IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 12, 2018, accepted March 13, 2018, date of publication March 21, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817600

Cryptanalyzing and Improving a Novel Color Image Encryption Algorithm Using RT-Enhanced Chaotic Tent Maps

CONGXU ZHU[1,2,3] AND KEHUI SUN[4]

[1]School of Information Science and Engineering, Central South University, Changsha 410083, China
[2]Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing, Yulin Normal University, Yulin 537000, China
[3]School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China
[4]School of Physics and Electronics, Central South University, Changsha 410083, China

Corresponding author: Congxu Zhu (zhucx@csu.edu.cn)

This work was supported by the Open Project of Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing under Grant 2016CSOBDP0103 and in part by the National Natural Science Foundation of China under Grant 61472451.

ABSTRACT In recent years, chaos-based image encryption algorithms have aroused extensive research interest. However, some image encryption algorithms still have several security defects, and the research on cryptanalysis is relatively inadequate. This paper performs the cryptanalysis of a newly proposed color image encryption scheme using RT-enhanced chaotic tent map. By using chosen-plaintext attacks, the equivalent keys of the cryptosystem are successfully broken, so that the target ciphertext image can be decoded. Based on the cryptanalysis, we then proposed an improved encryption algorithm. A new logistic-tent map is proposed and applied to the improved encryption algorithm, and a parameter related to the SHA-3 hash value of the plaintext image is introduced as a secret key parameter so that the improved algorithm can resist chosen-plaintext attacks. The security analysis and experimental tests for the improved algorithm are given in detail, which show that the improved algorithm can significantly increase the security of encryption images while still possessing all the merits of the original algorithm.

INDEX TERMS Chaotic cryptography, cryptanalysis, image encryption, logistic-tent map (LTM).

I. INTRODUCTION

The transmission of media information from the Internet has become very common in modern times. Consequently, it is important to ensure the security of the information transmission. In modern network communication, the commonly used information security technologies include data encryption [1], digital signature [2], trusted routing strategy [3], and so on. Among them, information encryption is the most basic technical means to protect information. Therefore, researches on encryption algorithms are particularly important. Image is a widely used information media on various occasions. However, Image encryption cannot be done in exactly the same way as text encryption. In image encryption, one need to consider some intrinsic characteristics of images [4], such as large amount of data, high data redundancy, and high correlation between adjacent pixels. Due to the good features of chaotic systems, such as the extreme sensitivity to initial conditions and control parameters, ergodicity and random-like behaviours, chaos has become an ideal tool for image encryption. As a result, chaos-based image encryption algorithm has become an attractive research area in recent years, and many image encryption algorithms have been proposed [5]–[12].

In different encryption schemes, a variety of strategies and different chaotic systems are adopted. Wu *et al*. [13] designed a high speed symmetric image encryption scheme by using a three-dimensional (3D) chaotic cat map. Wang *et al*. [14] proposed a fast image encryption scheme by using 3D chaotic baker maps. Chai [15] constructed an image encryption algorithm by using a new one-dimensional (1D) chaotic map, and simulate the Brownian motion of particles to confuse bit planes of the plain image. Huang [16] designed an image encryption algorithm by using chaotic Chebyshev generator. Wang *et al*. [17] proposed an image encryption scheme by using an intertwining logistic map and the PWLCM map. Ye [18] proposed an efficient symmetric image encryption algorithm based on an intertwining logistic map. Zhu [19] proposed a novel image

encryption scheme based on improved hyperchaotic sequences, which can achieve high key sensitivity and high plaintext sensitivity through only two rounds diffusion operation. Liu and Wang [20] proposed a color image encryption scheme, in which the piecewise linear chaotic map (PWLCM) and the Chebyshev maps.are used to generate the key stream sequence. the system parameter of PWLCM is modified by the perturbation sequence generated by the Chebyshev maps, and the initial condition of PWLCM is generated by the MD5 hash value of the mouse-position from entropy. However, the MD5 is not secure, which is cracked by Prof. Wang Xiaoyun at Tsinghua university. In [21], a color image encryption scheme is proposed by using spatial bit-level permutation and high-dimension chaotic system, which can achieve good encryption result and can resist common attack. But bit-level permutation and high-dimension chaotic system will increase the time overhead of the algorithm. In [22], an image encryption scheme by using DNA complementary rule and chaotic maps is proposed. The introduction of DNA coding principle into image encryption is a novel method, but DNA coding will also increase the time overhead of the algorithm. In [23], a chaotic image encryption system with a perceptron model is proposed, in which high-dimension Lorenz chaotic system and perceptron model within a neural network are used to enhanced the security of the cryptographic system. In [24], An image encryption scheme based on rotation matrix bit-level permutation and block diffusion is proposed, which has the suitability for a parallel mode and the robustness against noise attack. In [25], A double optical image encryption scheme is proposed by using discrete Chirikov standard map and chaos-based fractional random transform, which can achieve complete encryption for optical image. But fractional random transform increases the complexity of computing. In [26], a color image encryption scheme based on chaotic tent map (CTM) was proposed by C. Li *et al*, which only involves the diffusing phase, and the confusing phase has been omitted. As a result, there are some security defects in the pure CTM-based scheme. Very recently, a color image encryption scheme based on the rectangular transform and the CTM was proposed by Wu *et al*. [27], which is an enhanced CTM-based color image encryption scheme. Wu's scheme consists of two phases including confusion and diffusion, which are controlled by an improved 2D Arnold transform and the chaotic tent map, respectively. These works on the design of encryption schemes belong to the research category of cryptography, which is one of the branches of cryptology.

Compared with cryptography, cryptanalysis is the science of deciphering secret keys or plaintext [28]–[30], which is another branch of cryptology. Some recent studies have shown that there are security vulnerabilities in some chaos-based image encryption algorithms. Li *et al*. [28] developed the ciphertext-only attack, known-plaintext attack and chosen-plaintext attack on the Ye's scheme [18]. Li *et al*. [29] developed known-plaintext attack on the Zhu's scheme [19]. Wang *et al*. [30] cracked Huang's algorithm [16] by using chosen-plaintext attack. For some other examples, several chaos-based image encryption schemes that were cracked are mentioned in [31]. Cryptanalysis can not only reveal weaknesses in encryption algorithms, but also help the designers to improve the security of encryption algorithm. If the security bugs in encryption algorithms are not find out, then, the application of insecure algorithms to secure communications will bring serious security risks and losses to both sides of communications. Hence, works on cryptanalysis are of vital significance to promote the progress of cryptology.

As a typical color image encryption algorithm, Wu's encryption algorithm [27] has the merits of simple structure, fast encryption speed and fine cryptographic performance. As a consequence, it has advantages in dealing with large number of data and lessening redundant information, compared with the conventional image encryption algorithms. But some defects can also be found in Wu's encryption scheme. The present paper re-evaluates the security of the encryption algorithm proposed in [27], and discovers the following security problems: (1) it can't resist chosen-plaintext attack; (2) the encrption algorithm is also insensitive to all the chaotic secret keys; (3) the first pixel in the cipher image can not be decrypted in the decryption process; (4) there is an additional restriction on the parameter selection of the inverse rectangular transform system. To fix the security defects, we proposed an improved color image encryption algorithm.

The rest of this paper is organized as follows. Section II describes briefly the Wu's algorithm. Detailed cryptanalysis and attacks on the Wu's algorithm are presented in Section III. An improved encryption scheme was proposed in Section IV. Some experimental results and analysis for the improved scheme are given in Section V. Finally, concluding remarks are given in Section VI.

II. DESCRIPTION OF THE ORIGINAL ENCRYPTION ALGORITHM

The plaintext images to be encrypted in Wu's algorithm are color images with size of $m \times n \times 3$, which can be expressed by a matrix $\mathbf{P} = [P(i, j, k)]$, $P(i, j, k) \in \{0, 1, \ldots, 255\}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, $k = 1, 2, 3$. A color image $\mathbf{P}$ consists of R(Red), G(Green) and B(Blue) component images, and the red, green and blue component images can be expressed by $\mathbf{RP} = [RP(i, j)]$, $\mathbf{GP} = [GP(i, j)]$, and $\mathbf{BP} = [BP(i, j)]$. Where, $RP(i, j) = P(i, j, 1)$, $GP(i, j) = P(i, j, 2)$, $BP(i, j) = P(i, j, 3)$. Wu's algorithm includes two processing stages: (1) Confusion process, i.e. to permutate pixel positions. (2) Diffusion process, i.e. to encrypt pixel values. The main ideas of the Wu's algorithm can be redescribed briefly as follows.

A. THE CHAOTIC MAP AND SECRET KEYS

The chaotic system used in the Wu's algorithm to generate chaotic random sequence is the chaotic tent map (CTM), which is defined as

$$\begin{cases} x_{i+1} = \mu x_i, & \text{if } x_i < 0.5, \\ x_{i+1} = \mu(1 - x_i), & \text{else.} \end{cases} \quad (1)$$

where $x_i \in (0, 1)$. Note that when the parameter $\mu \in (0, 2]$ and the initial value $x_0 \in (0, 1)$, the tent map is chaotic and transforms an interval (0, 1) into itself [26].

The 2D rectangular transform used in the Wu's algorithm to permutate pixel positions is the improved 2D Arnold transform, which is defined as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \left[\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} r_m \\ r_n \end{pmatrix} \right] \bmod \begin{pmatrix} m \\ n \end{pmatrix} \quad (2)$$

where (a, b, c, d) are the elements of the transform matrix, (x, y) and (x', y') are the position of a pixel in the original image and its new position in the permutated image respectively, while m and n are the height and the width of the plain image respectively. The 2D rectangular transform has an inverse operation when the following condition is met, i.e.,

$$\begin{cases} p = \gcd(m, n), \quad p_m = p/m, \ p_n = p/n, \\ \gcd(a, p_m) = 1, \quad \gcd(d, p_n) = 1, \\ (b \mod p_m) = 0 \ or \ (c \mod p_n) = 0, \\ \gcd(ad - bc, p) = 1. \end{cases} \quad (3)$$

Then the inverse transform to Eq.(2) is expressed as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} x' - r_m \\ y' - r_n \end{pmatrix} \bmod \begin{pmatrix} m \\ n \end{pmatrix} \quad (4)$$

The secret keys of the Wu's algorithm are $(\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30})$ for the chaotic tent map and $(a, b, c, d, r_m, r_n, t)$ for the improved 2D-RT. Where (μ_1, μ_2, μ_3) and (x_{10}, x_{20}, x_{30}) are the control parameters and initial values of the CTM systems respective, and t is the iteration round number for permutation.

B. THE WU'S ALGORITHM

In order to describe the Wu's algorithm more clearly, we draw the flow chart of the algorithm, which is as shown in Fig.1. The encryption process consists of two phases, i.e., permutate pixel positions and encrypt pixel values. In cryptanalysis, an encryption scheme is like an encryption machinery. The dashed rectangle box in Fig.1 is equivalent to the encryption machinery of the Wu's algorithm.

We can explain the overall process of the encryption machinery briefly as follows. In the input port of the encryption machinery, a color plaintext image with size of $m \times n \times 3$ is input. In the output port of the encryption machinery, the encrypted color image with size of $m \times n \times 3$ is output. The encryption machinery includes confusion and diffusion processing stages. In the confusion process, firstly, the color plaintext image is transformed into a gray image. secondly, the gray image is permutated by the 2D Arnold transform. In the diffusion process, firstly, the permutated gray image is transformed into three color images. Secondly, three color images are encrypted by using CTM. Thirdly, the three encrypted color component images are merged into a color image, then the encrypted image is obtained.

The specific steps can be redescribed briefly as follows:

Step (1): Choose the secret keys $(a, b, c, d, r_m, r_n, t)$ and $(\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30})$.

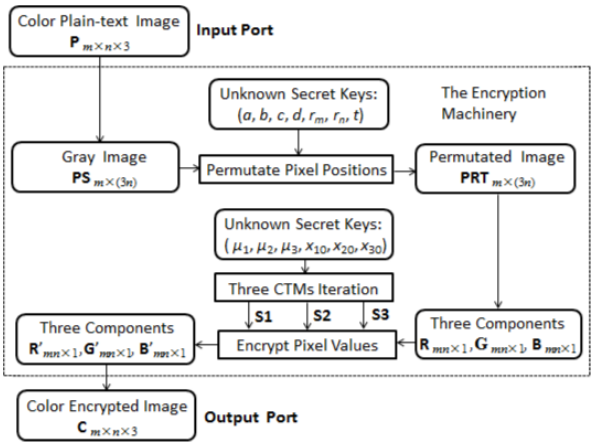

FIGURE 1. The flow chart of the Wu's algorithm.

Step (2): Read the $m \times n \times 3$ sized color plaintext image $\mathbf{P}_{m\times n\times 3} = [P(i, j, k)]$. Let $N = m \times n$, and denote the three components of $\mathbf{P}_{m\times n\times 3}$ as $\mathbf{RP}_{m\times n} = [RP(i, j)]$, $\mathbf{GP}_{m\times n} = [GP(i, j)]$ and $\mathbf{BP}_{m\times n} = [BP(i, j)]$, respectively. Where $i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, k = 1, 2, 3$.

Step (3): Stitch the three components $\mathbf{RP}_{m\times n}$, $\mathbf{GP}_{m\times n}$ and $\mathbf{BP}_{m\times n}$ together to form a gray image $\mathbf{PS}_{m\times 3n} = [PS(i, l)]$, where $i = 1, 2, \ldots, m, l = 1, 2, \ldots, 3\,n$.

Step (4): Permutate the gray image $\mathbf{PS}_{m\times 3n} = [PS(x, y)]$ by using Eq.(2) for t rounds, and get a permutated image as $\mathbf{PRT}_{m\times n} = [PRT(x', y')]$. Where, $PRT(x', y') = PS(x, y)$.

Step (5): Split $\mathbf{PRT}_{m\times 3n}$ into three matrices $\mathbf{RRT}_{m\times n}$, $\mathbf{GRT}_{m\times n}$, and $\mathbf{BRT}_{m\times n}$ with size of $m \times n$. Then further convert $\mathbf{RRT}_{m\times n}$, $\mathbf{GRT}_{m\times n}$, and $\mathbf{BRT}_{m\times n}$ to three 1D vectors $\mathbf{R}_{N\times 1}$, $\mathbf{G}_{N\times 1}$, and $\mathbf{B}_{N\times 1}$. Where $N = m \times n$.

Step (6): Iterate Equation (1) for $N + 1000$ times with the parameters (μ_1, x_{10}), (μ_2, x_{20}) and (μ_3, x_{30}) respectively, and take the final N values to form three chaotic sequences **X1**, **X2**, **X3** of length N.

Step (7): Calculate three key-streams **S1**, **S2**, **S3** with **X1**, **X2**, **X3** by

$$\begin{cases} \mathbf{S1} = \lfloor \mathbf{X1} \times 10^{10} \rfloor \mod 256, \\ \mathbf{S2} = \lfloor \mathbf{X2} \times 10^{10} \rfloor \mod 256, \\ \mathbf{S3} = \lfloor \mathbf{X3} \times 10^{10} \rfloor \mod 256. \end{cases} \quad (5)$$

Step (8): Encrypt $\mathbf{R}_{N\times 1}$, $\mathbf{G}_{N\times 1}$, and $\mathbf{B}_{N\times 1}$ to obtain their corresponding ciphertext images $\mathbf{R}' = [R'(i)]$, $\mathbf{G}' = [G'(i)]$, and $\mathbf{B}' = [B'(i)]$ as

$$\begin{cases} R'(i) = ((R(i)+G'(i-1)+B'(i-1)) \bmod 256) \oplus S1(i), \\ G'(i) = ((G(i)+R'(i-1)+B'(i-1)) \bmod 256) \oplus S2(i), \\ B'(i) = ((B(i)+R'(i-1)+G'(i-1)) \bmod 256) \oplus S3(i), \end{cases} \quad (6)$$

Where $i = 1, 2, \ldots, N$. When $i = 1$, $R'(i-1)$, $G'(i-1)$ and $B'(i-1)$ are replaced by three parameters R'_0, G'_0, and B'_0

respectively, which are calculated by

$$\begin{cases} R'_0 = (\frac{\sum_{i=1}^{i=m\times n} R(i)}{m\times n} + \delta) \mod 256, \\ G'_0 = (\frac{\sum_{i=1}^{i=m\times n} G(i)}{m\times n} + \delta) \mod 256, \\ B'_0 = (\frac{\sum_{i=1}^{i=m\times n} B(i)}{m\times n} + \delta) \mod 256. \end{cases} \tag{7}$$

and

$$\delta = \left\lfloor (\bar{\mathbf{P}} - \lfloor \bar{\mathbf{P}} \rfloor) \times 10^{10} \right\rfloor \mod 256. \tag{8}$$

and

$$\bar{\mathbf{P}} = \frac{\sum_{i=1}^{i=m} \sum_{j=1}^{j=n} \sum_{k=1}^{k=3} P(i,j,k)}{m\times n\times 3}. \tag{9}$$

Step (9): Reshape three 1D vectors $\mathbf{R}'$, $\mathbf{G}'$, $\mathbf{B}'$ to three matrices $\mathbf{RC}_{m\times n}$, $\mathbf{GC}_{m\times n}$, $\mathbf{BC}_{m\times n}$, and use these three components to compose the final color cipher image $\mathbf{C}$.

The decryption algorithm is the reverse operation of the encryption algorithm. Here, the two key operation steps of the decryption algorithm are briefly described as follows.

First, in the reverse diffusion process, the formula for recovering $\mathbf{R}$, $\mathbf{G}$, and $\mathbf{B}$ from $\mathbf{R}'$, $\mathbf{G}'$, and $\mathbf{B}'$ is as

$$\begin{cases} R(i) = (R'(i) \oplus S1(i) - G'(i-1) - B'(i-1)) \mod 256, \\ G(i) = (G'(i) \oplus S2(i) - R'(i-1) - B'(i-1)) \mod 256, \\ B(i) = ((B'(i) \oplus S3(i) - R'(i-1) - G'(i-1)) \mod 256. \end{cases} \tag{10}$$

Where $i = 1, 2, \ldots, N$. When $i = 1$, $R'(i-1)$, $G'(i-1)$ and $B'(i-1)$ are replaced by the three parameters R'_0, G'_0, and B'_0, respectively. It is worth noting that the first pixel values of $R(1)$, $G(1)$ and $B(1)$ can not be decrypted because the values of R'_0, G'_0 and B'_0 are unknown. These values need to be calculated by pixel values of the plain image.

Second, in the reverse confusion process, the formula for the recovery of the un-permutated gray image $\mathbf{PS}_{m\times 3n}$ from the permutated gray image $\mathbf{PRT}_{m\times 3n}$ is Eq.(4). As a result, the conditions expressed by the formula (3) must be satisfied.

C. MAIN DEFECTS OF THE ORIGINAL ALGORITHM

There are four main defects in the Wu's algorithm, which are summarized as follows:

(1) The secret keys used in Wu's algorithm are irrelevant to the plaintext image to be encrypted, and therefore Wu's algorithm can not resist chosen-plaintext attacks.
(2) When any one of the parameters (μ_1, μ_2, μ_3, x_{10}, x_{20}, x_{30}) changes, it only changes one of the sequences (**S1**, **S2**, **S3**). As a result, Wu's algorithm is insensitive to all the secret keys.
(3) In the decryption process, R'_0, G'_0 and B'_0 are unknown and can't be calculated. As a result, the first pixel can't be decrypted.
(4) In the reverse confusion process, Wu *et. al.* adopted the inverse 2D Arnold transform, and it requires the parameters (a, b, c, d) to meet the restrictive conditions expressed by Eq.(3).

III. THE CRYPTANALYSIS AND CHOSEN-PLAINTEXT ATTACKS

According to Kerchoff's principle [32], when analyzing an encryption algorithm, a hypothesis is that the cryptanalyst knows exactly the design and working of the cryptosystem except for the secret keys. Namely, the attacker knows all the working mechanisms of the cryptosystem, but does not know the secret keys. There are four classical types of attacks:

(1) Ciphertext only attack: the opponent possesses only the target ciphertext.
(2) Known plaintext attack: the opponent possesses a string of plaintext, and the corresponding ciphertext.
(3) Chosen-plaintext attack: the opponent has obtained temporary access to the encryption machinery. Hence, he or she can choose any plaintext, and obtain the corresponding ciphertext.
(4) Chosen-ciphertext attack: the opponent has obtained temporary access to the decryption machinery. Hence, he or she can choose any ciphertext, and obtain the corresponding plaintext.

Actually, in the confusion process of Wu's algorithm, the t rounds 2D rectangular transform can be equivalently replaced by a position traversing matrix $\mathbf{T} = [T(x, y)]$, where $x = 1, 2, \ldots, m$; $y = 1, 2, \ldots, 3\times n$; $T(x, y) = 1, 2, \ldots, m\times 3\times n$. If a pixel at the coordinate position (x, y) in the image **PS** is transformed to the coordinate position (x', y') in the image **PRT**, namely, $PRT(x', y') = PS(x, y)$. Then we let $T(x, y) = (y'-1)\times m + x'$. Here, $T(x, y)$ represents the one-dimensional ordinal number of the pixel $PS(x, y)$ in the image **PRT** according to the order of column priority. Conversely, $y' = \lceil T(x, y)/m \rceil$, $x' = T(x, y) - (y'-1)\times m$. It is worth noting that $\mathbf{T}$ is determined by parameters (a, b, c, d, r_m, r_n, t) and has nothing to do with the plaintext image. Therefore, $\mathbf{T}$ can be used as the equivalent key of the secret keys (a, b, c, d, r_m, r_n, t). Similarly, In the diffusion process of Wu's algorithm, the three key-streams **S1**, **S2** and **S3** are exactly equivalent to the secret keys (μ_1, μ_2, μ_3, x_{10}, x_{20}, x_{30}) and not related with the plaintext image.

Just because of the above reasons, the attacker can select some special plaintext images to encrypt and obtain the corresponding ciphertext images. Then he or she can uncover the keys **S1**, **S2**, **S3** and **T** by using the chosen-plaintext and its corresponding ciphertext. Finally, the attacker can utilize the cracked equivalent keys **S1**, **S2**, **S3** and **T** to decrypt the target ciphertext image without having to know the original keys (μ_1, μ_2, μ_3, x_{10}, x_{20}, x_{30}; a, b, c, d, r_m, r_n, t). Here we have three steps to implement the chosen-plaintext attacks on Wu's algorithm, each of the attacks is described separately in the following sub Sections ***A***, ***B*** and ***C***. In the following description, we use $\mathbf{U}$ to represent a chosen-plaintext image, and $\mathbf{V}$ to represent the ciphertext image corresponding to $\mathbf{U}$.

A. RECOVER THE ENCRYPTION KEY STREAMS

Choosing a plaintext image $\mathbf{U}$ consisting of all zero elements and obtain its corresponding ciphertext image $\mathbf{V}$ by

the encryption machinery. The 1D vectors corresponding to the permutated image of **U** are **R**, **G** and **B**. The 1D vectors corresponding to **V** are $\mathbf{R}'$, $\mathbf{G}'$ and $\mathbf{B}'$. Thanks to the permutating process that does not change the pixel values of the plaintext image, the pixel values in **R**, **G** and **B** are all zeros. Correspondingly, the values of R'_0, G'_0 and B'_0 can be obtained by formula (7) and $R'_0 = 0$, $G'_0 = 0$, $B'_0 = 0$. According to Eq.(6), then one can recover the key streams **S1**, **S2** and **S3** as

$$\begin{cases} S1(i) = R'(i) \oplus ((G'(i-1) + B'(i-1)) \bmod 256), \\ S2(i) = G'(i) \oplus ((R'(i-1) + B'(i-1)) \bmod 256), \\ S3(i) = B'(i) \oplus ((R'(i-1) + G'(i-1)) \bmod 256). \end{cases} \tag{11}$$

Where $i = 1, 2, \ldots, N$, and in the case of $i = 1, R'(0) = R'_0 = 0, G'(0) = G'_0 = 0, B'(0) = B'_0 = 0$.

B. RECOVER THE POSITION TRAVERSING MATRIX

A color image with size of $m \times n \times 3$ has $(3mn)$ pixels, and each pixel value is an integer ranging from 1 to 255. If $(3mn) \leq 255$, then only one chosen-plaintext image is required to recover the position traversing matrix **T**, so that each pixel in the chosen-plaintext image has a different value in the set $\{1, 2, \ldots, 255\}$. If $(3mn) > 255$, then the number of chosen-plaintext images required to recover the position traversing matrix **T** is $\lceil 3mn/255 \rceil$, so that any selected image have 255 pixels with different values ranging from 1 to 255, and the rest pixels have the same value zero.

For the case of $(3mn) > 255$, we let the 2D gray image **PS** derived from the I-th chosen-plaintext image **U** satisfies

$$\begin{cases} PS(i,j) = (j-1) \times m + i, \\ if\ (j-1) \times m + i \in [(I-1) \times 255 + 1,\ I \times 255], \\ PS(i,j) = 0, \\ if\ (j-1) \times m + i \notin [(I-1) \times 255 + 1,\ I \times 255]. \end{cases} \tag{12}$$

Where **PS** is the $m \times 3n$ gray image corresponding to **U**, and $I = 1, 2, \ldots, \lceil 3mn/255 \rceil$. By using the I-th chosen-plaintext image and its corresponding ciphertext image as well as the key streams **S1**, **S2** and **S3** obtained previously, we can recover the I-th permutated image **PRT**. Then we can recover at most 255 values of elements in matrix **T** by comparing the I-th chosen-plaintext image matrix **PS** and its corresponding permutated gray image matrix **PRT**. With $\lceil 3mn/255 \rceil$ chosen-plaintext images and their corresponding ciphertext images, all of the $(3mn)$ elements in **T** can be solved.

C. RECOVER THE TARGET PLAIN IMAGE

In sub Section A, we obtained the chaotic key streams **S1**, **S2** and **S3**, which are only related to the secret keys (μ_1, μ_2, μ_3, x_{10}, x_{20}, x_{30}) and unrelated to the plaintext image. In sub Section B, we also obtained the whole permutation matrix **T**, which is only related to the secret keys (a, b, c, d, r_m, r_n, t). Therefore, we can break any other ciphertext image **C** encrypted by the same encryption machinery which has the same parameters ($\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}; a, b, c, d, r_m, r_n, t$). The decryption process to recover **P** from **C** is as follows:

Step (1): Reshape the color cipher image **C** to three components of 1D vectors $\mathbf{R}'$, $\mathbf{G}'$ and $\mathbf{B}'$.

Step (2): For $i = 1, 2, \ldots, N$, by using Eq.(10) to recover the three components of 1D vectors **R**, **G** and **B** except for $R(1)$, G (1) and $B(1)$. Because the values of R'_0, G'_0 and B'_0 are unknown, therefore, $R(1)$, $G(1)$ and $B(1)$ can not be recovered. For simplicity, we let $R(1) = R'(1)$, $G(1) = G'(1)$, $B(1) = B'(1)$.

Step (3): The three components of 1D vectors **R**, **G** and **B** of size $(m \times n) \times 1$ are merged into a gray scale image matrix **PRT** of size $m \times 3n$.

Step (4): For every pixel position (x, y) in **PRT**, do inverse permutation operations by using **T** to obtain **PS** as follows:

$$y' = \lceil T(x,y)/m \rceil, \quad x' = T(x,y) - (y'-1) \times m. \tag{13}$$

$$PS(x,y) = PRT(x',y'). \tag{14}$$

Where $x = 1, 2, \ldots, m$; $y = 1, 2, \ldots, 3n$; $x' = 1, 2, \ldots, m$; $y' = 1, 2, \ldots, 3n$. (x, y) and (x', y') are coordinates of the same pixel in **PS** and **PRT**, respectively.

Step (5): Split $m \times 3n$ sized matrix $\mathbf{PS}_{m\times 3n}$ into three $m \times n$ sized matrices, i.e. $\mathbf{RP}_{m\times n}$, $\mathbf{GP}_{m\times n}$, $\mathbf{BP}_{m\times n}$.

Step (6): Combine the three components of $\mathbf{RP}_{m\times n}$, $\mathbf{GP}_{m\times n}$ and $\mathbf{BP}_{m\times n}$, and the final deciphered color image **P** is obtained.

D. AN EXAMPLE OF THE ATTACKS

In this example, the color plaintext image Baboon with size of 256×256×3 is encrypted by Wu's algorithm. The secret keys of the encryption machinery are ($\mu_1 = 1.9$, $\mu_2 = 1.7$, $\mu_3 = 1.6, x_{10} = 0.201, x_{20} = 0.301, x_{30} = 0.401; a = 1, b = 3$, $c = 5, d = 16, r_m = 4, r_n = 7, t = 5$). The plaintext image and the ciphered image are shown in Figs. 2(a) and 2(b) respectively. The recovered image is the one in Fig. 2(c), which coincides with the original plain image in Fig. 2(a). It takes about 12 minutes to break the color ciphertext image with size of 256×256×3 by our desktop PC. The cost of this time is acceptable. Therefore, the Wu's algorithm can not resist chosen-plaintext attacks, and it can't be used for secure communications with high security requirements.

IV. THE IMPROVED SCHEME

The improved scheme retains the main advantages of Wu's algorithm, but overcomes its security defects mentioned above.

A. THE NEW CHAOTIC SYSTEM AND ITS BASIC DYNAMIC BEHAVIORS

Chaotic systems play an important role in chaos-based image encryption algorithms. The performance of a chaotic system, such as the uniformity and randomness of the distribution of state values, and the size of parameter intervals that generate chaotic characteristics, can help to improve the security

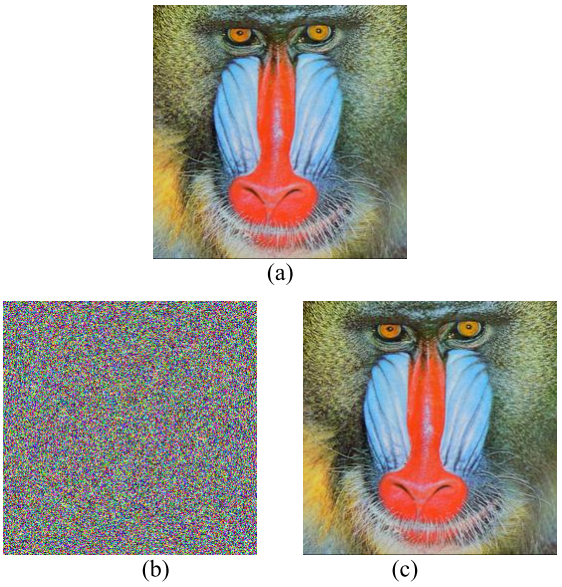

FIGURE 2. The experimental results of the chosen-plaintext attacks. (a) The plaintext image Baboon. (b) The ciphered image. (c) The recovered image.

of encryption schemes. Tent map can only produce chaotic behavior in a small parameter interval, and the distribution of its state values are not uniform. In order to improve the chaotic performance of tent map, we propose a new chaotic system by combining Logistic map and tent map, which has the following mathematical models:

$$\begin{cases} x_{i+1}=f_1(x_i)=(4-\mu)x_i(1-x_i)+\dfrac{\mu}{2}x_i, & if\ x_i<0.5, \\ x_{i+1}=f_2(x_i)=(4-\mu)x_i(1-x_i)+\dfrac{\mu}{2}(1-x_i), & if\ x_i\geq 0.5. \end{cases} \tag{15}$$

When $\mu = 0$, the new system degenerates into the Logistic map, and when $\mu = 4$, the new system degenerates into the tent map. We named the new system (15) as Logistic-tent map (LTM).

Proposition 1: If $\mu \in (0, 4)$ and $x_i \in (0, 1)$, then system (15) is a map $f: x_i \in(0, 1)\rightarrow x_{i+1} \in (0, 1)$.

Proof Function $f_1(x)$ can be converted into a standard quadratic function form as: $f_1(x) = (\mu - 4)x^2 + (4 - \mu/2)x$. For $\mu < 4$, $(\mu-4) < 0$, hence, function $f_1(x)$ has a maximum value at $x_m = (4 - \mu/2)/(8 - 2\mu) = 1/2 + \mu/(16 - 4\mu) > 1/2$. Therefore, when $x < 0.5 < x_m$, function $f_1(x)$ is monotonically increasing. Namely, $f_1(x < 0.5) < f_1(x = 0.5) = (2 - \mu/4) - (4 - \mu)/4 = 1$, $f_1(x > 0) > f_1(x = 0) = 0$.

Similarly, function $f_2(x)$ can be converted into a standard quadratic function form as: $f_2(x) = (\mu - 4)x^2 + (4 - 3\mu/2)x + \mu/2$. For $\mu < 4$, $(\mu - 4) < 0$, hence, function $f_2(x)$ has a maximum value at $x_m = (4 - 3\mu/2)/(8 - 2\mu) = 1/2 - \mu/(16 - 4\mu) < 1/2$. Therefore, when $x \geq 0.5 > x_m$, function $f_2(x)$ is monotonically decreasing. Therefore, $f_2(x \geq 0.5) < f_2(x = 0.5) = (4 - 3\mu/2)/2 - (4 - \mu)/4 + \mu/2 = 1$, and $f_2(x < 1) > f_2(x = 1) = 0$. To sum up, we come to the following conclusions: If $x_i \in(0, 0.5)$, then $x_{i+1} = f_1(x_i) \in (0, 1)$. If $x_i \in [0.5, 1.0)$, then $x_{i+1} = f_2(x_i) \in(0, 1)$. The proof is complete.

TABLE 1. The NIST-800-22 test results of LTM.

NIST tests	Min *P*-value	Results
Frequency	0.4272	Passed
Block Frequency	0.0632	Passed
The Run Test	0.5231	Passed
Longest Run of Ones in a block	0.0830	Passed
Binary Matrix Rank	0.0785	Passed
DFT Spectral	0.0422	Passed
Non-Overlapping Template Matching	0.0952	Passed
Overlapping Template Matching	0.1815	Passed
Maurer's Universal Statistical Test	0.0563	Passed
Linear Complexity	0.0756	Passed
Serial Test	0.2412	Passed
Approximate Entropy	0.0902	Passed
Cumulative Sums	0.2570	Passed
Random Excursions	0.0625	Passed
Random Excursions Variant	0.1236	Passed

To compare the chaotic characteristics of the new system and the tent map system, the chaotic dynamic behaviors of the two systems are described by using bifurcation and Lyapunov exponent diagrams. Figs. 3(a) and 3(b) are the bifurcation and Lyapunov exponent diagram of tent map respectively. Figs. 3(c) and 3(d) are the bifurcation and Lyapunov exponent diagram of the LTM respectively. From Fig. 3, one can see that tent map has positive Lyapunov exponents and is in a chaotic state when μ in the range of (1, 2], and the range is very small. Furthermore, the distribution of state values of chaotic sequence $\{x_i\}$ in the range of [0, 1] is very uneven. However, the new LTM system has positive Lyapunov exponent and is in a chaotic state when μ in the range of (0, 4), and the range of μ value is much larger than that of the tent map. Furthermore, the distribution of state values $\{x_i\}$ of the proposed new LTM system is more uniform in the range of [0, 1].

Furthermore, to evaluate whether the random numbers generated by the LTM system are proper for encryptions, the NIST test is performed. The NIST SP800-22 test suite consists of 15 statistical tests. Each test calculates a *P*-value and compares it with a given significance level to determine whether the sequence is random. When applying the NIST test suite, a significance level $\alpha = 0.01$ is chosen for testing. If all the *P*-value $> \alpha$, then the sequence is considered to be random. We generate three sequences by using LTM and turn them into three binary sequences of length 1000000, 15 indicators were tested by using the NISTSP800-22 suite, and the minimum *P*-values of these sequences are listed in Table 1. From Table 1, we can see that the minimum *P*-value results are greater than the significance level α, indicating that the tests meet the requirements of SP800-22 randomness. Therefore, the random numbers generated by the LTM system are proper for encryptions.

B. THE IMPROVED ENCRYPTION ALGORITHM

In the improved image encryption algorithm, the SHA-3 hash value of the plaintext image is adopted and a new secret key

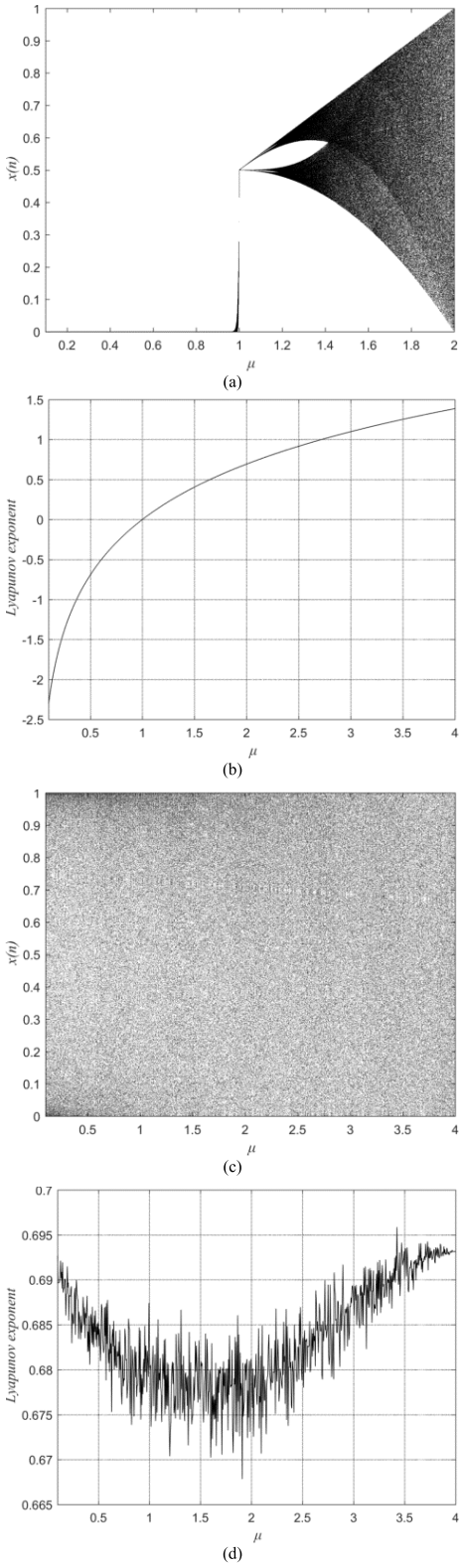

FIGURE 3. The bifurcation and Lyapunov exponent diagram of tent map and Logistic-tent map. (a) bifurcation diagram of tent map. (b) Lyapunov exponent diagram of tent map. (c) bifurcation diagram of Logistic-tent map. (d) Lyapunov exponent diagram of Logistic-tent map.

is added to the key set, which is related to the SHA-3 hash value. As a result, the key-streams (**S1**, **S2**, **S3**) are related to the plaintext image to be encrypted. The operation steps are as follows.

Step (1): Choose the secret keys $\{a, b, c, d, r_m, r_n, t, \mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}\}$.

Step (2): Read the $m \times n \times 3$ sized color plaintext image $\mathbf{P}_{m\times n\times 3} = [P(i, j, k)]$ $(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, k = 1, 2, 3)$, and convert the 3D matrix $\mathbf{P}_{m\times n\times 3}$ into a 2D matrix to get the gray image $\mathbf{PS}_{m\times 3n} = [PS(i, l)]$, where $i = 1, 2, \ldots, m$, $l = 1, 2, \ldots, 3 \times n$. The operation methods are the same as step (2) and (3) of the Wu's encryption algorithm.

Step (3): Use the SHA-3 hash algorithm to generate a 256-bit hash value H of the plaintext image, which can be divided into 32 blocks with the same size of 8-bit. Namely, $H = h_1 h_2 \ldots h_{32}$, and $h_i \in [0, 255]$, $i = 1, 2, \ldots, 32$. Calculate the parameter δ by using the hash value H as

$$\delta = (\sum_{i=1}^{i=32} h_i)/(32 \times 256), \tag{16}$$

and δ is also used as a secret key.

Step (4): Modify the initial values (x_{10}, x_{20}, x_{30}) as

$$\begin{cases} x_{10} = (x_{10} + \delta)/2, \\ x_{20} = (x_{20} + \delta)/2, \\ x_{30} = (x_{30} + \delta)/2. \end{cases} \tag{17}$$

Hence, the updated parameters (x_{10}, x_{20}, x_{30}) are related to the content of the plaintext image $\mathbf{P}_{m\times n\times 3}$.

Step (5): Permutate the gray image $\mathbf{PS}_{m\times 3n}$ to get a permutated image $\mathbf{PRT}_{m\times 3n}$. The specific operation method is as follows: For any position (x, y) of a pixel in $\mathbf{PS}_{m\times 3n}$, by iterating Eq.(2) for t times to get the position (x', y') of the pixel in $\mathbf{PRT}_{m\times 3n}$. Therefore, we obtain PRT(x', y') as $PRT(x', y') = PS(x, y)$. After the pixels in all positions are processed, the permutated image $\mathbf{PRT}_{m\times 3n}$ is obtained. Then split $\mathbf{PRT}_{m\times 3n}$ into three matrices $\mathbf{RRT}_{m\times n}$, $\mathbf{GRT}_{m\times n}$, and $\mathbf{BRT}_{m\times n}$ with size of $m\times n$. Further, convert $\mathbf{RRT}_{m\times n}$, $\mathbf{GRT}_{m\times n}$, and $\mathbf{BRT}_{m\times n}$ to three 1D vectors $\mathbf{R}_{N\times 1}$, $\mathbf{G}_{N\times 1}$, and $\mathbf{B}_{N\times 1}$ with size of $N \times 1$. Where $N = m \times n$.

Step (6): Calculate the three parameters R'_0, G'_0, and B'_0 as

$$\begin{cases} R'_0 = \left\lfloor (\sum_{i=2}^{i=N} R(i))/(N-1) \right\rfloor, \\ G'_0 = \left\lfloor (\sum_{i=2}^{i=N} G(i))/(N-1) \right\rfloor, \\ B'_0 = \left\lfloor (\sum_{i=2}^{i=N} B(i))/(N-1) \right\rfloor. \end{cases} \tag{18}$$

Step (7): Iterate the new chaotic system Eq.(15) for N+1000 times with the parameters $\{\mu_1, \mu_2, \mu_3\}$ and modified values $\{x_{10}, x_{20}, x_{30}\}$ respectively, and take the final N values to form three chaotic sequences **X1**, **X2**, **X3** of length N.

Step (8): Calculate three key-streams **S1**, **S2**, **S3** with **X1**, **X2**, **X3** by Eq.(5).

Step (9): Modify the key streams **S1**, **S2**, and **S3** as

$$\begin{cases} \mathbf{S1} = \mathbf{S1} \oplus [(\mathbf{S2} + \mathbf{S3}) \mod 256], \\ \mathbf{S2} = \mathbf{S2} \oplus [(\mathbf{S3} + \mathbf{S1}) \mod 256], \\ \mathbf{S3} = \mathbf{S3} \oplus [(\mathbf{S1} + \mathbf{S2}) \mod 256]. \end{cases} \tag{19}$$

Step (10): Encrypt the three components $\{R(i), G(i), B(i)\}$ for each pixel to obtain their corresponding cipher values $\{R'(i), G'(i), B'(i)\}$ as

$$\begin{cases} R'(i) = [(R(i) + G'(i-1)) \mod 256] \\ \qquad \oplus [(S1(i) + B'(i-1)) \mod 256], \\ G'(i) = [(G(i) + B'(i-1)) \mod 256] \\ \qquad \oplus [(S2(i) + R'(i-1)) \mod 256], \\ B'(i) = [(B(i) + R'(i-1)) \mod 256] \\ \qquad \oplus [(S3(i) + G'(i-1)) \mod 256]. \end{cases} \tag{20}$$

When $i = 1$, $R'(i-1)$, $G'(i-1)$ and $B'(i-1)$ are replaced by three parameters R'_0, G'_0, and B'_0 respectively, which are calculated by Eq. (18). By using Eq. (20), we make the relationship between ciphertext and plaintext more complex.

Step (11): Reshape three 1D vectors $\mathbf{R}'_{N\times1} = [R'(i)]$, $\mathbf{G}'_{N\times1} = [G'(i)]$, and $\mathbf{B}'_{N\times1} = [B'(i)]$ to three matrices $\mathbf{RC}_{m\times n}$, $\mathbf{GC}_{m\times n}$, $\mathbf{BC}_{m\times n}$, and use these three components to compose the final color cipher image $\mathbf{C}_{m\times n\times3}$.

C. THE IMPROVED DECRYPTION ALGORITHM

The decryption procedure is similar to that of the encryption, whereas with the reverse operational orders.

Step (1): Receive the secret keys, i.e. parameters set $\{a, b, c, d, r_m, r_n, t, \mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}, \delta\}$.

Step (2): Receive the color cipher image $\mathbf{C}_{m\times n\times3}$, and decompose the 3D matrix $\mathbf{C}_{m\times n\times3}$ into three 2D component matrices denoted as $\mathbf{RC}_{m\times n}$, $\mathbf{GC}_{m\times n}$, $\mathbf{BC}_{m\times n}$, respectively.

Step (3): Modify the initial parameters $\{x_{10}, x_{20}, x_{30}\}$ with δ by using Eq.(17).

Step (4): Iterate the new chaotic system Eq.(15) for N+1000 times with the modified initial values $\{x_{10}, x_{20}, x_{30}\}$ and system parameters $\{\mu_1, \mu_2, \mu_3\}$ respectively, and take the final N values to form three chaotic sequences **X1, X2, X3** of length N.

Step (5): Calculate three key-streams **S1**, **S2**, **S3** with **X1, X2, X3** by using Eq.(5) respectively, and modify **S1**, **S2**, **S3** by using Eq.(19) respectively.

Step (6): Reshape the three 2D matrices $\mathbf{RC}_{m\times n}$, $\mathbf{GC}_{m\times n}$, $\mathbf{BC}_{m\times n}$ to three 1D vectors $R'_{mn\times1}$, $G'_{mn\times1}$, $B'_{mn\times1}$ respectively.

Step (7): Inversely diffuse $R'_{mn\times1}$, $G'_{mn\times1}$, $B'_{mn\times1}$ and obtain partially three decrypted vectors $R_{mn\times1}, G_{mn\times1}, B_{mn\times1}$ as

$$\begin{cases} R(i) = [R'(i) \oplus [(S1(i) + B'(i-1)) \mod 256] \\ \qquad -G'(i-1)] \mod 256, \\ G(i) = [G'(i) \oplus [(S2(i) + R'(i-1)) \mod 256] \\ \qquad -B'(i-1)] \mod 256, \\ B(i) = [B'(i) \oplus [(S3(i) + G'(i-1)) \mod 256] \\ \qquad -R'(i-1)] \mod 256, \end{cases} \tag{21a}$$

Where $i = N, N-1, \ldots, 2$.

Step (8): Because now $\{R(2), R(3), \ldots, R(N)\}$, $\{G(2), G(3), \ldots, G(N)\}$, and $\{B(2), B(3), \ldots, B(N)\}$ are known, therefore, we can calculate the values of (R'_0, G'_0, B'_0) by using Eq.(18).

Step (9): Decrypte the first pixel values of $\{R(1), G(1), B(1)\}$ as

$$\begin{cases} R(1) = [R'(1) \oplus [(S1(1) + B'_0) \mod 256] \\ \qquad -G'_0] \mod 256 \\ G(1) = [G'(1) \oplus [(S2(1) + R'_0) \mod 256] \\ \qquad -B'_0] \mod 256 \\ B(1) = [B'(1) \oplus [(S3(1) + G'_0) \mod 256] \\ \qquad -R'_0] \mod 256 \end{cases} \tag{21b}$$

Step (10): Reshape $R_{mn\times1}, G_{mn\times1}, B_{mn\times1}$ to three matrices $\mathbf{RRT}_{m\times n}$, $\mathbf{GRT}_{m\times n}$, $\mathbf{BRT}_{m\times n}$, then stitch them to form a gray image $\mathbf{PRT}_{m\times3n}$.

Step (11): Inversely permutate the gray image $\mathbf{PRT}_{m\times3n}$ to get the un-permutated gray image $\mathbf{PS}_{m\times3n}$. Different from Wu's method, here we still use the improved 2D Arnold transform Eq.(2) instead of the inverse transform Eq.(4). The specific operation method is as follows: For any position (x, y) of a pixel in $\mathbf{PS}_{m\times3n}$, by iterating Eq.(2) for t times to get the position (x', y') of the pixel in $\mathbf{PRT}_{m\times3n}$. Therefore, we obtain $PS(x, y)$ as $PS(x, y) = PRT(x', y')$. After the pixels in all positions are processed, the un-permutated gray image $\mathbf{PS}_{\mathrm{m}\times3\mathrm{n}}$ is obtained.

Step (12): Split $\mathbf{PS}_{m\times3n}$ into three matrices, i.e. $\mathbf{RP}_{m\times n}$, $\mathbf{GP}_{m\times n}$, and $\mathbf{BP}_{m\times n}$.

Step (13): Finally, the decrypted color image $\mathbf{P}_{m\times n\times3}$ can be composed by its three components $\mathbf{RP}_{m\times n}$, $\mathbf{GP}_{m\times n}$, $\mathbf{BP}_{m\times n}$.

D. THEORETICAL ANALYSIS OF THE IMPROVED SCHEME

According to the processing principle, the improved scheme can overcome the following weakness of Wu's scheme.

1) The new combined chaotic system, Logistic-tent map (LTM), has better chaotic performance than tent map.
2) By introducing a parameter related to the SHA-3 hash value of the plaintext image as a secret key, the key-streams are associated with the image to be encrypted so that the improved algorithm can resist chosen-plaintext attacks.
3) By improving the method of generating parameters R'_0, G'_0, and B'_0, then these parameters can be accurately calculated on the decryption end so that the image can be completely decrypted.
4) By improving the method of generating key-streams **S1**, **S2**, and **S3**, make the improved algorithm more sensitive to each initial key in (μ_1, μ_2, μ_3, x_{10}, x_{20}, x_{30}).
5) By improving the encryption formula in diffusion process, we make the relationship between ciphertext and plaintext more complex.

V. TEST AND ANALYSIS FOR THE IMPROVED SCHEME

In our experimental tests, we choose the standard color plain images, such as Lena, Baboon, Peppers, *et al.*, with different sizes as the testing subjects. The secret keys are set as ($\mu_1 = 1.9$, $\mu_2 = 1.7$, $\mu_3 = 1.6$, $x_{10} = 0.1049306640625$, $x_{20} = 0.2049306640625$, $x_{30} = 0.3049306640625$; $a = 1$, $b = 3$,

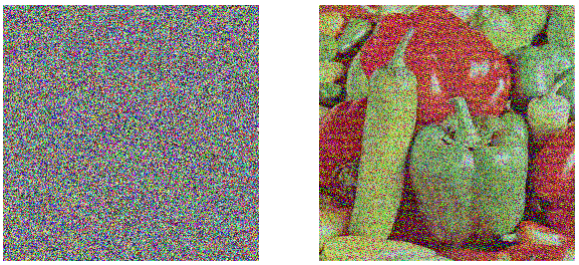

FIGURE 4. The decrypted image corresponding to one of the decryption keys $\{\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}\}$ with 10^{-10} error. (a) The improved algorithm. (b) The Wu's algorithm.

$c = 5$, $d = 16$, $r_m = 4$, $r_n = 7$, $t = 5$), and δ will be determined by the plaintext image to be encrypted.

A. KEY SPACE ANALYSIS

Key space size is the total number of different keys which can be used in a cryptosystem. For a good encryption algorithm, the key space should be large enough to make brute-force attack impossible. In the improved encryption scheme, the secret keys are $\mathbf{K1} = \{\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}; a, b, c, d, r_m, r_n, t\}$ and δ. The total number of different keys in **K1** is the same as those in Ref. [27], and it is (5×10^{102}). But δ is a new double-precision number introduced by the improved scheme, then the key space is $(5\times10^{102})\times10^{15}$, which is more than 2^{390}. Therefore the improved scheme has the bigger key space than Wu's scheme, and the key space is large enough to resist brute-force attacks.

B. KEY SENSITIVITY

A good encryption algorithm should be sensitive to the secret keys, namely, when the keys used in decryption are slightly different from the keys used in encryption, the plaintext image can not be decrypted correctly. To test the sensitivity of the improved encryption algorithm to the secret keys, we encrypted the $256\times256\times3$ color image Peppers by using the keys ($\mu_1 = 1.9$, $\mu_2 = 1.7$, $\mu_3 = 1.6$, $x_{10} = 0.201$, $x_{20} = 0.301$, $x_{30} = 0.401$; $a = 1$, $b = 3$, $c = 5$, $d = 16$, $r_m = 4$, $r_n = 7$, $t = 5$) and $\delta = 0.472900390625$. In the decryption process, we slightly change one of the parameters of $\{\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}\}$ and the change is only 10^{-10}, namely, $\mu_i' = \mu_i + 10^{-10}$ or $x_{i0}' = x_{i0} + 10^{-10}$ ($i = 1, 2, 3$), where μ_i' and x_{i0} are the encryption keys, while μ_i' and x_{i0}' are the decryption keys. For example, when $\mu_1' = 1.9000000001$ and the rest of the key parameters remain unchanged, the decrypted image by our improved algorithm and Wu's algorithm are shown in Figs. 4(a) and 4(b) respectively. Changing one of the other parameters, the results of decryption are similar to Fig.4.

From Fig.4, one can see that the decrypted image obtained by our improved algorithm with one key in decryption process has a slight error is meaningless at all. However, the decrypted image obtained by the Wu's algorithm with one key in decryption process has a slight error exposes the plaintext information. Hence, the Wu's algorithm is insensitive to the secret keys and unsafe. The reasons why the Wu's algorithm is not sensitive to the secret keys are as follows: When one parameter in $\{\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}\}$ has a slight error in the decryption process, only one keystream in $\{\mathbf{S1}, \mathbf{S2}, \mathbf{S3}\}$ will change accordingly, and only one component in $\{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$ won't be decrypted accurately. However, in our improved algorithm, thanks to the introduction of Eq.(19), changing any parameter in $\{\mu_1, \mu_2, \mu_3, x_{10}, x_{20}, x_{30}\}$ will affect all of the key streams **S1**, **S2** and **S3**. Therefore, our improved algorithm is more secure than Wu's algorithm.

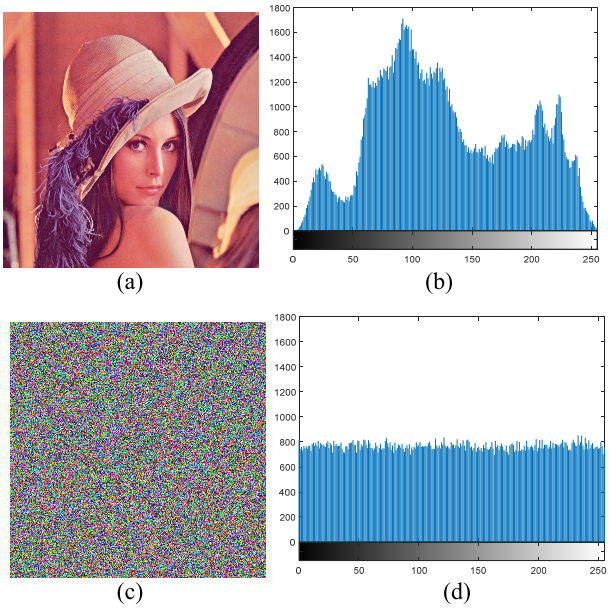

FIGURE 5. Experiment results of the proposed scheme. (a) The Lena image. (b) The histogram of the Lena image. (c) The encrypted Lena image. (d) The histogram of the encrypted Lena image.

C. DISTRIBUTION OF THE CIPHERTEXT

An image histogram displays the distribution of the values of its pixels, and it provide some statistical information of the image. Figs. 5(a) and 5(b) depict the plain-image Lena and its corresponding histogram respectively. While Figs. 5(c) and 5(d) depict the cipher-image Lena and its corresponding histogram respectively. It can be seen from Fig.5 that the distribution of the values of pixels in the plain-image is uneven. However, the distribution of the pixel values in the cipher-image is nearly uniform, and hence the cipher-image can well protect the information of the image to withstand the statistical analysis attack.

For quantity analyses of the performance of pixel values distribution, we introduce variances of histograms to evaluate uniformity of ciphered images. The variance of histograms is presented as [33]

$$\mathrm{var}(\mathbf{Z}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(z_i - z_j)^2 \tag{22}$$

where $\mathbf{Z}$ is the vector of the histogram values and $\mathbf{Z} = \{z_1, z_2, \ldots, z_{256}\}$, z_i and z_j are the numbers of pixels

TABLE 2. Variances of the histograms of the Lena (512 × 512).

Images	Variance
Plain image Lena	632097.4766
Cipher image of ours	3485.1953
Cipher image of Ref.[33]	5335.8309
Cipher image of Ref.[34]	5554.8219

which gray values are equal to i and j respectively. The lower value of variances indicates the higher uniformity of ciphered images.

In simulating experiments, we calculate variances of histograms of Lena plain image (size of 512×512) and its ciphered image by Eq. (22). The variances of the histograms of the Lena plain image and its corresponding cipher images encrypted by three different encryption algorithms are listed in Table 2. From the results, we can see that the variance of the cipher image Lena obtained with our algorithm is the lowest, that is 3485.1953, and is much less than that of Zhang's algorithm [33] and Zhu's algorithm [34]. Thus, our improved color image encryption algorithm is more efficient and secure.

D. CORRELATION ANALYSIS OF TWO ADJACENT PIXELS

A meaningful image usually has a large degree of correlation between any adjacent pixel pairs. To enhance the resistivity to statistical analysis attacks, a good encrypted image should reduce the correlation as much as possible. As a experimental test, we select all pairs of two-adjacent pixels (in vertical, horizontal, and diagonal direction) from Lena cipher-image, and calculate the correlation coefficients as Wu *et al*. did in [27]. The cipher-image which has smaller absolute values of correlation coefficient has better performance in resisting statistical attack. The test results about the correlation coefficients have been given in Table 3 and compared with some references. We can see that our improved scheme shows better performance than the other ones.

E. SHANNON ENTROPY ANALYSIS

Shannon entropy [35] is usually used to measure the randomness of the gray values of an image. For an 8-bit image, Shannon entropy is defined as

$$H(m) = -\sum_{i=0}^{255} P(m_i)\log_2[P(m_i)]. \tag{23}$$

Where m_i represents the ith gray value, while $P(m_i)$ is the probability of value m_i existing in the image. Obviously, for an 8-bit completely random image, $P(m_i) = 1/256$ and the entropy is 8. A good encryption algorithm should make the Shannon entropy of the ciphertext image very close to 8. For a color image with a size of $m \times n \times 3$, we convert it into a gray image with size of $m \times 3n$ to compute Shannon entropy.

TABLE 3. Correlation coefficients of two adjacent pixels.

Scheme	Directions	R	G	B
Ref.[12]	H	0.0049	0.0054	0.0053
	V	0.0031	0.0001	0.0022
	D	0.0007	0.0017	0.0007
Ref.[26]	H	-0.046355	0.043590	0.013696
	V	-0.058775	-0.068220	-0.068896
	D	-0.020083	-0.005259	0.012714
Ref.[27]	H	0.000538	0.001186	-0.002372
	V	-0.007058	0.000177	0.007818
	D	0.000573	-0.001693	-0.000927
This paper	H	-0.002026	-0.000836	-0.001374
	V	0.000805	-0.001615	0.002729
	D	-0.002971	-0.000780	0.000648

TABLE 4. Shannon entropy of different cipher image encrypted by different schemes.

Schemes	Lena	Baboon	Girl	Couple
This paper	7.999083	7.999784	7.998991	7.999001
Ref.[12]	7.9972	7.9972	N/A	N/A
Ref.[26]	7.990966	7.991296	7.991575	7.991349
Ref.[27]	7.997287	7.998973	7.998989	7.999136
Ref.[36]	7.9970	7.9969	N/A	N/A
Ref.[37]	7.9971	7.9974	N/A	N/A

The results of different ciphertext images encrypted by different related algorithms are listed in Table 4. Note that our improved algorithm obtains the highest entropy in most cases, which means that our improved algorithm leaks the least information among the three ones.

F. ROBUSTNESS AGAINST DIFFERENTIAL ATTACK

Sometimes, attackers encrypt two plaintext images with the same encryption algorithm, and the two plaintext images only have slight differences. Then attackers try to find out the relation between plain image and its cipher image by comparing the two encrypted images. We refer to this cryptanalysis method as differential attack. In order to verify whether the improved algorithm can resist differential attack, two commonly used metrics are cited, which are the number of pixel changing rate (*NPCR*) and the unified averaged changed intensity (*UACI*) [38]. Their definitions are as follows.

$$NPCR = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j)}{m \times n} \times 100\% \tag{24}$$

$$UACI = \frac{1}{m \times n}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|C2(i,j) - C1(i,j)|}{255}\right) \times 100\% \tag{25}$$

TABLE 5. Values of NPCR and UACI for Lena cipher images.

Position	$NPCR^{[27]}$	$UACI^{[27]}$	$NPCR$	$UACI$
(1,1,1)	0.9971	0.3345	1	0.3344
(1,256,1)	0.9973	0.3346	1	0.3340
(256,256,3)	0.9973	0.3332	1	0.3343
(256,1,1)	0.9971	0.3353	1	0.3348
(128,128,1)	0.9963	0.3342	0.9999	0.3340
(28,128,2)	0.9973	0.3345	1	0.3347

Where m, n represent the total number of rows and columns of pixels in the image respectively. $C1(i, j)$ and $C2(i, j)$ are pixel values at the same position (i, j) of the two encrypted images mentioned above, and $D(i, j)$ is computed by

$$D(i,j) = \begin{cases} 1, & if\ C1(i,j) \neq C2(i,j), \\ 0, & if\ C1(i,j) = C2(i,j). \end{cases} \quad (26)$$

The ideal value of $NPCR$ is close to 1, and the ideal value of $UACI$ is close to 0.3346 [38]. The greater the value of $NPCR$ and $UACI$, the better the performance of the algorithm resists differential attack.

In the experiment, We have randomly chosen six different pixels (one at a time, including the first and last positions) in Lena plain image and changed its value slightly (by adding 1). Table 5 shows the $NPCR$ and $UACI$ values. We can see that the $NPCR$ values are equal to 1 and $UACI$ values are also very close to 0.3346. Compared with the results in [27], our improved algorithm has better performance than Wu's algorithm in resisting differential attacks.

Table 6 compares the $NPCR$ and $UACI$ values of different encryption scheme for some classical standard test images under the circumstance of the first pixels with the changes of the least significant bits. From the results of Table 6, we know that our proposed scheme has a larger $NPCR$ value compared with other schemes, which implies that our improved scheme can resist stronger differential attack.

G. RESISTANCE TO CLASSICAL TYPES OF ATTACKS

According to the definitions of four classical types of attacks, chosen plaintext attack is the most powerful attack. If a cryptosystem can resist this attack, it can resist other types of attack [32].

In our improved scheme, the key-streams (**S1**, **S2**, **S3**) are related to the plaintext image to be encrypted. Even if the attacker cracked the key-streams (**S1**, **S2**, **S3**) with some special selected plaintext images, the key streams (**S1**, **S2**, **S3**) can not be used to decrypt the target ciphertext image, because different images have different key-streams (**S1**, **S2**, **S3**). Further more, in the diffusion process, the encrypted value is not only related to the corresponding plain value and the key but also related to the former plain value and former ciphered value. This means different ciphered image has different former plain value and former ciphered value. So, the improved algorithm can resist the chosen plain- text/ciphertext attack, and can well resist the four classical types of attacks.

TABLE 6. NPCR and UACI values comparison for different encryption schemes.

Schemes	Lena	Baboon	Peppers	Barbara
$NPCR$	1	1	1	1
$UACA$	0.3344	0.3346	0.3342	0.3342
$NPCR^{[1]}$	0.9962	0.9943	0.9964	0.9960
$UACA^{[1]}$	0.3377	0.3353	0.3353	0.3341
$NPCR^{[12]}$	0.9966	0.9965	0.9963	N/A
$UACA^{[12]}$	0.3344	0.3350	0.3347	N/A
$NPCR^{[27]}$	0.9971	0.9972	0.9974	0.9978
$UACA^{[27]}$	0.3345	0.3349	0.3353	0.3350
$NPCR^{[36]}$	0.9965	N/A	N/A	N/A
$UACA^{[36]}$	0.3348	N/A	N/A	N/A
$NPCR^{[37]}$	0.9965	0.9960	0.9959	0.9965
$UACA^{[37]}$	0.3348	0.3357	0.3351	0.3332

H. ANALYSIS OF SPEED

In applications, a practical algorithm should be fast. In our experimental tests, several 24 bits color images with different size have been used to measure the time cost of our improved algorithm in encrypting or decrypting an image.

Our experimental tests run on a desktop PC with Intel(R) Core(TM) i5-4590 3.30 GHz CPU, 4 GB RAM and 500 GB hard disk. The operating system is 64 bits Microsoft Windows 7, and the computational platform is Matlab R2016b. The average time taken by our improved algorithm for encrypting (or decrypting) the images with size of 256 $\times$ 256, 512 $\times$ 512 and 1024 $\times$ 1024 are 0.58, 2.28 and 9.16 seconds, respectively. Considering its high level of security, the speed of image encryption or decryption processing is acceptable.

VI. CONCLUSION

In this paper, a color image encryption algorithm is analyzed and cracked by using chosen-plaintext attacks. Further, we proposed an improved color image encryption algorithm. The improved algorithm includes the following three major improvements. Firstly, A new combined chaotic system called Logistic-tent map (LTM) is proposed, which has better chaotic performance than tent map. Secondly, the new chaotic system is applied to the improved encryption scheme. Thirdly, by improving the key generation method and encryption strategy, the new encryption scheme can overcome the security defects of the original encryption scheme. The analytical and experimental results show that the improved algorithm can significantly improve the security of encryption images while still possessing all the merits of the Wu's algorithm, which has a better potential for application. The improved image encryption algorithm proposed in this paper is suitable for encryption of color images with high security requirements, and is also suitable for gray images encryption.

ACKNOWLEDGMENT

The authors would like to thank their collaborator, Prof. Guojun Wang at Guangzhou University, who also contributed to this paper.

REFERENCES

[1] X.-J. Tong, M. Zhang, Z. Wang, and J. Ma, "A joint color image encryption and compression scheme based on hyper-chaotic system," *Nonlinear Dyn.*, vol. 84, no. 4, pp. 2333–2356, 2016.
[2] J. Tang, A. Liu, M. Zhao, and T. Wang, "An aggregate signature based trust routing for data gathering in sensor networks," *Secur. Commun. Netw.*, vol. 2018, Jan. 2018, Art. no. 6328504, doi: 10.1155/2018/6328504.
[3] X. Liu, N. Xiong, N. Zhang, A. Liu, H. Shen, and C. Huang, "A trust with abstract information verified routing scheme for cyber-physical network," *IEEE Access*, vol. 6, pp. 3882–3898, 2018, doi: 10.1109/ACCESS.2018.2799681.
[4] Y. Zhang, D. Xiao, Y. Shu, and J. Li, "A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations," *Signal Process., Image Commun.*, vol. 28, no. 3, pp. 292–300, 2013.
[5] L. Wang, H. Song, and P. Liu, "A novel hybrid color image encryption algorithm using two complex chaotic systems," *Opt. Lasers Eng.*, vol. 77, pp. 118–125, Feb. 2016.
[6] Y. Liu, X.-J. Tong, and J. Ma, "Image encryption algorithm based on hyper-chaotic system and dynamic S-box," *Multimedia Tools Appl.*, vol. 75, no. 13, pp. 7739–7759, 2015.
[7] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Opt. Lasers Eng.*, vol. 78, pp. 17–25, Mar. 2016.
[8] X. Wang and H.-L. Zhang, "A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems," *Nonlinear Dyn.*, vol. 83, nos. 1–2, pp. 333–346, 2016.
[9] T. Xiang, J. Hu, and J. Sun, "Outsourcing chaotic selective image encryption to the cloud with steganography," *Digit. Signal Process.*, vol. 43, pp. 28–37, Aug. 2015.
[10] Z. Hua and Y. Zhou, "Image encryption using 2D logistic-adjusted-sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.
[11] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.
[12] 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.
[13] J. Wu, X. Liao, and B. Yang, "Color image encryption based on chaotic systems and elliptic curve ElGamal scheme," *Signal Process.*, vol. 141, pp. 109–124, Dec. 2017.
[14] X. Wang, C. Liu, and H. Zhang, "An effective and fast image encryption algorithm based on Chaos and interweaving of ranks," *Nonlinear Dyn.*, vol. 84, no. 3, pp. 1595–1607, 2016.
[15] X. Chai, "An image encryption algorithm based on bit level Brownian motion and new chaotic systems," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 1159–1175, 2017.
[16] X. Huang, "Image encryption algorithm using chaotic Chebyshev generator," *Nonlinear Dyn.*, vol. 67, no. 4, pp. 2411–2417, 2012.
[17] X. Wang, C. Liu, and D. Xu, "Image encryption scheme using chaos and simulated annealing algorithm," *Nonlinear Dyn.*, vol. 84, no. 3, pp. 1417–1429, 2016.
[18] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognit. Lett.*, vol. 31, no. 5, pp. 347–354, 2010.
[19] C. Zhu, "A novel image encryption scheme based on improved hyper-chaotic sequences," *Opt. Commun.*, vol. 285, no. 1, pp. 29–37, 2012.
[20] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Comput. Math. Appl.*, vol. 59, no. 10, pp. 3320–3327, 2010.
[21] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Opt. Commun.*, vol. 284, nos. 16–17, pp. 3895–3903, 2011.
[22] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft. Comput.*, vol. 12, no. 5, pp. 1457–1466, 2012.
[23] X.-Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dyn.*, vol. 62, no. 3, pp. 615–621, 2010.
[24] Y. Zhang and D. Xiao, "An image encryption scheme based on rotation matrix bit-level permutation and block diffusion," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 1, pp. 74–82, 2014.
[25] Y. Zhang and D. Xiao, "Double optical image encryption using discrete Chirikov standard map and chaos-based fractional random transform," *Opt. Lasers Eng.*, vol. 51, no. 4, pp. 472–480, Apr. 2013.
[26] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 127–133, 2017.
[27] 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, 2017.
[28] 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, Mar. 2017.
[29] C. Li, Y. Liu, T. Xie, and M. Z. Q. Chen, "Breaking a novel image encryption scheme based on improved hyperchaotic sequences," *Nonlinear Dyn.*, vol. 73, no. 3, pp. 2083–2089, 2013.
[30] X. Wang, D. Luan, and X. Bao, "Cryptanalysis of an image encryption algorithm using Chebyshev generator," *Digit. Signal Process.*, vol. 25, pp. 244–247, Feb. 2014.
[31] Y.-Q. Zhang and X.-Y. Wang, "A new image encryption algorithm based on non-adjacent coupled map lattices," *Appl. Soft. Comput.*, vol. 26, pp. 10–20, Jan. 2015.
[32] 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.
[33] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice," *Inf. Sci.*, vol. 273, pp. 329–351, Jul. 2014.
[34] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, 2011.
[35] P. Zhen, G. Zhao, L. Min, and X. Jin, "Chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools Appl.*, vol. 75, no. 11, pp. 6303–6319, 2016.
[36] X.-Y. Wang, Y.-Q. Zhang, and X.-M. Bao, "A novel chaotic image encryption scheme using DNA sequence operations," *Opt. Lasers Eng.*, vol. 73, pp. 53–61, Oct. 2015.
[37] X. Wang, L. Liu, and Y. Zhang, "A novel chaotic block image encryption algorithm based on dynamic random growth technique," *Opt. Lasers Eng.*, vol. 66, pp. 10–18, Mar. 2015.
[38] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary J. Sci. Technol., J. Select. Areas Telecommu.*, vol. 1, pp. 31–38, Apr. 2011.

CONGXU ZHU received the Ph.D. degree in computer science and technology from Central South University, China, in 2006. He is currently a Professor with Central South University. His research interests include chaos theory and its applications in information security, chaos-based cryptography, image processing, and multimedia and network security.

KEHUI SUN received the Ph.D. degree in control theory and control engineering from Central South University, China, in 2005. He is currently a Professor with Central South University. His research interests include chaos theory and its applications in secure multimedia information communications.