entropy

MDPI

Article

Security Analysis and Improvement of an Image Encryption Cryptosystem Based on Bit Plane Extraction and Multi Chaos

Shuqin Zhu [1] and Congxu Zhu [2,*]

[1] School of Computer Science, Liaocheng University, Liaocheng 252059, China; shuqinzhu2008@163.com
[2] School of Computer Science and Engineering, Central South University, Changsha 410083, China
* Correspondence: zhucx@csu.edu.cn; Tel.: +86-0731-8882-7601

Abstract: This paper analyzes the security of image encryption systems based on bit plane extraction and multi chaos. It includes a bit-level permutation for high, 4-bit planes and bit-wise XOR diffusion, and finds that the key streams in the permutation and diffusion phases are independent of the plaintext image. Therefore, the equivalent diffusion key and the equivalent permutation key can be recovered by the chosen-plaintext attack method, in which only two special plaintext images and their corresponding cipher images are used. The effectiveness and feasibility of the proposed attack algorithm is verified by a MATLAB 2015b simulation. In the experiment, all the key streams in the original algorithm are cracked through two special plaintext images and their corresponding ciphertext images. In addition, an improved algorithm is proposed. In the improved algorithm, the generation of a random sequence is related to ciphertext, which makes the encryption algorithm have the encryption effect of a "one time pad". The encryption effect of the improved algorithm is better than that of the original encryption algorithm in the aspects of information entropy, ciphertext correlation analysis and ciphertext sensitivity analysis.

Keywords: information security; image encryption; security analysis; chosen-plaintext attack

Citation: Zhu, S.; Zhu, C. Security Analysis and Improvement of an Image Encryption Cryptosystem Based on Bit Plane Extraction and Multi Chaos. *Entropy* **2021**, *23*, 505. https://doi.org/10.3390/e23050505

Academic Editor: Amelia Carolina Sparavigna

Received: 5 April 2021
Accepted: 20 April 2021
Published: 22 April 2021

1. Introduction

With the rapid development of computer and internet technology, all kinds of multimedia data including digital images are transmitted through a network and stored on a disk, which greatly facilitates people's work and life. Image information can easily be illegally copied, tampered with, spread, and used for other malicious damage in the process of image transmission. Therefore, it is necessary to adopt reliable image encryption technology to ensure the safe transmission and storage of digital images. As described in [1], the main techniques used in image encryption algorithms include chaotic mapping, DNA computing, neural networks, compressed sensing, cellular automata, wavelet transformation, and so on. However, chaos has become an ideal tool for designing secure and efficient encryption schemes due to the sensitivity, ergodicity and randomness of chaotic systems under the initial conditions and the system parameters, which coincides with the two basic principles of cryptography: diffusion and confusion. In 1998, Friedrich [2] proposed an alternative diffusion encryption architecture, which was later developed into a classical scrambling diffusion encryption architecture [3–5]. Based on this structure, scholars have proposed many image encryption algorithms [6–14]. Chai et al. [6] designed a color image encryption algorithm based on a four-wing hyperchaotic system and DNA coding. The generation of random sequences and DNA coding sequences used in the algorithm is related to plaintext. Wang et al. [7] proposed a new image encryption algorithm in which the cipher pixel value depends on two random, nonadjacent pixels and a chaos interference value. A new chaos-based image encryption algorithm was designed by Li et al. [8], which adopts the orbit perturbation and the dynamic state variable selection mechanisms. Zhu et al. [9] constructed a five-dimensional, discrete, hyper-chaotic map by combining the logistic map and the 3D discrete Lorenz map, and designed a block-based image encryption scheme

related to a plain image based on this chaotic system. A chaotic image encryption, using the Hopfield model and Hindmarsh–Rose neurons implemented on FPGA, was presented in [15], which was focused on finding suitable coefficient values of neurons to generate robust random binary sequences that can be used in image encryption. In [16], a new algorithm to improve the randomness of five chaotic maps that were implemented on a PIC micro-controller was proposed. The improved chaotic maps were tested to encrypt digital images in a wireless communication scheme, particularly on a machine to machine (M2M) link, via ZigBee channels.

The operation of the above algorithms was based on the pixel level. At the same time, the chaotic image encryption algorithm based on the bit-level technique has also attracted the attention of researchers due to its reliability and effectiveness [17–19]. Wang et al. [17] proposed a hyperchaos-based image encryption algorithm based on bit-level permutation and DNA encoding. In [18], a symmetric color image encryption algorithm adopting bit-permutation was presented, in which the key streams are closely related to the plain image. In 2018, an image encryption algorithm with an avalanche effect based on bit-level substitution was proposed in [19]. With the improvement of cryptanalysis and design level, it is becoming increasingly difficult to decipher encryption algorithms. However, some algorithms are insecure against various common cryptanalysis methods [20–26]. Huang et al. [20] presented a simple color image encryption algorithm, in which the permutation process and diffusion process are all related to plaintext. The authors claimed that the algorithm could resist chosen- or known-plaintext attacks efficiently. However, in 2020, Lin et al. [21] found that Huang et al.'s algorithm [20] could not resist chosen-plaintext attacks and they proposed an enhanced algorithm to overcome the flaw. Diab and El-semary [22] broke an image encryption algorithm presented by Chen et al. [23]. An image block encryption algorithm with a sufficient security level and high encryption speed was proposed in [20], while Ma et al. [25] broke the equivalent secret keys successfully by giving five chosen plain images and the corresponding cipher images; Zhu et al. [26] cracked the equivalent key sequence for image obfuscation and image scrambling, respectively, by combining the chosen-plaintext attack and the chosen-ciphertext attack. In [27], Zhu et al. cracked a color image encryption scheme based on combined 1D chaotic maps [28]. An image encryption algorithm using an S-box generated by chaos [29] and a multiple chaotic S-boxes-based image encryption algorithm [30] was broken by Zhu et al. [31] and Lu et al. [32], respectively. It can be found from the literature [20,23,24,29,30] that the main reason why the above algorithms were cracked is that the equivalent key stream of the encryption system had nothing to do with plaintext.

In [33], an image encryption algorithm based on binary bit plane extraction and multiple chaotic maps was proposed, which includes a bit-level permutation for high, 4-bit planes and bit-wise XOR diffusion. It claimed that the algorithm has high security performance. However, the security analysis showed that the key in the bit plane permutation and the key in the diffusion phase are independent of the plain image or the cipher image; therefore, the equivalent diffusion key and the equivalent permutation key can both be obtained by adopting the chosen-plaintext attack. This paper is organized as follows. Section 2 concisely describes the original algorithm in [33]. In Section 3, the security of the algorithm is analyzed, and the equivalent key is cracked by the chosen-plaintext attack method. In Section 4, the experimental simulation is carried out. An improved image encryption algorithm is proposed in Section 5. Section 6 concludes the paper.

2. The Original Image Encryption Cryptosystem

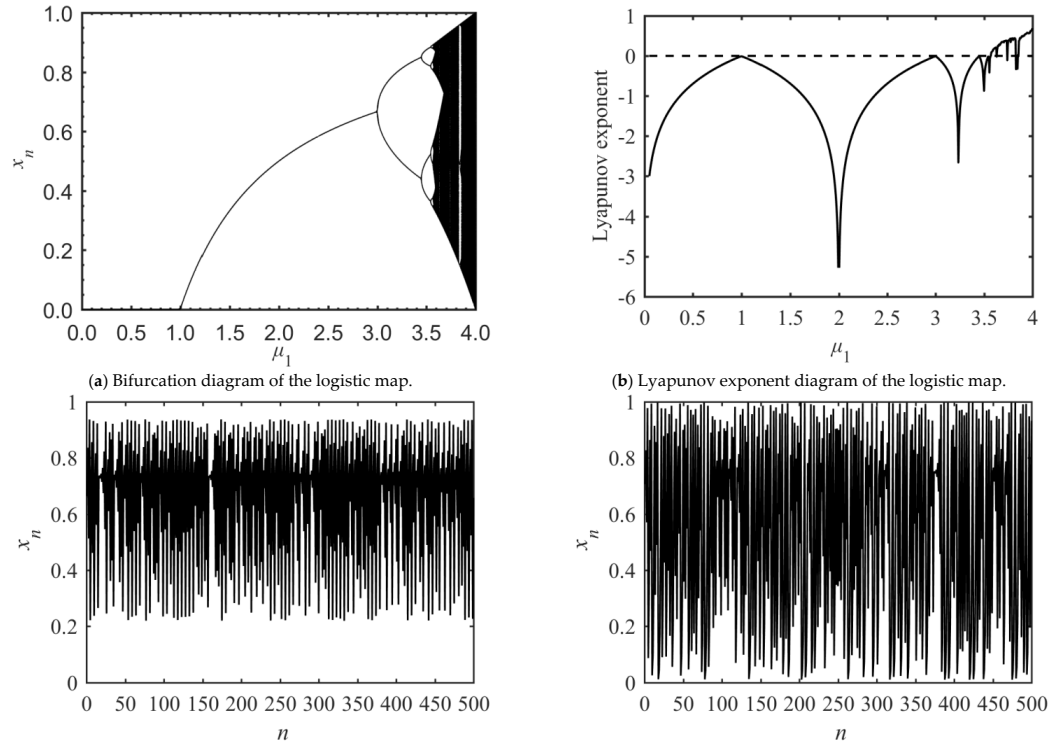
This section provides a brief introduction to the original encryption system of [33].

2.1. Logistic Chaotic Map and Cubic Logistic Chaotic Map

The logistic chaotic map and the cubic logistic chaotic map are used in the original algorithm. The logistic map is shown as

$$x_{n+1} = \mu_1 x_n (1 - x_n) \tag{1}$$

In order to further determine the value range of parameter μ_1 when the logistic map generates a chaotic sequence, the bifurcation diagram and the Lyapunov exponent diagram of the logistic map are given as Figure 1. It is found that system (1) is chaotic when the control parameter $\mu_1 \in (3.57, 4)$ and $x_n \in (0, 1)$.



(**a**) Bifurcation diagram of the logistic map.

(**b**) Lyapunov exponent diagram of the logistic map.

(**c**) Plot for the logistic sequence against 500 iterations for $\mu_1 = 3.75$.

(**d**) Plot for the logistic sequence against 500 iterations for $\mu_1 = 3.99$.

Figure 1. Dynamic system analysis of logistic map.

The cubic logistic map is defined as

$$y_{n+1} = \mu_2 y_n (1 - y_n)(2 + y_n) \tag{2}$$

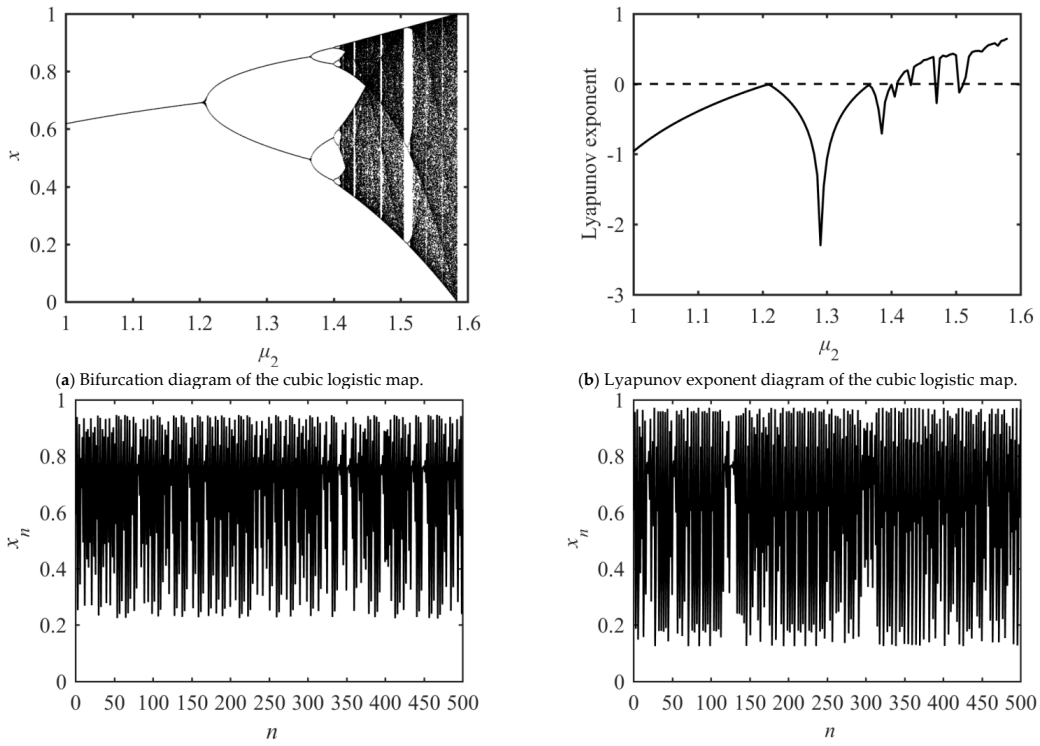
Similarly, in order to further determine the value range of parameter μ_2 when the cubic logistic map generates a chaotic sequence, the bifurcation diagram and Lyapunov exponent diagram of the cubic logistic map are given as Figure 2. It is found that system (2) is chaotic when the control parameter $\mu_2 \in (1.41, 1.59)$ and $y_n \in (0, 1)$.

2.2. Detailed Description of the Original Encryption Algorithm

The secret key of the original encryption algorithm contains four parameters: x_0, μ_1, y_0, μ_2. The encryption objects of the original algorithm are a gray image and an RGB color image with size of $H \times W$ (height $\times$ width). For the convenience of the description, only the gray image is discussed, and the encryption algorithm of the RGB color image is basically the same. The plain image is defined as $P = \{p(i,j)\}$, and the permuted image and the cipher image are defined as $P' = \{p'(i,j)\}$ and $C = \{c(i,j)\}$, respectively. The encryption process includes three stages, as follows:

Step 1: Bit plane decomposition. The plain image $P = \{p(i,j)\}$ is decomposed into 8-bit planes $P_k = \{p_k(i,j)\}$ (k = 1, 2,..., 8), given by

$$P = \sum_{k=1}^{8} 2^{k-1} p_k = p_1 + 2p_2 + 2^2 p_3 + 2^3 p_4 + 2^4 p_5 + 2^5 p_6 + 2^6 p_7 + 2^7 p_8 \tag{3}$$



(**a**) Bifurcation diagram of the cubic logistic map.

(**b**) Lyapunov exponent diagram of the cubic logistic map.

(**c**) Plot for the cubic logistic sequence against 500 iterations for $\mu_2 = 1.5$.

(**d**) Plot for the cubic logistic sequence against 500 iterations for $\mu_2 = 1.54$.

Figure 2. Dynamic system analysis of cubic logistic map.

Here, let $\mathbf{Z_m}$ represent the set $[0, m-1]$, so $p(i,j) \in \mathbf{Z}_{256}$, $p_k(i,j) \in \mathbf{Z}_2$, P_1 and P_8 are the lowest and highest bit planes, respectively;

Step 2: Bit-level permutation. The permutation process is only for the high, 4-bit planes, the 8-th bit plane is described as an example. Firstly, given the initial value y_0 and the control parameter μ_2, the cubic logistic map is iterated to get two real sequences, $\{y_1, y_2, \ldots, y_H\}$ with length H and $\{y_{H+1}, y_{H+2}, \ldots, y_{H+W}\}$ with length W, respectively. Then, the two real number sequences are sorted in ascending order to obtain the position index sequence $RS = \{rs(i)\}_{i=1}^{H}$ and $CS = \{cs(j)\}_{j=1}^{W}$, respectively. Then, using the two sequences RS and CS, the permutation bit plane $P_8' = \{p_8'(i,j)\}_{i=1,\,j=1}^{H,\,W}$ corresponding to the original 8-th bit plane $P_8 = \{p_8(i,j)\}_{i=1,j=1}^{H,W}$ is obtained by Equation (4)

$$p_8'(i,j) = p_8(rs(i), cs(j)) \tag{4}$$

Similarly, through Equation (4), the permuted bit planes P_5', P_6', P_7' can be obtained from P_5, P_6, P_7, respectively. Finally, the permuted image P' is obtained though Equation (5)

$$P' = P_1 + 2P_2 + 2^2P_3 + 2^3P_4 + 2^4P_5' + 2^5P_6' + 2^6P_7' + 2^7P_8' \tag{5}$$

where p_1, p_2, p_3, p_4 are the low 4-bit planes, and p_5', p_6', p_7', p_8' are the high 4-bit planes, respectively;

Step 3: Bit-wise XOR diffusion. Firstly, setting x_0 and μ_1 as the initial value and control parameter of the logistic map, respectively, a real matrix $R = \{r(i,j)\}_{i=1,j=1}^{H,W}$ is obtained by iterating the logistic map $H \times W$ times. Then, a mask image $M = \{m(i,j)\}_{i=1,j=1}^{H,W}$ is obtained by Equation (6)

$$m(i,j) = \mathrm{mod}\Big(floor\Big(r(i,j) \times 10^5\Big), 256\Big) \tag{6}$$

Then, through Equation (7), the ciphertext $C = \{c(i,j)\}_{i=1,j=1}^{H,W}$ can be obtained as

$$c(i,j) = p'(i,j) \oplus m(i,j) \tag{7}$$

It can be seen that the key set of the encryption system in [33] is keys = $\{\mu_1, x_n, \mu_2, y_n\}$. If we choose an accuracy of 10^{-14} for the four variables (μ_1, x_0, μ_2, y_0), we obtain a key space of $10^{56} \approx 2^{187}$. As [34–36] pointed out, the effective key space of the image encryption system should be greater than 2^{100} in order to prevent brute force attacks, so the key space of our algorithm is sufficiently large to resist against brute force attacks.

3. Security Analysis of the Original Algorithm and Chosen-Plaintext Attack

Through the security analysis, we found that the encryption system has the following security defects:

(1) The chaotic sequences used for encryption are independent of the plaintext image. In other words, when the keys are fixed, the chaotic sequences used for encryption are unchanged for different plaintext images of the same size;
(2) The diffusion part is too simple, as only XOR diffusion is adopted, in which neither a nonlinear function nor a complicated diffusion mechanism is involved. Therefore, the algorithm is not sensitive to plain images;
(3) Permutation and diffusion are independent of each other, and there is no relationship between them. Therefore, the permutation and diffusion parts of the original algorithm can be deciphered by the strategy of divide and conquer.

From the encryption process of the original algorithm, it can be found that two sequences, RS and CS, are used in the scrambling process, and the chaotic sequence, M, is used in the diffusion phase. Therefore, the equivalent key streams of the original algorithm are M, RS and CS. If the equivalent key streams are cracked, the original encryption system will be cracked.

The so-called chosen-plaintext attack and selective plaintext attack refer to the following process. In addition to not knowing the secret keys used by the cryptosystem, the attacker understands the working mechanism of the encryption algorithm and has the opportunity to use the encryption machine of the cryptosystem. Therefore, the attacker can choose some special plaintext images and obtain the corresponding ciphertext images, thereby deciphering the equivalent secret keys of the cryptosystem or the target ciphertext image.

3.1. Cracking of Equivalent Key M in the Diffusion Phase

For bit-level permutation, if all the bits of the input plaintext image are the same, that is, all are 0 or all are 1, then the corresponding permutation image is exactly the same as the plain image. For example, by choosing the image $P_0 = \{p_0(i,j) = 0\}_{i=1,j=1}^{H,W}$, whose pixel values are all 0, as the input plain image, the result, P_0', after bit-level permutation is exactly the same as the original plain image, that is $P_0' = P_0$. The attacker obtains the cipher image $C_0 = \{c_0(i,j)\}_{i=1,j=1}^{H,W}$ corresponding to P_0. Finally, according to formula (7), the attacker can obtain $M = \{m(i,j)\}_{i=1,j=1}^{H,W}$ as

$$m(i,j) = p'(i,j) \oplus c_0(i,j) = 0 \oplus c_0(i,j) = c_0(i,j) \tag{8}$$

Therefore, the equivalent key M is cracked in the diffusion phase.

3.2. Breaking Bit-Level Permutation

From the permutation of Formula (4), we find that the essence of permutation is to exchange the rows and columns of the bit plane matrix. After permutation, the elements of the same row are still in the same row, and the elements of the same column are in the same column.

For example, taking a plain image, P, of size 4×4 as an example, let RS = [3, 2, 4, 1], CS = [2, 1, 4, 3], and

$$P = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \tag{9}$$

Then, the permuted image, P', is obtained through Formula (4)

$$P' = \begin{bmatrix} 10 & 9 & 12 & 11 \\ 6 & 5 & 8 & 7 \\ 14 & 13 & 16 & 15 \\ 2 & 1 & 4 & 3 \end{bmatrix} \tag{10}$$

In the original algorithm, the bit plane whose element was 0 or 1 is permuted, so the sequences, RS and CS, can be recovered by constructing a special bit plane. Taking the 8-th bit plane as an example, a special plain image is constructed so that its 8-th bit plane has the following form:

$$P_8 = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & 1 & \dots & 1 \end{array}\right]_{H \times W} \tag{11}$$

Because the numbers of element 1 in each row (column) of P_8 are different, the sequences, RS and CS, can be obtained by comparing the numbers of element 1 in each row (column) in the 8-th bit plane P'_8 of the permuted image, P'. Suppose H = 4, W = 4, and then the scrambled P'_8 is

$$P'_8 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Based on P'_8, it can be inferred that RS = [3, 2, 4, 1] and CS = [2, 1, 4, 3].

3.3. Specific Steps of Chosen-Plaintext Attack

The specific steps of our chosen-plaintext attack are as follows:

Step 1: The chosen-plaintext attack means that the attacker has the access right of the encryptor and can construct the ciphertext corresponding to any plaintext. Thus, as shown in Section 3.1, by choosing the all-zero image, P_0, as the input plain image, one gets the corresponding cipher image, C_0, then $M = C_0$;

Step 2: Select a special plaintext image so that its 8-th bit plane has the form of matrix (11) and so the selected plain image can be the following matrix image, PP

$$PP = \left[\begin{array}{ccc:ccc} 128 & 0 & 0 & 0 & \dots & 0 \\ 128 & 128 & 0 & 0 & \dots & 0 \\ 128 & 128 & 128 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 128 & 128 & 128 & 128 & \dots & 128 \end{array}\right]_{H \times W} \tag{12}$$

After encryption, obtain the cipher image, CP, corresponding to PP, then obtain the permuted image PP' of PP is by using the cracked diffusion key M, the elements of which are shown in Equation (13)

$$pp'(i,j) = c_0(i,j) \oplus m(i,j) \tag{13}$$

Step 3: Extract the 8-th bit plane PP'_8 of PP', as shown in Section 3.2. By comparing the numbers of element 1 in each row of PP'_8, the vector RS is obtained. Similarly, the vector, CS, is also obtained by comparing the numbers of element 1 in each column of PP'_8;

Step 4: As for a given cipher image, C, firstly the permuted image, P', is obtained by using the diffusion key M though Equation (14)

$$p'(i,j) = c(i,j) \oplus m(i,j) \tag{14}$$

Extract the 5–8th bit planes of P', then perform reverse permutation on the 5–8th bit planes of P' to obtain the 5–8th bit planes of the plaintext image P by using the sequences CS and RS. The 1–4th bit planes of P are exactly the same as that of P'. In this way, all the eight-bit planes of P are obtained, and then the plain image P can be obtained by Formula (3).

3.4. The Discussion

In [37], the algorithm in [33] is cracked, but for an 8-bit grayscale image of size 256×256, the data complexity of the attack method required for breaking the algorithm is $O(\log_2(H \times W)) = O(19)$, while only one special plaintext image and its corresponding ciphertext image is needed to decode the scrambling sequences, RS and CS, in our method. As such, the complexity of our attack algorithm is greatly reduced.

4. Experimental Simulations of Cracking

The experimental image is an 8-bit grayscale image, Cameraman, of size 256×256. The process is as follows: choose keys $x_0 = 0.8578$, $\mu_1 = 3.6832$, $y_0 = 0.3476$, $\mu_2 = 1.5866$; encrypt the image, Cameraman, size 256×256, with a pixel value of 0, and a special image of size 256×256 as Formula (12) to obtain the corresponding encrypted image, as shown in Figure 3b,d,f. According to Figure 3d, the mask matrix, M, can be decrypted without knowing the key. The permutation sequences, RS and CS, can be decrypted by combining the matrix, M, and Figure 3f. Therefore, all the equivalent keys can be decrypted. Furthermore, any encrypted image can be decrypted by using the equivalent keys. After attacking the encrypted image, Figure 3b, the recovered image is shown in Figure 4.

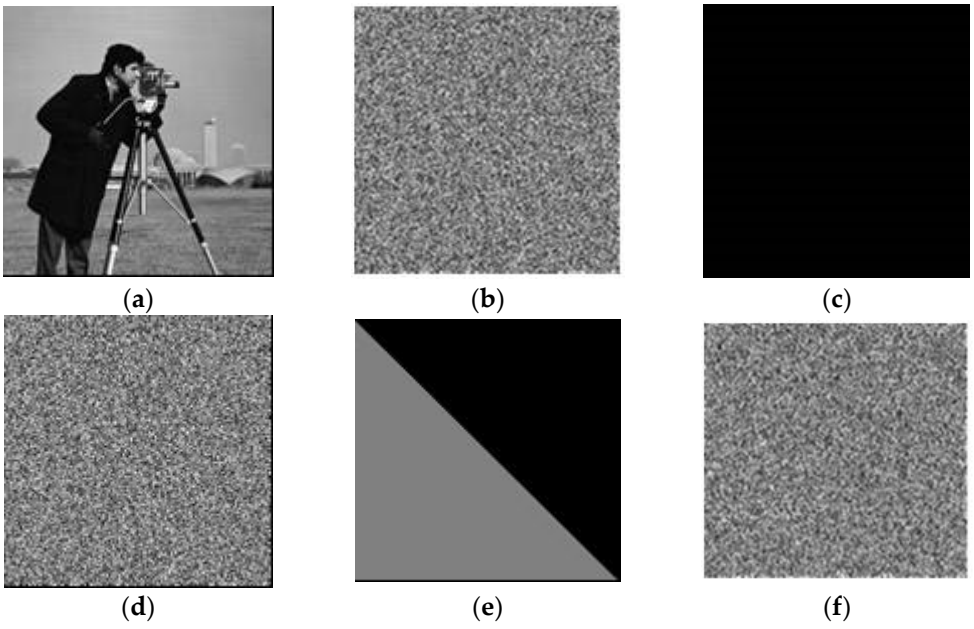


Figure 3. The results of the test images. (**a**) "Cameraman" plain image. (**b**) The encrypted "Cameraman". (**c**) All black plain image. (**d**) Encrypted "black image". (**e**) The special image as Formula (12). (**f**) The cipher image of (**e**).

Figure 4. The decrypted image.

5. The Improved Algorithm and Security Analysis

The main reason why the original algorithm is cracked is that the equivalent keys M, RS and CS of the original encryption algorithm are independent of the plain image. According to the five suggestions given in [37], we propose the improved algorithm of [33]. The key set of the improved algorithm is exactly the same as that of the original algorithm. Compared with the original algorithm of [33], the improved algorithm can resist chosen-plaintext attack and has better security performance.

5.1. The Improved Encryption Algorithm

The specific steps of the improved encryption algorithm are as follows:

Step 1: In the permutation phase, the initial value y_0 of chaotic map (2) is related to the sum of the pixel values of the 8-th bit plane of the plain image

$$sum = \sum_{i=1}^{H} \sum_{j=1}^{W} p_8(i,j) \tag{15}$$

Then, y_0 is updated with sum:

$$y_0 = (\mathrm{mod}(floor(y_0 \times 100 \times sum), W \times H))/(W \times H) \tag{16}$$

In this way, the two permutation sequences, RS and CS, will be related to the plain image. The permutation process is exactly the same as the original algorithm;

Step 2: The permuted image, $P' = \{p'(i,j)\}$, and the real matrix, $R = \{r(i,j)\}$, of the original algorithm are all transformed into one-dimensional vector sequences $P' = \{p'(1),\ p'(2), \cdots,\ p'(H \times W)\}$ and $R = \{\ r(1),\ \ r(2),\ \ \ldots,\ \ r(H \times W)\ \}$, respectively shown from left to right and from top to bottom. The ciphertext sequence $C = \{c(1), c(2), \ldots, c(H \times W)\}$ is generated according to Formulas (17)–(21), then sequence C is transformed into a matrix of size $H \times W$ to obtain the final cipher image;

$$m(1) = \mathrm{mod}\left(floor\left(r(1) \times 10^5\right), 256\right) \tag{17}$$

$$c(1) = p'(1) \oplus m(1) \tag{18}$$

$$m(i) = \mathrm{mod}\left(floor\left(r(i) \times c(i-1) \times 10^5\right), 256\right) \tag{19}$$

$$kt(i) = floor(m(i) \times (i-1)/256) + 1 \tag{20}$$

Obviously, $kt(i) \in [1, i-1]$. $i = 2, 3, 4, \ldots, H \times W$.

$$c(i) = \mathrm{mod}(p'(i) + m(i), 256) \oplus c(kt(i)) \tag{21}$$

It can be seen from Equation (19) that the generation of the key, M, is related to the cipher image, so the key, M, used for encrypting different plaintext is different. Therefore, the improved algorithm can resist a chosen-plaintext attack. Furthermore, the ciphertext

feedback mechanism is adopted in the Formulas (20) and (21), which overcomes the weakness that the original algorithm is not sensitive to the plain image.

5.2. The Improved Decryption Algorithm

The specific steps of the improved decryption algorithm are as follows:

Step 1: Transform the ciphertext image matrix to a sequence $C = \{c(1), c(2), \ldots, c(H \times W)\}$, and the real chaotic sequence $R = \{r(1), r(2), \ldots, r(H \times W)\}$;

Step 2: Decrypt the first pixel value $c(1)$ to obtain the first pixel value $p'(1)$ by

$$\begin{cases} m(1) = \mathrm{mod}(\mathrm{floor}(r(1) \times 10^5), 256) \\ p'(1) = \mathrm{bitxor}(m(1), c(1)) \end{cases} \tag{22}$$

Step 3: Decrypt the i-th pixel value $C(i)$ to obtain the i-th pixel value $P'(i)$ by

$$\begin{cases} m(1) = \mathrm{mod}(\mathrm{floor}(r(i) \times c(i-1) \times 10^5), 256) \\ kt(i) = \mathrm{floor}(m(i) \times (i-1)/256) + 1 \\ p'(i) = \mathrm{mod}(\mathrm{bitxor}(c(i), c(kt(i))) - m(i), 256) \end{cases} \tag{23}$$

where i = 1,2,..., $H \times W$;

Step 4: Transform the 1D sequence P' = [p'(1), p'(2),..., $p'(H \times W)$] to a matrix size of $H \times W$ as $P' = \{p'(\mathrm{i}, \mathrm{j}) \mid i = 1, 2, \ldots, H, j = 1, 2, \ldots, W\}$;

Step 5: Bit plane decomposition. Decompose the intermediate version image $P' = \{p'(i,j)\}_{i=1,j=1}^{H,W}$ with the size of $H \times W$ into 8-bit planes, PP_1 to PP_8, and calculate the sum of pixel values of the 8-th bit plane of the binary image PP_8

$$sum = \sum_{i=1}^{H} \sum_{j=1}^{W} pp_8(i,j) \tag{24}$$

The result of the above equation is equal to the result of Equation (15);

Step 6: Calculate y_0 with Equation (16), and generate the two permutation sequences, *RS* and *CS*, by using the same method as the original algorithm;

Step 7: Perform an inverse scrambling operation on the planes of (PP_5, PP_6, PP_7, PP_8) to obtain the planes of (P_5, P_6, P_7, P_8) by using *RS* and *CS* as:

$p_5(rs(i, j), cs(i, j)) = pp_5(i, j)$, $p_6(rs(i, j), cs(i, j)) = pp_6(i, j)$, $p_7(rs(i, j), cs(i, j)) = pp_7(i, j)$, $p_8(rs(i, j), cs(i, j)) = pp_8(i, j)$. $i = 1, 2, \ldots, H$, $j = 1, 2, \ldots, W$.

Step 8: Combine 8-bit planes to obtain the restored original image, P, by

$$P = 2^7 \times P8 + 2^6 \times P7 + 2^5 \times P6 + 2^4 \times P5 + 2^3 \times PP4 + 2^2 \times PP3 + 2 \times PP2 + PP1 \tag{25}$$

5.3. Analysis of Improved Algorithm to Resist Chosen-Plaintext Attack

The improved algorithm can resist the chosen-plaintext attack, which is reflected in two aspects. Firstly, from Formulas (15) and (16), it can be seen that the scrambling sequences, *RS* and *CS*, produced in the scrambling stage are related to the plaintext image, and the sequences, *RS* and *CS*, used to encrypt the different images are different. Secondly, from Equations (19) and (20), we can see that the generation of m(i) is related to the previous ciphertext value c ($\mathrm{i} - 1$), and the generation of kt(i) is related to m(i). Therefore, the sequences, m(i) and kt(i), used to encrypt different images are different. In short, the improved algorithm has the effect of "one-time pad".

5.4. Comparison of Ciphertext Security Performance between Improved Algorithm and Original Algorithm

In order to further highlight the advantages of the improved algorithm, we will compare it with the original algorithm from the aspects of information entropy, ciphertext correlation analysis and ciphertext sensitivity.

(1) Comparison of Information Entropy

The ideal value of entropy for an 8-bit gray-scale image is 8. The closer the value is to 8, the more uncertain the image is, and the more uniform the distribution of image pixel value is. Table 1 shows the information entropy of the cipher images Rice, Cameraman, Lena and Pepper, which were encrypted by the improved algorithm and the original algorithm. Compared to the original algorithm and other algorithms, the improved algorithm is closer to the ideal situation, that is, the encryption effect of this algorithm is better.

Table 1. Information entropy of ciphertext image.

Images	The Improved Algorithm	The Original Algorithm	Ref. [38]	Ref. [39]
Cameraman	7.9972	7.8716	7.9921	7.9987
Rice	7.9973	7.8921	7.9945	7.9965
Pepper	7.9990	7.8608	7.9934	7.9929
Lena	7.9989	7.8769	7.9968	7.9973

(2) Comparison of Correlation Coefficient

In general, there is a strong correlation between the adjacent pixels of the plaintext image, while the correlation between the adjacent pixels of the ciphertext image is close to zero. Table 2 shows the correlation coefficient of the cipher images of Cameraman and Peppers encrypted by the improved algorithm and the original algorithm in the horizontal direction, the vertical direction and the diagonal direction, respectively.

Table 2. Correlation coefficient analysis of two adjacent pixels.

The Test Image	Direction	The Improved Algorithm	The Original Algorithm
Cameraman	Horizontal	0.0266	0.2237
Cameraman	Vertical	−0.0088	−0.0504
Cameraman	Diagonal	−0.0049	−0.0131
Peppers	Horizontal	0.0078	−0.0490
Peppers	Vertical	0.0148	0.3377
Peppers	Diagonal	0.0113	−0.0318

(3) Comparison of Plaintext Sensitivity

The number of pixels change rate (NPCR) and unified average changing intensity (UACI) are commonly used to measure the sensitivity of encryption algorithms to plaintext. The formulas for the calculation of NPCR and UACI are found in [10]. For the 256 levels of the grayscale images, the expected values of NPCR and UACI are 99.6094% and 33.4635%, respectively.

We have performed 20 groups of tests. In each test, we randomly selected one pixel in the plain image Cameraman, changed its value with 1 bit and encrypted it. Finally, we calculated the NPCR and UACI values between any two pairs of ciphertext image. The results are shown in Table 3. From Table 3, one can see that the NPCR and UACI values are very close to the ideal values in the improved algorithm, while in the original algorithm, the values of NPCR and UACI are close to 0. This is mainly because the original algorithm does not adopt the ciphertext feedback mechanism in the diffusion stage, so the original algorithm is not sensitive to plaintext.

Table 3. Comparison of plaintext sensitivity between the improved algorithm and the original algorithms.

Algorithms	NPCR (%)		UACI (%)	
	Min	Average	Min	Average
The improved algorithm	99.48	99.68	32.43	33.36
The original algorithm	0	0.00001	0	0.0000001
Ref. [10]	99.53	99.67	33.52	33.64

6. Conclusions

In this paper, the security performance of a recent chaotic image encryption cryptosystem based on bit planes extraction and multiple chaotic maps is cryptanalyzed in detail. It is found that the equivalent key streams *M*, *RS* and *CS* can be recovered separately in the scenario of a chosen-plaintext attack. In order to overcome the shortcomings of the original algorithm, which cannot resist the chosen-plaintext attack and is not sensitive to plaintext, we propose an improved encryption algorithm. The innovation of the improved algorithm lies in that the key set of the encryption system is the same as that of the original algorithm, but the equivalent sequences, *M*, *RS* and *CS*, used to encrypt different images, are different, which has the effect of a one-time pad.

The improved algorithm has the advantages of high security and resistance to chosen-plaintext attacks. However, it also has the following defects: from Formulas (19) and (20), we can see that in the encryption process, we need to switch back and forth between the floating-point operation and the integer operation (that is, one floating-point operation, one integer operation, switching back and forth), which is not conducive to hardware implementation. Therefore, it is still necessary to design a secure and efficient image encryption algorithm based on chaos.

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

Funding: This work was supported by the National Natural Science Foundation of China grant number No. 62071496 and in part by the Shan Dong Province Nature Science Foundation grant number No. ZR2017MEM019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The authors are thankful to the reviewers for their comments and suggestions to improve the quality of the manuscript.

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

References

1. Liu, M.Z.; Zhao, F.X.; Jiang, X.; Liu, X.H.; Liu, Y.N. A Novel Image Encryption Algorithm Based on Plaintext-related Hybrid Modulation Map. *J. Internet Technol.* **2019**, *20*, 2141–2155. [CrossRef]
2. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [CrossRef]
3. Chen, G.R.; Mao, Y.B.; Chui, C.K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761. [CrossRef]
4. Lian, S.G.; Sun, J.S.; Wang, Z.Q. A block cipher based on a suitable use of the chaotic standard map. *Chaos Solitons Fractals* **2005**, *26*, 117–129. [CrossRef]
5. Wong, K.W.; Kwok, B.S.H.; Law, W.S. A fast image encryption scheme based on chaotic standard map. *Phys. Lett. A* **2008**, *372*, 2645–2652. [CrossRef]
6. Chai, X.; Fu, X.; Gan, Z.; Lu, Y.; Chen, Y. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process.* **2019**, *155*, 44–62. [CrossRef]
7. Wang, X.; Zhao, H.; Feng, L.; Ye, X.; Zhang, H. High-sensitivity image encryption algorithm with random diffusion based on dynamic-coupled map lattices. *Opt. Lasers Eng.* **2019**, *122*, 225–238. [CrossRef]
8. Li, H.J.; Wang, Y.R.; Zuo, Z.W. Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms. *Opt. Lasers Eng.* **2019**, *115*, 197–207. [CrossRef]
9. Zhu, S.; Zhu, C. Plaintext-Related Image Encryption Algorithm Based on Block Structure and Five-Dimensional Chaotic Map. *IEEE Access* **2019**, *7*, 147106–147118. [CrossRef]
10. Zhu, S.; Zhu, C.; Wang, W. A New Image Encryption Algorithm Based on Chaos and Secure Hash SHA-256. *Entropy* **2018**, *20*, 716. [CrossRef] [PubMed]

11. Chen, J.X.; Zhu, Z.L.; Fu, C.; Yu, H.; Zhang, L.B. An efficient image encryption scheme using gray code based permutation approach. *Opt. Lasers Eng.* **2015**, *67*, 191–204. [CrossRef]
12. Chen, J.X.; Zhu, Z.L.; Fu, C.; Yu, H. Optical image encryption scheme using 3-D chaotic map based joint image scrambling and random encoding in gyrator domains. *Opt. Commun.* **2015**, *341*, 263–270. [CrossRef]
13. Dhall, S.; Pal, S.K.; Sharma, K. Cryptanalysis of image encryption scheme based on a new 1D chaotic system. *Signal Process.* **2019**, *146*, 22–32. [CrossRef]
14. Belazi, A.; Talha, M.; Kharbech, S.; Xiang, W. Novel medical image encryption scheme based on chaos andDNA encoding. *IEEE Access* **2019**, *7*, 36667–36681. [CrossRef]
15. Tlelo-Cuautle, E.; Díaz-Muñoz, J.D.; González-Zapata, A.M.; Li, R.; León-Salas, W.D.; Fernández, F.V.; Cruz-Vega, I. Chaotic Image Encryption Using Hopfield and Hindmarsh–Rose Neurons Implemented on FPGA. *Sensors* **2020**, *20*, 1326. [CrossRef]
16. García-Guerrero, E.E.; Inzunza-González, E.; López-Bonilla, O.R.; Cárdenas-Valdez, J.R.; Tlelo-Cuautle, E. Randomness improvement of chaotic maps for image encryption in a wireless communication scheme using PIC-microcontroller via Zigbee channels. *Chaos Solitons Fractals* **2020**, *133*, 109646. [CrossRef]
17. Wang, T.; Wang, M.-h. Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. *Opt. Laser Technol.* **2020**, *132*, 106355. [CrossRef]
18. Cai, S.; Huang, L.; Chen, X.; Xiong, X. A Symmetric Plaintext-Related Color Image Encryption System Based on Bit Permutation. *Entropy* **2018**, *20*, 282. [CrossRef]
19. Zhu, S.; Zhu, C. Image encryption algorithm with an avalanche effect based on a six-dimensional discrete chaotic system. *Multimed. Tools Appl.* **2018**, *77*, 29119–29142. [CrossRef]
20. Huang, L.; Cai, S.; Xiao, M.; Xiong, X. A Simple Chaotic Map-Based Image Encryption System Using Both Plaintext Related Permutation and Diffusion. *Entropy* **2018**, *20*, 535. [CrossRef]
21. Lin, C.Y.; Wu, J.L. Cryptanalysis and Improvement of a Chaotic Map-Based Image Encryption System Using Both Plaintext Related Permutation and Diffusion. *Entropy* **2020**, *22*, 589. [CrossRef]
22. Diab, H.; El-semary, A.M. Cryptanalysis and improvement of the image cryptosystem reusing permutation matrix dynamically. *Signal Process.* **2018**, *148*, 172–192. [CrossRef]
23. Chen, J.X.; Zhu, Z.L.; Fu, C.; Yu, H.; Zhang, Y.S. Reusing the permutation matrix dynamically for efficient image cryptographic algorithm. *Signal Process.* **2015**, *111*, 294–307. [CrossRef]
24. Liu, L.; Hao, S.; Lin, J.; Wang, Z.; Hu, X.; Miao, S. Image block encryption algorithm based on chaotic maps. *IET Signal Process.* **2018**, *12*, 22–30. [CrossRef]
25. Ma, Y.; Li, C.; Ou, B. Cryptanalysis of an image block encryption algorithm based on chaotic maps. *J. Inf. Secur. Appl.* **2020**, *54*, 102566.
26. Zhu, C.; Liao, C.; Deng, X. Breaking and improving an image encryption scheme based on total shuffling scheme. *Nonlinear Dyn.* **2013**, *71*, 25–34. [CrossRef]
27. Zhu, C.; Wang, G.; Sun, K. Improved Cryptanalysis and Enhancements of an Image Encryption Scheme Using Combined 1D Chaotic Maps. *Entropy* **2018**, *20*, 843. [CrossRef]
28. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [CrossRef]
29. Çavuşoğlu, Ü.; Kaçar, S.; Pehlivan, I.; Zengin, A. Secure image encryption algorithm design using a novel chaos based S-Box. *Chaos Solitons Fractals* **2017**, *95*, 92–101. [CrossRef]
30. Wang, X.; Çavuşoğlu, Ü.; Kacar, S.; Akgul, A.; Pham, V.-T.; Jafari, S.; Alsaadi, F.; Nguyen, X. S-Box Based Image Encryption Application Using a Chaotic System without Equilibrium. *Appl. Sci.* **2019**, *9*, 781. [CrossRef]
31. Zhu, C.X.; Wang, G.J.; Sun, K.H. Cryptanalysis and Improvement on an Image Encryption Algorithm Design Using a Novel Chaos Based S-Box. *Symmetry* **2018**, *10*, 399. [CrossRef]
32. Lu, Q.; Zhu, C.; Deng, X. An Efficient Image Encryption Scheme Based on the LSS Chaotic Map and Single S-Box. *IEEE Access* **2020**, *8*, 25664–25678. [CrossRef]
33. Shafique, A.; Shahid, J. Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. *Eur. Phys. J. Plus* **2018**, *133*, 1–16. [CrossRef]
34. Li, Z.; Peng, C.; Tan, W.; Li, L. An Efficient Plaintext-Related Chaotic Image Encryption Scheme Based on Compressive Sensing. *Sensors* **2021**, *21*, 758. [CrossRef] [PubMed]
35. Li, S.; Chen, G.; Mou, X. On the dynamical degradation of digital piecewise linear chaotic maps. *Int. J. Bifurc. Chaos* **2005**, *15*, 3119–3151. [CrossRef]
36. Curiac, D.; Iercan, D.; Dranga, O.; Dragan, F.; Banias, O. Chaos-Based Cryptography: End of the Road? In Proceedings of the International Conference on Emerging Security Information, Systems, and Technologies (SECUREWARE 2007), Valencia, Spain, 14–20 October 2007; pp. 71–76. [CrossRef]
37. Wen, H.; Yu, S. Cryptanalysis of an image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. *Eur. Phys. J. Plus* **2019**, *134*, 337. [CrossRef]

38. Zhang, Y.; Tang, Y. A plaintext-related image encryption algorithm based on chaos. *Multimed. Tools Appl.* **2018**, *77*, 6647–6669. [CrossRef]
39. Xu, L.; Gou, X.; Li, Z.; Li, J. A novel chaotic image encryption algorithm using block scrambling and dynamicindex based diffusion. *Opt. Lasers Eng.* **2017**, *91*, 41–52. [CrossRef]