryption standard). Appl. Cryptogr. Netw. Secur. Proc. 2005, 3531, 393–406.

4. Ap, WS.; Phan, R.C.W.; Goi, B.M. Cryptanalysis of a highdefinition image encryption based on AES modification. Wirel. Pers. Commun. 2016, 88, 685–699.

5. Hua, Z.Y.; Zhou, Y.C.; Pun, C.M.; Chen, C.I.P. 2D Sine Logistic modulation map for image encryption. Inf. Sci. 2015, 297, 80–94. [CrossRef]

6. Ua, Z.Y.; Zhou, Y.C. Image encryption using 2D logisticadjusted-sine map. Inf. Sci. 2016, 339, 237–253.

7. Enayatifar, R.; Abdullah, A.H.; Isnin, I.F.; Altameem, A.; Lee, M. Image encryption using a synchronous permutation-diffusion technique. Opt. Lasers Eng. 2017, 90, 146–154. [CrossRef]

8. Irfan, Y.; Majid, K. A New Efficient Digital Image Encryption Based on Inverse Left Almost Semi Group and Lorenz Chaotic System. Entropy 2018, 20, 913.

9. Diab, H.; El-Semary, A.M. Secure image cryptosystem with unique key streams via hyper-chaotic system. Signal Process. 2018, 142, 53–68. [CrossRef]

10. Chen, J.X.; Zhu, Z.L.; Zhang, L.B.; Zhang, Y.S.; Yang, B.Q. Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption. Signal Process. 2018, 142, 340–353. [CrossRef]

11. Zou, C.; Wang, X.; Li, H. Image encryption algorithm with matrix semi-tensor product. Nonlinear Dyn. 2021, 105, 859–876. [CrossRef]

12. Xinyuan, W.; Maochang, Z. An image encryption algorithm based on hyperchaotic system and DNA coding. Opt. Laser Technol. 2021, 143, 107316.

13. Fridrich, J. Image encryption based on chaotic maps. IEEE Int. Conf. Syst. 1997, 2, 1105–1110.

14. Zhou, Y.C.; Bao, L.; Chen, C.I.P. A new 1D chaotic system for image encryption. Signal Process. 2014, 97, 172–182. [CrossRef]

15. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. Signal Process. 2017, 138, 129–137. [CrossRef]

16. Hua, Z.; Zhou, Y. Design of image cipher using block-based scrambling and image filtering. Inf. Sci. 2017, 396, 97–113. [CrossRef]

17. Niyat Yaghooti, A.; Moattar, M.H.; Torshiz Niazi, M. Color image encryption based on hybrid hyper-chaotic system and cellular automata. Opt. Lasers Eng. 2017, 90, 225–237. [CrossRef]

18. Wu, J.; Liao, X.; Yang, B. Image encryption using 2D Hénon-Sine map and DNA approach. Signal Process. 2018, 153, 11–23. [CrossRef]

19. Li, M.; Wang, P.C.; Liu, Y.F.; Fan, H.J. Cryptanalysis of a novel bit-level color image encryption using improved 1D chaotic map. IEEE Access. 2019, 7, 145798–145806. [CrossRef]

20. Wang, H.; Xiao, D.; Chen, X.; Huang, H. Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map. Signal Process. 2018, 144, 444–452. [CrossRef]

21. Feng, Y.; Xinhui, G.; Hanpeng, L.; Shihong, W. Differential cryptanalysis of image cipher using block-based scrambling and image filtering. Inf. Sci. 2021, 554, 145–156. [CrossRef]

22. Li, M.; Lu, D.D.; Wen, W.; Ren, H.; Zhang, Y. Cryptanalyzing a color image encryption scheme based on hybrid hyper-chaotic system and cellular automata. IEEE Access 2018, 6, 47102–47111. [CrossRef]

23. Chen, J.; Chen, L.; Zhou, Y. Cryptanalysis of a DNA-based image encryption scheme. Inf. Sci. 2020, 520, 130–141. [CrossRef]

24. Hu, G.Q.; Xiao, D.; Wang, Y.; Li, X. Cryptanalysis of a chaotic image cipher using Latin square-based confusion and diffusion. Nonlinear Dyn. 2017, 88, 1305–1316. [CrossRef]

25. Li, M.; Lu, D.D.; Xiang, Y.; Zhang, Y.; Ren, H. Cryptanalysis and improvement of a chaotic image cipher using two-round permutation and diffusion. Nonlinear Dyn. 2019, 96, 31–47. [CrossRef]

26. Hsiao, H.J.; Lee, J. Color image encryption using chaotic nonlinear adaptive filter. Signal Process. 2015, 117, 281–309. [CrossRef]

27. Fan, H.; Li, M.; Liu, D.; Zhang, E. Cryptanalysis of a colour image encryption using chaotic APFM nonlinear adaptive filter. Signal Process. 2018, 143, 28–41. [CrossRef]

28. Sheela, S.J.; Suresh, K.V.; Tandur, D. Image encryption based on modified Henon map using hybrid chaotic shift transform. Multimed. Tools Appl. 2018, 77, 25223–25251. [CrossRef]

29. Zhou, K.; Xu, M.; Luo, J.; Fan, H.; Li, M. Cryptanalyzing an image encryption based on a modified Henon map using hybrid chaotic shift transform. Digit. Signal Process. 2019, 93, 115–127. [CrossRef]

30. Mondal, B.; Behera, P.K.; Gangopadhyay, S. A secure image encryption scheme based on a novel 2D sines-cosine cross-chaotic (SC3) map. J. Real-Time Image Process. 2020, 18, 1–18. [CrossRef]

31. Li, M.; Wang, P.; Yue, Y.; Liu, Y. Cryptanalysis of a secure image encryption scheme based on a novel 2D sine–cosine cross-chaotic map. J. Real-Time Image Process. 2021. (prepublish). [CrossRef]

32. Khan, M.; Masood, F. A novel chaotic image encryption technique based on multiple discrete dynamical maps. Multimed. Tools Appl. 2019, 78, 26203–26222. [CrossRef]
33. Arroyo, D.; Diaz, J.; Rodriguez, F.B. Cryptanalysis of a one round chaos-based substitution permutation network. *Signal Process.* 2012, *93*, 1358–1364. [CrossRef]

34. Zhang, L.Y.; Liu, Y.; Pareschi, F.; Zhang, Y.; Wong, K.W.; Rovatti, R.; Setti, G. On the security of a class of diffusion mechanisms for image encryption. *IEEE Trans. Cybern.* 2017, *99*, 1163–1175. [CrossRef] [PubMed]

35. Jolfaei, A.; Wu, X.; Muthukkumarasamy, V. On the security of permutation-only image encryption schemes. *IEEE Trans. Inf. Forensics Secur.* 2016, *11*, 235–246. [CrossRef]