IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 25, 2018, accepted August 20, 2018, date of publication August 24, 2018, date of current version September 21, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2867111

Cryptanalyzing a Color Image Encryption Scheme Based on Hybrid Hyper-Chaotic System and Cellular Automata

MING LI[1], DANDAN LU[1], WENYING WEN[2], HUA REN[1], AND YUSHU ZHANG[3,4], (Member, IEEE)

[1]College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
[2]School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China
[3]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[4]School of Information Technology, Deakin University, Geelong, VIC 3125, Australia

Corresponding author: Yushu Zhang (yushuboshi@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602158, in part by the Science Foundation for the Excellent Youth Scholars of Henan Normal University under Grant YQ201607, in part by the Science and Technology Research Project of Henan Province under Grant 182102210374, in part by the Natural Science Foundation of Jiangxi Province under Grant 20171BAB202015, and in part by the Research Foundation of the Education Department of Jiangxi Province under Grant GJJ170322.

ABSTRACT Recently, a new color image encryption scheme based on hybrid hyper-chaotic system and cellular automata was proposed. In order to generate different key streams according to different plaintexts, the sum of pixel values of each component of the original color image apart from the secret keys is used to determine the initial value of logistic map for encryption. It was claimed that the scheme can resist kinds of attacks. However, this paper found out three security drawbacks about the original encryption scheme and proposed an effective attack method using chosen-plaintext attack by cryptanalysis. The equivalent permutation key stream can be obtained by adjusting individual pixel values of the chosen plaintexts while keeping the sum of pixel values of each color channel unchanged, and all 256 possibilities of equivalent diffusion key streams can be obtained by using 512 specific chosen plaintexts. In addition, the main stages of the proposed attack method, including breaking diffusion and breaking permutation, are exchangeable. The effectiveness of our method is supported by both theoretical analyses and experimental results.

INDEX TERMS Chaotic image encryption, cryptanalysis, cellular automata, chosen-plaintext attack.

I. INTRODUCTION

Nowadays, a large number of users exploit Internet for communicating, shopping and finding information, and massive data are generated each day, in which the digital images have a large proportion. The huge amount of image data need to be protected against malicious manipulation. Encryption technology can turn a meaningful image into a noise-like image for protecting the confidentiality of the image in storing and transmission on the open and shared networks. It is well known that chaotic map is a powerful tool for image encryption since there are many similar intrinsic features between chaos and cryptography, such as ergodicity, sensitivity to initial condition and control parameters, etc. Compared with the traditional encryption schemes, the chaotic image encryption algorithms have the advantages of high security, high efficiency, reasonable calculation cost [1]–[3]. Therefore, the application of chaos in the field of image encryption has been a hotspot [4]–[12]. In recent researches, most chaotic encryption algorithms [13]–[25] are designed based on Fridrich's structure of permutation-substitution [9].

On the other hand, cryptanalysis which aims to promote the perfection of image cryptosystems verified that some encryption schemes were not secure enough to resist attacks [26]–[37]. For example, the Fridrich's scheme has been attacked by chosen-plaintext attack (CPA) in [26]. A cryptographic analysis has been done in [27] to crack an encryption algorithm based on auto-blocking and electrocardiography, which uses Logistic map and 2-D Arnold map to generate pseudo random number sequences. Diab and El-Semary [28], an efficient cryptographic algorithm by reusing the permutation matrix dynamically based on Baker map has been broken . The security of a novel color image encryption algorithm based on rectangular transformed 2-D

Arnold map has been analyzed and attacked in [29]. Ahmad *et al.* [30] have broken an image encryption algorithm based on the piecewise linear chaotic map (PWLCM) and inertial delayed neural network. An image encryption algorithm was cryptanalyzed and improved by Wang *et al.* [31] which utilizes DNA addition to scramble pixel values of image and then masks the scrambled image with two logistics maps. In [32], a color image encryption scheme based on skew tent map and hyper chaotic system of 6th-order cellular neural network (CNN) was cryptanalyzed. In these schemes, the used Logistic map, 2D Arnold map, skew tent map, Baker map or even PWLCM, are all 1D or 2D chaotic systems. Some researchers pointed out that the security of the image encryption schemes using low-dimensional chaotic systems is not enough, and the image cryptosystems based on hyper-chaotic systems have strong robustness to resist more attacks because they have larger key space and higher key sensitivity [13]–[16], [19]–[22].

In the current research of image encryption, scholars often combine the hyper-dimension chaos with other techniques to obtain a more ideal encryption effect. Wu *et al.* [13] by combining a hyper-chaotic system with the 2D discrete wavelet transform (DWT) technique, proposed a lossless encryption scheme for color images. Two color image encryption algorithms based on DNA sequence operation and different hyper-chaotic systems were introduced in [14] and [19]. In [20], Dong designed a color image encryption scheme based on the 3-D Rabinovich chaotic system, and the one-time keys are generated by both the initial keys and the hash value of the plain image. In [21], a color image encryption algorithm based on a complex chaotic system which combines two high-dimensional chaotic systems was proposed. However, the hyper-chaos-based cryptosystems are still not secure enough. Özkaynak [35], a new analysis roadmap proposed shows a checklist including 12 steps for testing security of an encryption method. Fatih said that many hyperchaos-based proposals involve various weaknesses because the steps of the checklist have not been followed. This checklist can test most plaintext-irrelated encryption schemes including those mentioned above. However, it does not work with the plaintext-related encryption algorithms.

In the latest Niyat *et al.* [22] firstly combine Cellular Automata (CA) with hyper-dimensional chaotic systems in the color encryption. CA, which has a simple structure, random performance and complex behavior, is easy for hardware/software implementation [22], [38]–[40]. The cryptosystem [22] uses the structure of permutation-diffusion. In the permutation stage, a three-dimensional Arnold map was employed to break the spatial correlation of the color image by rearranging the rows and columns of each color component. In the diffusion phase, a one-dimensional Logistic map and a one-dimensional non-uniform CA was used to create an image key, then hyper-chaotic Chen system was used to select image key values to encrypt each components of color image. It was claimed that the approach is robust against different kinds of attacks, because the initial value of Logistic map is plaintext-related, that is, it is determined by the sum of pixel values of each component of the original color image. However, in this paper, we found that there are three defects in the original scheme by theoretical analysis and experimental verification. Also, we proposed an attack scheme based on CPA. The effectiveness of the attack is supported by theoretical analysis and simulation experiments.

The rest of the paper is organized as follows. Section II introduces the original color image encryption scheme. Three security vulnerabilities of the original scheme are analyzed in Section III. Section IV gives a commutable attack scheme and the corresponding experimental results. Conclusion is drawn in the last section.

II. REVIEW OF THE COLOR IMAGE CRYPTOSYSTEM

A. THE THREE CHAOTIC MAPS

In [22], three chaotic systems, which comprised of one-dimensional Logistic chaotic map, 3-D cat map and Chen-hyper chaotic system, are used to generate the key streams.

The Logistic map is used to initialize cellular automata. It is described as

$$X_{n+1} = \alpha X_n(1 - X_n). \tag{1}$$

Parameter α and initial value of X_0 are considered as secret keys. For $\alpha \in (3.57, 4)$ and $X_0 \in (0, 1)$, the Logistic map is chaotic. The initial value X_0 used in the scheme is changed with the different plaintext, that is, X_0 is not a fixed value, while the parameter α is a fixed value.

The 3-D Arnold cat system is used to generate six pseudo-random sequences for permuting the rows and columns of each color component in permutation phase of the original scheme. This map is governed by

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod 1, \tag{2}$$

where A, as shown at the bottom of this page, a_x, a_y, a_z, b_x, b_y and b_z are six control parameters of 3-D cat map. When $a_x = b_x = 1$, $a_y = b_y = 2$, $a_z = b_z = 3$, the map of (2) has chaotic behavior. In the encryption scheme, the six control parameters and the initial values are all fixed.

$$\mathbf{A} = \begin{bmatrix} 1 + a_x a_z b_y & a_z & a_z + a_x a_z + a_x a_y a_z b_y \\ b_z + a_x b_y + a_x a_z b_y b_z & a_z b_z + 1 & a_y b_z + a_x a_y a_z b_y b_z + a_x a_z b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & a_x a_y b_x b_y + a_x b_x + a_y b_y + 1 \end{bmatrix}$$

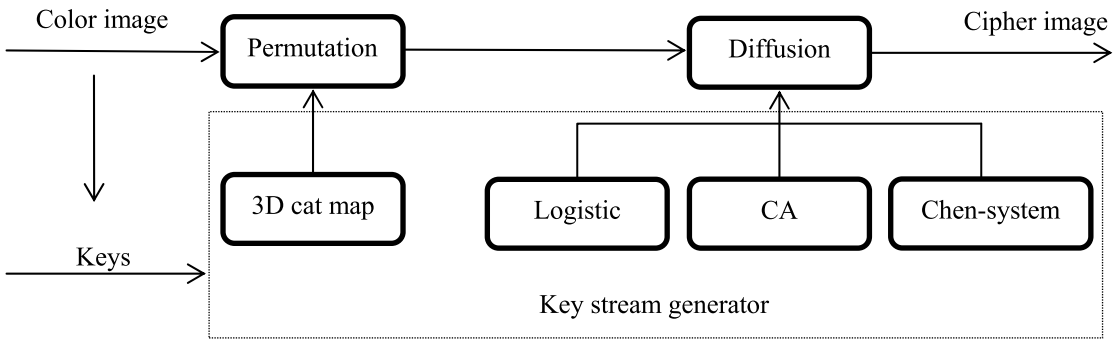

FIGURE 1. The sketch of the original scheme.

Chen-system used to select image key values to encrypt each component of color image is described as

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - xz + cy \\ \dot{z} = xy - bz, \end{cases} \tag{3}$$

where a, b and c are parameters of the Chen-system. When setting $a = 35$, $b = 3$ and $c \in [20, 28.4]$, the system is chaotic. The initial and parameter values of the Chen system are fixed values.

B. CELLULAR AUTOMATA

CA is a discrete time model composed of a grid of cells with a finite number of states, such as on and off states. In a one-dimensional CA, each cell has two adjacent cells, each of which has two possible values, that is, 0 and 1. Thus, there are $2 \times 2 \times 2 = 2^3$ possible binary states for the three neighboring cells. Also, for an 8 bit binary CA, there are $2^8 = 256$ total cellular automata. Let S_i denotes the current state of the i^{th} cell at time t, and f is a Boolean state function that specifies the local rule. At time $t + 1$, the next state value of S_i is produced by the rule f:

$$S_i^{t+1} = f(S_{i-1}^t, S_i^t, S_{i+1}^t). \tag{4}$$

If all CA cells obey the same rule, the CA is said to be uniform; otherwise, it is non-uniform. The CA in the original design has 8 cells which are generated by the logistic chaotic map. The image key is produced according to the eight rules shown in TABLE 1.

C. THE ENCRYPTION SCHEME

The sketch of the original scheme [22] is shown in Fig. 1. The color image encryption scheme can be described as follows.

Step1: The input is a color image $\mathbf{P}(M, N, 3)$ where M and N denote the rows and columns of the image, respectively.

Step2: The color image is divided into three components, red, green and blue, denoted by $\mathbf{P}_\mathrm{R}$, $\mathbf{P}_\mathrm{G}$ and $\mathbf{P}_\mathrm{B}$ respectively.

Step3: By using the 3-D cat map, three chaos sequences $\{xr\}, \{yg\}, \{zb\}$ of length M and three chaos sequences $\{rr\}$, $\{sg\}$, $\{tb\}$ of length N are generated. Initial values of chaotic 3-D cat map are shown in TABLE 2. The six sequences are sorted by

$$\begin{cases} [f1, hr] = sort(xr) \\ [f2, hg] = sort(yg) \\ [f3, hb] = sort(zb), \end{cases} \tag{5}$$

$$\begin{cases} [f4, lr] = sort(rr) \\ [f5, lg] = sort(sg) \\ [f6, lb] = sort(tb), \end{cases} \tag{6}$$

Then, use the sorted sequences $\{hr\}$, $\{hg\}$, $\{hb\}$ and $\{lr\}$, $\{lg\}$, $\{lb\}$ to permute all of the pixels of $\mathbf{P}$ according to (7) for the rows and (8) for the columns of each color component

$$\begin{cases} \mathrm{P}_\mathrm{R}^\mathrm{R}(i, j) = \mathrm{P}_\mathrm{R}(hr(i), j) \\ \mathrm{P}_\mathrm{G}^\mathrm{R}(i, j) = \mathrm{P}_G(hg(i), j) \\ \mathrm{P}_\mathrm{B}^\mathrm{R}(i, j) = \mathrm{P}_B(hb(i), j), \end{cases} \tag{7}$$

TABLE 1. The eight rules of CA in the scheme.

R_i	Rule number	Boolean function
1	30	$S_i^{t+1} = S_{i-1}^t xor\left[S_i^t or S_{i+1}^t\right]$
2	90	$S_i^{t+1} = S_{i-1}^t xor S_{i+1}^t$
3	150	$S_i^{t+1} = S_{i-1}^t xor S_i^t xor S_{i+1}^t$
4	153	$S_i^{t+1} = S_i^t xnor S_{i+1}^t$
5	165	$S_i^{t+1} = S_{i-1}^t xnor S_{i+1}^t$
6	86	$S_i^{t+1} = \left[S_{i-1}^t nor S_i^t\right] xor\left[not(S_i^t)\right]$
7	105	$S_i^{t+1} = not\left[S_{i-1}^t xor S_i^t xor S_{i+1}^t\right]$
8	101	$S_i^{t+1} = \left[S_{i-1}^t nor S_{i+1}^t\right] or\left[S_i^t xor(S_{i+1}^t) and S_{i-1}^t\right]$

TABLE 2. The initial values of chaotic 3D cat map.

xr_0	yg_0	zb_0
0.98146532417548	0.37845691433245	0.334213224587656
rr_0	zg_0	tb_0
0.85245635768455	0.36925874135788	0.98765432115966

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2867111, IEEE Access

$$\begin{cases} \mathrm{P}_{\mathrm{R}}^{\mathrm{RC}}(i,j) = \mathrm{P}_{\mathrm{R}}^{\mathrm{R}}(i, lr(j)) \\ \mathrm{P}_{\mathrm{G}}^{\mathrm{RC}}(i,j) = \mathrm{P}_{G}^{\mathrm{R}}(i, lg(j)) \\ \mathrm{P}_{\mathrm{B}}^{\mathrm{RC}}(i,j) = \mathrm{P}_{B}^{\mathrm{R}}(i, lb(j)), \end{cases} \tag{8}$$

where $i = 1, 2, \ldots, \mathrm{M}$ and $j = 1, 2, \ldots, \mathrm{N}$.

Step4: Logistic map with the initial value X_0 and the parameter value $\alpha = 4$ are iterated 200 times to initialize the CA, and then a key stream **image_key** of size $\mathrm{M} \times \mathrm{N}$ is created according to the eight rules in TABLE I. The initial value of X_0 is calculated by

$$X_0 = k_0 + k_1 + k_2 + k_3, \tag{9}$$

where k_0, k_1, k_2, k_3 can be calculated by using

$$\begin{cases} k_0 \in \{\mathrm{mod}([0, 1, \ldots, 128], 256\} \\ k_1 = \dfrac{1}{256} \mathrm{mod}\left(\sum_{i=1}^{M}\sum_{j=1}^{N} \mathrm{P_R}(i,j), 256\right) \\ k_2 = \dfrac{1}{256} \mathrm{mod}\left(\sum_{i=1}^{M}\sum_{j=1}^{N} \mathrm{P_G}(i,j), 256\right) \\ k_3 = \dfrac{1}{256} \mathrm{mod}\left(\sum_{i=1}^{M}\sum_{j=1}^{N} \mathrm{P_B}(i,j), 256\right), \end{cases} \tag{10}$$

where $\mathrm{P_{R,G,B}}(i,j)$ are the values of the pixels of each color component of image in the position (i,j), M is the width of the image and N is the height. If $X_0 > 1$, $X_0 = \mathrm{mod}\ (X_0, 1)$.

Step5: After iterating (3) for n times, the iteration is continued for $\mathrm{M} \times \mathrm{N}$ times to get three sequences of $\{x\}$, $\{y\}$, $\{z\}$. By permuting the **image_key** based on $\{x\}$, $\{y\}$, $\{z\}$, one can get $\mathbf{image\_key}_{1,2,3}$ for encrypting the red, green and blue components of $\mathbf{P}_{\mathrm{R,G,B}}^{\mathrm{RC}}$ image, and $\mathbf{C_R}$, $\mathbf{C_G}$, $\mathbf{C_B}$ are obtained by

$$\begin{cases} \mathbf{C}_R = \mathbf{P}_{\mathrm{R}}^{\mathrm{RC}} \oplus \mathbf{image\_key}_1 \\ \mathbf{C}_G = \mathbf{P}_{\mathrm{G}}^{\mathrm{RC}} \oplus \mathbf{image\_key}_2 \\ \mathbf{C}_B = \mathbf{P}_{\mathrm{B}}^{\mathrm{RC}} \oplus \mathbf{image\_key}_3, \end{cases} \tag{11}$$

Step6: $\mathbf{C_R}$, $\mathbf{C_G}$ and $\mathbf{C_B}$ are combined and then the encrypted image $\mathbf{C}$ is obtained.

III. SECURITY VULNERABILITIES OF THE ORIGINAL SCHEME

The cryptosystem [22] uses the structure of permutation-diffusion. It was claimed that the approach is robust against different kinds of attacks. However, three security vulnerabilities of the scheme are found and verified by analysis. The drawbacks are shown as follows.

A. SECURITY VULNERABILITY 1

Generally, the encryption scheme needs to follow the confusion-diffusion principle of Shannon, in which the diffusion refers to the fact that changing one bit of the plaintext can cause a widespread change in the ciphertext image by at least 50%. A relationship between the secret key and the image

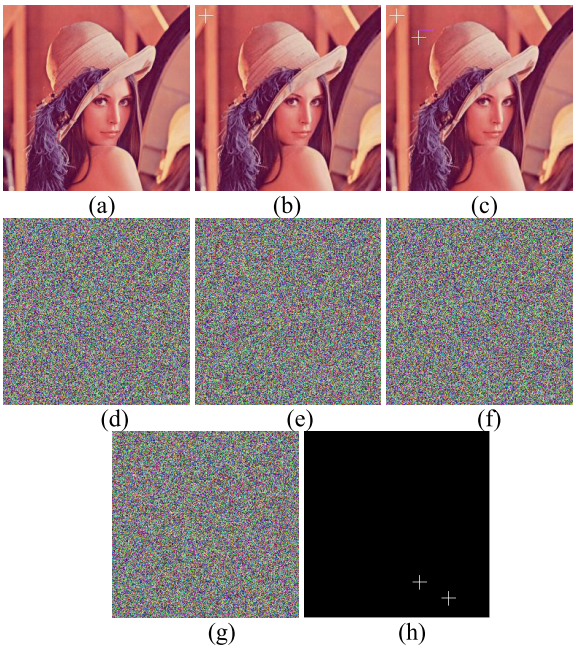

FIGURE 2. Diffusion in success and failure situations: (a) plain-image 'Lena'; (b) one changed pixel of plain-image 'L1'; (c) two changed pixels of plain-image 'L2'; (d) cipher-image corresponding to 'Lena'; (e) cipher-image corresponding to'L1'; (f) cipher-image corresponding to 'L2'; (g) XOR between (d) and (e); (h) XOR between (d) and (f).

plaintext was established in Step 4 of the original scheme to achieve diffusion effect.

However, it was found by analyzing (10) that when the changes of $k_{1,2,3}$ can offset each other, the diffusion effect of the scheme would be eliminated. For example, if any two pixels of a plaintext image $\mathbf{P}$, denoted by $\mathrm{P}(i_1, j_1)$ and $\mathrm{P}(i_2, j_2)$, are changed to get $\mathbf{P}'$ by $\mathrm{P}(i_1, j_1)$ plus ∂ and $\mathrm{P}(i_2, j_2)$ minus ∂. It is apparent that the initial value X_0 of Logistic map has no change when $\mathbf{P}$ was transformed into $\mathbf{P}'$. Therefore, the **image_key** created in Step 4 of the original scheme gets no change as well. Finally, $\mathbf{image\_key}_{1,2,3}$ selected from the **image_key** by Chen-system with the fixed parameter and initial values also gets no alternate. Consequently, there are only two different pixel values for the two cipher images $\mathbf{C}$ and $\mathbf{C}'$ obtained by encrypting $\mathbf{P}$ and $\mathbf{P}'$ respectively.

The diffusion in a failure situation can be verified experimentally. Firstly, copy the plaintext 256×256 '**Lena**' into duplicate images $\mathbf{L}_1$ and $\mathbf{L}_2$. Then any one pixel of $\mathbf{L}_1$ is changed by

$$\mathrm{L}_1(i_1, j_1) = \mathrm{mod}(\mathrm{L}_1(i_1, j_1) - \sigma, 256), \tag{12}$$

and any two pixels of $\mathbf{L}_2$ are changed by

$$\begin{cases} \mathrm{L}_2(i_1, j_1) = \mathrm{mod}(\mathrm{L}_2(i_1, j_1) + \sigma, 256) \\ \mathrm{L}_2(i_2, j_2) = \mathrm{mod}(\mathrm{L}_2(i_2, j_2) - \sigma, 256). \end{cases} \tag{13}$$

Next, cipher images $\mathbf{C}$, $\mathbf{C}_1$ and $\mathbf{C}_2$ are obtained by encrypting '**Lena**', $\mathbf{L}_1$ and $\mathbf{L}_2$ respectively. Finally, the result of $\mathbf{C}$ XOR $\mathbf{C}_1$ is Y_1, and similarly, $\mathbf{C}$ XOR $\mathbf{C}_2$ is Y_2. Experimental process is shown in Fig 2. As we can see, Y_1 has

TABLE 3. Diffusion in success and failure situations.

Plain-image	Change Plain-image	X_0	Diffusion
Lena	$L_1(1,3) = L_1(1,3) + 1;$	Change	Success
Lena	$\begin{cases} L_2(1,1) = L_2(1,1) + 1; \\ L_2(1,2) = L_2(1,2) - 1; \end{cases}$	Un-change	Fail
Fruits	$\begin{cases} F_1(2,1) = F_1(2,1) + 3; \\ F_1(2,2) = F_1(2,2) - 1; \\ F_1(2,3) = F_1(2,3) - 2; \end{cases}$	Un-change	Fail

some noise pixels, so there are obviously more than 50% different pixels between $\mathbf{C}$ and $\mathbf{C}_1$. While Y_2 has only two nonzero values, there are only two different pixels between $\mathbf{C}$ and $\mathbf{C}_2$.

In order to avoid accidental events, we have done a lot of experiments to verify that the scheme has this security vulnerability. Here is the main data shown in TABLE 3. A conclusion can be obtained from TABLE 3 that when the change of plain-image affects the value of X_0, the **image_key** generated in the diffusion phase get change as well, therefore the cryptosystem has diffusion effect; otherwise when the change of plain-image does not affect the value of X_0, the **image_key** gets no change, and the diffusion of the scheme fails.

B. SECURITY VULNERABILITY 2

In the permutation phase of the scheme, the three-dimensional Arnold mapping was used to create six pseudo-random sequences which were employed to rearrange the rows and columns of each color component. In the diffusion phase, the one-dimensional Logistic map and a one-dimensional non-uniform CA were used to create an **image_key**, then hyper-chaotic Chen system was used to obtain $\mathbf{image\_key}_{1,2,3}$ from **image_key** by permutation for encrypting each component of color image.

However, the initial value and parameter of the hyper-chaotic map is fixed, that is, the generated chaotic key-streams will keep unchanged with the different plain images.

Consequently, there is a one-to-one map of pixel locations between plain image and cipher image, and there is a bijective relationship between $\mathbf{image\_key}_{1,2,3}$ and **image_key**. Suppose a known gray image $\mathbf{P}$ are denoted by

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & p_{m3} & \cdots & p_{mn} \end{bmatrix}, \quad (14)$$

We change the two values in the first row of $\mathbf{P}$ to obtain $\mathbf{P}_\mathrm{m}$ as

$$\mathbf{P}_\mathrm{m} = \begin{bmatrix} p_{11}+1 & p_{12}-1 & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & p_{m3} & \cdots & p_{mn} \end{bmatrix}, \quad (15)$$

and ensure X_0 unchanged, then encrypt $\mathbf{P}$ and $\mathbf{P}_\mathrm{m}$ to obtain $\mathbf{C}$ and $\mathbf{C}_\mathrm{m}$ respectively and execute $\mathbf{C}$ XOR $\mathbf{C}_\mathrm{m}$ to obtain $\mathbf{Y}_\mathrm{m}$. Obviously, there are two non-zero values in $\mathbf{Y}_\mathrm{m}$ based on Security vulnerability 1. We can get the rank value of two non-zero values. Thus, the corresponding row's location in cipher image of $\mathbf{P}(1;:)$ can be obtained. For other rows or columns, the situation is the same. This leads to the vulnerability of CPA.

C. SECURITY VULNERABILITY 3

The Logistic map coupled with a one-dimensional non-uniform CA was used to create an **image_key** for encrypting a plain image as shown in the step 4 of the diffusion phase. The initial value X_0 of Logistic map depends on each component of the color image according to (9) and (10). That is, the value X_0 isn't fixed, and the **image_key** will change with X_0.

However, the number of possible values of X_0 with the different plain images is finite according to (9) and (10). Firstly, the value of k_0 is fixed and selected from

$$\left\{ \frac{0}{256}, \frac{1}{256}, \cdots, \frac{128}{256} \right\}.$$

Furthermore, the values of k_1, k_2 and k_3 all belong to

$$\left\{ \frac{0}{256}, \frac{1}{256}, \cdots, \frac{255}{256} \right\},$$

since all image pixel values are integers. Thus

$$\{k_0 + k_1 + k_2 + k_3\} = \left\{ \frac{0}{256}, \frac{1}{256}, \cdots, \frac{255 * 3 + 128}{256} \right\}.$$

And the Logistic map has chaotic behavior when $X_0 \in (0, 1)$. Finally, the number of possible values of X_0 is 256 in total, that is

$$X_0 \in \left\{ \frac{0}{256}, \frac{1}{256}, \frac{2}{256}, \cdots, \frac{255}{256} \right\}.$$

Consequently, the number of possibilities of **image_key** is 256 totally from what has been discussed above. As stated before, there is a bijective relation between $\mathbf{image\_key}_{1,2,3}$ and **image_key**; therefore, the number of $\mathbf{image\_key}_{1,2,3}$ is also 256 totally. We chose a plain image $\mathbf{O}_r^\mathrm{R}$ with all 0 pixels

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2867111, IEEE Access

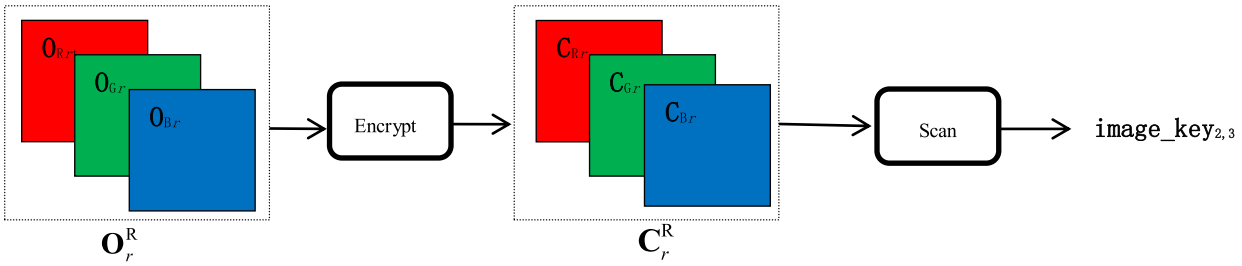

FIGURE 3. The flowchart of obtaining image_key$_{2,3}$.

except $\mathrm{O}_{\mathrm{R}r}(1,1) = r$, $r \in [0, 255]$, which is denoted by

$$\mathbf{O}_r^{\mathrm{R}} = \begin{cases} \mathbf{O}_{\mathrm{R}r} = \begin{bmatrix} r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{m\times n} \\ \mathbf{O}_{\mathrm{G}r} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{m\times n} \\ \mathbf{O}_{\mathrm{B}r} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{m\times n,} \end{cases} \tag{16}$$

and encrypt $\mathbf{O}_r^{\mathrm{R}}$ to get cipher image $\mathbf{C}_r^{\mathrm{R}}$. Then, $\mathbf{C}_r^{\mathrm{R}}$ is divided into red, green and blue components, i.e., $\mathbf{C}_{\mathrm{R}r}$, $\mathbf{C}_{\mathrm{G}r}$ and $\mathbf{C}_{\mathrm{B}r}$, in which $\mathbf{C}_{\mathrm{R}r}$ is equivalent to **image_key**$_1$ except the only non-zero r, and $\mathbf{C}_{\mathrm{G}r}$ and $\mathbf{C}_{\mathrm{B}r}$ are exactly equivalent to **image_key**$_2$ and **image_key**$_3$ respectively, because of the property of XOR operation ($0 \oplus A = A$) and the fact that permutation does not affect the image component with all-0s. The flowchart of this process is shown in Fig. 3.

IV. CRYPTANALYSIS

The attacker knows everything about the scheme except the secret keys based on Kerckhoff's principle. The security vulnerabilities proposed above show that the approach can be broken by CPA, which means that the attacker gains access to the encryption scheme and performs cryptanalysis by selecting adequate plaintexts [34].

We propose a flexible attack scheme which includes breaking the diffusion stage and breaking the permutation stage based on the three security drawbacks. The two main parts of the proposed attack are exchangeable, that is, one can break the diffusion firstly and then the permutation or break the permutation firstly and then the diffusion based on different situations. The attack scheme and simulation experiments are described in detail as follows.

A. BREAKING DIFFUSION STAGE

If a given cipher image $\mathbf{C}$ needs to be recovered without the secret keys, the attacker needs to know the equivalent permutation key streams and **image_key**$_{1,2,3}$ to recover the plain image $\mathbf{P}$. Thus, our purpose is to reveal all the equivalent key streams. If we get **image_key**$_{1,2,3}$, we can see that

$$\begin{cases} \mathbf{C}_{\mathrm{R}} \oplus \textbf{image\_key}_1 = \mathbf{P}_{\mathrm{R}}^{\mathrm{RC}} \\ \mathbf{C}_{\mathrm{G}} \oplus \textbf{image\_key}_2 = \mathbf{P}_{\mathrm{G}}^{\mathrm{RC}} \\ \mathbf{C}_{\mathrm{B}} \oplus \textbf{image\_key}_3 = \mathbf{P}_{\mathrm{B}}^{\mathrm{RC}} \end{cases}$$

according to (11) and the reversible property of XOR ($A \oplus B \oplus B = A$). Thus, the diffusion of encryption algorithm can be cracked. However, the number of possibilities of **image_key**$_{1,2,3}$ is all 256 based on the security vulnerability 3. The attacker does not know which **image_key**$_{1,2,3}$ is used in the scheme. Therefore, the attacker needs to obtain all 256 possible cracked images first, and then perform a further correlation analysis on the images to select the correct cracked image. The correct cracked image can be easily identified because it must be the only meaningful image in the 256 possibilities due to the strong randomicity, ergodicity and sensitivity of the used chaotic maps in the encryption scheme. If all **image_key**$_{1,2,3}$ denoted by

$$\begin{cases} \{\textbf{image\_key}_1^r\}_{r=0}^{255} \\ \{\textbf{image\_key}_2^r\}_{r=0}^{255} \\ \{\textbf{image\_key}_3^r\}_{r=0}^{255} \end{cases}$$

could be obtained by lunching CPA, the three components of cipher image are then XORed with $\{\textbf{image\_key}_1^r\}_{r=0}^{255}$, $\{\textbf{image\_key}_2^r\}_{r=0}^{255}$ and $\{\textbf{image\_key}_3^r\}_{r=0}^{255}$ in turn, with the only correctly recovered three components in the final 256×3 results.

From what have been discussed, the core idea of breaking diffusion phase is to find out all 256 **image_key**$_{1,2,3}$. Different **image_key**$_{1,2,3}$ results from the changing X_0 based on the Security vulnerability 3. According to (10), the reason for the change of X_0 is the sum of plain pixels. Therefore, 256 special plain images denoted by $\{\mathbf{O}_r^{\mathrm{R}}\}_{r=0}^{255}$ (shown as Eq. (16)) with all 0s except the first pixels r in the red component ranging from 0 to 255 are chosen to break the diffusion stage. Suppose the corresponding cipher images are $\{\mathbf{C}_r^{\mathrm{R}}\}_{r=0}^{255}$. According to Section III C, the extracted green component $\{\mathbf{C}_{\mathrm{G}r}\}_{r=0}^{255}$ and the blue component $\{\mathbf{C}_{\mathrm{B}r}\}_{r=0}^{255}$ are exactly equivalent to $\{\textbf{image\_key}_2^r\}_{r=0}^{255}$ and $\{\textbf{image\_key}_3^r\}_{r=0}^{255}$ respectively.

Similarly, when $\{\mathbf{O}_r^{\mathrm{G}}\}_{r=0}^{255}$ with all 0s but the first pixel r in the green component are chosen for encryption to obtain $\{\mathbf{C}_r^{\mathrm{G}}\}_{r=0}^{255}$, the extracted red component $\{\mathbf{C}_{\mathrm{R}r}\}_{r=0}^{255}$ and the blue component $\{\mathbf{C}_{\mathrm{B}r}\}_{r=0}^{255}$ are exactly equivalent to $\{\mathbf{image\_key}_1^r\}_{r=0}^{255}$ and $\{\mathbf{image\_key}_3^r\}_{r=0}^{255}$ respectively. Thus, all $\mathbf{image\_key}_{1,2,3}$ can be obtained.

The steps of obtaining $\mathbf{image\_key}_{1,2,3}$ are described as follows.

Step1: Encrypt all the 256 images $\{\mathbf{O}_r^{\mathrm{R}}\}_{r=0}^{255}$ to obtain 256 cipher images $\{\mathbf{C}_r^{\mathrm{R}}\}_{r=0}^{255}$ respectively, and decompose $\{\mathbf{C}_r^{\mathrm{R}}\}_{r=0}^{255}$ to get green and blue channel images $\{\mathbf{C}_{\mathrm{G}r}\}_{r=0}^{255}$ and $\{\mathbf{C}_{\mathrm{B}r}\}_{r=0}^{255}$.

Step2: Encrypt $\{\mathbf{O}_r^{\mathrm{G}}\}_{r=0}^{255}$ to obtain $\{\mathbf{C}_r^{\mathrm{G}}\}_{r=0}^{255}$, and extract the red component $\{\mathbf{C}_{\mathrm{R}r}\}_{r=0}^{255}$ from $\{\mathbf{C}_r^{\mathrm{G}}\}_{r=0}^{255}$.

Step3: The $\{\mathbf{C}_{\mathrm{R}r}\}_{r=0}^{255}$, $\{\mathbf{C}_{\mathrm{G}r}\}_{r=0}^{255}$ and $\{\mathbf{C}_{\mathrm{B}r}\}_{r=0}^{255}$ are equivalent to $\{\mathbf{image\_key}_1^r\}_{r=0}^{255}$, $\{\mathbf{image\_key}_2^r\}_{r=0}^{255}$ and $\{\mathbf{image\_key}_3^r\}_{r=0}^{255}$ respectively. Save 256×3 $\mathbf{image\_key}_{1,2,3}$ as three key tables (KTs). For example, every cipher image of $\{\mathbf{C}_{\mathrm{R}r}\}_{r=0}^{255}$ is scanned column-wisely from top to bottom, left to right, and saved to the key table 1 (KT_1) with size $(256, \mathrm{M} \times \mathrm{N})$ as shown in Table IV.

TABLE 4. $\mathbf{KT_1}$ **of** $\mathbf{image\_key_1}$**.**

r	$\mathrm{Image\_key}_1$				
0	$C_{R0}(1,1)$	$C_{R0}(2,1)$	$C_{R0}(3,1)$	$\cdots$	$C_{R0}(m,n)$
1	$C_{R1}(1,1)$	$C_{R1}(2,1)$	$C_{R1}(3,1)$	$\cdots$	$C_{R1}(m,n)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
255	$C_{R255}(1,1)$	$C_{R255}(2,1)$	$C_{R255}(3,1)$	$\cdots$	$C_{R255}(m,n)$

In this phrase, all values in KTs can be obtained using the special 256×2 plain-cipher image pairs. Thus, the computational complexity of this phase is $\mathrm{O}(256 \times 2 \times 3 \times \mathrm{M} \times \mathrm{N})$.

B. BREAKING PERMUTATION STAGE

Three sequences of length M and three sequences of length N are separately generated based on 3-D cat map in the permutation stage of the encryption scheme, then one can perform rows and columns scramble for the red, green and blue components of the color image respectively. As described in security vulnerabilities 1, when the sum of image pixel values remains unchanged, the diffusion effect will be eliminated. Moreover, the initial values and parameters for 3-D cat map are fixed based on security vulnerabilities 2. Therefore, the permutation key streams (***hr***, ***hg***, ***hb*** and ***lr***, ***lg***, ***lb***) can be obtained by launching CPA according to the two drawbacks.

Supposing the components $\mathbf{P}_{\mathrm{R}}$, $\mathbf{P}_{\mathrm{G}}$, $\mathbf{P}_{\mathrm{B}}$ of a known image $\mathbf{P}$, we change the two values in each row in order to obtain $\mathbf{P}_{\mathrm{R}}^r$, $\mathbf{P}_{\mathrm{G}}^r$, $\mathbf{P}_{\mathrm{B}}^r$ according to

$$\begin{cases} \mathrm{P}_{\mathrm{R}}^r(r,1) = \mathrm{mod}(\mathrm{P}_{\mathrm{R}}(r,1) - 1, 256), \\ \mathrm{P}_{\mathrm{R}}^r(r,2) = \mathrm{mod}(\mathrm{P}_{\mathrm{R}}(r,2) + 1, 256). \end{cases}$$

$$\begin{cases} \mathrm{P}_{\mathrm{G}}^r(r,1) = \mathrm{mod}(\mathrm{P}_{\mathrm{G}}(r,1) - 1, 256), \\ \mathrm{P}_{\mathrm{G}}^r(r,2) = \mathrm{mod}(\mathrm{P}_{\mathrm{G}}(r,2) + 1, 256). \end{cases}$$

$$\begin{cases} \mathrm{P}_{\mathrm{B}}^r(r,1) = \mathrm{mod}(\mathrm{P}_{\mathrm{B}}(r,1) - 1, 256), \\ \mathrm{P}_{\mathrm{B}}^r(r,2) = \mathrm{mod}(\mathrm{P}_{\mathrm{B}}(r,2) + 1, 256). \end{cases} \quad (17)$$

then merge the changed components to get $\mathbf{P}^r(r = 1, 2, \ldots, \mathrm{M})$.

There is no difference between the values of X_0 for images $\mathbf{P}$ and $\mathbf{P}^r$. Two pixels at least determine the position of a row or a column for one matrix, this is the reason why we change the first two values of each line in each channel image. After that, we encrypt $\mathbf{P}$ and $\mathbf{P}^r$ to get $\mathbf{C}$ and $\mathbf{C}^r$, here the result of $\mathbf{C}$ XOR $\mathbf{C}^r$ is $\mathbf{Y}^r$, and $\mathbf{Y}^r$ is divided into $\mathbf{Y}_{\mathrm{R}}^r$, $\mathbf{Y}_{\mathrm{G}}^r$, $\mathbf{Y}_{\mathrm{B}}^r$. Obviously, there are two non-zero elements in $\mathbf{Y}_{\mathrm{R}}^r$, $\mathbf{Y}_{\mathrm{G}}^r$ and $\mathbf{Y}_{\mathrm{B}}^r$ respectively based on Security vulnerability 1. The rank values of two non-zero elements in each channel is the first rule of row permutation map, which are denoted by $hr\,(1)$, $hg\,(1)$, $hb\,(1)$. In this way, all the row disorganized sequences ***hr***, ***hg***, ***hb*** can be obtained. Similarly, the column scramble sequences ***lr***, ***lg***, ***lb*** also can be obtained.

The detailed approach of breaking permutation stage can be described as follows.

- Cracking the row scrambling sequences ***hr***, ***hg***, ***hb***:

Step1: Encrypt a known image $\mathbf{P}$ and the changed $\mathbf{P}^r$ to obtain the cipher images $\mathbf{C}$ and $\mathbf{C}^r$ respectively.

Step2: Execute bit-XOR operation between $\mathbf{C}$ and $\mathbf{C}^{\mathrm{r}}$ to get $\mathbf{Y}^r$.

Step3: Decompose $\mathbf{Y}^r$ for getting the three components $\mathbf{Y}_{\mathrm{R}}^{\mathrm{r}}$, $\mathbf{Y}_{\mathrm{G}}^r$, $\mathbf{Y}_{\mathrm{B}}^r$. There would be two non-zero elements in each component. The locations of two non-zero elements for $\mathbf{Y}_{\mathrm{R}}^r$, $\mathbf{Y}_{\mathrm{G}}^r$ and $\mathbf{Y}_{\mathrm{B}}^r$ are denoted by $(x_{\mathrm{R}1}, y_{\mathrm{R}1})$ and $(x_{\mathrm{R}1}, y_{\mathrm{R}2})$, $(x_{\mathrm{G}1}, y_{\mathrm{G}1})$ and $(x_{\mathrm{G}1}, y_{\mathrm{G}2})$, $(x_{\mathrm{B}1}, y_{\mathrm{B}1})$ and $(x_{\mathrm{B}1}, y_{\mathrm{B}2})$, where x denotes the row number and y denotes the column number. The values of $x_{\mathrm{R}1}$, $x_{\mathrm{G}1}$, and $x_{\mathrm{B}1}$ are used to set the values of row permutation sequences $hr(r)$, $hg(r)$ and $hb(r)$ respectively.

Step4: Alter r and repeat Step1–3 until all the elements of row permutation sequences are revealed.

- Cracking the column scrambling sequences ***lr***, ***lg***, ***lb***:

Step1: Encrypt a known image $\mathbf{P}$ and the changed $\mathbf{P}^c$ to obtain cipher image $\mathbf{C}$ and $\mathbf{C}^c$ respectively. $\mathbf{P}^c$ is calculated by

$$\begin{cases} \mathrm{P}_{\mathrm{R}}^{\mathrm{c}}(1,c) = \mathrm{mod}(\mathrm{P}_{\mathrm{R}}(1,c) - 1, 256), \\ \mathrm{P}_{\mathrm{R}}^{\mathrm{c}}(2,c) = \mathrm{mod}(\mathrm{P}_{\mathrm{R}}(2,c) + 1, 256). \end{cases}$$

$$\begin{cases} \mathrm{P}_{\mathrm{G}}^{\mathrm{c}}(1,c) = \mathrm{mod}(\mathrm{P}_{\mathrm{G}}(1,c) - 1, 256), \\ \mathrm{P}_{\mathrm{G}}^{\mathrm{c}}(2,c) = \mathrm{mod}(\mathrm{P}_{\mathrm{G}}(2,c) + 1, 256). \end{cases}$$

$$\begin{cases} \mathrm{P}_{\mathrm{B}}^{\mathrm{c}}(1,c) = \mathrm{mod}(\mathrm{P}_{\mathrm{B}}(1,c) - 1, 256), \\ \mathrm{P}_{\mathrm{B}}^{\mathrm{c}}(2,c) = \mathrm{mod}(\mathrm{P}_{\mathrm{B}}(2,c) + 1, 256). \end{cases} \quad (18)$$

Step2: Execute bit-XOR operation between $\mathbf{C}$ and $\mathbf{C}^c$ to get $\mathbf{Y}^c$.

Step3: Decompose $\mathbf{Y}^c$ for getting the three components $\mathbf{Y}_{\mathrm{R}}^{\mathrm{c}}$, $\mathbf{Y}_{\mathrm{G}}^{\mathrm{c}}$, $\mathbf{Y}_{\mathrm{B}}^{\mathrm{c}}$. There would be two nonzero values in each component. The locations of two nonzero values for $\mathbf{Y}_{\mathrm{R}}^{\mathrm{c}}$, $\mathbf{Y}_{\mathrm{G}}^{\mathrm{c}}$, $\mathbf{Y}_{\mathrm{B}}^{\mathrm{c}}$ are denoted by $(x_{\mathrm{R}1}, y_{\mathrm{R}1})$ and $(x_{\mathrm{R}2}, y_{\mathrm{R}1})$, $(x_{\mathrm{G}1}, y_{\mathrm{G}1})$ and $(x_{\mathrm{G}2}, y_{\mathrm{G}1})$, $(x_{\mathrm{B}1}, y_{\mathrm{B}2})$ and $(x_{\mathrm{B}2}, y_{\mathrm{B}1})$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2867111, IEEE Access

where x denotes the row number and y denotes the column number. The values of y_{R1}, y_{G1} and y_{B1} are used to set the values of column permutation sequences $lr(c)$, $lg(c)$ and $lb(c)$ respectively.

Step4: Alter c and repeat Step1–3 until all the elements of column permutation sequences are revealed.

The permutation of original scheme is based on row and column scrambling. Therefore, the computational complexity of this phrase for a color image sized M × N × 3 is O(3 × (M + N)).

C. THE COMMUTATIVE ATTACK SCHEMES

From the above we can see, the core idea of the breaking diffusion phase is to build all the values of $\mathbf{image\_key}_{1,2,3}$ into KTs While the key point for breaking permutation stage is to find out six scrambling sequences ***hr***, ***hg***, ***hb***, ***lr***, ***lg***, and ***lb***. Since the two phases of the attack are independent with each other, the order of them can be exchanged. That is, the proposed attack procedure is of high flexibility. The two attack schemes in different order are as follows.

- Scheme I: breaking diffusion and then permutation.

Step1: Decompose a given color cipher image $\mathbf{C}$ into three components $\mathbf{C}_R$, $\mathbf{C}_G$, $\mathbf{C}_B$, and scan the three two-dimensional images from top to bottom, left to right respectively.

Step2: Obtain the KTs including $\mathbf{image\_key}_{1,2,3}$ by breaking diffusion stage shown in Sec. III. A.

Step3: Get the permutation key streams ***hr***, ***hg***, ***hb*** and ***lr***, ***lg***, ***lb*** by using the CPA proposed in Sec. III. B.

Step4: 256 $\mathbf{image\_key}_{1,2,3}$ in KTs are sequentially XORed with $\mathbf{C}_R$, $\mathbf{C}_G$ and $\mathbf{C}_B$ to get 256 $\{\mathbf{C'}_{Rr}\}_{r=0}^{255}$, $\{\mathbf{C'}_{Gr}\}_{r=0}^{255}$ and $\{\mathbf{C'}_{Br}\}_{r=0}^{255}$ respectively.

Step5: All possible red decryption images $\{\mathbf{P}_{Rr}\}_{r=0}^{255}$ can be obtained based on reverse permutation on $\{\mathbf{C'}_{Rr}\}_{r=0}^{255}$ using key streams ***hr*** and ***lr***. The green and blue components are processed similarly to obtain $\{\mathbf{P}_{Gr}\}_{r=0}^{255}$ and $\{\mathbf{P}_{Br}\}_{r=0}^{255}$ respectively.

Step6: Find out the most relevant image $\mathbf{P}_{Rr}$ which is the correct red decryption map from $\{\mathbf{P}_{Rr}\}_{r=0}^{255}$ by utilizing vector correlation analysis, and then return r value. In correlation analysis, the correlation coefficient of two vectors $\mathbf{V}_1$, $\mathbf{V}_2$ of a matrix calculated by

$$\rho = \frac{\langle \mathbf{V}_1, \mathbf{V}_2 \rangle}{\|\mathbf{V}_1\| \times \|\mathbf{V}_2\|}, \tag{19}$$

where $\langle \mathbf{V}_1, \mathbf{V}_2 \rangle$ denotes the dot product between $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$. $\|\mathbf{V}_1\|$ and $\|\mathbf{V}_2\|$ indicate the length of a vector. The correct decipher image $\mathbf{P}_{Rr}$ are selected from $\{\mathbf{P}_{Rr}\}_{r=0}^{255}$ by operations (i)-(ii) as follows:

(i) Select two columns $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ randomly from the image $\mathbf{P}_{Rr}$ and calculate their correlation coefficient $\boldsymbol{\rho}_r$. Save $\boldsymbol{\rho}_r$ to $\boldsymbol{\rho}_r = [\boldsymbol{\rho}_r]_{r=0}^{255}$.

(ii) Find the maximum value $\boldsymbol{\rho}_r$ and return k. Then the deciphered image is $\mathbf{P}_{Rr}$.

The green and blue components are processed in the same way to obtain $\mathbf{P}_{Gr}$ and $\mathbf{P}_{Br}$ respectively.

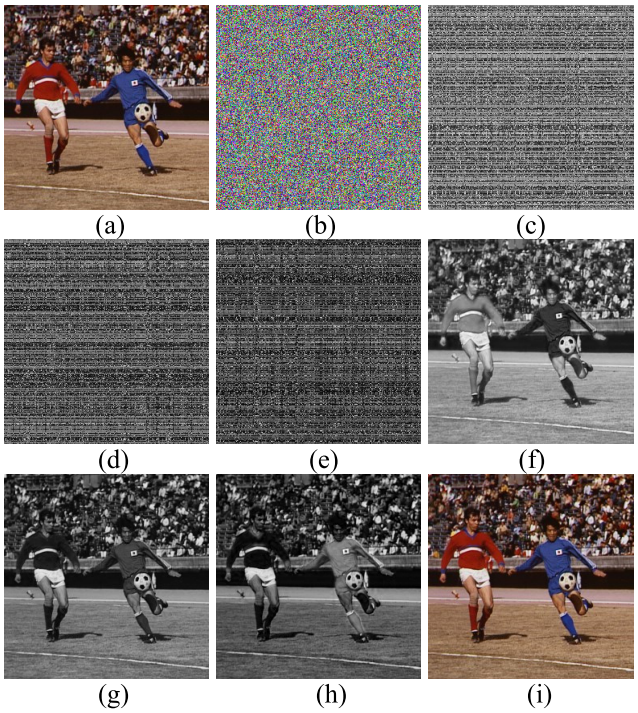

FIGURE 4. Experiment of the Scheme I: (a) plain image; (b) cipher image of (a); (c)(d)(e) de-diffusion components of (b); (f)(g)(h) de-permutation images corresponding to (c)(d)(e); (i) the final recovered image.

Step7: Merge the three channel images $\mathbf{P}_{Rr}$, $\mathbf{P}_{Gr}$ and $\mathbf{P}_{Br}$ for obtaining the decrypted color image $\mathbf{P}$.

The simulation on real natural image is performed to test the effectiveness of the proposed method. In Fig. 4, the encrypted image Fig. 4 (b) is obtained by encrypting Fig. 4(a). Fig.4(c), (d) and (e) are the three components from Fig. 4 (b) by de-diffusion stage firstly of Scheme I. When the three images are further decrypted, the decrypted three channel images are shown as Fig. 4(f), (g) and (h). The recovery result is shown in Fig. 4(i). Therefore, the original encryption scheme has been broken by the proposed Scheme I.

- Scheme II: breaking permutation and then diffusion:

Step1: Decompose a given color cipher image $\mathbf{C}$ into three components $\mathbf{C}_R$, $\mathbf{C}_G$, $\mathbf{C}_B$.

Step2: Get the permutation key streams ***hr***, ***hg***, ***hb*** and ***lr***, ***lg***, ***lb*** by using the CPA proposed in Sec. III. B.

Step3: Obtain KTs including $\mathbf{image\_key}_{1,2,3}$ by breaking diffusion stage shown in Sec. IV. A.

Step4: $\mathbf{C}_R$, $\mathbf{C}_G$, $\mathbf{C}_B$ are de-scrambled based on the permutation rules to obtain $\mathbf{C}'_R$, $\mathbf{C}'_G$, $\mathbf{C}'_B$ respectively. 256 $\mathbf{image\_key}_{1,2,3}$ in KTs are changed sequentially as $\mathbf{C}_R$, $\mathbf{C}_G$ and $\mathbf{C}_B$ de-scrambled operation to get the changed KT′s.

Step5: 256 $\mathbf{image\_key'}_{1,2,3}$ in KT′s are sequentially XORed with $\mathbf{C}'_R$, $\mathbf{C}'_G$ and $\mathbf{C}'_B$ to get all 256 possible red decryption images $\{\mathbf{P}_{Rr}\}_{r=0}^{255}$, $\{\mathbf{P}_{Gr}\}_{r=0}^{255}$ and $\{\mathbf{P}_{Br}\}_{r=0}^{255}$ respectively.

Step6: Find out the most relevant image $\mathbf{P}_{Rr}$, $\mathbf{P}_{Gr}$ and $\mathbf{P}_{Br}$ which are the correct red, green and blue decryption

map from $\{\mathbf{P}_{Rr}\}_{r=0}^{255}$, $\{\mathbf{P}_{Gr}\}_{r=0}^{255}$ and $\{\mathbf{P}_{Br}\}_{r=0}^{255}$ respectively by utilizing vector correlation analysis. The process is the same as that of Step 6 in Scheme I.

Step7: Merge the three channel images $\mathbf{P}_{Rr}$, $\mathbf{P}_{Gr}$ and $\mathbf{P}_{Br}$ for getting the decrypted color image $\mathbf{P}$.

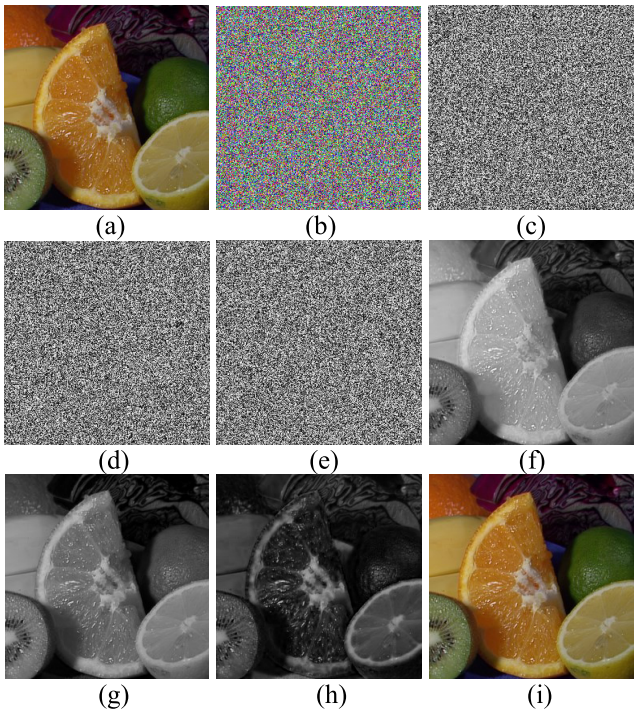

FIGURE 5. Experiment of the Scheme II: (a) plain image; (b) cipher image of (a); (c)(d)(e) de-permutation components of (b);(f)(g)(h) de-diffusion images corresponding to (c)(d)(e);(i) the final recovered image.

To verify the above attack Scheme II, some experiments were further performed. A plain image 'fruits' sized 256 × 256 as shown in Fig. 5 (a) is encrypted by the original encryption scheme, and the corresponding cipher image is shown in Fig. 5 (b). The de-permutation results of the three components of Fig. 5 (b) are shown in Fig. 5 (c), (d) and (e) respectively, and the further de-diffused three components are shown in Fig. 5 (f), (g) and (h) respectively. The decrypted color image is depicted in Fig.5 (i), which is the same as Fig. 5(a), showing that the original encryption scheme is entirely broken by Scheme II.

D. ANALYSIS OF THE PROPOSED SCHEMES

As shown in Fig. 4 and 5, both of the two schemes proposed above could recover the encrypted images with 100% recovery rate. However, the computational complexities of these two schemes are different, because Scheme II needs to additionally change the rows and columns position of all $\mathbf{image\_key}_{1,2,3}$ of KTs to get the updated KT′s. Thus, the computational complexity of scheme II is $O(3 \times (M + N))$ more than that of Scheme I. It is noted that the KTs for breaking diffusion can be stored in advance in order to reduce the computational complexity of both schemes in the attack process. In [37], there are three breaking schemes by ciphertext-only Attack (COA), known-plaintext attack (KPA) and CPA on an image scrambling encryption algorithm (ISEA) respectively. The permutation of the original scheme can be seen as IESA after de-diffusion firstly (in scheme I), and it is easily attacked. Therefore, Scheme I is better than Scheme II in terms of computational complexity and flexibility.

V. CONCLUSION

In this paper, we analyzed an encryption algorithm based on hybrid hyper-chaos and cellular automata. Three security flaws about the original cryptosystem are found. The leading weakness among them is that the diffusion effect can be offset by maintaining the sum of pixel values unchanged. Moreover, two commutative attack schemes are proposed. All the equivalent key streams used for encryption can be revealed. Based on the cryptanalysis of this paper, we also proposed three guidelines for improving the security of the image encryption scheme:

1) The relevance between secret key and plaintext should be complex instead of determining the key simply based on the sum of pixel values.

2) It is suggested to add an effective diffusion function after performing XOR.

3) Multiple rounds of permutation-diffusion should be adopted.

REFERENCES

[1] 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.

[2] T. Gao and Z. Chen, "A new image encryption algorithm based on hyper-chaos," *Phys. Lett. A*, vol. 372, no. 4, pp. 394–400, 2008.

[3] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.

[4] S. M. Seyedzadeh and S. Mirzakuchaki, "A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map," *Signal Process.*, vol. 92, no. 5, pp. 1202–1215, May 2012.

[5] W. Liu, K. Sun, and C. Zhu, "A fast image encryption algorithm based on chaotic map," *Opt. Lasers Eng.*, vol. 84, pp. 26–36, Sep. 2016.

[6] Z. Ni, X. Kang, and L. Wang, "A novel image encryption algorithm based on bit-level improved Arnold transform and hyper chaotic map," in *Proc. IEEE Int. Conf. Signal Image Process. (ICSIP)*, Aug. 2016, pp. 156–160.

[7] H. Fan, M. Li, D. Liu, and E. Zhang, "Cryptanalysis of a colour image encryption using chaotic APFM nonlinear adaptive filter," *Signal Process.*, vol. 143, pp. 28–41, Feb. 2018.

[8] M. Li, D. Xiao, Y. Zhang, and H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism," *Signal Process., Image Commun.*, vol. 39, pp. 234–248, Nov. 2015.

[9] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.

[10] H. Liu, A. Kadir, and Y. Niu, "Chaos-based color image block encryption scheme using S-box," *AEU-Int. J. Electron. Commun.*, vol. 68, no. 7, pp. 676–686, Jul. 2014.

[11] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.

[12] H. Liu and A. Kadir, "Asymmetric color image encryption scheme using 2D discrete-time map," *Signal Process.*, vol. 113, pp. 104–112, Aug. 2015.

[13] X. Wu, D. Wang, J. Kurths, and H. Kan, "A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system," *Inf. Sci.*, vols. 349–350, pp. 137–153, Jul. 2016.

[14] X. Wei, L. Guo, Q. Zhang, J. Zhang, and S. Lian, "A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system," *J. Syst. Softw.*, vol. 85, no. 2, pp. 290–299, 2012.

[15] X.-J. Tong, M. Zhang, Z. Wang, Y. Liu, H. Xu, and J. Ma, "A fast encryption algorithm of color image based on four-dimensional chaotic system," *J. Vis. Commun. Image Represent.*, vol. 33, pp. 219–234, Nov. 2015.

[16] B. Norouzi and S. Mirzakuchaki, "A fast color image encryption algorithm based on hyper-chaotic systems," *Nonlinear Dyn.*, vol. 78, no. 2, pp. 995–1015, Oct. 2014.

[17] Z. Liu, Y. Zhang, W. Liu, F. Meng, Q. Wu, and S. Liu, "Optical color image hiding scheme based on chaotic mapping and Hartley transform," *Opt. Lasers Eng.*, vol. 51, no. 8, pp. 967–972, Aug. 2013.

[18] A. Kadir, A. Hamdulla, and W.-Q. Guo, "Color image encryption using skew tent map and hyper chaotic system of 6th-order CNN," *Opt.-Int. J. Light Electron Opt.*, vol. 125, no. 5, pp. 1671–1675, Mar. 2014.

[19] X. Li, L. Wang, Y. Yan, and P. Liu, "An improvement color image encryption algorithm based on DNA operations and real and complex chaotic systems," *Opt.-Int. J. Light Electron Opt.*, vol. 127, no. 5, pp. 2558–2565, Mar. 2016.

[20] C. Dong, "Color image encryption using one-time keys and coupled chaotic systems," *Signal Process., Image Commun.*, vol. 29, no. 5, pp. 628–640, May 2014.

[21] T. Qi, J. Jun-Min, and J. Jun-Li, "An image encryption algorithm based on high-dimensional chaotic systems," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Nov. 2016, pp. 1–4.

[22] A. Y. Niyat, M. H. Moattar, and M. N. Torshiz, "Color image encryption based on hybrid hyper-chaotic system and cellular automata," *Opt. Lasers Eng.*, vol. 90, pp. 225–237, Mar. 2017.

[23] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Process.*, vol. 92, no. 4, pp. 1101–1108, Apr. 2012.

[24] L. Liu, Q. Zhang, and X. Wei, "A RGB image encryption algorithm based on DNA encoding and chaos map," *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1240–1248, Sep. 2012.

[25] X. Wu, B. Zhu, Y. Hu, and Y. Ran, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access*, vol. 5, pp. 6429–6436, Apr. 2017.

[26] E. Y. Xie, C. Li, S. Yu, and J. Lü, "On the cryptanalysis of Fridrich's chaotic image encryption scheme," *Signal Process.*, vol. 132, pp. 150–154, Mar. 2017.

[27] C. Li, D. Lin, J. Lö, and F. Hao. (Jun. 2018). "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography." [Online]. Available: https://arxiv.org/abs/1711.01858

[28] H. Dia and A. M. El-Semary, "Cryptanalysis and improvement of the image cryptosystem reusing permutation matrix dynamically," *Signal Process.*, vol. 148, pp. 172–192, Jul. 2018.

[29] C. Zhu and K. Sun, "Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps," *IEEE Access*, vol. 6, pp. 18759–18770, 2018.

[30] M. Ahmad, M. Z. Alam, S. Ansari, D. Lambić, and H. D. AlSharari, "Cryptanalysis of an image encryption algorithm based on PWLCM and inertial delayed neural network," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1323–1332, 2018.

[31] Y. Wang, P. Lei, H. Yang, and H. Cao, "Security analysis on a color image encryption based on DNA encoding and chaos map," *Comput. Electr. Eng.*, vol. 46, pp. 433–446, Aug. 2015.

[32] W. Wen, "Security analysis of a color image encryption scheme based on skew tent map and hyper chaotic system of 6th-order CNN against chosen-plaintext attack," *Multimedia Tools Appl.*, vol. 75, no. 6, pp. 3553–3560, Mar. 2016.

[33] C. Li, L. Y. Zhang, R. Ou, K.-W. Wong, and S. Shu, "Breaking a novel colour image encryption algorithm based on chaos," *Nonlinear Dyn.*, vol. 70, no. 4, pp. 2383–2388, 2012.

[34] D. Arroyo, J. Diaz, and F. B. Rodriguez, "Cryptanalysis of a one round chaos-based substitution permutation network," *Signal Process.*, vol. 93, no. 5, pp. 1358–1364, May 2013.

[35] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dyn.*, vol. 92, no. 2, pp. 305–313, Apr. 2018.

[36] L. Chen, B. Ma, X. Zhao, and S. Wang, "Differential cryptanalysis of a novel image encryption algorithm based on chaos and Line map," *Nonlinear Dyn.*, vol. 87, no. 3, pp. 1797–1807, Feb. 2017.

[37] C. Li, D. Lin, and J. Lü, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE MultiMedia*, vol. 24, no. 3, pp. 64–71, Aug. 2017.

[38] A. L. Abu Dalhoum, A. Madain, and H. Hiary, "Digital image scrambling based on elementary cellular automata," *Multimedia Tools Appl.*, vol. 75, no. 24, pp. 17019–17034, Dec. 2015.

[39] A. Bakhshandeh and Z. Eslami, "An authenticated image encryption scheme based on chaotic maps and memory cellular automata," *Opt. Lasers Eng.*, vol. 51, no. 6, pp. 665–673, 2013.

[40] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations," *Signal Process., Image Commun.*, vol. 52, pp. 6–19, Mar. 2017.

MING LI received the master's degree in science from the College of Physics and Information Engineering, Henan Normal University, Henan, China, in 2010, and the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2014. He is currently an Associate Professor with the College of Computer and Information Engineering, Henan Normal University. His research interests include multimedia security, information hiding, and compressive sensing.

DANDAN LU received the B.S. degree from the Department of Information Science and Technology, Bohai University, Jinzhou, China, in 2016. She is currently pursuing the M.S. degree in computer technology from Henan Normal University, Xinxiang, China. Her research interests include multimedia security and information hiding.

WENYING WEN received the Ph.D. degree in computational mathematics from Chongqing University, Chongqing, China, in 2013. She is currently an Associate Professor with the School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China. Her research interests include image processing and multimedia security.

HUA REN received the B.S. degree from the Department of Software, Henan Normal University, Xinxiang, China, in 2016, where she is currently pursuing the M.S. degree in computer science and technology. Her research interests include information hiding and multimedia security.

YUSHU ZHANG (M'17) received the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2014. He held various research positions at the City University of Hong Kong and the University of Macau. He is currently an Alfred Deakin Post-Doctoral Research Fellow with the School of Information Technology, Deakin University, Australia. He is also a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. He has published over 70 refereed journal articles and conference papers in his research areas. His research interests include multimedia security, compressive sensing security, cloud computing, and big data security.