Hindawi
Security and Communication Networks
Volume 2018, Article ID 2708532, 13 pages
https://doi.org/10.1155/2018/2708532

WILEY | Hindawi

Research Article

A New Chaos-Based Color Image Encryption Scheme with an Efficient Substitution Keystream Generation Strategy

Chong Fu, Gao-yuan Zhang, Mai Zhu, Zhe Chen, and Wei-min Lei

School of Computer Science and Engineering, Northeastern University, Shenyang 110004, China

Correspondence should be addressed to Chong Fu; fuchong@mail.neu.edu.cn

Received 10 August 2017; Accepted 3 January 2018; Published 20 February 2018

Academic Editor: Leo Y. Zhang

This paper suggests a new chaos-based color image cipher with an efficient substitution keystream generation strategy. The hyperchaotic Lü system and logistic map are employed to generate the permutation and substitution keystream sequences for image data scrambling and mixing. In the permutation stage, the positions of colored subpixels in the input image are scrambled using a pixel-swapping mechanism, which avoids two main problems encountered when using the discretized version of area-preserving chaotic maps. In the substitution stage, we introduce an efficient keystream generation method that can extract three keystream elements from the current state of the iterative logistic map. Compared with conventional method, the total number of iterations is reduced by 3 times. To ensure the robustness of the proposed scheme against chosen-plaintext attack, the current state of the logistic map is perturbed during each iteration and the disturbance value is determined by plain-pixel values. The mechanism of associating the keystream sequence with plain-image also helps accelerate the diffusion process and increase the degree of randomness of the keystream sequence. Experimental results demonstrate that the proposed scheme has a satisfactory level of security and outperforms the conventional schemes in terms of computational efficiency.

1. Introduction

Nowadays, digital image information has been widely communicated over the Internet and wireless networks owing to the rapid advancements in the multimedia and communication technology. Meanwhile, the protection of digital image information against illegal usage has become an important issue. A direct and obvious way to protect image data from unauthorized eavesdropping is to employ an encryption algorithm. Unfortunately, the renowned block ciphers, such as Triple-DES, AES, and IDEA, are not suitable for practical image encryption. This is because the security of these algorithms is mainly ensured by their high computational cost, making them hard to meet the demand for online communications when dealing with digital images characterized by bulk data capacity. To meet this challenge, many different encryption technologies have been proposed. Among them, the chaos-based algorithms provide an optimal trade-off between security and efficiency. The first chaos-based image encryption scheme was suggested by Fridrich in 1998 [1]. The permutation-substitution network, introduced by Claude Shannon in his classic 1949 paper, Communication Theory of Secrecy Systems, and now a guiding principle for the design of a secure cipher, is adopted in her approach. In each round of the cipher, the pixel positions are firstly scrambled in a secret way, which leads to a great reduction in the correlation among neighboring pixels. Then, the pixel values are altered sequentially and the influence of each pixel is diffused to all its succeeding ones during the modification process. With such a structure, a minor change in one pixel of the plain-image may result in a totally different cipher-image with several overall rounds of encryption.

Conventionally, three area-preserving invertible chaotic maps, that is, the cat map, the baker map, and the standard map, are widely used for image scrambling. Unfortunately, this kind of permutation strategy suffers from two main disadvantages, that is, the periodicity of discretized version of chaotic maps and applicability to only square images [2–4]. To address these two drawbacks, Fu et al. [5] suggested an image scrambling scheme using a chaotic sequence sorting mechanism. Unfortunately, this method takes a whole row/column of an image as the scrambling unit and results in

weaker confusion effect compared with many of the existing schemes working on individual pixels. In [6], inspired by the natural ripple-like phenomenon that distorts a reflection on a water surface, Wu et al. suggested a novel scrambling algorithm that shuffle images in an n dimensional (n D) space using wave perturbations. In [7], the original pixel level matrix is considered as a natural 3D bit matrix, and a new 3D bit-level permutation algorithm is proposed. During the permutation stage, the original and target bit locations are both randomly selected to further enhance the permutation effect.

In the substitution stage, various discrete chaotic maps and continuous chaotic systems can be employed to generate keystream sequences with desired statistical properties, including the most commonly used ones like the logistic map [2], the Lorenz system [8], the Chen system [9], and varieties of high dimensional chaotic systems [10]. Obviously, low dimensional chaotic maps, especially the logistic map, have the advantages of simplicity and high efficiency but suffer from small key space; in contrast, high dimensional chaotic systems, especially the hyperchaotic systems, provide sufficiently large key space but at the expense of computations. Recently, it has been reported that many existing image encryption schemes have been successfully broken by using known/chosen-plaintext attacks [11–14]. This is due to the fact that the substitution keystream sequences used in these schemes are solely determined by the secret key. That is, the same keystream sequence is used to encrypt different plain-images unless a different secret key is used. Consequently, the keystream sequence may be determined by encrypting some specially created images (e.g., an image with all pixels having the same value) and then comparing them with their corresponding outputs. Obviously, if a keystream sequence depends on both the secret key and the plaintext, then such analysis may become impractical. For instance, in [15], the keystream elements are extracted from multiple-time iteration of the logistic map, and the iteration times are determined by plain-pixel values. Unfortunately, the redundant iteration operations downgrade the efficiency of the cryptosystem to some extent. In [16], the value of each keystream element is dynamically altered according to the plain-pixel values during the substitution process.

To better meet the challenge of online secure image communications, much research has been done on improving the efficiency of chaos-based image ciphers. For instance, in [17], Xiang et al. investigated the feasibility of selective image encryption on a bit-plane. It is concluded that only selectively encrypting the higher four bit-planes of an image can achieve an acceptable level of security. As only 50% of the whole image data are encrypted, the execution time is reduced. In [18], Wong et al. proposed a more efficient diffusion mechanism using simple table lookup and swapping techniques as a light-weight replacement of the 1D chaotic map iteration. Following this work, Chen et al. [19] presented an efficient image encryption scheme with confusion and diffusion operations being both performed based on a lookup table. The other advantage of their approach is that it can effectively tolerate the channel errors, which may lead to the corruption of cipher data. It has been demonstrated that images recovered from the damaged cipher data have satisfactory visual perception. In [20], Fu et al. introduced a novel bidirectional diffusion strategy to minimize the number of encryption rounds needed to spread the influence of each individual pixel over the entire cipher-image. Experimental results have demonstrated that their scheme takes one round of permutation and two rounds of substitution to obtain a satisfactory diffusion effect. In [16, 21–23], chaos-based image ciphers using a bit-level permutation were suggested. Owing to the substitution effect introduced in the permutation stage, the number of iteration rounds required by the time-consuming substitution procedure is reduced, and hence a shorter encryption time is needed. In [8], Fu et al. suggested a fast chaos-based image cipher with the permutation key determined by the hash value of the original image. Owing to the avalanche property of hash function, completely different shuffled images will be produced even if there is a tiny difference between the original ones, thereby accelerating the diffusion process. In [24], Chen et al. presented a novel image encryption scheme using a Gray-code-based permutation. Taking full advantage of (n, p, k)-Gray-code achievements, the new permutation strategy provides superior computational efficiency. In [25], Hua et al. introduced an image encryption algorithm based on a new two-dimensional sine logistic modulation map (2D-SLMM). Compared with corresponding seed maps, the new map has wider chaotic range, more parameters, and complex chaotic properties while remaining of relatively low implementation cost. Accordingly, the algorithm provides a good trade-off between security and efficiency.

Conventionally, in the substitution stage, one keystream element is obtained from the current value of a state variable of an iterative chaotic system. That is, to generate a keystream sequence of length m, a n-dimensional chaotic system should be iterated $T = \text{round}(m/n)$ times. In the present paper, we introduce an efficient logistic map-based keystream generation strategy that can simultaneously extract three keystream elements from the current state of the map. As a result, the total number of iterations is reduced by 3 times and the encryption time is shortened. In the permutation stage, the positions of subpixel in each color channel of the plain-image are scrambled across the entire color space using a pixel-swapping strategy under the control of a keystream sequence generated from the hyperchaotic Lü system. To ensure the robustness of the proposed scheme against chosen-plaintext attack, the current state of the logistic map is perturbed during each iteration and the disturbance value is determined by plain-pixel values. The mechanism of associating the keystream sequence with plain-image also helps accelerate the diffusion process and increase the degree of randomness of the keystream sequence. Experimental results demonstrate that the proposed scheme has a satisfactory level of security and outperforms the conventional schemes in terms of computational efficiency.

The rest of this paper is organized as follows. The proposed permutation and substitution algorithms are thoroughly described in Sections 2 and 3, respectively. In Section 4, the degree of randomness of the substitution keystream sequences generated using our proposed method

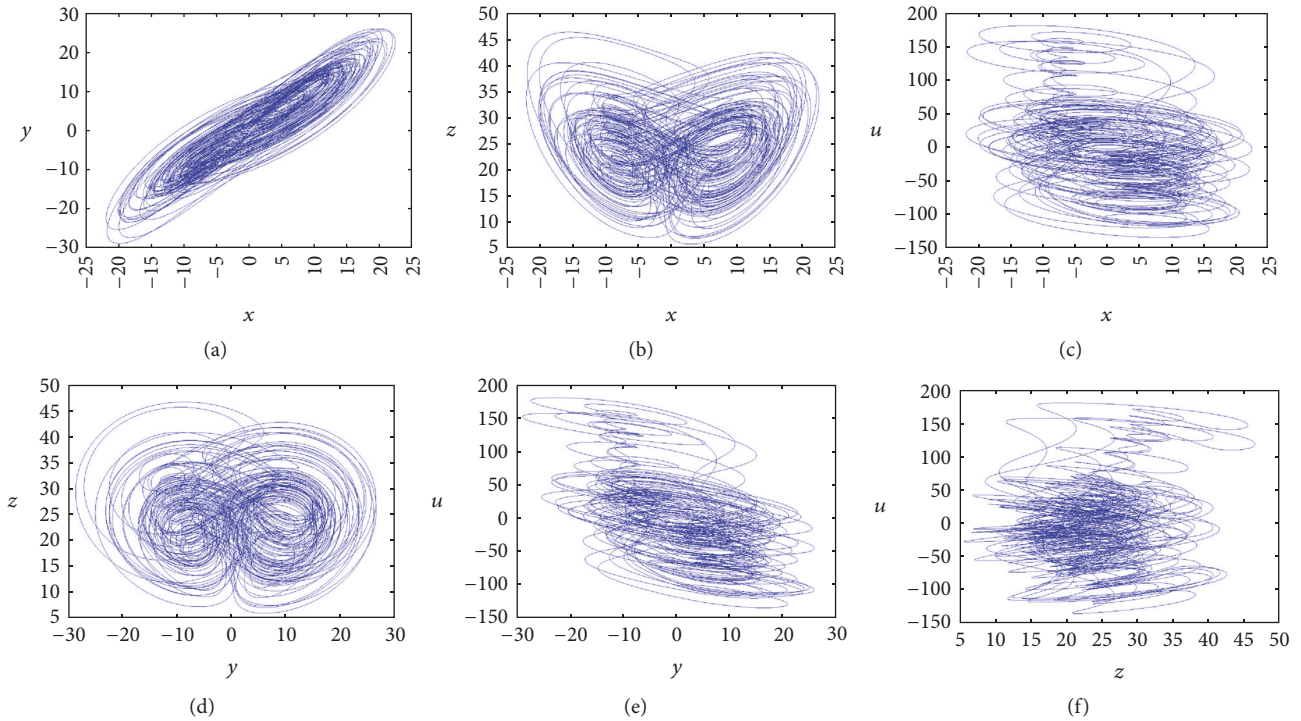

FIGURE 1: The projections of phase portrait of system (1) with $a = 36$, $b = 3$, $c = 20$, and $d = 1.3$. (a) x-y plane. (b) x-z plane. (c) x-u plane. (d) y-z plane. (e) y-u plane. (f) z-u plane.

is evaluated by using NIST test suite. In Section 5, the confusion and diffusion performance of the proposed cryptosystem are analyzed. The security and efficiency of the cryptosystem are analyzed in Sections 6 and 7, respectively. Finally, Section 8 concludes the paper.

2. Color Image Permutation Using a Pixel-Swapping Strategy

The hyperchaotic Lü system [26], which is employed in our scheme to generate the permutation keystream sequence, is described by

$$\begin{aligned} \dot{x} &= a(y - x) + u, \\ \dot{y} &= -xz + cy, \\ \dot{z} &= xy - bz, \\ \dot{u} &= xz + du, \end{aligned} \tag{1}$$

where a, b, c are the constants of Lü system [27] and d is a control parameter. When $a = 36$, $b = 3$, $c = 20$, and $-0.35 < d \leq 1.30$, the system exhibits a hyperchaotic behavior, and the projections of its phase portrait are shown in Figure 1. Evidently, the initial conditions (x_0, y_0, z_0, u_0) of the system are the immediate candidate for the secret key for permutation, as they uniquely determine a chaotic trajectory from which the permutation keystream is extracted.

Without loss of generality, a 24-bit RGB color image of size $H \times W$ is used as an input. The detailed permutation process is described as follows.

Step 1. Arrange the colored subpixels in the input image to a one-dimensional byte array imgData $= \{p_0, p_1, \ldots, p_{3\times H\times W-1}\}$ in the order from left to right, top to bottom.

Step 2. Generate a chaotic sequence of length $L_{\text{perm}} =$ len (imgData) -1 by iterating system (1), where len(x) returns the length of sequence x.

Step 2.1. Preiterate system (1) for N_0 times to avoid the harmful effect of transitional procedure, where N_0 is a constant. The system can be numerically solved by using fourth-order Runge-Kutta method, as given by

$$\begin{aligned} x_{n+1} &= x_n + \left(\frac{h}{6}\right)(K_1 + 2K_2 + 2K_3 + K_4), \\ y_{n+1} &= y_n + \left(\frac{h}{6}\right)(L_1 + 2L_2 + 2L_3 + L_4), \\ z_{n+1} &= z_n + \left(\frac{h}{6}\right)(M_1 + 2M_2 + 2M_3 + M_4), \\ u_{n+1} &= u_n + \left(\frac{h}{6}\right)(N_1 + 2N_2 + 2N_3 + N_4), \end{aligned} \tag{2}$$

where

$$
\begin{aligned}
K_j &= a\left(y_n - x_n\right) + u_n,\\
L_j &= -x_n z_n + c y_n,\\
M_j &= x_n y_n - b z_n,\\
N_j &= x_n z_n + d u_n,\\
&\qquad\qquad \left(\text{with } j = 1\right),\\
K_j &= a\left[\left(y_n + \frac{hL_{j-1}}{2}\right) - \left(x_n + \frac{hK_{j-1}}{2}\right)\right]\\
&\quad + \left(u_n + \frac{hN_{j-1}}{2}\right),\\
L_j &= -\left(x_n + \frac{hK_{j-1}}{2}\right)\left(z_n + \frac{hM_{j-1}}{2}\right)\\
&\quad + c\left(y_n + \frac{hL_{j-1}}{2}\right),\\
M_j &= \left(x_n + \frac{hK_{j-1}}{2}\right)\left(y_n + \frac{hL_{j-1}}{2}\right)\\
&\quad - b\left(z_n + \frac{hM_{j-1}}{2}\right),\\
N_j &= \left(x_n + \frac{hK_{j-1}}{2}\right)\left(z_n + \frac{hM_{j-1}}{2}\right)\\
&\quad + d\left(u_n + \frac{hN_{j-1}}{2}\right),\\
&\qquad\qquad \left(\text{with } j = 2, 3\right),\\
K_j &= a\left[\left(y_n + hL_{j-1}\right) - \left(x_n + hK_{j-1}\right)\right]\\
&\quad + \left(u_n + hN_{j-1}\right),\\
L_j &= -\left(x_n + hK_{j-1}\right)\left(z_n + hM_{j-1}\right) + c\left(y_n + hL_{j-1}\right),\\
M_j &= \left(x_n + hK_{j-1}\right)\left(y_n + hL_{j-1}\right) - b\left(z_n + hM_{j-1}\right),\\
N_j &= \left(x_n + hK_{j-1}\right)\left(z_n + hM_{j-1}\right) + d\left(u_n + hN_{j-1}\right),\\
&\qquad\qquad \left(\text{with } j = 4\right),
\end{aligned}
\tag{3}
$$

and the step h is chosen as 0.005.

Step 2.2. Continue the iteration for $I_{\text{perm}} = \lceil L_{\text{perm}}/4 \rceil$ times, where $\lceil x \rceil$ returns the least integer greater than or equal to x. For each iteration, the current values of the four state variables x, y, z, and u are in turn stored into an array permSeq $= \{ps_0, ps_1, ps_2, \ldots, ps_{3\times H\times W-2}\}$. Obviously, the last $R_{\text{perm}} = 4 \times I_{\text{perm}} - (3 \times H \times W - 1)$ element(s) produced at the final iteration step is redundant and should be discarded.

Step 3. Extract a permutation keystream sequence permKstr $= \{pk_0, pk_1, \ldots, pk_{3\times H\times W-2}\}$ from permSeq according to

$$
\begin{aligned}
pk_m = \text{pos}\left(pk_m\right) + \big(1 + \text{mod}\big(\text{sig}\left(\text{abs}\left(ps_m\right), \alpha\right),\\
\left(\left(\text{len}\left(\text{imgData}\right) - 1\right) - \text{pos}\left(pk_m\right)\right)\big)\big),
\end{aligned}
\tag{4}
$$

where $\text{pos}(pk_m)$ returns the position of pk_m in permKstr, that is, m, $\text{abs}(x)$ returns the absolute value of x, $\text{sig}(x, \alpha)$ returns the α most significant decimal digits in x, and $\text{mod}(x, y)$ divides x by y and returns the remainder of the division. An α value of 15 is recommended as all the state variables in our scheme are declared as double-precision type. From (4) it can be seen that pk_m is in the range between $(\text{pos}(pk_m) + 1)$ and $(\text{len}(\text{imgData}) - 1)$. That is, the swapping target for each colored subpixel (except the last one) in imgData will be pseudorandomly chosen from all its succeeding ones.

Step 4. Scramble imgData by swapping each subpixel p_m $(m = 0, 1, \ldots, 3\times H\times W-2)$ with another subpixel located at pk_m.

As can be seen from the above description, the proposed permutation scheme well addresses the two problems encountered when using the discretized version of area-preserving chaotic maps. First, the proposed scheme can be applied to images of arbitrary size, whereas the area-preserving chaotic maps can be only applied to square images. Secondly, though the aperiodicity nature of a chaotic system will be deteriorated in computer realization with finite computation precision, the period length of pseudorandom keystream sequence generated by a chaotic system is by far longer than that of its discretized version. A keystream sequence with a very long period can be considered practically aperiodic when applied to images of reasonable size. That is, image scrambled by the proposed method will not return to its original state even after a huge number of iterations.

3. Color Image Substitution Using a Fast Plaintext-Dependent Keystream Generation Strategy

In the substitution stage, the logistic map [28], which is the most studied example of discrete nonlinear dynamical systems that exhibit chaotic behavior, is employed to generate the keystream sequence for mixing the subpixel values. Mathematically, the map is described by

$$
x_{n+1} = r x_n \left(1 - x_n\right), \quad x_n \in [0, 1], \; r \in [0, 4], \tag{5}
$$

where x is the state variable and r is the parameter. The logistic map behaves chaotically, interrupted by small periodic windows for values of r between about 3.57 and 4. In our scheme, r is set to 4 to avoid those nonchaotic regions. Similarly, the initial condition x_0 of the map is used as the secret key for substitution.

The detailed substitution process is described as follows.

Step 1. Preiterate the logistic map for N_0 times for the same purpose mentioned above. It should be noticed that the values of 0.5 and 0.75 are two "bad" points, trapping the iterations to the fixed points 0 and 0.75, respectively. If this case is encountered, a slight perturbation should apply.

Step 2. The logistic map is iterated continuously. For each iteration, a 24-bit (3 byte) integer can be obtained from the current state of the map according to

$$\text{pseRandInt} = \text{mod}\left[\text{sig}\left(x_{n+1}, \alpha\right), (1 \ll 24)\right], \tag{6}$$

where $n = 0, 1, \ldots, H \times W - 1$ and "$\ll s$" denotes a left shift by s bit. For instance, $(1 \ll 24)$ returns bitwise representation of 1 shifted to the left by 24 bits, which is equivalent to 2^{24}.

Step 3. Extract three keystream elements from pseRandInt according to

$$\begin{aligned} \text{kstrEle}_{(\text{red})} &= \left(\text{pseRandInt} \gg 16\right) \ \& \ \text{0xFF}, \\ \text{kstrEle}_{(\text{green})} &= \left(\text{pseRandInt} \gg 8\right) \ \& \ \text{0xFF}, \\ \text{kstrEle}_{(\text{blue})} &= \text{pseRandInt} \ \& \ \text{0xFF}, \end{aligned} \tag{7}$$

where "$\gg s$" denotes a right shift by s bit and & denotes a bitwise AND operation. It can be seen from (7) that the upper, middle, and lower 8 bits of pseRandInt are assigned to $\text{kstrEle}_{(\text{red})}$, $\text{kstrEle}_{(\text{green})}$, $\text{kstrEle}_{(\text{blue})}$, respectively.

Step 4. Convert the plain-pixel to its cipher form according to

$$\begin{aligned} c_{3n} &= \text{kstrEle}_{(\text{red})} \oplus \text{mod}\left(\left(p_{3n} + \text{kstrEle}_{(\text{red})}\right), G_L\right) \\ &\quad \oplus c_{3n-1}, \\ c_{3n+1} &= \text{kstrEle}_{(\text{green})} \\ &\quad \oplus \text{mod}\left(\left(p_{3n+1} + \text{kstrEle}_{(\text{green})}\right), G_L\right) \oplus c_{3n}, \\ c_{3n+2} &= \text{kstrEle}_{(\text{blue})} \\ &\quad \oplus \text{mod}\left(\left(p_{3n+2} + \text{kstrEle}_{(\text{blue})}\right), G_L\right) \oplus c_{3n+1}, \end{aligned} \tag{8}$$

where $(p_{3n}, p_{3n+1}, p_{3n+2})$ and $(c_{3n}, c_{3n+1}, c_{3n+2})$ are the three colored subpixels of the currently operated pixel and its output cipher-pixel, respectively, $\oplus$ performs bitwise exclusive OR operation, and G_L is the number of gray levels in the input image (for a 24-bit RGB color image, $G_L = 256$).

As can be seen from (8), the modification made to a subpixel depends not only on the keystream element but also on its previous ciphered subpixel, and thereby the influence of each subpixel can be spread over all its succeeding ones. For the first subpixel p_0, the initial value c_{-1} can be set as a constant.

Step 5. Make the keystream elements depend on the plain-pixel by perturbing the state variable of logistic map according to

$$\begin{aligned} x_{n+1} &= x_{n+1} + \beta \quad \text{for } 0 < x_n < 0.5, \\ x_{n+1} &= x_{n+1} - \beta \quad \text{for } 0.5 < x_n < 1, \end{aligned} \tag{9}$$

where

$$\beta = 0.1 \times \sum_{j=0}^{2} \frac{p_{3n+j}}{\left[3 \times \left(G_L - 1\right)\right]} \tag{10}$$

is the disturbance value determined by the original values of the three colored subpixels of the currently operated pixel. It is clear that β falls within the range between 0 and 0.1, keeping the logistic map operating in chaotic region.

Step 6. Return to Step 2 until all the subpixels in imgData are encrypted.

Step 7. Perform several rounds of the overall permutation-substitution operations so as to spread the influence of each individual subpixel over the entire cipher-image.

Step 8. Produce the final output by adding a file header identical to that of the input image to imgData.

The decryption procedure is similar to that of the encryption process except that some steps are followed in a reversed order. Particularly, the inverse of (8) is given by

$$\begin{aligned} p_{3n} &= \text{mod}\left[\left(k\text{strEle}_{(\text{red})} \oplus c_{3n} \oplus c_{3n-1} + G_L \right.\right. \\ &\quad \left.\left. - \text{kstrEle}_{(\text{red})}\right), G_L\right], \\ p_{3n+1} &= \text{mod}\left[\left(\text{kstrEle}_{(\text{green})} \oplus c_{3n+1} \oplus c_{3n} + G_L \right.\right. \\ &\quad \left.\left. - \text{kstrEle}_{(\text{green})}\right), G_L\right], \\ p_{3n+2} &= \text{mod}\left[\left(\text{kstrEle}_{(\text{blue})} \oplus c_{3n+2} \oplus c_{3n+1} + G_L \right.\right. \\ &\quad \left.\left. - \text{kstrEle}_{(\text{blue})}\right), G_L\right]. \end{aligned} \tag{11}$$

In addition, as can be seen from the above description of the substitution algorithm, the perturbing of the state of the logistic map is performed after a pixel is enciphered and the disturbance value is calculated from the original values of the ciphered pixel. Accordingly, in the decryption procedure, the same disturbance value can be calculated out after a cipher-pixel is decrypted.

4. Analysis of the Randomness of the Substitution Keystream

The randomness of the keystream sequence is crucial to the security of a chaotic cipher. A cryptographically secure keystream generator should generate the keystream sequence without repetition or predictability, thus preventing different parts of a messages encrypted with the repeated parts of the keystream sequence from being intercepted or generated by an attacker. The degree of randomness of a keystream

sequence may be determined by statistical tests, and the most authoritative one is the test suite designed by the National Institute of Standards & Technology (NIST). The test suite is a statistical package consisting totally of 16 tests, which are carried out as follows: For each statistical test, a set of P values (corresponding to the set of sequences) is produced. A sequence passes a statistical test whenever the P value $\geq \alpha$ and fails otherwise, where $\alpha \in (0.001, 0.01]$ is the significance level. For each statistical test, compute the proportion of sequences that pass. The range of acceptable proportions is determined using the confidence interval defined as

$$(1-\alpha) - \sigma\sqrt{\frac{\alpha(1-\alpha)}{m}} \leq p_\alpha \leq 1.0, \tag{12}$$

where σ is the number of standard deviations and m is the sample size. In our experiments, 200 keystream sequences ($m = 200$), each with a length, in bits, as long as that of a 24-bit RGB color image of size 512×512, are generated with randomly selected substitution keys. Together with the chosen standard parameters, $\alpha = 0.01$ and $\sigma = 3$, we have $0.968893 \leq P_\alpha \leq 1.0$. If the proportion falls outside of this interval, then there is evidence that the data is nonrandom. Table 1 lists the test results for the keystream sequences generated using the proposed and conventional methods. As can be seen from this table, the proposed method has a higher pass rate in 13 of the 16 test items, and hence it generates keystream sequences with a higher degree of randomness over the conventional method.

5. Analysis of the Confusion and Diffusion Performance of the Proposed Scheme

5.1. Analysis of Confusion Performance. In order to evaluate the confusion performance of the proposed permutation method, we apply it to the standard "peppers" test image (512×512 pixels, 24-bit RGB color) and the result is compared with that of three most widely used methods based on area-preserving invertible chaotic maps, as demonstrated in Figure 2. Figure 2(a) shows the test image and Figure 2(b) is the resulting image after applying the proposed permutation method once. The permutation key, that is, the initial conditions of the hyperchaotic Lü system, is randomly chosen to be $\{x_0 = 8.14723686393179,\ y_0 = -3.13375856139019,\ z_0 = 15.0794726517402,\ u_0 = -47.8753417717149\}$. Figures 2(c)–2(e), 2(f)–2(h), and 2(i)–2(k) show the resulting images after applying the cat map, the baker map, and the standard map once, twice, and three times, respectively. The permutation keys, that is, the control parameters of the three maps, are chosen to be $\{p = 40, q = 8\}$, $n_i = \{32, 16, 32, 64, 16, 32, 64, 16, 32, 32, 64, 16, 64, 32\}$, and $K = 1024$, respectively. As can be seen from Figure 2, the proposed method takes only one round to achieve a satisfactory scrambling effect, whereas more (≥2) is needed by the three conventional ones. Moreover, the colors in the scrambled image produced by the proposed permutation method are much more uniformly distributed than that produced by the three conventional methods. This is because the proposed scheme scrambles the subpixels across all the three color channels, whereas the schemes using area-preserving chaotic maps have to scramble each color channel separately, making the dominant colors of the resulting scrambled image similar to those of its original version.

5.2. Analysis of Diffusion Performance. As is known, the diffusion property is essential to ensure the security of a cryptographic algorithm against chosen-plaintext attack. The differential analysis is the most common way to implement the chosen-plaintext attack. To do this, an opponent may firstly create two plain-images with only one-bit difference and then encrypt the two images using the same secret key. By observing the differences between the two resulting cipher-images, some meaningful relationship between plain-image and cipher-image could be found out, and it further facilitates determining the keystream. Obviously, this kind of cryptanalysis may become impractical if a cryptosystem is highly sensitive to plaintext; that is, changing one bit of the plaintext affects every bit in the ciphertext.

To measure the diffusion property of an image cryptosystem, two criteria, that is, NPCR (the number of pixel change rate) and UACI (the unified average changing intensity), are commonly used. The NPCR is used to measure the percentage of different pixel numbers between two images. Let $I_1(i, j, k)$ and $I_2(i, j, k)$ be the (i, j)th pixel in kth color channel ($k = 1, 2, 3$ denotes the red, green, and blue color channels, resp.) of two images I_1 and I_2, where $0 \leq i \leq H$, $0 \leq j \leq W$; the NPCR can be defined as

$$\text{NPCR} = \frac{\sum_{k=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W} D(i, j, k)}{3 \times H \times W} \times 100\%, \tag{13}$$

where $D(i, j, k)$ is defined as

$$D(i, j, k) = \begin{cases} 0 & \text{if } I_1(i, j, k) = I_2(i, j, k), \\ 1 & \text{if } I_1(i, j, k) \neq I_2(i, j, k). \end{cases} \tag{14}$$

The second criterion, UACI, is used to measure the average intensity of differences between the two images. It is defined as

$$\text{UACI} = \frac{1}{3 \times H \times W}\left[\sum_{k=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{|I_1(i, j, k) - I_2(i, j, k)|}{G_L - 1}\right] \times 100\%. \tag{15}$$

Clearly, no matter how similar the two input images are, a good image cryptosystem should produce outputs with NPCR and UACI values ideally being equal to those of two random images, which are given by

$$\begin{aligned} \text{NPCR}_{\text{random}} &= \left(1 - \frac{1}{2^{\log_2 G_L}}\right) \times 100\%, \\ \text{UACI}_{\text{expected}} &= \frac{1}{{G_L}^2}\left(\frac{\sum_{i=1}^{G_L - 1} i(i+1)}{G_L - 1}\right) \times 100\%. \end{aligned} \tag{16}$$

For instance, the NPCR and UACI values for two random color images in 24-bit RGB format ($G_L = 256$) are 99.609% and 33.464%, respectively.

Table 1: Results of NIST statistical test.

Test items	Pass rate	
	Proposed method	Conventional method
Frequency	100.0%	97.50%
Block frequency	99.00%	99.00%
Cusum-forward	100.0%	97.00%
Cusum-reverse	100.0%	97.50%
Runs	99.50%	98.50%
Long runs of ones	99.50%	97.50%
Rank	99.00%	98.50%
Spectral DFT	99.00%	99.00%
Nonoverlapping templates	99.50%	98.50%
Overlapping templates	99.00%	97.00%
Universal	98.50%	98.00%
Approximate entropy	99.00%	97.50%
Random excursions	98.21%	99.43%
Random excursions variant	99.40%	100.0%
Linear complexity	99.50%	99.50%
Serial	99.50%	99.00%

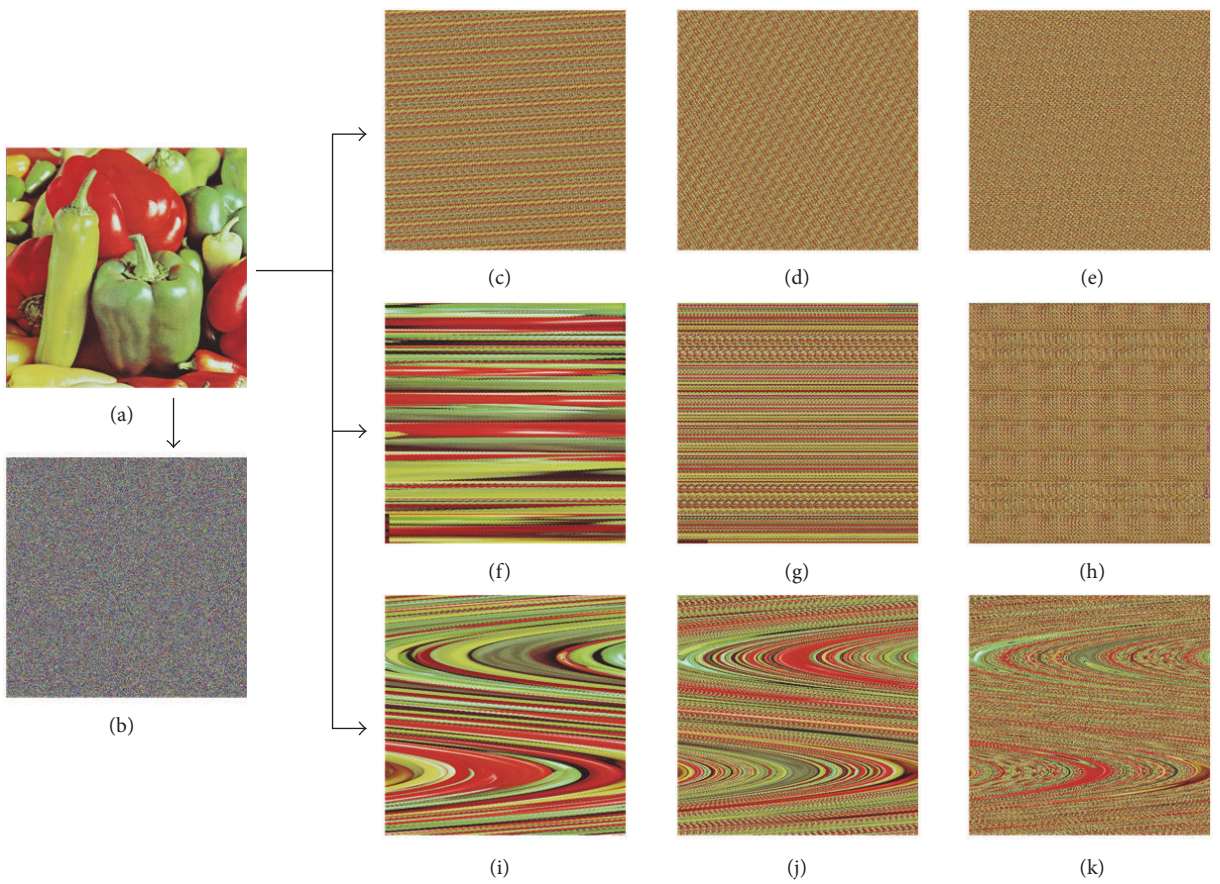

Figure 2: The applications of the proposed and three conventional chaos-based permutation methods. (a) The test image; (b) the test image after applying the proposed permutation method once; (c)–(e), (f)–(h), and (i)–(k) are the test images after applying the cat map, the baker map, and the standard map once, twice, and three times, respectively.

Table 2: Differential images used in NPCR and UACI tests.

Test image name	Color channel	Pixel position (x, y)	Pixel value	
			Original	Modified
Baboon	G	(29, 130)	66	65
House	G	(182, 179)	35	36
Lena	B	(425, 39)	121	122
Peppers	R	(428, 144)	123	122
Portofino	R	(306, 294)	64	65

Table 3: Results of NPCR and UACI tests.

Test image name	Number of encryption rounds							
	1		2		3		4	
	NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
Baboon	0.90905	0.30349	0.99614	0.33460	0.996095	0.33445	0.99609	0.33457
House	0.83127	0.27808	0.99611	0.33483	0.9961407	0.33466	0.99608	0.33446
Lena	0.06218	0.02089	0.99620	0.33469	0.996074	0.33487	0.99619	0.33447
Peppers	0.81662	0.27345	0.99607	0.33448	0.996147	0.33457	0.99608	0.33466
Portofino	0.44400	0.14861	0.99600	0.33443	0.995989	0.33425	0.99603	0.33487

The NPCR and UACI of the proposed cryptosystem are evaluated using five standard 24-bit color test images of size 512×512 taken from the USC-SIPI image database. The differential images are created by randomly changing 1 bit in the original ones, as listed in Table 2. The two images in each test pair are encrypted using the same secret key, and their NPCR and UACI values under different number of cipher rounds are given in Table 3. It can be concluded from Table 3 that the diffusion efficiency of the proposed scheme is competitive with that of existing optimal substitution strategies, which take a minimum of two encryption rounds to achieve desired NPCR and UACI values.

6. Security Analysis

In this section, thorough security analysis has been carried out, including the most important ones like brute-force analysis, statistical analysis, and key sensitivity analysis, to demonstrate the high security of the proposed scheme.

6.1. Brute-Force Analysis. In cryptography, a brute-force attack is a cryptanalytic attack that attempts to break a cipher by systematically checking all possible keys until the correct one is found. Obviously, a cipher with a key length of n bits can be broken in a worst-case time proportional to 2^n and an average time of half that. A key should therefore be long enough that this line of attack is impractical, that is, would take too long to execute. As mentioned above, the initial conditions of the hyperchaotic Lü system and the logistic map, which consist of four and one state variables, are used as the permutation key and substitution keys, respectively. The two keys are independent of each other and a 64-bit double-precision type gives 53 bits of precision, and therefore the key length of the proposed scheme is $5 \times 53 = 265$ bits. Generally, cryptographic algorithms using keys with a length greater than 100 bits are considered to be "computational security" as the number of operations required to try all possible 2^{100} keys is widely considered out of reach for conventional digital computing techniques for the foreseeable future. Therefore, the proposed scheme is secure against brute-force attack.

6.2. Statistical Analysis

6.2.1. Frequency Distribution of Pixel Values. A good image cryptosystem should sufficiently mask the distribution of pixel values in the plain-images so as to make frequency analysis infeasible. That is, the redundancy of plain-image or the relationship between plain-image and cipher-image should not be observed from the cipher-image as such information has the potential to be exploited in a statistical attack. The frequency distribution of pixel values in an image can be easily determined by using histogram analysis. An image histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. The histograms of the RGB color channels of the "peppers" test image and its output cipher-image produced by the proposed scheme are shown in Figure 3. It is clear from Figures 3(l)–3(n) that the pixel values in all the three color channels of the output cipher-image are fairly evenly distributed over the whole intensity range, and therefore no information about the plain-image can be gathered through histogram analysis.

The distribution of pixel values can be further quantitatively determined by calculating the information entropy of the image. Information entropy, introduced by Shannon in his classic paper "A Mathematical Theory of Communication" [29], is a key measure of the randomness or unpredictability of information content. The information entropy is usually expressed by the average number of bits needed to store or communicate one symbol in a message, as described by

$$H(S) = -\sum_{i=1}^{N} P(s_i) \log_2 P(s_i), \quad (17)$$

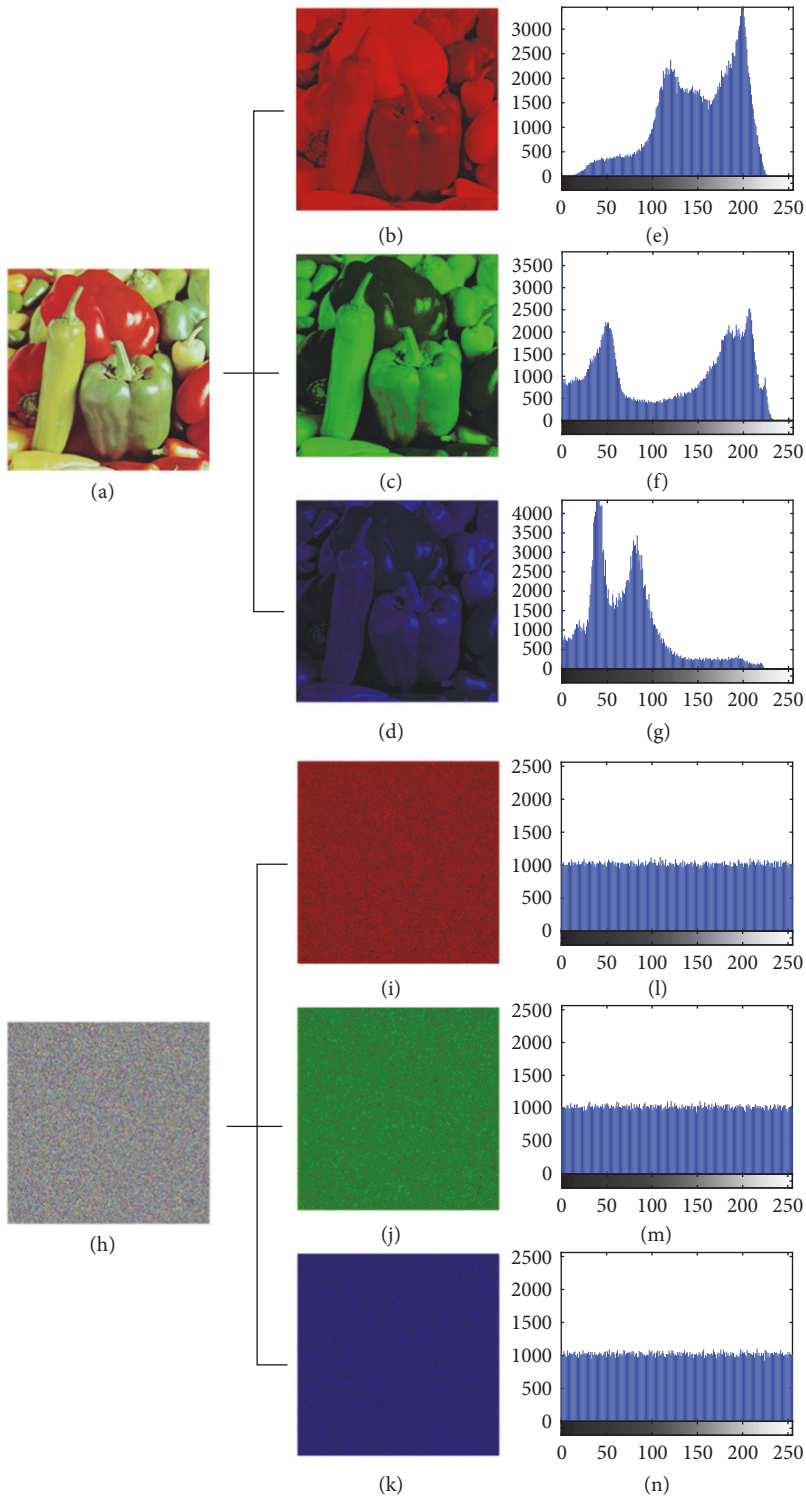

FIGURE 3: Histogram analysis. (a) and (h) are the test image and its output cipher-image, respectively. (b)–(d) and (i)–(k) are the three color channels of (a) and (h), respectively. (e)–(g) and (l)–(n) are the histograms of (b)–(d) and (i)–(k), respectively.

Table 4: Information entropies of the test images and their output cipher-images.

Test image name	Information entropy	
	Plain-image	Cipher-image
Baboon	7.762436	7.999778
House	7.485787	7.999747
Lena	7.750197	7.999772
Peppers	7.669826	7.999788
Portofino	7.306934	7.999766

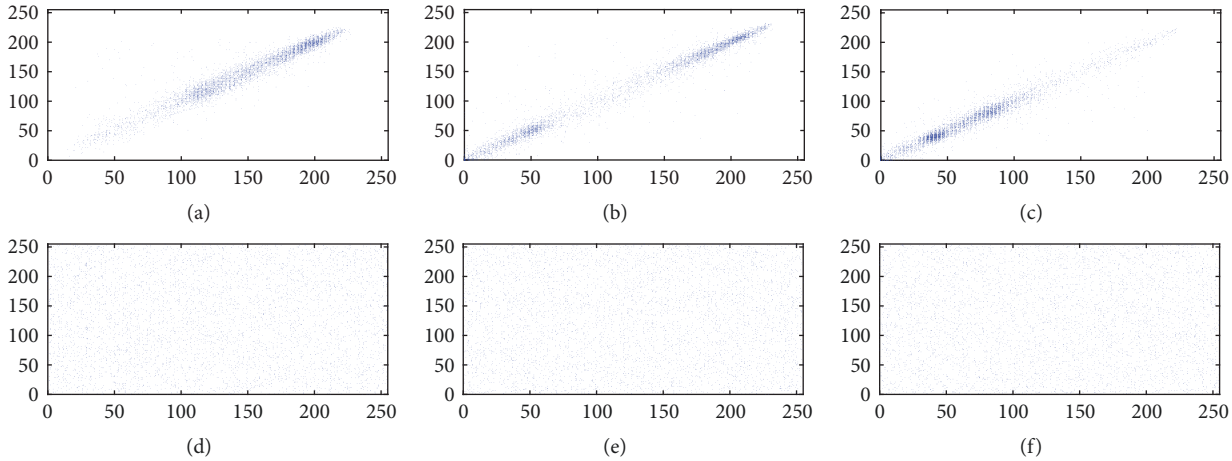

Figure 4: Graphical analysis for correlation of neighboring pixels. (a)–(c) and (d)–(f) are scatter diagrams for horizontally neighboring pixels in the three color channels of the "peppers" test image and its output cipher-image, respectively.

where S is a random variable with N outcomes $\{s1, \ldots, s_N\}$ and $P(s_i)$ is the probability mass function of outcome s_i. It is obvious from (17) that the entropy for a random source emitting N symbols is $\log_2 N$. For instance, for a ciphered image with 256 color levels per channel, the entropy should ideally be 8; otherwise, there exists certain degree of predictability which threatens its security.

The information entropies of the five test images and their output cipher-images are calculated, and the results are listed in Table 4. As can be seen from Table 4, the entropy of all the output cipher-images are very close to the theoretical value of 8. This means the proposed scheme produces outputs with perfect randomness and hence is robust against frequency analysis.

6.2.2. Correlation between Neighboring Pixels. Pixels in an ordinary image are usually highly correlated with their neighbors either in horizontal, vertical, or diagonal direction. However, an effective image cryptosystem should procedure cipher-images with sufficiently low correlation between neighboring pixels. Scatter diagram is commonly used to qualitatively explore the possible relationship between two data sets. To plot a scatter diagram for image data, the following procedures are carried out. First, randomly select S_n pairs of neighboring pixels in each direction from a color channel of the image. Then, the selected pairs is displayed as a collection of points, each having the value of one pixel determining the position on the horizontal axis and the value of the other pixel determining the position on the vertical axis.

Figures 4(a)–4(c) and 4(d)–4(f) show the scatter diagrams for horizontally neighboring pixels in the three color channels of the "peppers" test image and its output cipher-image with $S_n = 5000$, respectively. Similar results can be obtained for the other two directions. As can be seen from this figure, most points in (a)–(c) are clustered around the main diagonal, whereas those in (d)–(f) are fairly evenly distributed. The results indicate that the proposed scheme can effectively eliminate the correlation between neighboring pixels in an original image.

To further quantitatively measure the correlation between neighboring pixels in an image, the correlation coefficients r_{xy} for the sampled pairs are calculated according to the following three formulas:

$$r_{xy} = \frac{(1/S_n)\sum_{i=1}^{S_n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\left((1/S_n)\sum_{i=1}^{S_n}(x_i - \overline{x})^2\right)\left((1/S_n)\sum_{i=1}^{S_n}(y_i - \overline{y})^2\right)}},$$
$$\overline{x} = \frac{1}{S_n}\sum_{i=1}^{S_n} x_i, \quad (18)$$
$$\overline{y} = \frac{1}{S_n}\sum_{i=1}^{S_n} y_i,$$

where x_i and y_i form the ith pair of neighboring pixels.

Table 5 lists the calculated correlation coefficients for neighboring pixels in the three color channels of the five test

Table 5: Correlation coefficients for neighboring pixels in the test images and their output cipher-images.

Test image name	Direction	Plain-image			Cipher-image		
		R	G	B	R	G	B
Baboon	horizontal	0.8719	0.7544	0.8853	−0.0324	−0.0125	−0.0129
	vertical	0.9245	0.8642	0.9140	−0.0142	0.0027	−0.0036
	diagonal	0.8607	0.7324	0.8481	0.0173	0.0207	0.0034
House	horizontal	0.9601	0.9442	0.9619	−0.0055	0.0363	0.0140
	vertical	0.9570	0.9414	0.9688	0.0132	−0.0080	−0.0001
	diagonal	0.9260	0.8959	0.9340	0.0003	−0.0107	−0.0375
Lena	horizontal	0.9892	0.9833	0.9586	0.0033	0.0294	0.0086
	vertical	0.9796	0.9700	0.9357	0.0155	0.0146	−0.0229
	diagonal	0.9690	0.9571	0.9142	0.0158	0.0102	−0.0366
Peppers	horizontal	0.9640	0.9853	0.9698	0.0133	0.0016	−0.0112
	vertical	0.9622	0.9835	0.9691	0.0146	−0.0082	0.0115
	diagonal	0.9545	0.9737	0.9525	0.0008	−0.0255	0.0109
Portofino	horizontal	0.9530	0.9527	0.9209	0.0141	0.0011	−0.0109
	vertical	0.9423	0.8918	0.9119	0.0023	0.0332	0.0120
	diagonal	0.9346	0.8639	0.8954	−0.0020	−0.0048	0.0106

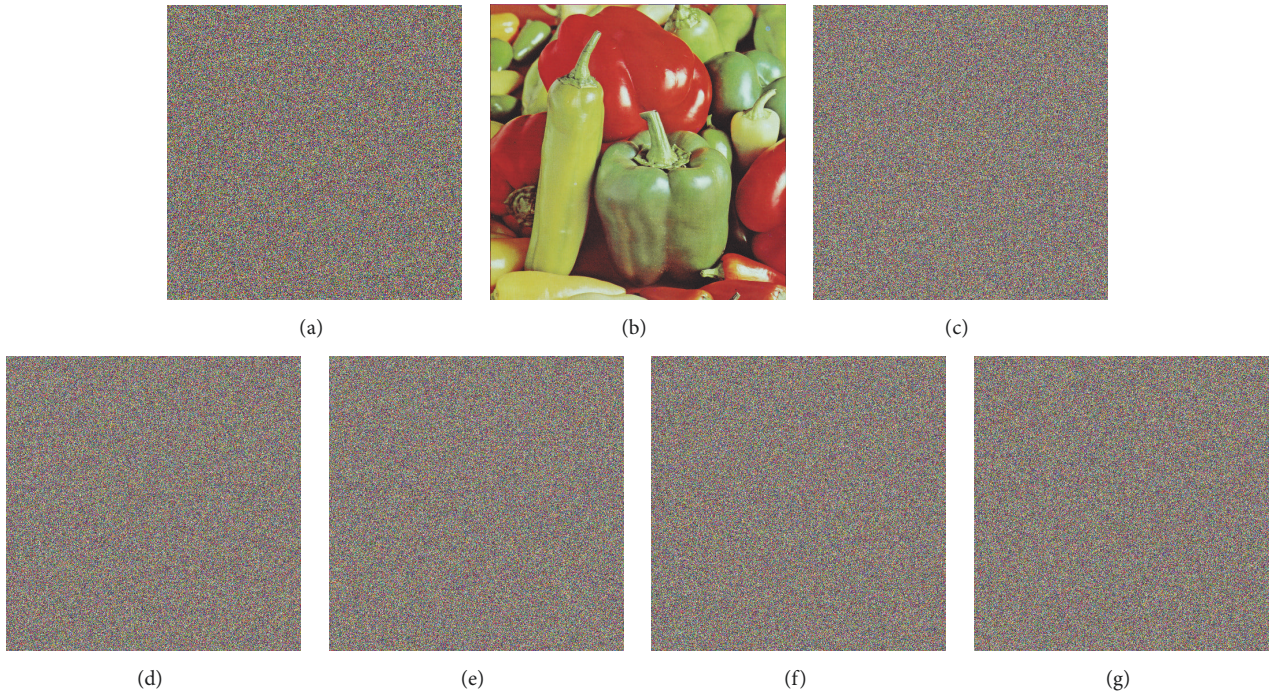

Figure 5: Results of key sensitivity test.

images and their output cipher-images. As can be seen from this table, the correlation coefficients for neighboring pixels in all the three color channels of the output cipher-images are practically zero. The results further support the conclusion drawn from Figure 4.

6.3. Key Sensitivity Analysis. Key sensitivity, another basic design principle of cryptographic algorithms, ensures that no information about the plaintext can be revealed even if there is only a slight difference between the decryption and encryption keys. To evaluate the key sensitivity property of the proposed scheme, the "peppers" test image is firstly encrypted with a randomly generated secret key: hyperchaotic Lü system with initial conditions (x_0 = 9.05791937075619, y_0 = 2.53973632587012, z_0 = 25.2943698490164, u_0 = −28.5802537020945) and logistic map with initial condition x_0 = 0.278498218867048, and the resulting cipher-image is shown in Figure 5(a). Then the ciphered image is tried to be decrypted using six decryption keys, one of which is exactly the same as the encryption key and the other five have only one-bit difference from it, as listed in Table 6. The resulting deciphered images are shown in Figures 5(b)–5(g), respectively, from which we can see that even an almost perfect guess of the key does not reveal any

Table 6: Decryption keys used for key sensitivity test.

Figure	Decryption key		
	Permutation part		Substitution part
5(b)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(c)	$\mathbf{x_0 = 9.05791937075618}$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(d)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$\mathbf{y_0 = 2.53973632587011}$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(e)	$x_0 = 9.05791937075619$ $\mathbf{z_0 = 25.2943698490163}$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(f)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $\mathbf{u_0 = -28.5802537020944}$	$x_0 = 0.278498218867048$
5(g)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$\mathbf{x_0 = 0.278498218867047}$

Table 7: Performance comparison of two schemes using different keystream generation methods.

Image size	File size (KB)	Running speed (ms)	
		Proposed method	Conventional method
256×256	192	27.7	37.0
512×512	768	62.0	77.9
1024×1024	3072	200.9	260.1

information about the original image. It can, therefore, be concluded that the proposed scheme fully satisfies the key sensitivity requirement.

7. Speed Performance

As can be seen from the above description of the substitution keystream generation method, three keystream elements can be simultaneously extracted from the current state of the logistic map, whereas only one can be obtained using the conventional method. As a result, the total number of iterations is reduced by 3 times and the encryption time is shortened. We use three 24-bit RGB test images of different sizes to evaluate the computational efficiency of the proposed scheme and compare it with that of an identical copy of the proposed scheme except using a conventional keystream generation method. Each test image is ciphered 10 times with two rounds of permutation-substitution operations, and the average execution times are listed in Table 7. Both schemes have been implemented using C programming language on Windows 7 64-bit platform, and the tests have been done on a personal computer with an Intel Xeon E3-1230 v3 3.3 GHz processor and 8 GB RAM. As can be seen from Table 7, the proposed scheme significantly outperforms the one using a conventional keystream generation method in terms of computational efficiency, and therefore it provides a good candidate for online secure image transmission over public networks.

8. Conclusions

This paper has proposed a new permutation-substitution type color image cipher to better meet the increasing demand for real-time secure image communications. To confuse the relationship between the ciphertext and the secret key, the positions of colored subpixels in the input image are scrambled using a pixel-swapping mechanism, which avoids two main problems encountered when using the discretized version of area-preserving chaotic maps. To improve the computational efficiency of the substitution process, we introduced an efficient keystream generation method that can simultaneously extract three keystream elements from the current state of the iterative logistic map. Compared with the conventional method, the total number of iterations is reduced by 3 times. The computational efficiency comparison results have shown the superior performance of the proposed encryption scheme. To ensure the robustness of the proposed scheme against chosen-plaintext attack, the current state of the logistic map is perturbed during each iteration and the disturbance value is determined by plain-pixel values. The mechanism of associating the keystream sequence with plain-image also helps accelerate the diffusion process and increase the degree of randomness of the keystream sequence. The results of NPCR and UACI tests indicate that the proposed scheme takes only two encryption rounds to achieve a satisfactory diffusion effect. The results of NIST statistical test indicated that the substitution keystream sequences generated using the proposed method have a higher degree of randomness than that generated by conventional method. We have carried out an extensive security analysis, which demonstrates the satisfactory security level of the new scheme. It can therefore be concluded that the proposed scheme provides a good candidate for online secure image communication applications.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (no. N150402004) and the Online Education Research Fund of MOE Research Center for Online Education (Qtone Education) (no. 2016YB123).

References

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

[2] C. Fu, W.-H. Meng, Y.-F. Zhan et al., "An efficient and secure medical image protection scheme based on chaotic maps," *Computers in Biology and Medicine*, vol. 43, no. 8, pp. 1000–1010, 2013.

[3] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and Y. Zhang, "Reusing the permutation matrix dynamically for efficient image cryptographic algorithm," *Signal Processing*, vol. 111, pp. 294–307, 2015.

[4] K. Wong, B. S. Kwok, and W. Law, "A fast image encryption scheme based on chaotic standard map," *Physics Letters A*, vol. 372, no. 15, pp. 2645–2652, 2008.

[5] C. Fu, B. Lin, Y. Miao, X. Liu, and J. Chen, "A novel chaos-based bit-level permutation scheme for digital image encryption," *Optics Communications*, vol. 284, no. 23, pp. 5415–5423, 2011.

[6] Y. Wu, Y. Zhou, S. Agaian, and J. P. Noonan, "A symmetric image cipher using wave perturbations," *Signal Processing*, vol. 102, pp. 122–131, 2014.

[7] W. Zhang, H. Yu, Y.-L. Zhao, and Z.-L. Zhu, "Image encryption based on three-dimensional bit matrix permutation," *Signal Processing*, vol. 118, pp. 36–50, 2016.

[8] C. Fu, O. Bian, H.-Y. Jiang, L.-H. Ge, and H.-F. Ma, "A new chaos-based image cipher using a hash function," in *Proceedings of the 15th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2016*, Japan, June 2016.

[9] J. Chen, Z. Zhu, C. Fu, H. Yu, and L. Zhang, "A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism," *Communications in Nonlinear Science and Numerical Simulation*, 2014.

[10] C. Fu, Z.-K. Wen, Z.-L. Zhu, and H. Yu, "A security improved image encryption scheme based on chaotic Baker map and hyperchaotic Lorenz system," *International Journal of Computational Sciences and Engineering*, vol. 12, no. 2-3, pp. 113–123, 2016.

[11] C. Li, S. Li, and K.-T. Lo, "Breaking a modified substitution-diffusion image cipher based on chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 2, pp. 837–843, 2011.

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

[13] C. Li, T. Xie, Q. Liu, and G. Cheng, "Cryptanalyzing image encryption using chaotic logistic map," *Nonlinear Dynamics*, vol. 78, no. 2, pp. 1545–1551, 2014.

[14] C. Li, Y. Liu, L. Y. Zhang, and M. Z. Q. Chen, "Breaking a chaotic image encryption algorithm based on modulo addition and XOR operation," *International Journal of Bifurcation and Chaos*, vol. 23, no. 4, Article ID 1350075, 2013.

[15] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, and G. Chen, "A chaos-based image encryption algorithm with variable control parameters," *Chaos, Solitons & Fractals*, vol. 41, no. 4, pp. 1773–1783, 2009.

[16] C. Fu, J.-B. Huang, N.-N. Wang, Q.-B. Hou, and W.-M. Lei, "A symmetric chaos-based image cipher with an improved bit-level permutation strategy," *Entropy*, vol. 16, no. 2, pp. 770–788, 2014.

[17] T. Xiang, K.-W. Wong, and X. Liao, "Selective image encryption using a spatiotemporal chaotic system," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, Article ID 023115, 2007.

[18] K.-W. Wong, B. S.-H. Kwok, and C.-H. Yuen, "An efficient diffusion approach for chaos-based image encryption," *Chaos, Solitons & Fractals*, vol. 41, no. 5, pp. 2652–2663, 2009.

[19] J.-X. Chen, Z.-L. Zhu, C. Fu, L.-B. Zhang, and Y. Zhang, "An efficient image encryption scheme using lookup table-based confusion and diffusion," *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1151–1166, 2015.

[20] C. Fu, J. Chen, H. Zou, W. Meng, Y. Zhan, and Y. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Optics Express*, vol. 20, no. 3, pp. 2363–2378, 2012.

[21] Z. Zhu, W. Zhang, K. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, 2011.

[22] J.-x. Chen, Z.-l. Zhu, C. Fu, L.-b. Zhang, and Y. Zhang, "An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 23, no. 1-3, pp. 294–310, 2015.

[23] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, and B.-Q. Yang, "Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption," *Signal Processing*, vol. 142, pp. 340–353, 2018.

[24] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and L.-B. Zhang, "An efficient image encryption scheme using gray code based permutation approach," *Optics and Lasers in Engineering*, vol. 67, pp. 191–204, 2015.

[25] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Information Sciences*, vol. 297, pp. 80–94, 2015.

[26] A. Chen, J. Lu, J. Lü, and S. Yu, "Generating hyperchaotic Lü attractor via state feedback control," *Physica A: Statistical Mechanics and its Applications*, vol. 364, pp. 103–110, 2006.

[27] J. Lü, G. Chen, and S. Zhang, "Dynamical analysis of a new chaotic attractor," *International Journal of Bifurcation and Chaos*, vol. 12, no. 5, pp. 1001–1015, 2002.

[28] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[29] C. E. Shannon, "A mathematical theory of communication," *Bibliometrics*, vol. 5, no. 1, pp. 3–55, 2001.