IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 7, 2020, accepted January 28, 2020, date of publication January 31, 2020, date of current version February 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970806

An Efficient Image Encryption Scheme Based on the LSS Chaotic Map and Single S-Box

QING LU[1], CONGXU ZHU[2], AND XIAOHENG DENG[2], (Member, IEEE)
[1]Hunan Police Academy, Changsha 410138, China
[2]School of Computer Science and Engineering, Central South University, Changsha 410083, China

Corresponding authors: Congxu Zhu (zhucx@csu.edu.cn) and Xiaoheng Deng (dxh@csu.edu.cn)

This work was supported in part by the Open Research Fund of Key Laboratory of Network Crime Investigation of Hunan Provincial Colleges under Grant 2018WLFZZC002, in part by the Science and Technology Project of Hunan Province of China under Grant 2017SK1040, in part by the National Natural Science Foundation of China under Grant 61772553 and Grant 61379058, and in part by the Scientific Research Excellent Youth Project of Hunan Education Department under Grant 16B048 and Grant 18B549.

ABSTRACT This paper presents an efficient and secure chaotic S-Box based image encryption algorithm. Firstly, by cryptanalyzing a multiple chaotic S-Boxes based image encryption algorithm, we successfully cracked the cryptosystem by using chosen-plaintext attack (CPA). Secondly, we put forward a new image encryption scheme based on a novel compound chaotic map and single S-Box. In the new scheme, a novel discrete compound chaotic system, Logistic-Sine system (LSS), is proposed, which has wider chaotic range and better chaotic properties. And a new S-Box is constructed by using LSS, which has satisfactory cryptographic performance. Based on the new S-Box and the chaotic key stream, the new image encryption algorithm is designed, which consist of a round of permutation and two rounds of substitution process. The permutation and substitution key sequences are related to the plaintext image content, this strategy enables the cryptosystem to resist CPA. The simulation results and security analysis verified the effectiveness of the proposed image encryption scheme. Especially, the new scheme has obvious efficiency advantages, showing that it has better application potential in real-time image encryption.

INDEX TERMS Image encryption, chaos, S-box, Logistic-Sine system, substitution, chosen-plaintext attack.

I. INTRODUCTION

With the rapid development of network communication and big data applications, information security has become a very important hot issue. The most important means to protect information is data encryption. Therefore, cryptography is the core technology in information security. In modern cryptographic systems, block encryption algorithms is widely used, such as in Data Encryption Standard (DES), Advanced Encryption Standard (AES) and other systems. However, the traditional encryption methods encountered difficulties in encrypting images because of the large amounts of data in images. As a random phenomenon in nonlinear systems, chaos has many natural relations with cryptography. Chaos system provides a very convenient source for the generation of plentiful pseudo-random sequences and the construction of nonlinear encryption components. Chaotic systems have the advantages in generating a large number of keys quickly. As a supplementary technology of traditional encryption, chaotic systems are very suitable for real-time encryption of image media. As the only nonlinear component in a block cipher system, S-Boxes (substitution boxes) have an important influence on the security performance of the cipher system. AES is considered to be an effective cryptosystem to a large extent. An important components of AES is its S-Box, which is based on the inversion and affine transformation of GF(2^8) elements. Because of the popularity of AES in communication systems, S-Box has attracted the attention of cryptosystem designers. Employing chaotic system to generate S-boxes and apply them to image encryption is the most promising field of chaos system, and many related works were proposed [1]–[5].

With regard to design S-Box in block cipher system, strong cryptographic performance of an S-Box has always been the goal of cryptosystem designers. Researchers have proposed a variety of S-Box construction methods. D. Lambić [5] proposed a novel method of S-Box design based on discrete chaotic maps. Tanyildizi and Erkan [6] proposed a chaotic S-Box generation method using

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

parameter optimization of one dimensional chaotic maps. Çavuşoğlu *et al.* [4] proposed a strong s-box generation algorithm based on a chaotic scaled zhongtang system. Hussain *et al.* [7] proposed an S-Box construction method based on chaotic maps and algebraic structures. Rafiq and Khan *et al.* [8] proposed a new S-Box construction method based on triangle groups. Zahid and Arshad *et al.* [9] proposed an innovative method to design S-Boxes by using cubic polynomial mapping. Liu *et al.* [10] proposed a novel method for constructing S-Boxes based on spatiotemporal chaotic systems. In recent years, chaotic systems have been widely applied in designing of S-boxes and image encryption scheme because of their good cryptographic characteristics, such as pseudo randomness [11], ergodicity or non-periodicity [12], extreme sensitivity to the initial conditions [13] and easy to implement with software or hardware [14]. Ullah *et al.* [15] proposed a novel method for constructing S-Box by using a combination of chaotic maps with improved chaotic range. Belazi and El-Latif [16] proposed a very simple method for constructing S-Boxes based on chaotic sine map. Al Solami *et al.* [17] proposed a novel method to construct cryptographically strong bijective S-Boxes by using a 5D hyper-chaotic system. Khan *et al.* [18] proposed a chaotic Boolean functions method for constructing S-Boxes. Belazi *et al.* [19] proposed an efficient method for constructing S-Boxes by using chaotic logistic-sine map. Wang *et al.* [20] constructed S-Boxes by using a hyper-chaotic system with infinite equilibria. With regard to design image encryption scheme based on S-Boxes, more schemes use multiple S-Boxes to realize image encryption system. Khan [21] proposed a novel image encryption scheme based on multiple chaotic S-Boxes. Wang and Wang [22] proposed a novel image encryption algorithm based on dynamic S-boxes constructed by chaos. Hussain *et al.* [23] proposed an image encryption algorithm that is also based on multiple chaotic S-Boxes generating by chebyshev chaotic map. However, no matter multiple chaotic S-Boxes method or dynamic S-Boxes method, it takes a lot of time to construct multiple S-Boxes. In order to improve the efficiency of image encryption algorithm, some researchers have proposed image encryption algorithms based on double S-Boxes [24], [25]. Some researchers use high-dimensional continuous time chaotic systems to generate S-Boxes, for example, it can be seen in Ref.s [26], [27] and other image encryption algorithms. However, using high-dimensional continuous time chaotic systems to generate S-Boxes will require more time overhead. Compared with high-dimensional continuous-time chaotic systems, low-dimensional discrete chaotic systems can generate S-Boxes with less time overhead. Moreover, some studies show that the complexity of discrete systems is higher than that of continuous systems [28]–[30]. Considering the anti attack performance, some image encryption algorithms based on S-Boxes exist security defects, such as the inability to resist the chosen-plaintext attack. For examples, Zhang and Xiao [31] proposed a general model for attacking S-Box-only image ciphers. The S-Box based image encryption algorithm in [27] has been cracked by us [32]. Zhang *et al.* [33] cracked an image encryption algorithm based on hyper-chaotic systems and dynamic S-Boxes [34].

In order to contribute the cryptanalysis studies and improvement of future schemes, this paper analyzes a recently proposed image encryption algorithm based on three S-Boxes generated with a 3D continuous time chaotic system without equilibrium [26]. Furthermore, an improved image encryption scheme based on single S-Box is proposed. The innovations of this work are as follows:

(1) Analyzed an image encryption algorithm based on three chaotic S-Boxes, and a chosen-plaintext attack method which can crack the image encryption algorithm is proposed.

(2) In order to improve the efficiency and security of image encryption algorithms based on chaotic systems. a new discrete compound chaotic system LSS is proposed, which has wider chaotic range and better chaotic performance than the original chaotic systems.

(3) By using the new chaotic system, a strong S-Box is constructed, and the cryptographic strength of the proposed S-Box is tested. It is verified that the proposed S-Box has satisfactory cryptographic performance.

(4) A new image encryption algorithm based on chaos and single S-Box is proposed, which can overcome the security defect of the original S-Box based image encryption scheme. In the permutation and substitution process, the strategy of key associated with image content is proposed. This strategy enables the algorithm to resist chosen-plaintext attack (CPA). In addition, the proposed image encryption scheme has other better cryptographic performance specially the faster encryption speed, which means that it has better application potential in real-time image encryption.

The rest of this paper is organized as follows. Section II describes the cryptanalysis of an S-Box based image encryption algorithm and the CPA algorithm. Section III presents the new compound chaotic map and the construction of S-Box. Section IV proposed an improved S-Box based image encryption scheme. Section V shows the experimental results of the proposed image encryption scheme, and makes a comparison with some S-Box based image encryption schemes. Section VI completes the research paper with conclusions.

II. CRYPTANALYSIS OF AN S-BOX BASED IMAGE ENCRYPTION SCHEME

A. DESCRIPTION OF THE ORIGINAL SCHEME

Recently, Wang *et al.* [26] proposed an image encryption scheme based on three S-Boxes. In Wang's scheme, three S-Boxes were generated by using a 3D chaotic system. The mathematical model of the system is described by the following differential equations

$$\begin{cases} \dot{x} = ay, \\ \dot{y} = -x + bz + xz, \\ \dot{z} = x + cy - z - xy + dxz + 1, \end{cases} \tag{1}$$

where, $\{a, b, c, d\}$ are system parameters. When $a = 2$, $b = 2.5$, $c = 0.2$, and $d = 0.3$, system (1) does not have any real equilibrium and is chaotic. By using system (1), three S-Boxes were generated and applied to the substitution encryption of image pixels. The sketch of Wang's encryption algorithm can be simply redescribed as follows.

Step 1: Input the plain image and transform its matrix into a 1D array $\mathbf{P} = [P(1), P(2), \ldots, P(L)]$. Input the parameters and initial state values $\{x(0), y(0), z(0)\}$ of the chaotic system (1), which are used as secret keys.

Step 2: Generate three real value chaotic sequences **x**, **y**, and **z** by solving the chaotic system with the secret parameters of $\{a, b, c, d, x(0), y(0), z(0)\}$ and RK-4 (Runge–Kutta 4) algorithm with the specified sampling step interval 0.001.

Step 3: Transform the three real value sequences **x**, **y** and **z** into three integer sequences **X**, **Y** and **Z** by chaos based pseudo random number generator (PRNG) as

$$\begin{cases} \mathbf{X} = \text{mod}(\text{round}(\mathbf{x} \times 10^3), 256) \\ \mathbf{Y} = \text{mod}(\text{round}(\mathbf{y} \times 10^3), 256) \\ \mathbf{Z} = \text{mod}(\text{round}(\mathbf{z} \times 10^3), 256). \end{cases} \quad (2)$$

Step 4: Then three S-Boxes {**S1, S2, S3** } are generated by the S-Box generation algorithm with sequences **X**, **Y** and **Z**. Each of {**S1, S2, S3** } is a 16 × 16 matrix. Namely, $\mathbf{S1} = [S1(i, j)]$, $\mathbf{S2} = [S2(i, j)]$, and $\mathbf{S3} = [S3(i, j)]$, $i = 1, 2, \ldots, 16$, $j = 1, 2, \ldots, 16$. In fact, three 2D matrices can be expressed by one 3D matrix as $\mathbf{S} = [S(i, j, k)]$, where, $S(i, j, k) \in \{0, 1, \ldots, 255\}$, $i = 1, 2, \ldots, 16$, $j = 1, 2, \ldots, 16$, and $k = 1, 2, 3$.

Step 5: The intermediate cipher image array $\mathbf{P}' = [P'(1), P'(2), \ldots, P'(L)]$ is generated by using one of the integer sequences **X**, **Y** or **Z.** Since the authors do not clearly indicate which sequence is used for encryption, we assume that the sequence $\mathbf{X} = [X(1), X(2), \ldots, X(L)]$ is used for encryption, and this assumption will not affect the analysis results. Therefore, the intermediate cipher image array is obtained as

$$P'(i) = P(i) \oplus X(i), \quad i = 1, 2, \ldots, L \quad (3)$$

where $\oplus$ denotes bitwise XOR. The decryption operation corresponding to Eq. (3) can be expressed as Eq. (4):

$$P(i) = P'(i) \oplus X(i), \quad i = 1, 2, \ldots, L \quad (4)$$

Step 6: Perform sub-byte operation on $\mathbf{P}'$ with the 16 × 16 × 3 sized S-box **S,** and obtain the cipher image array $\mathbf{C} = [C(1), C(2), \ldots, C(L)]$. The sub-byte operation can be expressed as

$$C(i) = \text{sub\_byte}[S(:, :, k), P'(i)], \quad i = 1, 2, \ldots, L \quad (5)$$

where, $S(:, :, k)$ represents the k-th S-Box. sub_byte[$S(:, :, k)$, $P'(i)$] is a function, which return an element value in the position (row, col) of the k-th S-Box. Where, k is determined by the i-th element $X(i)$ in the sequence **X**, that is $k = \text{mod}(X(i), 3) + 1$, namely $k = 1$, 2, or 3. row and col is determined by the value of $P'(i)$. For example, if $X(i) = 33$, then $k = \text{mod}(X(i), 3) + 1 = 1$. And if $P'(i) = 151 = (10010111)_2$, then $row = (1001)_2 + 1 = 10$, $col = (0111)_2 + 1 = 8$. Consequently, $C(i) = \text{sub\_byte}[S(:, :, k), P'(i)] = S(row, col, k) = S(10, 8, 1)$.

We note that in Wang's encryption algorithm, $X(i)$ and $S(:, :, k)$ become the equivalent keys of the cryptosystem, which are independent of the image to be encrypted.

B. ATTACK ON THE ORIGINAL ENCRYPTION ALGORITHM BY USING CPA

Chosen-plaintext attack (CPA) is the strongest one of the four classical cipher attack methods [35]. The CPA attack model considers that attackers have the opportunity to use the encryption machine of the cryptosystem, and they can choose special plaintext to obtain the corresponding ciphertext. Thus, the attackers use these known plaintext-ciphertext pairs to decipher the equivalent keys of the cryptosystem or the target ciphertext. The Wang's S-Box based image encryption algorithm has the following defects:

(1) The chaotic sequences {X, **Y, Z}** and S-Boxes {**S1**, **S2**, **S3**} are not related with the image to be encrypted. Hence, once an attacker decodes the sequences {X, **Y, Z** } and S-Boxes (**S1**, **S2**, **S3**} with CPA method, the attacker can use these sequences and S-Boxes to decrypt the target ciphertext.

(2) The algorithm has no diffusion effect. When one pixel at the position (i, j) of the plain image is changed, there is only one pixel at the position (i, j) of the cipher image will be changed.

(3) The cryptosystem lacks the permuting procedure, which results in the reduction of the anti attack ability of the cryptosystem.

Based on careful investigation, we find a CPA algorithm to crack Wang's encryption algorithm. Suppose the target cipher image pixel array to be decrypted is $\mathrm{C} = [C(1), C(2), \ldots, C(L)]$, we can launch chosen-plaintext attack on Wang's encryption algorithm to decode its corresponding plain image pixel array $\mathrm{P} = [P(1), P(2), \ldots, P(L)]$. The simplest CPA algorithm requires 256 selected plaintext images. Our CPA algorithm can be described as Algorithm 1 in MATLAB pseudo code. The correctness and effectiveness of the attack algorithm has been verified by our experiment.

III. THE NEW COMPOUND CHAOTIC MAP AND THE CONSTRUCTION OF S-BOX

Compared with continuous time chaotic systems, discrete chaotic maps have the following advantages [36]: their mathematical structure is simpler, they can generate chaotic sequence at a faster speed, and they are easier to implement with hardware. In order to improve the time efficiency of image encryption system, our new S-Box based image encryption algorithm employs a discrete compound chaotic map, and only one single S-Box and one chaotic sequence are generated by the new proposed discrete compound chaotic map.

Algorithm 1 The CPA Algorithm Pseudo Code

Input: $\mathbf{C} = [C(1), C(2), \ldots, C(L)]$
Output: $\mathbf{P} = [P(1), P(2), \ldots, P(L)]$
1: Let $n \leftarrow 1$; $\mathbf{F} \leftarrow$ zeros(256, L).
2: While ($n <= 256$) do
3: Choose the n-th plain image $\mathbf{Pn} = [n-1, n-1, \ldots, n-1]$.
4: Get the n-th corresponding cipher image $\mathbf{Cn} = [Cn(1), Cn(2), \ldots, Cn(L)]$ by using Wang's encryption machine. Then we have the following equations: $Cn(i) = \text{sub\_byte}[S(:, :, k), (n-1) \oplus X(i)]$. Where, $k = \text{mod}(X(i), 3) + 1$; $i = 1, 2, \ldots, L$; and $k = 1, 2, 3$.
5: For all $i = 1, 2, \ldots, L$, let $F(n, i) \leftarrow Cn(i)$;
6: Let $n \leftarrow n + 1$;
7: End While
8: For $i \leftarrow 1, 2, \ldots, L$, do
9: For $n \leftarrow 1, 2, \ldots, 256$ do
10: If $C(i) = \text{F}(n, i)$
$P(i) \leftarrow n - 1$;
End If
11: End For n
12: End For i

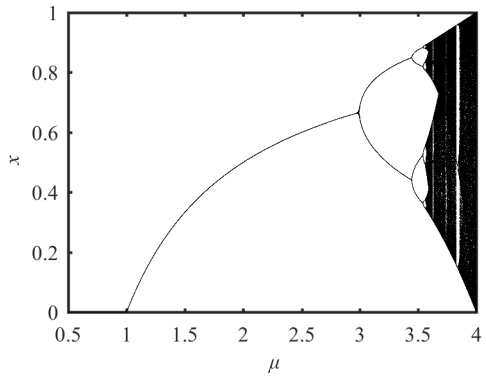


FIGURE 1. Bifurcation diagrams of logistic map.

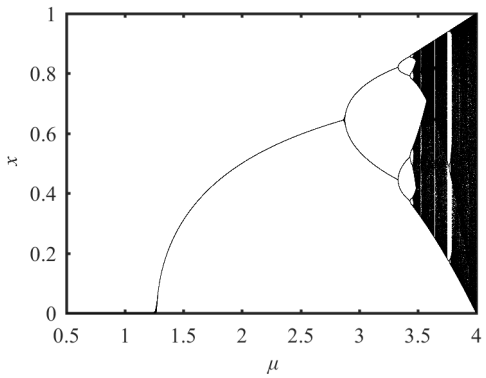


FIGURE 2. Bifurcation diagrams of sine map.

A. CONSTRUCT THE NEW DISCRETE COMPOUND CHAOTIC MAP LSS

Some studies have been devoted to construct discrete compound chaotic systems with more complex properties and better chaotic behavior. The most common method of constructing compound discrete chaotic maps is using multiple discrete seed maps. Our new discrete chaotic map is a compound system composed of Logistic map and Sine map.

1) DISCRETE LOGISTIC MAP

Logistic map is a one-dimensional discrete chaotic map with very simple mathematical structure, which is defined as:

$$x(n+1) = h[x(n)] = \mu \times x(n) \times (1 - x(n)). \tag{6}$$

where, μ is the system parameter, $x(n) \in (0, 1)$ are state values. When $3.57 < \mu \leq 4$, the Logistic map is in chaotic state. h[$x(n)$] is a function map: $x(n) \in (0, 1) \rightarrow x(n+1) \in (0, 1)$. The bifurcation diagrams of Logistic map can be plotted and shown in Figure 1. In Figure 1, the step size chosen for the numerical solution is 0.01. Figure 1 shows that the chaotic range of Logistic map is $3.57 < \mu \leq 4$. So, Logistic map has a narrow chaotic range. Moreover, there are some narrow periodic windows in the chaotic region.

2) DISCRETE SINE MAP

The 1D discrete Sine map is defined as:

$$x(n+1) = g[x(n)] = \frac{\mu}{4} \times \sin[\pi \times x(n)] \tag{7}$$

where, μ is the system parameter, $x(n) \in (0, 1)$ are state values. When $3.48 < \mu \leq 4$, the Sine map is in chaotic state. g[$x(n)$] is a function map: $x(n) \in (0, 1) \rightarrow x(n+1) \in (0, 1)$. The bifurcation diagrams of Sine map can be plotted and shown in Figure 2. In Figure 2, the step size chosen for the numerical solution is 0.01. Figure 2 shows that Sine map has very similar shape of bifurcation diagram to Logistic map. Sine map has a narrow chaotic range, too. Moreover, there are also some narrow periodic windows in the chaotic region.

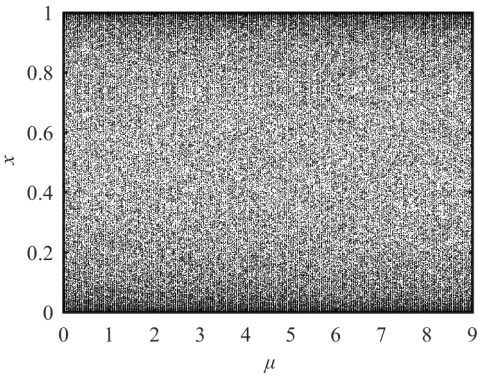


FIGURE 3. Bifurcation diagrams of logistic-sine map.

3) THE NEW DISCRETE COMPOUND CHAOTIC MAP

To expand the chaotic range of discrete chaotic Logistic and Sine map, we propose a discrete compound chaotic map composed of Logistic map and Sine map. The new system has the following mathematical form:

$$\begin{aligned} x(n+1) &= f[x(n)] \\ &= \frac{4\mu}{9} \times x(n) \times (1 - x(n)) \\ &\quad + \frac{9-\mu}{9} \times \sin[\pi \times x(n)]. \end{aligned} \tag{8}$$

where μ is the system parameter and $0 \leq \mu \leq 9$. When $\mu = 0$, the compound system degenerates to the Sine map. While $\mu = 9$, the compound system degenerates to the Logistic map. We call system (8) Logistic-Sine system (LSS).

Proposition: f$[x(n)]$ of system (8) is a function map: $x(n) \in (0, 0.5) \cup (0.5, 1) \rightarrow x(n+1) \in (0, 0.5) \cup (0.5, 1)$.

Proof: The derivative function of the function f$[x(n)]$ is

$$\mathrm{f}'[x(n)] = \frac{4\mu}{9} - \frac{4\mu}{9}x(n) + \frac{(9-\mu)\pi}{9} \times \cos[\pi \times x(n)]$$

When $x(n) \in (0, 0.5)$, f$'[x(n)]$ is a strictly monotonically decreasing function. Therefore, f$'[x(n) < 0.5] >$ f$'[x(n) = 0.5] = \frac{4\mu}{9} - \frac{4\mu}{9} \times 0.5 + \frac{(9-\mu)\pi}{9} \times 0 = 0$. Namely, f$'[x(n) < 0.5] > 0$. Therefore, f$[x(n)]$ is a strictly monotonically increasing function. So, f$[x(n) > 0] >$ f$[x(n) = 0] = 0$ and f$[x(n) < 0.5] <$ f$[x(n) = 0.5] = 1$. Namely, $0 <$ f$[x(n)] < 1$ if $0 < x(n) < 0.5$.

When $x(n) \in (0.5, 1)$, f$'[x(n)]$ is also a strictly monotonically decreasing function. Therefore, f$'[x(n) > 0.5] <$ f$'[x(n) = 0.5] = \frac{4\mu}{9} - \frac{4\mu}{9} \times 0.5 + + \frac{(9-\mu)\pi}{9} \times 0 = 0$. Namely, f$'[x(n) > 0.5] < 0$. Therefore, f$[x(n)]$ is a strictly monotonically decreasing function. So, f$[x(n) > 0.5] <$ f$[x(n) = 0.5] = \frac{4\mu}{9} \times \frac{1}{2} \times \frac{1}{2} + \frac{9-\mu}{9} \times 1 = 1$ and f$[x(n) < 1] >$ f$[x(n) = 1] = 0$. Namely, $0 <$ f$[x(n)] < 1$ if $0.5 < x(n) < 1$. The proof is over.

The bifurcation diagrams of Logistic-Sine system is shown in Figure 3. In Figure 3, the step size chosen for the numerical solution is 0.01. It can be seen that the new compound chaotic system has wider chaotic range in $0 \leq \mu \leq 9$. Moreover, there are no narrow periodic windows in the chaotic region as long as the step size of parameter μ is small enough. Therefore, the new Logistic-Sine system has better chaotic performance than Logistic map and Sine map. In this paper, we employ the chaotic Logistic-Sine system (8) to generate the S-Box and chaotic sequence for image encryption.

The advantage of Logistic-Sine system (8) is that no modular operation required in it, which can speed up the solution speed. As far as we know, modular operations have been used in the compound chaotic maps proposed in some previous literatures, such as in Ref. [19]. As an alternative to our approach, the adaptive chaotic maps obtained by symmetric integration and the maps with disturbed parameters and chains of synchronized maps, are some recent developments.

B. THE NEW S-BOX GENERATED WITH THE LSS

1) THE S-BOX AND CHAOTIC SEQUENCE GENERATION ALGORITHM

Based on the new compound chaotic system LSS, we proposed a new algorithm of generating S-Boxes and chaotic sequences. The basic idea of constructing S-Box includes the following two steps: first, generate an initial S-box based on cube mapping. Secondly, the initial S-box is scrambled based on the position index sequence obtained by chaotic sequence sorting, and the final S-box is obtained. The detailed steps of generating S-boxes and chaotic sequence are given below.

Input : μ; $X(0)$; L; $q \in \mathrm{N}^+$ and $1 \leq q \leq 256$; $d \in \mathrm{N}^+$ and $1 \leq d \leq L - 255$. L is larger than or equal to the number of pixels in the image to be encrypted.

Output: S-Box $\mathbf{S} = [S(i, j)]$, $S(i, j) \in \{0, 1, 2, \ldots, 255\}$, $i = 1, 2, \ldots, 16$; $j = 1, 2, \ldots, 16$. The chaotic sequence $\mathbf{X} = [X(1), X(2), \ldots, X(\mathrm{L})]$.

Step 1: Iterate chaotic map (1) L times to generate a chaotic sequence $\mathbf{X} = [X(1), X(2), \ldots, X(\mathrm{L})]$.

Step 2: Get 256 elements from $X(d)$ to $X(d+255)$ of sequence $\mathbf{X}$ to form a sub-sequence $\mathbf{x} = [x(1), x(2), \ldots, x(256)] = [X(d), X(d+1), \ldots, X(d+255)]$.

Step 3: Sort the chaotic sub-sequence $\mathbf{x}$, then we can get a position index array $\mathbf{I} = \{I(1), I(2), \ldots, I(256)\}$, $I(i) \in \{1, 2, \ldots, 256\}$. Because of the non-periodicity and ergodicity of chaotic sequences, it will inevitably lead to that $I(i) \neq I(j)$ as long as $i \neq j$.

Step 4: Let $\mathbf{U} \leftarrow [U(1), U(2), \ldots, U(256)] = [1, 2, \ldots, 256]$, $U(i) = i, i = 1, 2, \ldots, 256$.

Step 5: Based on $\mathbf{U}$ and the integer q to get a new array $\mathbf{V} = [V(1), V(2), \ldots, V(L)]$ by the following cube mapping

$$V(i) = mod((q \times U(i)^3), 257) - 1, \quad i = 1, 2, \ldots, 256. \tag{9}$$

Because of $1 \leq U(i) \leq 256$ and $1 \leq q \leq 256$, as a result, mod$((q \times U(i)^3), 257) \neq 0$. It can be proved by theory and experiment that Eq. (9) is a one to one map, that is, $V(i) \neq V(j)$ as long as $U(i) \neq U(j)$. Therefore, Eq.(9) is a map: $U(i) \in \{1, 2, \ldots, 256\} \rightarrow V(i) \in \{0, 1, \ldots, 255\}$, $i = 1, 2, \ldots, 256$.

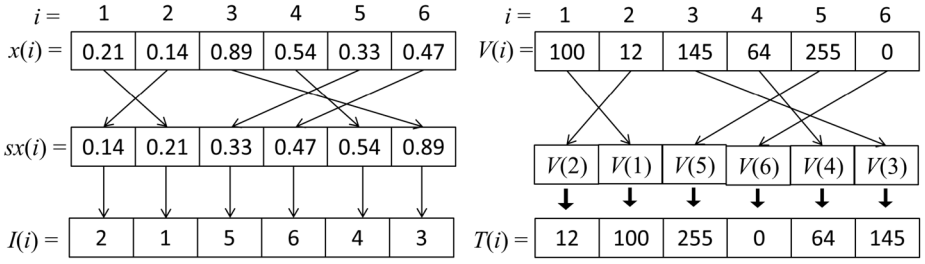


FIGURE 4. The diagrammatic sketch of chaotic sequence sorting and array element scrambling.

TABLE 1. The proposed new S-Box.

*i**j*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	141	249	24	52	144	166	183	139	209	59	19	49	156	219	57	34
2	245	48	239	21	12	237	117	1	29	202	6	65	113	66	185	168
3	246	27	248	226	132	17	127	22	92	31	78	253	178	225	158	140
4	208	74	136	255	128	194	218	42	160	126	121	33	80	159	16	134
5	198	82	196	189	203	149	38	227	41	64	207	108	191	222	204	3
6	119	214	97	7	135	190	172	104	100	15	123	242	107	210	60	46
7	155	157	148	115	37	51	89	11	223	98	143	13	69	152	44	114
8	221	125	120	25	181	197	63	131	173	73	188	170	124	174	250	70
9	171	145	56	76	105	106	142	229	184	195	8	211	68	231	130	133
10	54	79	85	213	95	109	32	179	61	4	235	236	244	238	224	20
11	206	118	5	58	176	163	154	36	2	81	150	192	182	83	164	252
12	23	165	75	9	47	39	72	147	193	28	234	90	103	96	94	200
13	43	151	35	110	129	18	99	10	138	167	232	175	101	247	251	240
14	186	112	91	77	243	169	87	161	111	71	233	199	146	212	116	153
15	220	67	88	241	228	55	254	93	30	122	180	62	50	217	230	177
16	162	45	14	84	215	187	201	216	102	40	26	53	86	0	205	137

Step 6: Permute $\mathbf{V}$ with position index array $\mathbf{I}$ to get a new array $\mathbf{T} = [T(1), T(2), \ldots, T(256)]$, where $T(i)$ is as:

$$T(i) = V(I(i)), \quad i = 1, 2, \ldots, 256 \tag{10}$$

Step 7: Transform the 1D array $\mathbf{T}$ into a 16×16 matrix $\mathbf{S}$, the proposed S-Box is obtained.

Step 8: Output the matrix $\mathbf{S}$ and the chaotic sequence $\mathbf{X} = [X(1), X(2), \ldots, X(L)]$. The algorithm is over.

The diagrammatic sketch of sorting chaotic sequence $\mathbf{x}$ and scrambling array $\mathbf{V}$ to obtain array $\mathbf{T}$, for the example of six elements, is shown in Figure 4.

A new S-Box is generated by the proposed S-Box generation algorithm with parameters set as $\{\mu = 8.8, X(0) = 0.1618, q = 9, L = 65536, d = 46593\}$, which is shown in Table 1.

2) PERFORMANCE TEST OF THE PROPOSED S-BOX

In order to test the cryptographic performance of the S-Box generated by the proposed algorithm, we employed the widely used standard S-Box performance evaluation criteria [37], [38]. In addition, the cryptographic performance of our proposed S-Box with some recently proposed S-boxes are compared, and the results are listed in Table 2. For the performance evaluation criteria, the ideal values of SAC and BIC-SAC are 0.5. The greater the values of Nonlinearity and BIC-NL, the better the performance of an S-Box against linear cryptanalysis attack. The smaller the values of LP and DP, the better the performance of an S-Box against linear cryptanalysis and differential cryptanalysis. From Table 2, one can see that our S-Box has smaller value of DP than most of the S-boxes, which indicates that our S-box has better performance in resisting the attacks of linear cryptanalysis and differential cryptanalysis. Our S-Box has average value of nonlinearity greater than most of the other S-Boxes. Table 2 also indicates that the SAC value (0.503) and the BIC-SAC value (0.507) of our proposed S-box is very near to the ideal value of SAC (0.5).

IV. THE NEW ENCRYPTION SCHEME

The improved S-Box based image encryption scheme in this paper includes the following four stages: The first stage is the generation of S-Box and chaotic sequence, the specific

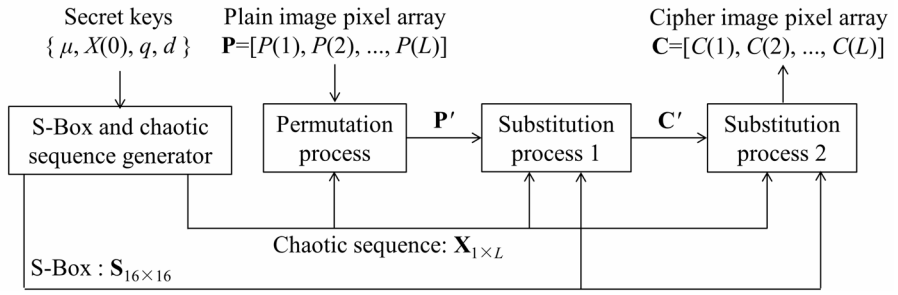


FIGURE 5. The overall flow chart of the improved encryption algorithm.

operations of generating S-Box and chaotic sequence are described in Section III. The second stage is the permutation process, in which the pixels of the image are scrambled. The third stage is the first round of substitution process. The fourth stage is the second round of substitution process. In the substitution process stage, pixel values are replaced by element values of S-Box. The overall flow chart of the proposed new image encryption algorithm is shown in Figure 5. Next, we will elaborate on the specific algorithm of image pixel position scrambling and pixel value substitution.

A. THE PERMUTATION PROCESS

The operation steps of the permutation process are as follows:

(1) Based on the chaotic real value sequence $\mathbf{X}$, we can get a integer index sequence $\mathbf{J}$=$\{J(1), J(2), \ldots, J(L))$ as follows

$$J(i) = 1 + \mathrm{mod}(floor(X(i) \times 10^{15}), L), \quad i = 1, 2, \ldots, L. \tag{11}$$

where, $J(i) \in \{1, 2, \ldots, L\}$.

(2) Initialize two arrays of $\mathbf{P}'$ and $\mathbf{H}$: $\mathbf{P}' = \mathbf{P}$, $\mathbf{H} = \mathbf{J}$.

(3) The permuted image sequence $\mathbf{P}' = [P'(1), P'(2), \ldots, P'(L)]$ is obtained by swapping the i-th element and the $H(i)$-th element. The swapping algorithm is described by the following formulas (12) and (13).

$$t = P'(1),\ P'(1) = P'(H(1)),\ P'(H(1)) = t. \tag{12}$$

$$\begin{cases} H(i) = \mathrm{mod}(P'(i-1) + J(i),\ L) + 1; \\ t = P'(i),\ P'(i) = P'(H(i)),\ P'(H(i)) = t; \\ i = 2,\ 3,\ \ldots,\ L. \end{cases} \tag{13}$$

Eq. (13) shows that the generation of permutation sequence $H(i)$ is related to the (i-1)-th pixel of the permuted image pixel. As the result, the permutation sequence $H(i)$ used to encrypt different plaintext images is distinct, so that the algorithm can resist chosen-plaintext attack.

B. THE SUBSTITUTION PROCESS

There two rounds of substitution process in the algorithm.

1) THE FIRST ROUND OF SUBSTITUTION PROCESS

The specific steps of the process are as follows.

Input: S-Box matrix $\mathbf{S}$, chaotic real value sequence $\mathbf{X}$, and the permuted image array $\mathbf{P}'$.

Output: The temporary ciphertext image array $\mathbf{C}'$.

Step 1: The chaotic real value sequence $\mathbf{X}$ is transformed into a key sequence $\mathbf{K}$ consist of 8-bit integers $\{K(i)\}$, $i = 1, 2, \ldots, L$. The conversion formula is

$$K(i) = \mathrm{mod}[floor[X(i) \times 10^{15}], 256], \quad i = 1, 2, \ldots, L. \tag{14}$$

Step 2: Each pixel in the permuted sequence $\mathbf{P}'$ is substituted with the S-Box matrix $\mathbf{S}$ and the key sequence $\mathbf{K}$. The formula for encrypting the first pixel is

$$\begin{cases} j = \mathrm{mod}[(1 + P'(2)), L] + 1, \\ C'(1) = \mathrm{sub\_byte}[\mathbf{S}, \mathrm{C0} \oplus P'(1) \oplus K(j)]. \end{cases} \tag{15}$$

where, C0 $\in \{1, 2, \ldots, 255\}$ is a constant to be set.

Step 3: Encrypt the i-th pixel by

$$\begin{cases} j = \mathrm{mod}[(i + P'(i+1)), L] + 1; \\ C'(i) = \mathrm{sub\_byte}[\mathbf{S}, C'(i-1) \oplus P'(i) \oplus K(j)]; \\ i = 2, 3, \ldots, L-1. \end{cases} \tag{16}$$

Step 4: Encrypt the last L-th pixel by

$$\begin{cases} j = \mathrm{mod}[(L + C0 + CL), L] + 1; \\ C'(L) = \mathrm{sub\_byte}[\mathbf{S}, C'(L-1) \oplus P'(L) \oplus K(j)]. \end{cases} \tag{17}$$

where, $CL \in \{1, 2, \ldots, 255\}$ is a constant to be set. $\mathbf{S}$ is the matrix of S-Box and $\mathbf{S} = [S(row, col)]$, $S(row, col) \in \{0, 1, 2, \ldots, 255\}$; $row = 1, 2, \ldots, 16$, $col = 1, 2, \ldots, 16$. sub_byte[$\mathbf{S}$, z] represents the byte substitution operation function that uses $\mathbf{S}$ substitute for z. The algorithm of sub_byte[$\mathbf{S}$, z] can be described by Algorithm 2.

Algorithm 2 The Pseudo Code of Sub_Byte[S, z]

Input:	$\mathbf{S} = [S(row, col)]$ an z, $S(row, col) \in \{0, 1, 2, \ldots, 255\}$; $z \in \{0, 1, 2, \ldots, 255\}$.
Output:	c = sub_byte[S, z].
1:	Let $col \leftarrow$ mod(z, 16)+1;
2:	Let $row \leftarrow$ (z-mod(z, 16))/16+1;
3:	c $\leftarrow S(row, col)$.

TABLE 2. Performance comparison of different S-Boxes.

S-Box Methods	SAC	Nonlinearity			BIC-SAC	BIC-NL	LP	DP
		Min.	Max.	Average				
Ref. [6]	0.500	98	106	103.3	0.504	104.0	0.133	0.047
Ref. [9]	0.507	104	108	106.8	0.507	103.9	0.140	0.054
Ref. [10]	0.498	102	108	104.5	0.508	104.6	0.125	0.047
Ref. [16]	0.500	100	110	105.5	0.497	103.8	0.125	0.047
Ref. [19]	0.496	102	108	105.3	0.499	103.8	0.156	0.039
Ref. [20]	0.520	104	110	106.3	0.501	104.2	0.133	0.039
Ref. [26]	0.495	104	110	106.5	0.498	103.8	0.141	0.039
Ref. [38]	0.505	104	110	106.3	0.499	103.8	0.125	0.039
Ref. [39]	0.499	100	110	105.5	0.506	106.0	0.133	0.125
Ref. [40]	0.501	108	112	109.3	0.506	108.2	0.094	0.031
This paper	0.503	104	110	106.3	0.507	103.9	0.133	0.039

2) THE SECOND ROUND OF SUBSTITUTION PROCESS

The specific steps of the algorithm are as follows.

Input: S-Box matrix **S**, secret key sequence **K**, and the temporary ciphertext image array **C**′.

Output: The final ciphertext image array **C**.

Step 1: Encrypt the first pixel by

$$\begin{cases} j = \mathrm{mod}[(1 + C'(2)), L] + 1, \\ C(1) = \mathrm{sub\_byte}[\mathbf{S}, C'(1) \oplus C'(L) \oplus K(j)]. \end{cases} \tag{18}$$

Step 2: Encrypt the i-th pixel by

$$\begin{cases} j = \mathrm{mod}[(i + C'(i + 1)), L] + 1; \\ C(i) = \mathrm{sub\_byte}[\mathbf{S}, C(i - 1) \oplus C'(i) \oplus K(j)]; \\ i = 2, 3, \ldots, L - 1. \end{cases} \tag{19}$$

Step 3: Encrypt the last L-th pixel by

$$\begin{cases} j = \mathrm{mod}[(L + C0 + CL), L] + 1; \\ C(L) = \mathrm{sub\_byte}[\mathbf{S}, C(L - 1) \oplus C'(L) \oplus K(j)].. \end{cases} \tag{20}$$

C. THE DECRYPTION ALGORITHM

The image decryption process is the inverse operation of image encryption process. Firstly, generate S-Box **S** and chaotic sequence **X** with secret key parameters. Then perform the following operations.

1) THE FIRST ROUND OF INVERSE SUBSTITUTION PROCESS

In this stage, each pixel in the cipher sequence **C** is inverse substituted with the S-Box and the key sequence **K**. The inverse substitution procedure processes pixels in the reverse order of the substitution process.

Input: The final ciphertext image array **C**.

Output: The temporary ciphertext image array **C**′.

Step 1: Decrypt the last pixel by

$$\begin{cases} j = \mathrm{mod}[(L + C0 + CL), L] + 1, \\ C'(L) = \mathrm{sub\_byte1}[\mathbf{S}, C(L)] \oplus C(L - 1) \oplus K(j). \end{cases} \tag{21}$$

Algorithm 3 The Pseudo Code of Sub_Byte_1[S, C]

Input: $\mathbf{S} = [S(row, col)]$ and C,
$S(row, col) \in \{0, 1, 2, \ldots, 255\}$;
$C \in \{0, 1, 2, \ldots, 255\}$.

Output: $z = \mathrm{sub\_byte\_1}[S, C]$.

1: Find the position (row, col) in **S** such that $S(row, col) = \mathrm{c}$;

2: Let $row \leftarrow row$ -1; $col \leftarrow col$ -1;

3: $\mathrm{z} \leftarrow row \times 16 + col$.

Step 2: Decrypt the i-th pixel by

$$\begin{cases} j = \mathrm{mod}[(i + C'(i + 1)), L] + 1, \\ C'(i) = \mathrm{sub\_byte\_1}[\mathbf{S}, C(i)] \oplus C(i - 1) \oplus K(j)] \\ i = L - 1, L - 2, \ldots, 2. \end{cases} \tag{22}$$

Step 3: Decrypt the first pixel by

$$\begin{cases} j = \mathrm{mod}[(1 + C'(2)), L] + 1, \\ C'(1) = \mathrm{sub\_byte\_1}[\mathbf{S}, C(1)] \oplus C'(L) \oplus K(j)]. \end{cases} \tag{23}$$

where, sub_byte_1[**S**, C] represents the inverse process of byte substitution. The algorithm of sub_byte_1[**S**, C] function can be described by **Algorithm** 3.

2) THE SECOND ROUND OF INVERSE SUBSTITUTION PROCESS

Input: The temporary ciphertext image array **C**'.

Output: The permuted image array **P**'.

Step 1: Decrypt the last pixel by

$$\begin{cases} j = \mathrm{mod}[(L + C0 + CL), L] + 1, \\ P'(L) = \mathrm{sub\_byte\_1}[\mathbf{S}, C'(L)] \oplus C'(L - 1) \oplus K(j). \end{cases} \tag{24}$$

Step 2: Decrypt the i-th pixel by

$$\begin{cases} j = \text{mod}[(i + P'(i+1)), L] + 1, \\ P'(i) = \text{sub\_byte\_1}[\mathbf{S}, C'(i)] \oplus C'(i-1) \oplus K(j)] \\ i = L-1, L-2, \ldots, 2. \end{cases} \quad (25)$$

Step 3: Decrypt the first pixel by

$$\begin{cases} j = \text{mod}[(1 + P'(2)), L] + 1, \\ P'(1) = \text{sub\_byte\_1}[\mathbf{S}, C'(1)] \oplus C0 \oplus K(j)]. \end{cases} \quad (26)$$

3) INVERSE PERMUTATION PROCESS

In the inverse permutation process, the processing direction of pixels is backward. The steps of the algorithm are as follows.

Input: S-Box matrix $\mathbf{S}$, chaotic sequence $\mathbf{X}$, and the permuted image array $\mathbf{P}'$.

Output: The recovered plaintext image array $\mathbf{P}$.

Step 1: Based on the chaotic real value sequence $\mathbf{X}$, we can get a integer index sequence $\mathbf{J} = \{J(1), J(2), \ldots, J(L))$ by using Eq. (11).

Step 2: Initialize two arrays of $\mathbf{P}$ and $\mathbf{H}$: $\mathbf{P} = \mathbf{P}'$, $\mathbf{H} = \mathbf{J}$.

Step 3: The recovered image sequence $\mathbf{P} = [P(1), P(2), \ldots, P(L)]$ is obtained by swapping the i-th element and the $H(i)$-th element. The swapping algorithm is described by the following formulas (27) and (28).

$$\begin{cases} H(i) = \text{mod}(P(i-1) + J(i),\ L) + 1; \\ t = P(i),\ P(i) = P(H(i)),\ P(H(i)) = t; \\ i = L, L-1,\ \ldots,\ 3, 2. \end{cases} \quad (27)$$

$$t = P(1),\ P(1) = P(H(1)),\quad P(H(1)) = t. \quad (28)$$

V. EXPERIMENTAL SIMULATION AND SECURITY ANALYSIS FOR THE ENCRYPTION ALGORITHM

To examine the security performance of the proposed encryption scheme, some experimental tests are carried out. The standard test images with size of 256 × 256 come from USC-SIPI test image database, such as lena, tree, baboon, cameraman, are used in the simulation experiments. The secret keys are set as follows: $\mu = 8.80, X(0) = 0.1618, q = 9, d = 46593$. The constant parameters are C0 = 176, CL = 168.

A. THE ENCRYPTION EFFECT

The plain images tested in the experiment and their corresponding encrypted images are shown in Figure 6, respectively. It can been seen that the encrypted images are completely confused and unidentified. Therefore, the proposed encryption scheme achieves satisfactory encryption effect.

B. KEY SPACE ANALYSIS

The number of keys available in an encryption algorithm is the size of its key space. In order to resist brute force attack, the key space of an encryption system must be large enough. In our proposed encryption scheme, the key set is $\{\mu, X(0), q, d\}$. There are 10 possible values for the integer part of μ, and the decimal part can be accurate to 15 decimal places. So, there are 10^{16} values for μ. $X(0)$ has 10^{15} values, d has 10^4 values, q has 256 values, So, the key space of our encryption scheme can reach $10^{16} \times 10^{15} \times 10^4 \times 256 \approx 2^{124}$. The key space of a secure cryptosystem should be greater than 2^{100}, which is pointed out in [41]. Both ENISA and NIST guidelines requires the key size to be at least 2^{112}. The key space of our encryption scheme is larger than the above standards. Therefore, the key space of our proposed algorithm is large enough to resist brute force attack.

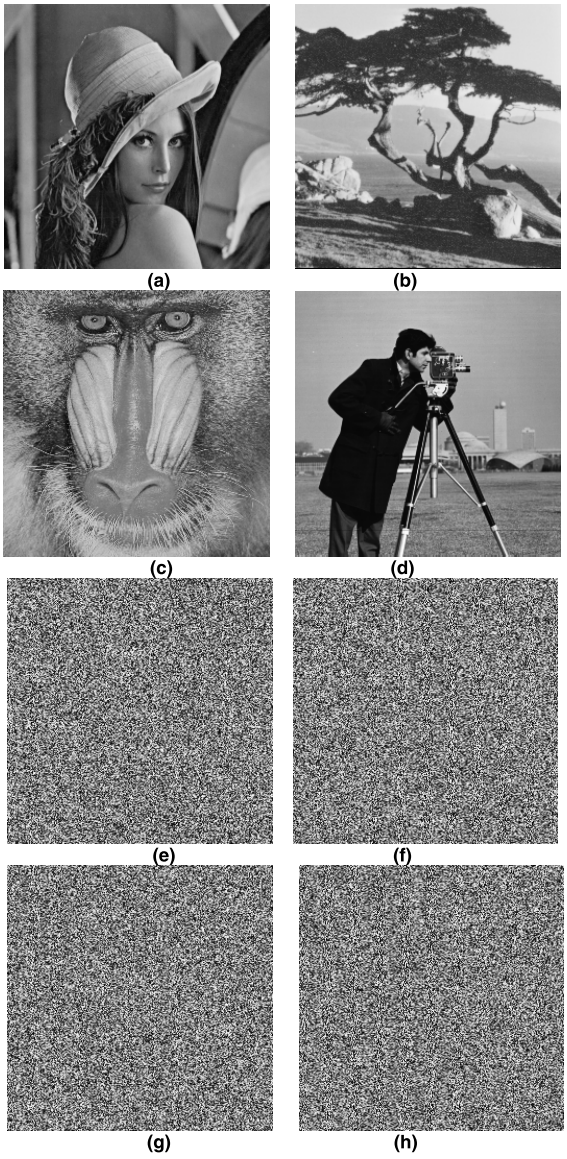


FIGURE 6. Experimental simulation results. (a) Plaint image of lena. (b). Plaint image of tree. (c) Plaint image of baboon. (d) Plaint image of cameraman. (e) Cipher image of lena (f) Cipher image of tree. (g) Cipher image of baboon. (h) Cipher image of cameraman.

C. STATISTICAL CHARACTERISTIC ANALYSIS

It is necessary to carry out statistical analysis. In this part, the security evaluation of the encryption algorithm is carried out from the aspects of image histogram analysis, adjacent pixel correlation and information entropy.

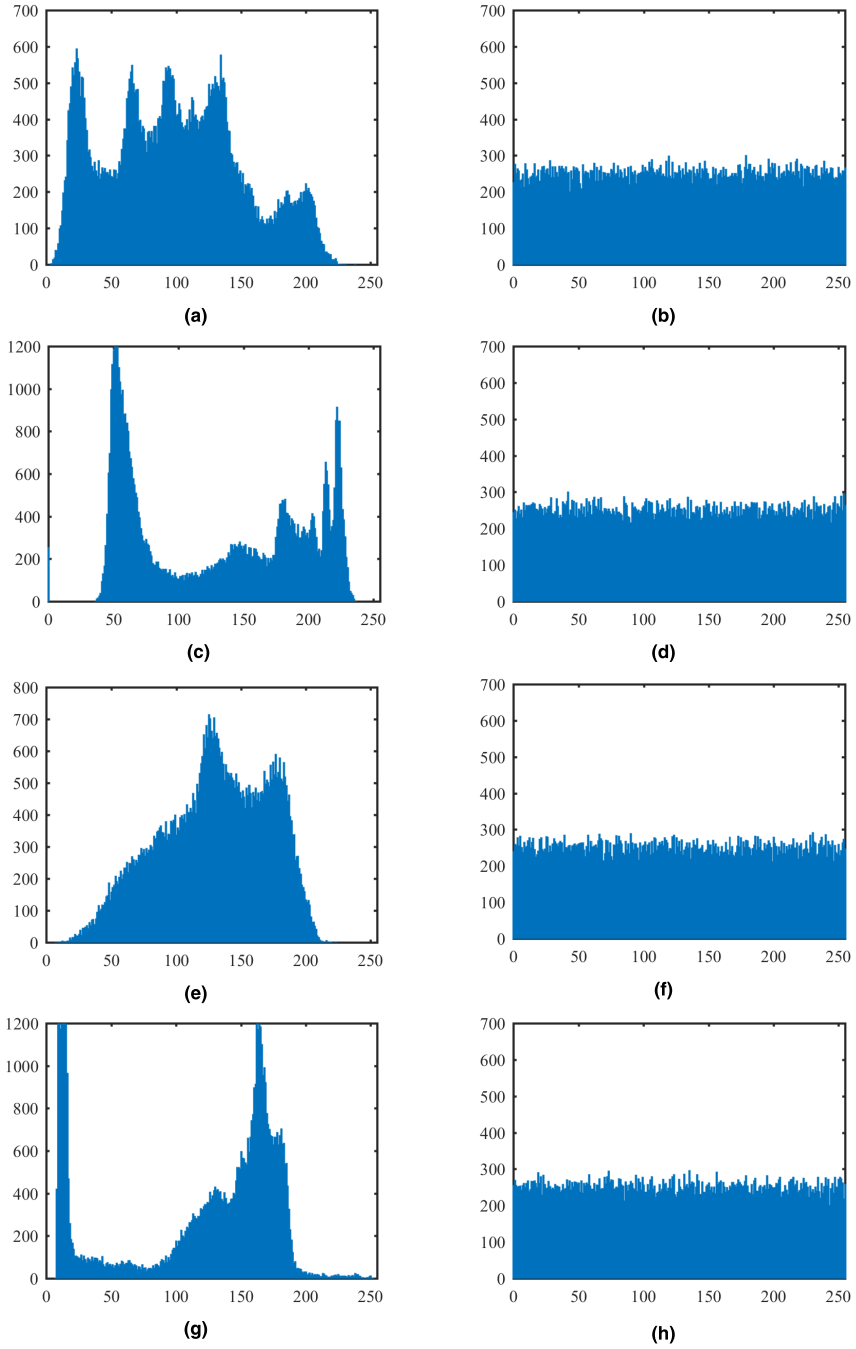


FIGURE 7. Histogram analysis. (a) Histogram of plain image lena. (b) Histogram of encrypted image lena. (c) Histogram of plain image tree. (d) Histogram of encrypted image tree. (e) Histogram of plain image baboon. (f) Histogram of encrypted image baboon. (g) Histogram of plain image cameraman. (h) Histogram of encrypted image cameraman.

1) HISTOGRAM ANALYSIS

A histogram of an image exposes the pixel distribution characteristics of the image. The more uniform the histogram distribution of an encrypted image, the better the performance of ciphertext resistance statistical analysis. Figure 7 shows the histograms of the three test images and their corresponding cipher images obtained by our improved encryption algorithm. From Figure 7, we can see that the histogram distribution of each plain image is clearly not uniform but the histogram distribution of each encrypted image is very uniform.

The variance of a histogram can also be employed to describe quantitatively the distribution characteristics of a histogram, which is calculated by [42]

$$\mathrm{var}(\mathbf{Z}) = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{1}{2}(z_i - z_j)^2 \tag{29}$$

TABLE 3. Variances of histograms of the encrypted images.

Images	Plain image	Cipher image	Cipher image [24]	Cipher image [26]	Cipher image [27]
lena	30,665.703	262.5000	284.5781	280.1172	609.7266
tree	69,330.922	270.5859	257.6875	310.6641	496.1797
baboon	47,799.055	267.8125	268.2109	272.9219	405.9375
cameraman	110,973.305	201.7891	223.3594	339.5938	1074.2000
Average	64,692.246	**250.6719**	258.4590	300.8242	646.5109

TABLE 4. Results of applying nist test to the cipher image lena with size 512 × 512.

NIST statistical test	P-value	Results
Frequency (monobit)	0.777944	PASS
Block Frequency (m = 128)	0.379951	PASS
Cumulative Sums (Forward)	0.707351	PASS
Cumulative Sums (Reverse)	0.886012	PASS
Runs	0.362864	PASS
Longest Run of Ones	0.235317	PASS
Rank	0.554236	PASS
FFT	0.755036	PASS
Non-Overlapping Templates	0.010808	PASS
Overlapping Templates (m = 9)	0.098332	PASS
Universal	0.313259	PASS
Approximate Entropy(m = 10)	0.105506	PASS
Random-Excursions	/	TEST NOT APPLICABLE
Random-Excursions Variant	/	TEST NOT APPLICABLE
Serial Test 1 (m = 16)	0.261001	PASS
Serial Test 2 (m = 16)	0.213789	PASS
Linear complexity (M = 500)	0.548827	PASS

where, N denotes the number of gray levels of an image. For an 8-bit gray image, $N = 256$. $\mathbf{Z}$ is a vector and $\mathbf{Z} = \{z_0, z_1, \ldots, z_{N-1}\}$, z_i and z_j are the numbers of pixels with gray values equal to i and j respectively. The smaller the variance of a histogram is, the flatter the histogram is, and the number of each gray level pixel in the image tends to be equal. The ideal value of var($\mathbf{z}$) is 0, which means $z_i = z_j$ for all i and j. The variances of histograms of the tested plain images (size of 256 × 256) and their corresponding cipher images are listed in Table 3. The variances of histograms of cipher images obtained by other algorithms are also listed in Table 3. The average variance of the cipher images encrypted by our improved algorithm is 250.6719, which is the smallest one among the results of four algorithms. Thus, our improved scheme has better performance in resisting statistical attacks. We also applied the Kolmogorov-Smirnov Goodness of Fit Test to check if the distributions are uniform. Let X_1 represent our ciphertext, X_2 represent completely evenly distributed ciphertext, by using MATLAB command H = kstest2(X1, X2), and we get the results H = 0 in all the case. The test results show that the ciphertext images obtained by our encryption algorithm are uniformly distributed.

Further, the NIST statistical tests are provided to check if the ciphertext image are random. We take the ciphertext image Lena with size 512 × 512 × 8 bits as example. The NIST test results are listed in Table 4. The ''Non-Overlapping Templates'' contains multiple tests, the listed is the worst case (P-value = 0.010808 > 0.01). The tests of ''Random-Excursions'' and '' Random-Excursions Variant'' are not applicable to binary sequences with insufficient number of cycles. From Table 4, one can see that all the applicable tests successfully passed, which verified that the ciphertext image distribution is random.

2) PIXEL CORRELATION ANALYSIS

In general, there is a strong correlation between adjacent pixels in the horizontal, vertical, and diagonal directions of a plaintext image, and there should be no correlation between adjacent pixels in the ciphertext image. We can introduce correlation coefficient to measure the correlation between adjacent pixels. We randomly select m pairs of adjacent pixels from the analyzed image, denote the i-th pairs of the adjacent pixel gray-scale values as (A_i, B_i), $i = 1, 2, \ldots, m$, and the correlation coefficient between vectors $\mathbf{A} = \{A_1, A_2, \ldots, A_m\}$ and $\mathbf{B} = \{B_1, B_2, \ldots, B_m\}$ can be calculated by

$$r_{\mathbf{AB}} = \frac{\sum_{i=1}^{m} \left(A_i - \overline{\mathbf{A}}\right)\left(B_i - \overline{\mathbf{B}}\right)}{\sqrt{\sum_{i=1}^{m} \left(A_i - \overline{\mathbf{A}}\right)^2} \sqrt{\sum_{i=1}^{m} \left(B_i - \overline{\mathbf{B}}\right)^2}} \tag{30}$$

TABLE 5. Correlation coefficients of cipher images encrypted by different algorithms.

Images	Directions	Plain image	Cipher image	Cipher image [24]	Cipher image [26]	Cipher image [27]
lena	Horizontal	0.940080	-0.005649	**-0.000582**	0.004414	0.009721
	Vertical	0.969460	**0.000610**	0.001336	0.008837	0.087607
	Diagonal	0.91797	0.001835	-0.004690	-0.004907	**0.001279**
tree	Horizontal	0.967700	**-0.000756**	-0.001377	0.000863	0.004004
	Vertical	0.944820	**-0.002332**	0.003585	0.011904	0.083118
	Diagonal	0.92933	-0.003625	-0.003176	**0.000121**	0.001288
baboon	Horizontal	0.731510	0.000741	-0.003773	**0.000014**	-0.001829
	Vertical	0.654480	**-0.000415**	-0.002788	0.012304	0.013993
	Diagonal	0.641190	-0.001437	**-0.000712**	0.001539	0.004556
cameraman	Horizontal	0.933480	**0.002941**	-0.003117	0.007432	0.004339
	Vertical	0.959220	**-0.000775**	-0.003952	0.013917	0.108790
	Diagonal	0.908660	-0.001903	**-0.000479**	-0.003196	0.001813

where, $\overline{\mathbf{A}}$ is the average value of vector $\mathbf{A} = \{A_1, A_2, \ldots, A_m\}$, and $\overline{\mathbf{B}}$ is the average value of vector $\mathbf{B} = \{B_1, B_2, \ldots, B_m\}$. We randomly selected $m = 2000$ pairs of pixel along a certain direction (horizontal or vertical or diagonal) from an image, then calculated the correlation coefficients of cipher images encrypted by our proposed algorithm, the results are listed in Table 5. Table 5 also lists the correlation coefficients of the cipher images encrypted with Zhang's, Wang's and Çavuşoğlu's algorithm. The data with the lowest absolute value in each row is displayed in bold, indicating the best result. Our algorithm has 6 best results in total, 3 best results in Ref. [24], 2 best results in Ref. [26], and 1 best result in Ref. [27]. So, in general, our algorithm is superior to the other three algorithms.

3) INFORMATION ENTROPY ANALYSIS OF CIPHERTEXT IMAGE

Information entropy of an image reflects the uncertainty of the image information. The larger the information entropy of an image is, the more uncertain the distribution of the image information is, and the visual information is less. The formula for calculating the information entropy $H(s)$ of an image is as follows

$$H(s) = -\sum_{i=0}^{2^n-1} P(s_i)\log_2[P(s_i)]. \tag{31}$$

where $P(s_i)$ denotes the occurrence probability of the gray level i, and $i = 0, 1, 2, \ldots, 2^n - 1$. Here, 2^n is the number of grayscale levels of an image. If each s_i has the same occurrence probability in an image, then $P(s_i) = 1/2^n$. For 8-bit grey images ($n = 8$), the theoretical value of the information entropy is 8. The calculated information entropy values of the encrypted images by our encryption algorithm are shown in Table 6. From the test results in Table 6, one can see that the information entropy values are very close to the theoretical value 8. The information entropy values of cipher images encrypted by our proposed algorithm are obviously more close to 8 and better than those of the other three algorithms.

D. ANALYSIS OF SENSITIVITY TO PLAINTEXT

When only one bit of the plaintext image changes, it can change almost all pixels of the ciphertext image, so the algorithm is sensitive to plaintext. Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are commonly used to measure the sensitivity of encryption algorithms to plaintext. Suppose $\mathbf{C} = [C(i,j)]$ represents the ciphertext image corresponding to the original plaintext image, and $\mathbf{C}' = [C'(i,j)]$ represents the ciphertext image corresponding to the plaintext image that there is one bit changed. Then, *NPCR* and *UACI* can be calculated by

$$D(i,j) = \begin{cases} 1, & if\ C(i,j) \neq C'(i,j) \\ 0, & f\ C(i,j) = C'(i,j) \end{cases} \tag{32}$$

$$NPCR = \left(\frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n} D_{ij}\right) \times 100\% \tag{33}$$

$$UACI = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\left|C(i,j) - C'(i,j)\right|}{255} \times 100\% \tag{34}$$

where, $m \times n$ is the size of the image. $C(i, j)$ represent the pixel at the i-th row and j-th column of the ciphertext image corresponding to the original plaintext image, and $C'(i, j)$ represent the pixel at the i-th row and j-th column of the ciphertext image corresponding to the changed plaintext image. For 8-bit grayscale images, the expected values of *NPCR* and *UACI* are 99.6094% and 33.4635%, respectively. We performed four groups of encryption experiments with lena image in size of 256 × 256. Each time the value of one pixel in the plaintext image was increased by 1, then the image was encrypted. Then compare with the ciphertext image of the original plaintext image. The changed pixel includes positions of 1, $L/4$, $L/2$, $3L/4$, and L. The test results of *NPCR* and *UACI* corresponding to the slightly changed plaintext image in our algorithm are shown in Table 7. The experimental results show that our encryption algorithm is extremely sensitive to plaintext.

TABLE 6. Information entropy of encrypted image.

Algorithms	lena	tree	baboon	cameraman
This paper	**7.9971**	7.9970	**7.9971**	**7.9978**
Ref. [24]	7.9969	**7.9972**	7.9970	7.9975
Ref. [26]	7.9969	7.9966	7.9970	7.9963
Ref. [27]	7.9932	7.9946	7.9955	7.9883

TABLE 7. *NPCR* and *UACI* test results of slight change of plaintext in our algorithm.

Position	*NPCR* (%)	*UACI* (%)
1	99.6216	33.5848
$L/4$	99.7101	33.4457
$L/2$	99.7406	33.4577
$3L/4$	99.6307	33.4551
L	99.5743	33.3941

TABLE 8. Error keys used in decryption.

Keys	δ of μ	δ of $X(0)$	δ of q	δ of d
key1	10^{-15}	0	0	0
key2	0	10^{-15}	0	0
key3	0	0	1	0
key4	0	0	0	1

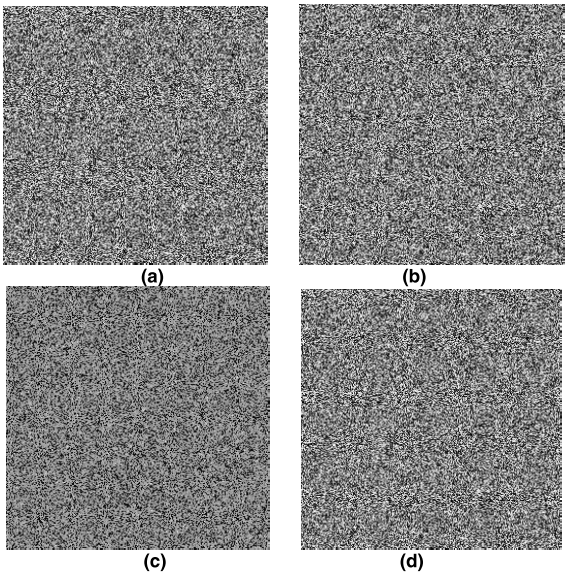


FIGURE 8. The image decrypted by the wrong key. (a) the decrypted image with key1. (b) the decrypted image with key2. (c) the decrypted image with key3. (d) the decrypted image with key4.

E. ANALYSIS OF SENSITIVITY TO KEY

A secure encryption algorithm should be sensitive to the key in order to resist exhaustive attack. Key sensitivity means that if there is a tiny difference between the decryption key and the encryption key, the useful information cannot be decrypted at all. We select the following error key sets {Key1, Key2, Key3, Key4} in Table 8 to decrypt the original ciphertext image. Only one key in each set has a small error δ and the decrypted images are shown in Figure 8. It can be seen that no useful information of the original image can be obtained from any of the decrypted image, which confirms that our algorithm is sensitivity to the key.

F. ANALYSIS OF CHOSEN-PLAINTEXT ATTACK

An attacker could construct a special image to obtain ciphertext so that to obtain information about the encryption scheme, and crack the target cipher image encrypted by the scheme. This kind of attack is called chosen-plaintext attack. In the permutation process of our proposed scheme, the key value $H(i)$ of the proposed cryptosystem for swapping the i-th pixel is related to the $(i-1)$-th permuted image pixel $P'(i-1)$. In the substitution process of our proposed scheme, the key value $K(j)$ of the proposed cryptosystem for encrypting the i-th pixel is related to the $(i+1)$-th permuted image pixel $P'(i+1)$. So that different plaintext images will generate different sequence of **H** and **K**, which makes the encryption process related to the content of image to be encrypted. In other words, although the initial secret keys are the same, the key-streams **H** and **K** are different when different plaintext images are encrypted, so that our proposed algorithm can effectively resist chosen-plaintext attack.

TABLE 9. Encryption time and comparisons (unit: second).

Algorithms	Encryption time (256×256)	Encryption time (512×512)
This paper	0.382	1.489
Ref. [24]	1.212	4.749
Ref. [26]	1.245	4.826
Ref. [27]	0.959	3.253

G. SPEED TEST

Encryption speed is very important for the practicability of an image encryption algorithm. In order to compare the encryption speed of several S-Box based image encryption algorithms, we run four algorithms on the same software and hardware computer platform. The hardware environment is a PC with 3.3 GHz CPU, and 4 GB memory, and the software platform is MATLAB R2016b with Microsoft Windows 7 operating system. We take the average value of five experiments, and the results are listed in Table 9. The improved image encryption algorithm in this paper has the fast encryption speed, its average time for encrypting a 256×256 size image is 0.382 seconds, and the average time for encrypting a 512×512 size image is 1.489 seconds. The speed of our improved image encryption algorithm is much faster than those of the other three algorithms. According to reference [41], if the frequency of the CPU is f_z, the encryption speed should be greater than $f_z/100$. Considering that MATLAB implementation is typically slower than other implementations, we implement the proposed encryption algorithm with C++ to test the speed. It takes about 0.003 seconds to encrypt a $256\times256\times8$ bits gray image. So, the encryption speed is equivalent to 174.76 Mbps. Considering the frequency of the CPU $f_z = 3.3$ GHz, therefore, the speed is about $f_z/18.88$ and greater than $f_z/100$. If the code is further optimized, the speed will be further improved.

VI. CONCLUSION

Based on the security analysis of an image encryption algorithm using chaotic S-Boxes, an efficient new image encryption scheme based on chaos and single S-Box is proposed. Firstly, in order to improve the cryptography performance

of chaotic systems, a new discrete compound chaotic system, LSS, is proposed, which has wider chaotic range and better chaotic performance than Logistic and Sine chaotic systems. Secondly, a strong S-Box is constructed by using the new chaotic system, and the cryptographic strength of the proposed S-Box is tested. It is verified that the proposed S-Box has satisfactory cryptographic performance. Thirdly, the new image encryption algorithm based on LSS and single S-Box is proposed. The new encryption scheme consists of permutation and substitution process, and the strategy of key associated with image content is introduced in the encryption process. This strategy can bring about the effect of ''one-time pad'' [43] and enables the algorithm to resist chosen-plaintext attack (CPA). Using the new chaotic system can improve the security of chaotic key sequence, and using single S-Box can improve the encryption speed. Experimental results and security analysis verified the effectiveness of the proposed image encryption scheme. It has obvious efficiency advantages, which means that the new image encryption scheme has better application potential in real-time image encryption.

REFERENCES

[1] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu, and J. Liu, ''A novel block encryption algorithm based on chaotic S-box for wireless sensor network,'' *IEEE Access*, vol. 7, pp. 53079–53090, 2019.

[2] H. Liu, A. Kadir, X. Sun, and Y. Li, ''Chaos based adaptive double-image encryption scheme using hash function and S-boxes,'' *Multimed Tools Appl.*, vol. 77, no. 1, pp. 1391–1407, Jan. 2018.

[3] 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 Process.*, vol. 142, pp. 340–353, Jan. 2018.

[4] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, ''A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system,'' *Nonlinear Dyn.*, vol. 87, no. 2, pp. 1081–1094, Jan. 2017.

[5] D. Lambić, ''A novel method of S-box design based on discrete chaotic map,'' *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2407–2413, Mar. 2017.

[6] E. Tanyildizi and F. Ozkaynak, ''A new chaotic S-box generation method using parameter optimization of one dimensional chaotic maps,'' *IEEE Access*, vol. 7, pp. 117829–117838, 2019.

[7] I. Hussain, A. Anees, T. Al-Maadeed, and M. Mustafa, ''Construction of S-box based on chaotic map and algebraic structures,'' *Symmetry*, vol. 11, no. 3, p. 351, Mar. 2019.

[8] A. Rafiq and M. Khan, ''Construction of new S-boxes based on triangle groups and its applications in copyright protection,'' *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 15527–15544, Jun. 2019.

[9] A. Zahid and M. Arshad, ''An innovative design of substitution-boxes using cubic polynomial mapping,'' *Symmetry*, vol. 11, no. 3, p. 437, Mar. 2019.

[10] L. Liu, Y. Zhang, and X. Wang, ''A novel method for constructing the S-box based on spatiotemporal chaotic dynamics,'' *Appl. Sci.*, vol. 8, no. 12, p. 2650, Dec. 2018.

[11] S. Zhu, C. Zhu, H. Cui, and W. Wang, ''A class of quadratic polynomial chaotic maps and its application in cryptography,'' *IEEE Access*, vol. 7, pp. 34141–34152, 2019.

[12] S. Zhu, C. Zhu, and W. Wang, ''A new image encryption algorithm based on chaos and secure hash SHA-256,'' *Entropy*, vol. 20, no. 9, p. 716, Sep. 2018.

[13] S. Zhu and C. Zhu, ''A new image compression-encryption scheme based on compressive sensing and cyclic shift,'' *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 20855–20875, Aug. 2019.

[14] L. G. De La Fraga, E. Torres-Pérez, E. Tlelo-Cuautle, and C. Mancillas-López, ''Hardware implementation of pseudo-random number generators based on chaotic maps,'' *Nonlinear Dyn.*, vol. 90, no. 3, pp. 1661–1670, Nov. 2017.

[15] A. Ullah, S. S. Jamal, and T. Shah, ''A novel construction of substitution box using a combination of chaotic maps with improved chaotic range,'' *Nonlinear Dyn.*, vol. 88, no. 4, pp. 2757–2769, Jun. 2017.

[16] A. Belazi and A. A. A. El-Latif, ''A simple yet efficient S-box method based on chaotic sine map,'' *Optik*, vol. 130, pp. 1438–1444, Feb. 2017.

[17] E. Al Solami, M. Ahmad, C. Volos, M. Doja, and M. Beg, ''A new hyperchaotic system-based design for efficient bijective substitution-boxes,'' *Entropy*, vol. 20, no. 7, p. 525, Jul. 2018.

[18] M. Khan, T. Shah, and S. I. Batool, ''Construction of S-box based on chaotic Boolean functions and its application in image encryption,'' *Neural Comput. Appl.*, vol. 27, no. 3, pp. 677–685, Apr. 2016.

[19] A. Belazi, M. Khan, A. A. A. El-Latif, and S. Belghith, ''Efficient cryptosystem approaches: S-boxes and permutation-substitution-based encryption,'' *Nonlinear Dyn.*, vol. 87, no. 1, pp. 337–361, 2017.

[20] X. Wang, A. Akgul, U. Cavusoglu, V.-T. Pham, D. Vo Hoang, and X. Nguyen, ''A chaotic system with infinite equilibria and its S-box constructing application,'' *Appl. Sci.*, vol. 8, no. 11, p. 2132, Nov. 2018.

[21] M. Khan, ''A novel image encryption scheme based on multiple chaotic S-boxes,'' *Nonlinear Dyn.*, vol. 82, nos. 1–2, pp. 527–533, Oct. 2015.

[22] X. Wang and Q. Wang, ''A novel image encryption algorithm based on dynamic S-boxes constructed by chaos,'' *Nonlinear Dyn.*, vol. 75, no. 3, pp. 567–576, Feb. 2014.

[23] I. Hussain, A. Anees, A. H. Alkhaldi, M. Aslam, N. Siddiqui, and R. Ahmed, ''Image encryption based on Chebyshev chaotic map and S8 S-boxes,'' *Opt. Appl.*, vol. 49, no. 2, pp. 317–330, 2019.

[24] X.-P. Zhang, R. Guo, H.-W. Chen, Z.-M. Zhao, and J.-Y. Wang, ''Efficient image encryption scheme with synchronous substitution and diffusion based on double S-boxes,'' *Chin. Phys. B*, vol. 27, no. 8, Aug. 2018, Art. no. 080701.

[25] S. Zhu, G. Wang, and C. Zhu, ''A secure and fast image encryption scheme based on double chaotic S-boxes,'' *Entropy*, vol. 21, no. 8, p. 790, Aug. 2019.

[26] X. Wang, Ü. Çavuşoğlu, S. Kacar, A. Akgul, V.-T. Pham, S. Jafari, F. Alsaadi, and X. Nguyen, ''S-box based image encryption application using a chaotic system without equilibrium,'' *Appl. Sci.*, vol. 9, no. 4, p. 781, Feb. 2019.

[27] Ü. Çavuşoğlu, S. Kaçar, I. Pehlivan, and A. Zengin, ''Secure image encryption algorithm design using a novel chaos based S-Box,'' *Chaos, Solitons Fractals*, vol. 95, pp. 92–101, Feb. 2017.

[28] K.-H. Sun, S.-B. He, L.-Z. Yin, and A. D.-L. D. Li-Kun, ''Application of fuzzyen algorithm to the analysis of complexity of chaotic sequence,'' *Acta Physica Sinica*, vol. 61, no. 13, Jul. 2012, Art. no. 130507.

[29] K.-H. Sun, S.-B. He, Y. He, and L.-Z. Yin, ''Complexity analysis of chaotic pseudo-random sequences based on spectral entropy algorithm,'' *Acta Phys. Sinica*, vol. 62, no. 1, Jan. 2013, Art. no. 010501.

[30] S.-B. He, K.-H. Sun, and C.-X. Zhu, ''Complexity analyses of multi-wing chaotic systems,'' *Chin. Phys. B*, vol. 22, no. 5, May 2013, Art. no. 050506.

[31] Y. Zhang and D. Xiao, ''Cryptanalysis of S-box-only chaotic image ciphers against chosen plaintext attack,'' *Nonlinear Dyn.*, vol. 72, no. 4, pp. 751–756, Jun. 2013.

[32] C. Zhu, G. Wang, and K. Sun, ''Cryptanalysis and improvement on an image encryption algorithm design using a novel chaos based S-box,'' *Symmetry*, vol. 10, no. 9, p. 399, Sep. 2018.

[33] X. Zhang, W. Nie, Y. Ma, and Q. Tian, ''Cryptanalysis and improvement of an image encryption algorithm based on hyper-chaotic system and dynamic S-box,'' *Multimedia Tools Appl.*, vol. 76, no. 14, pp. 15641–15659, Jul. 2017.

[34] Y. Liu, X. Tong, and J. Ma, ''Image encryption algorithm based on hyperchaotic system and dynamic S-box,'' *Multimedia Tools Appl.*, vol. 75, no. 13, pp. 7739–7759, Jul. 2016.

[35] C. Li, Y. Zhang, and E. Y. Xie, ''When an attacker meets a cipher-image in 2018: A year in review,'' *J. Inf. Secur. Appl.*, vol. 48, Oct. 2019, Art. no. 102361.

[36] C. Zhu, G. Wang, and K. Sun, ''Improved cryptanalysis and enhancements of an image encryption scheme using combined 1D chaotic maps,'' *Entropy*, vol. 20, no. 11, p. 843, Nov. 2018.

[37] C. Adams and S. Tavares, ''The structured design of cryptographically good S-boxes,'' *J. Cryptol.*, vol. 3, no. 1, pp. 27–41, 1990.

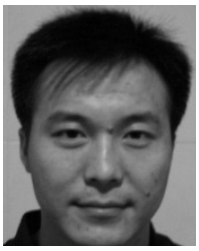
[38] Q. Lu, C. Zhu, and G. Wang, ''A novel S-box design algorithm based on a new compound chaotic system,'' *Entropy*, vol. 21, no. 10, p. 1004, Oct. 2019.

[39] S. Mahmood, S. Farwa, M. Rafiq, S. M. J. Riaz, T. Shah, and S. S. Jamal, ''To study the effect of the generating polynomial on the quality of nonlinear components in block ciphers,'' *Secur. Commun. Netw.*, vol. 2018, pp. 1–8, 2018.

[40] D. Lambić, "A novel method of S-box design based on chaotic map and composition method," *Chaos, Solitons Fractals*, vol. 58, pp. 16–21, Jan. 2014.
[41] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, Aug. 2006.
[42] C. Zhu and K. Sun, "Cryptanalyzing and improving a novel color image encryption algorithm using rt-enhanced chaotic tent maps," *IEEE Access*, vol. 6, pp. 18759–18770, 2018.
[43] S. Zhu and C. Zhu, "Plaintext-related image encryption algorithm based on block structure and five-dimensional chaotic map," *IEEE Access*, vol. 7, pp. 147106–147118, 2019.

QING LU was born in China. He received the bachelor's degree in measurement and control engineering from The Second Artillery Engineering College, Xi'an, China, in 2008, and the master's degree in computer science and technology from Central South University, Changsha, China, in 2016. He is currently a Senior Engineer with the Hunan Police Academy, Changsha. His research interests include the applications of chaos-based cryptography in multimedia information security and image processing.

CONGXU ZHU was born in China. He 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 multimedia information security, chaos-based cryptography, and image processing. He has published over 80 articles in international journals and conferences.



XIAOHENG DENG (Member, IEEE) was born in China. He received the Ph.D. degree in computer science from Central South University, Changsha, Hunan, China, in 2005. He is currently a Professor with the Central South University. His research interests include wireless networking, network QoS optimization, online social network modeling and analysis, and multimedia information security. He has published over 120 articles in international journals and conferences. He is also a Senior Member of the CCF, and a member of the ACM.