

Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption

Akram Belazi · Majid Khan ·
Ahmed A. Abd El-Latif · Safya Belghith

Received: 7 August 2015 / Accepted: 23 August 2016 / Published online: 30 August 2016
© Springer Science+Business Media Dordrecht 2016

Abstract In this paper, two efficient cryptosystem schemes in the form of permutation–substitution based on chaotic systems are proposed. Firstly, a simple and efficient S-box method is introduced in order to use this S-box designed scheme in secure color image encryption technique. The major advantage of the proposed strategy is the dynamic aspect of keys used by chaotic map to generate strong S-boxes. Secondly, an efficient color encryption scheme based on chaotic maps and S-boxes in the form of permutation–substitution network is developed. Experimental results show the effectiveness of the proposed schemes. The suggested cryptosystems have superior performance and great potential for prominent prevalence in cryptographic applications compared to previous schemes.

Keywords Cryptography · S-box · Chaos · Substitution · Permutation

A. Belazi (✉) · S. Belghith
National Engineering School of Tunis, Tunis, Tunisia
e-mail: belazi.akram@gmail.com

M. Khan
Department of Applied Mathematics & Statistics, Institute
of Space Technology, Islamabad 44000, Pakistan

A. A. A. El-Latif
Computer Science Laboratory, Mathematics Department,
Faculty of Science, Menoufia University, Shebin,
El-Koom 32511, Egypt

1 Introduction

The chaotic system has various ultimate features, such as ergodicity, sensitivity to preliminary condition. It also exhibits random behavior, which can be applied to the field of cryptography [1]. The behavior of chaotic maps is predictable only if the control parameters and the initial conditions are known to the observer [2, 3]. The chaotic behaviors in the dynamical systems can be used to infer the diffusion and confusion in the plain images, thus safely transmitting the confidential data over telecommunication channels. Therefore, it is a standard indication to use chaos to enhance the strategy of novel cryptosystems.

In block cryptosystem schemes, the original data are distributed into blocks of the same size, and the encryption is carried out for the complete block. Two wide-ranging ideas of block encryption which were proposed by Shannon are confusion and diffusion. The diffusion consists in scattering the effect of plaintext bits to ciphertext bits to obscure the statistical configuration of the plaintext. Confusion is the transformation in which information of ciphertext alters according to the alteration of the plaintext information. In most cryptosystems structures, diffusion and confusion are attained by means of round recurrence. Modern block encryptions comprise four conversions: substitution, permutation, mixing, and key adding [4–10, 13–17].

Substitution boxes, or simply S-boxes, are used in substitution permutation cipher structures as the essential nonlinear element that ensures confusion property

of the block ciphers [18–22]. A robust block cipher must be hardy to numerous attacks, such as linear and differential cryptanalysis. In strong substitution permutation systems, S-boxes should satisfy a number of measures. The S-box functioning in encryption procedure could be selected under the control of key, as a substitute of being static. Several random key-dependent and bijective S-boxes are generated for encryption applications, which satisfy the selected standards [23–28, 30–36].

Previous image encryption approaches have many defects in their internal structures and most of them present vulnerabilities and hence attacks become easier and practically feasible [11, 19, 45, 47, 48]. Li et al., analyzed the encryption scheme proposed in [12] and found that the position permutation-only part and the substitution part can be broken by chosen-plaintext attack [11]. Zhang et al. [19] analyzed the security of an image encryption algorithm based on perception model [21] and found that the equivalent secret key can be reconstructed with only one pair of known plaintext/ciphertext. Norouzi et al. proposed a hyperchaotic system-based image cipher with only one round diffusion process [44]. However, in [45], Zhang et al. found that the scheme can be effectively broken with known-plaintext and chosen-plaintext attacks. The combination of chaos and DNA is employed for many image ciphers. SaberiKamarposhti et al. [46] introduced an image cipher based on three-cell chaotic map and DNA. Recently, Zhang et al. [47] have analyzed the scheme and found that it can be deciphered by a chosen-plaintext attack. Lately, Liu et al. [48] have analyzed the encryption scheme that uses a single-round modified permutation–diffusion pattern (ICMPD) in their internal structure. They report that ICMPD suffers from the weakness against chosen-plaintext attack. Therefore, the need of designing better cipher schemes based on S-boxes and chaos is essential for encryption applications and stimulating further development.

In this paper, an efficient S-box method based on chaotic logistic-sine map is proposed. Then, using the proposed S-box method, a new color image encryption scheme based on chaotic permutation–substitution network and S-boxes is presented. First, a chaotic cat map is used to shuffle the original image. Then, a substitution algorithm based on the suggested S-box is introduced to substitute the shuffled image in order to guarantee the nonlinearity in the generated image.

To enhance the security, a diffusion process is done using keystream extracted from a combination of a chaotic logistic-Chebyshev map and the substituted image. Finally, a permutation method is carried out by chaotic sine-Chebyshev map in increasing the performance of the resulted image. Conducted tests show that the proposed S-box method has better performance than other S-boxes. Moreover, the results of cryptographic analyses demonstrate that the proposed chaos-based image encryption algorithm outperforms the current image cipher algorithms in terms of security and performance.

The rest of this paper is organized as follows: Sect. 2 outlines the preliminary work for the proposed approaches. In Sect. 3, we introduce the proposed method for the construction of strong S-box based on chaotic map. In Sect. 4, the criteria for evaluating S-box are briefly presented and the performance of the proposed S-box is evaluated and compared with other chaos-based S-boxes. The proposed encryption scheme and its performance analyses are presented in Sect. 5. Security and performance analysis of S-box-only chaotic image ciphers are investigated in Sect. 6. Finally, Sect. 7 concludes the paper.

2 The preliminary work

2.1 The logistic-sine map

The logistic-sine map is 1-D chaotic map defined as:

$$x_{n+1} = \left(\alpha(x_n - x_n^2) + (4 - \alpha) \sin(\pi x_n) / 4 \right) \bmod 1, \quad (1)$$

where α is the system parameter $\alpha \in [0, 4]$, and x_0 is the initial condition. In [37], many experiments were conducted to prove that the logistic-sine map is chaotic. Figure 1 shows the bifurcation diagram of a logistic-sine map with $\alpha \in [0, 4]$. It is well revealed from the bifurcation diagram that the logistic-sine is purely chaotic. The reason for choosing logistic-sine map is its simplicity compared to some other chaotic systems with assurance of higher level of security.

Note: Iterating the logistic-sine map with (α, x_0) pair means iterating it with control parameter α and initial condition x_0 .

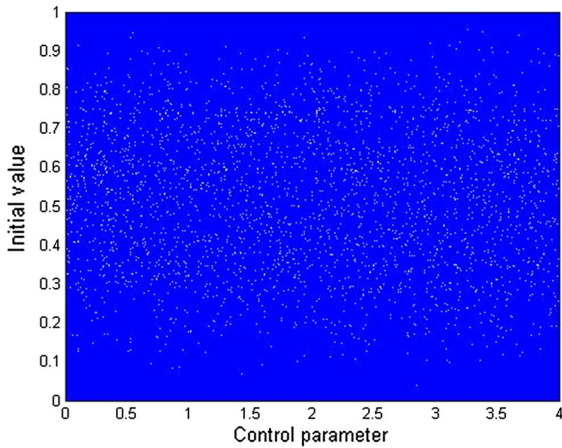


Fig. 1 Bifurcation diagram of the logistic-sine map

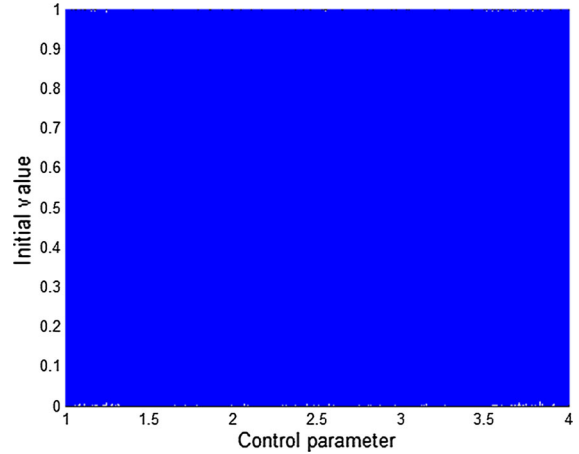


Fig. 3 Bifurcation diagram of the sine-Chebyshev map

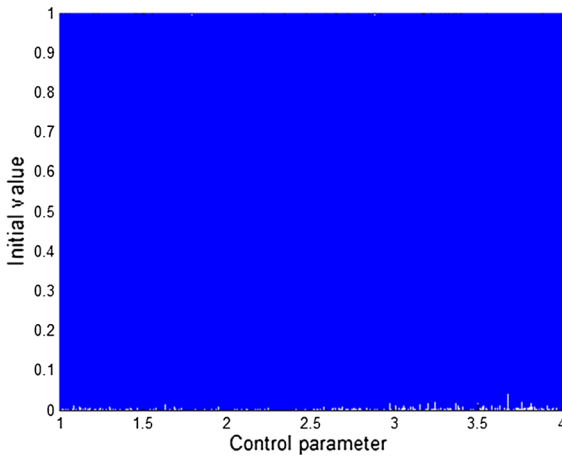


Fig. 2 Bifurcation diagram of the logistic-Chebyshev map

2.2 The logistic-Chebyshev

The logistic-Chebyshev map (denoted as LCbv) is a chaotic system that uses the logistic and Chebyshev as seed maps. The definition of LCbv can be described by Eq. (2). Figure 2 shows the Hopf bifurcation diagram of the LCbv.

$$y_{n+1} = (\lambda y_n(1 - y_n) + (4 - \lambda) \cos(b \cdot \arccos(y_n))/4) \text{ mod } 1, \quad (2)$$

where λ is the system parameter $\lambda \in [0, 4]$, y_0 is an initial seed, and $b \in \mathbb{N}$ denotes the degree of the Chebyshev map.

2.3 The sine-Chebyshev

The sine-Chebyshev map (denoted as SCbv) is a chaotic system that uses the sine and Chebyshev as seed maps. The definition of SCbv can be described by Eq. (3). Figure 3 shows the Hopf bifurcation diagram of the SCbv.

$$y_{n+1} = (\lambda \sin(\pi y_n) + (4 - \lambda) \cos(b \cdot \arccos(y_n))/4) \text{ mod } 1, \quad (3)$$

where λ is the system parameter $\lambda \in [0, 4]$, y_0 is an initial seed, and $b \in \mathbb{N}$ denotes the degree of the Chebyshev map.

2.4 2D toral automorphism

The 2D toral automorphism is a function defined as [29]:

$$\begin{pmatrix} u_{i+1} \\ v_{i+1} \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix} \text{ mod } (1), \quad i = 0, 1, 2, \dots \quad (4)$$

If the continuous coordinates u_i, v_i in the unit square is replaced by the indices x_i, y_i in the discrete lattice of width M , the generalized discretized toral automorphism can be defined as

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \text{ mod } (M) \quad (5)$$

If $b_{11} = 1, b_{22} = 1 + b_{12}b_{21}$ then the generalized discretized toral automorphism is reduced to the generalized discretized cat map as follows:

Table 1 The example of four-bit gray code and bit-reversed order transformations

Rank	Binary representation	Gray number	Bit-reversed order	Decimal representation
0	0000	0000	0000	0
1	0001	0001	1000	8
2	0010	0011	1100	12
3	0011	0010	0100	4
4	0100	0110	0110	6
5	0101	0111	1110	14
6	0110	0101	1010	10
7	0111	0100	0010	2

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & b_{12} \\ b_{21} & b_{12}b_{21} + 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \text{ mod } (M) \quad (6)$$

The above chaotic map has a period Γ for the parameters b_{12} , b_{21} and M , which mean:

$$\begin{pmatrix} 1 & b_{12} \\ b_{21} & b_{12}b_{21} + 1 \end{pmatrix}^{\Gamma} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ mod } (M) \quad (7)$$

To avoid the weak keys of the above map Eq. (6), the following equation is desirable [30]

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \text{ mod } (M) \quad (8)$$

Where M is the modulo of the map and it is preferable to be square like 64, 128, 256 or 512 in order to have a good performance.

2.5 Gray code

The gray code is an alternative binary representation so that each two successive values differ only in a single bit. This property has been found gainful in security field. The gray code representation of a number c is given by the transformation G defined by Eq. (9):

$$G(c) = c \oplus (c \gg 1), \quad (9)$$

where \oplus is the binary XOR operation, and \gg represents the binary right shift. Let Γ be the bit reversal order function that reverses the order of bits, so the least significant bit becomes the most significant one as shown in Eq. (10):

$$\begin{cases} \Gamma : \Sigma^n \rightarrow \Sigma^n \\ c_0c_1 \dots c_{n-1} \rightarrow \Gamma(c_0c_1 \dots c_{n-1}) = c_{n-1}c_{n-2} \dots c_0, \end{cases} \quad (10)$$

The gray codes corresponding to the first eight nonnegative integers and their bit-reversed order transformations are given in Table 1.

3 The proposed substitution box

In this section, we propose a novel method for the construction of S-boxes based on chaotic map. The new method makes full use of the traits of the chaotic logistic-sine map to construct strong S-boxes. Experimental results and performance analyses demonstrate that the proposed method is efficient and has good cipher properties compared to previous methods.

3.1 Review of the recently proposed S-boxes related to the proposed scheme

A short overview of the main chaos-based S-boxes proposed recently is given hereafter. In [1], Jakimoski and Kocarev presented a four-step method to generate S-boxes based on chaotic maps. In another construction method based on a chaotic map [23], Tang et al. proposed an approach for generating S-boxes based on a 2D discretized chaotic map. In [24], Chen et al. improved the scheme proposed in [23] using a three-dimensional Backer map. In [25], Ozkaynak et al. proposed a method for designing an S-box based on chaotic Lorenz system. Another method for designing substitution boxes based on chaotic Lorenz system was introduced in [26]. Hussain et al. presented

a projective general linear group-based algorithm for the generation of S-boxes for block ciphers [27]. In [28], Khan et al. proposed another method for the construction of S-boxes to be used in block cipher with multi-chaotic systems. Furthermore, several methods of S-boxes have been proposed very recently as a development of image cryptography field based on chaotic systems [31–36].

3.2 The proposed method for generating $n \times n$ S-box

The proposed approach of S-box construction is described by the following steps:

Step 1 Iterate the logistic-sine map 256 times with (λ_0, f_0) pair to produce the chaotic sequence $A(1 \times 256)$.

Step 2 Calculate the sequence $P(1 \times 256)$ according to Eq. (11).

$$P(i) = A(1) - A(i) \quad 1 \leq i \leq 256, \tag{11}$$

Step 3 Put in ascending order the elements of $P(1 \times 256)$ to obtain the sequence $M(1 \times 256)$. Then, determine the sequence $Q(1 \times 256)$, which contains the order of each element of $M(1 \times 256)$ in $P(1 \times 256)$.

Step 4 Calculate the code sequence $B(1 \times 256)$ and the sequence $N(1 \times 256)$ according to Eqs. (12), (13), respectively.

$$B(i) = Q(i) - 1 \quad 1 \leq i \leq 256, \tag{12}$$

$$N(i) = A(256) - A(i) \quad 1 \leq i \leq 256, \tag{13}$$

Step 5 Put in ascending order the elements of $N(1 \times 256)$ and produce the sequence $K(1 \times 256)$. Then, determine the order of each element of $K(1 \times 256)$ in $N(1 \times 256)$ in order to obtain the sequence $R(1 \times 256)$. After that, calculate the code sequence $S(1 \times 256)$ according to Eq. (14).

$$S(i) = R(i) - 1 \quad 1 \leq i \leq 256 \tag{14}$$

Step 6 Permute the elements of $B(1 \times 256)$ according to the elements of sequence $S(1 \times 256)$ and generate the sequence $Z(1 \times 256)$. In fact, the first element of B is assigned to element number $S(1)$ in sequence Z , the second element of B is assigned to element number S

(2) in sequence Z ...the i th element of B is assigned to element number $S(i)$ in sequence Z .

Step 7 Apply the gray code defined in Eq. (9), and the bit reversal order function given by Eq. (10) to each element of sequence Z to obtain a new sequence $W(1 \times 256)$.

Step 8 Translate $W(1 \times 256)$ into $n \times n$ S-box $SB(16 \times 16)$.

In our experiments, $\lambda_0 = 3.6034099541280193$; $f_0 = 0.22784293570604186$.

The block diagram of the proposed S-box is shown in Fig. 4.

The generated S-box is presented in Table 2.

4 Performance analysis of the proposed S-box

Efficient substitution boxes should satisfy some specific cryptographic criteria, such as bijective, nonlinearity, outputs bit independence, strict avalanche and linear approximation probability [23–27]. Here, we give a detailed analysis of the proposed S-box in terms of the aforementioned cryptographic properties. Some chaos-based S-boxes presented in [1, 23–26, 28], were chosen to be compared with our S-box.

4.1 Bijective property

An $n \times n$ S-box is bijective if all its different output values are within the interval $[0, 2^n - 1]$ [24–28]. Our generated S-box has different output values within the interval $[0, 255]$. Therefore, it accept the bijective criterion.

4.2 Nonlinearity criterion

The nonlinearity of a Boolean function $h(x)$ can be represented by the Walsh spectrum (WS)

$$N_h = 2^{n-1} \left(1 - 2^{-n} \max_{\chi \in GF(2^n)} |S_{<h>}(\chi)| \right) \tag{15}$$

The WS of $h(x)$ is defined as:

$$S_{<h>}(\chi) = \sum_{x \in GF(2^n)} (-1)^{h(x) \oplus x \cdot \chi} \tag{16}$$

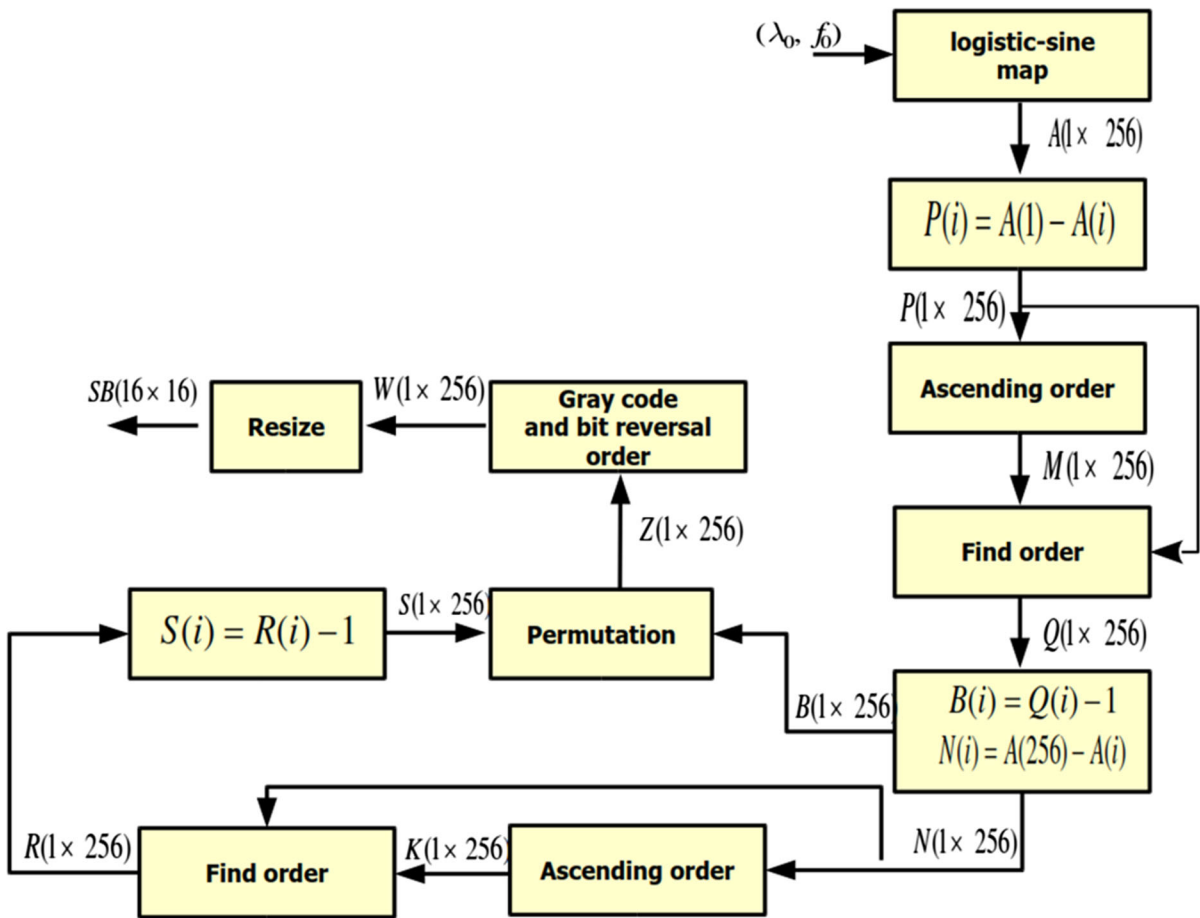


Fig. 4 Block diagram of the proposed S-box

where $\chi \in GF(2^n)$ and $x \cdot \chi$ means the dot product of x and χ , which is given by:

$$x \cdot \chi = x_1 \oplus \chi_1 + \dots + x_n \oplus \chi_n.$$

The nonlinearities of the generated S-box and S-boxes studied in [1,23–26,28] are shown in Table 3. It is observed that the mean value obtained from the suggested method is better than those of the other S-boxes.

4.3 Strict avalanche criterion

The strict avalanche criterion (SAC), firstly published by Webster and Tavares [38], indicate that when a single input bit is complemented, all of the output bits change with a probability of a half.

The S-box satisfies the SAC, if each value of the dependence matrix [38] is approximately equal to 0.5.

Table 4 gives the values of the dependence matrix for the generated S-box. From Table 4, we can see the average value is $0.4956 \approx 0.5$. The comparison of SAC of different S-boxes is displayed in Table 5. The results given in Tables 4 and 5 demonstrate the efficiency of the proposed S-box.

4.4 Statistical curve analysis of the proposed S-box

In this subsection, we mainly discuss the curve fitting of our proposed S-box. We drew the Chi-square and binomial distribution (Fig. 5a and b, respectively) of our suggested nonlinear component for the block cipher. Both these curves show symmetrical shapes, with respect to average values which clearly reflect the non-repeating and uniqueness of each values in the S-box design. The analyses of this S-box have not been devised in the literature so far.

Table 2 The S-box generated by the proposed algorithm

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

Table 3 Nonlinearity of the proposed S-box in comparison with other S-boxes

S-box	Nonlinearity		
	Min	Max	Avg.
Proposed S-box	102	108	105.25
Ref. [1]	98	108	103.2
Ref. [23]	99	106	103.3
Ref. [24]	100	106	103
Ref. [25]	100	106	103.2
Ref. [26]	102	108	104.7
Ref. [28]	96	106	103

Table 5 SAC of the proposed S-box in comparison with other S-boxes

S-box	SAC		
	Min	Max	Avg.
Proposed S-box	0.4297	0.5313	0.4956
Ref. [1]	0.3671	0.5975	0.5058
Ref. [23]	0.4140	0.6015	0.4987
Ref. [24]	0.4218	0.6093	0.5000
Ref. [25]	0.4218	0.5937	0.5048
Ref. [26]	0.3906	0.5937	0.5056
Ref. [28]	0.3906	0.6250	0.5039

4.5 Output bits independence criterion (BIC)

BIC means that all the avalanche variables must be pairwise independent for a given set of avalanche vectors by

complementing a single bit [38]. The correlation coefficient between the couples is applied to measure the degree of independence between the avalanche variable couples.

Table 4 The SAC values of the generated S-box

0.5000	0.4922	0.5000	0.5000	0.5313	0.5156	0.4844	0.5156
0.5000	0.5078	0.5000	0.4922	0.5000	0.5000	0.5000	0.5000
0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.4844	0.4844
0.4297	0.5000	0.5000	0.4922	0.5313	0.4844	0.4844	0.4844
0.4297	0.5000	0.4922	0.5000	0.5000	0.5156	0.4844	0.5156
0.5000	0.5000	0.5000	0.4922	0.5000	0.4844	0.4844	0.4844
0.4297	0.5000	0.5078	0.5000	0.4688	0.5000	0.5000	0.5000
0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5156

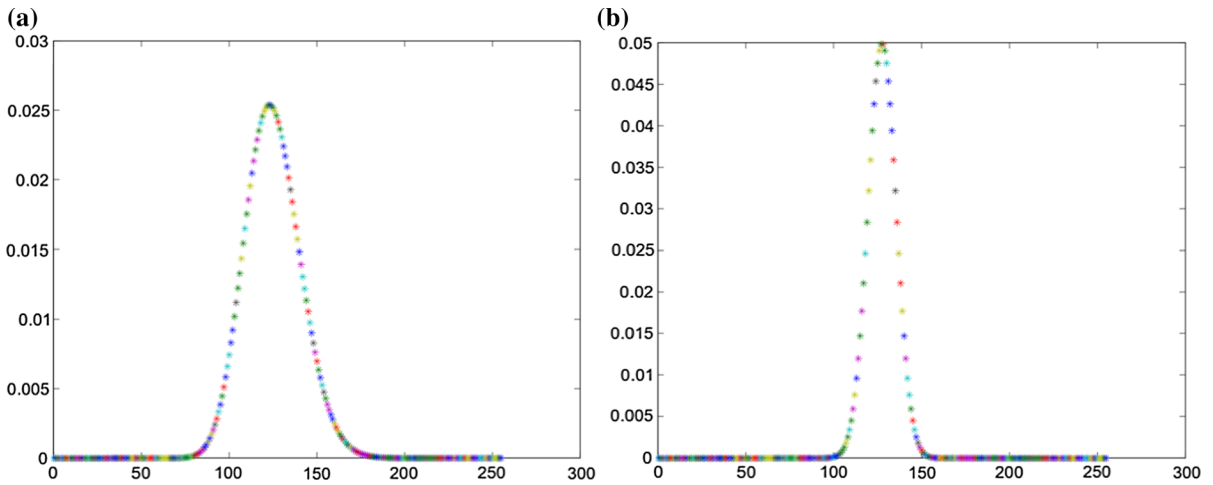


Fig. 5 a, b Chi-square and Binomial distributions for the proposed S-box

Table 6 BIC-nonlinearity results for the proposed S-box

–	106	104	102	100	102	102	104
106	–	102	108	104	104	106	102
104	102	–	104	104	104	104	106
102	108	104	–	104	102	104	108
100	104	104	104	–	100	106	102
102	104	104	102	100	–	106	106
102	106	104	104	106	106	–	100
104	102	106	108	102	106	100	–

Suppose the Boolean functions in the proposed S-box were h_1, h_2, \dots, h_n , it is denoted in [38] that if the S-box achieves BIC, $h_j \oplus h_k (j \neq k, 1 \leq j, k \leq n)$ should be highly nonlinear and fulfill the avalanche criterion. Thus, we can calculate the SAC and the non-linearity of $h_j \oplus h_k (j \neq k)$ to check the BIC of an S-box.

Table 7 BIC-SAC results for the proposed S-box

–	0.4911	0.4911	0.5134	0.5123	0.5033	0.5011	0.4944
0.5112	–	0.5089	0.5134	0.5045	0.5045	0.5000	0.5067
0.5067	0.4888	–	0.5045	0.5112	0.4978	0.5067	0.4933
0.4933	0.4799	0.4799	–	0.4955	0.4955	0.4933	0.4866
0.4989	0.4944	0.4967	0.5033	–	0.4955	0.4844	0.5112
0.4967	0.5145	0.4855	0.5078	0.5011	–	0.4944	0.4900
0.4810	0.5011	0.5145	0.4922	0.5045	0.5022	–	0.5156
0.4989	0.5056	0.4922	0.5011	0.5067	0.5000	0.4978	–

Tables 6 and 7 give the results of BIC-nonlinearity and BIC-SAC criteria for the proposed S-box, respectively. As can be seen from Table 6, the maximum value is 108, the average is 103.8 and the minimum is 100. Table 7 gives the results of BIC-SAC criterion for the proposed scheme. The mean value of this criterion is 0.4996. In addition, Table 8 gives the results of comparing BIC-SAC and BIC-nonlinearity. From this Table, we observe that the values of BIC-SAC of the presented S-box are more advantageous than those of the other S-boxes, while the proposed S-box and the S-boxes studied in [1, 23–26] have better BIC-nonlinearity property compared with that proposed in [28].

4.6 The equiprobable input/output XOR distribution

Biham and Shamir presented the idea behind the equiprobable input/output XOR distribution using imbalances in the input/output XOR distribution table

Table 8 BIC of the proposed S-box compared to previous S-boxes

S-box	Mean value of BIC-nonlinearity	Mean value of BIC-SAC
Proposed S-box	103.8	0.4996
Ref. [1]	104.2	0.5031
Ref. [23]	103.3	0.4995
Ref. [24]	103.1	0.5024
Ref. [25]	103.7	0.5009
Ref. [26]	104.1	0.5021
Ref. [28]	100.3	0.5010

[39]. Mathematically, the equiprobable input/output XOR distribution or differential approximation probability (DP) of a given S-box is estimated by calculating the differential uniformity as follows:

$$DP_h(\Delta d \rightarrow \Delta x) = \left(\frac{\#\{d \in D | h(d) \oplus h(d \oplus \Delta d) = \Delta x\}}{2^n} \right) \quad (17)$$

where D represents the set of all possible input values, and 2^n is the number of its elements.

Tables 9 and 10 give the differential approach table and its DP for the proposed S-box. As can be seen from these tables, the DP of the proposed S-box is 0.039062, which proves its efficiency against differential attacks.

Table 9 Results of DP for the generated S-box

6	8	8	6	6	6	6	6	6	6	6	6	6	6	6	6
6	6	6	6	6	6	6	6	8	6	6	6	8	8	6	6
8	6	6	8	8	8	6	8	6	8	6	6	8	8	6	6
8	6	6	6	6	8	8	8	6	8	6	6	6	6	6	6
6	8	6	6	8	6	6	6	6	6	8	6	8	6	8	6
8	4	6	6	8	6	6	6	8	6	6	6	6	6	8	8
10	8	6	6	8	6	6	6	6	6	8	6	6	8	6	8
6	6	6	6	8	6	6	6	8	6	8	6	6	6	8	6
6	6	8	6	8	6	8	6	4	6	6	8	6	10	8	6
8	6	8	10	8	8	6	6	6	8	8	6	6	6	6	6
8	8	10	8	6	6	6	6	6	8	6	6	8	6	8	6
6	6	6	6	6	6	8	6	4	8	8	6	8	8	6	6
8	6	6	8	6	6	6	6	6	4	6	6	8	6	6	6
6	8	6	6	8	10	10	6	10	4	8	6	6	6	6	6
8	10	6	6	4	6	6	8	10	6	6	6	6	8	8	8
8	6	6	6	6	8	6	6	10	6	6	8	6	8	8	—

Table 10 DP for different S-boxes

S-box	Maximum DP
Proposed S-box	0.039062
Ref. [1]	0.046875
Ref. [23]	0.039062
Ref. [24]	0.054687
Ref. [25]	0.039062
Ref. [26]	0.046875
Ref. [28]	0.046875

Table 10 shows the DPs of previous S-boxes. To conclude, the introduced S-box has a higher performance than competitive S-boxes.

4.7 Linear approximation probability

The linear approximation probability (LP) is the maximum value of the imbalance of an event. The parity of the input bits selected by mask a is equal to the parity of the output bits selected by mask b . According to Matsui’s original definition [22], linear approximation probability is defined by:

$$LP = \max_{u,v \neq 0} \left| \frac{\#\{d \in D | d \cdot u = h(d) \cdot v\}}{2^n} - \frac{1}{2} \right| \quad (18)$$

Table 11 LP for the proposed S-box in comparison with other S-boxes

S-box	LP
Proposed S-box	0.156200
Ref. [1]	0.128906
Ref. [23]	0.132813
Ref. [24]	0.128906
Ref. [25]	0.128906
Ref. [26]	0.125000
Ref. [28]	0.125000

where u and v are the input and output masks. D is defined as the set of all possible inputs and 2^n is the number of its elements. The linear approximation probability of the presented S-box and previous S-boxes [1, 23–26, 28] are recorded in Table 11. From this table, we can conclude that the S-box generated by the proposed method has better LP performance than the other S-boxes.

5 The proposed image encryption algorithm

5.1 Encryption scheme

In this section, we present the proposed encryption scheme. This cryptosystem is intended to color images and it contains three rounds; we firstly split the RGB image into R, G and B components and we set $cmpt = 1$. Secondly, we permute the R, G and B components by the cat map function and we get three shuffled matrixes. In order to guarantee the nonlinearity of the proposed scheme, we substituted each shuffled matrix by the proposed S-box. Then, we got three substituted matrixes. Thirdly, to scramble the pixels of the different components, we carried out bitwise XOR operation between each substituted matrix and three random matrixes generated by logistic-Chebyshev map, then we obtained three scrambled matrixes. Fourthly, we permute each scrambled component using three chaotic matrixes obtained by iterating sine-Chebyshev map. So, we got the results of the first round. Moreover, we updated the parameters of the logistic-Chebyshev and the sine-Chebyshev maps, and we set $cmpt = cmpt + 1$. After that, we repeated all these operations in a loop until $cmpt \leq 3$. Finally, if $cmpt = 4$, we combined the R, G, B components and we get the encrypted image.

The block diagram of the cryptosystem is presented in Fig. 6.

The whole encryption process consists of the following operation steps.

Step 1 Input 24-bit color image $P(M, M, 3)$, where $M \times M$ are the image dimensionalities of rows and columns, respectively and set $cmpt = 1$.

Step 2 Split the RGB image into R, G and B components.

Step 3 Block division and permutation

Decompose the matrixes of R, G and B to blocks of size $(m \times m)$. The number of blocks is $\frac{M \times M}{m \times m}$. The results of this decomposition were the matrixes R_d, G_d and B_d each created by these blocks with size $(\frac{M}{m}, \frac{M}{m})$. Next, permute each matrix R_d, G_d and B_d by the use of cat map function (Eq. 6) and get three permuted matrixes R_p, G_p and B_p . The permutation procedure is given by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{pmatrix} 1 & b_{12} \\ b_{21} & b_{12}b_{21} + 1 \end{pmatrix}^k \begin{bmatrix} x \\ y \end{bmatrix} \bmod \frac{M}{m} \quad (19)$$

where, $x, y = 1, \dots, \frac{M}{m}$. The $(x, y), (x', y')$ pairs represent the coordinates of the block in each decomposed component and in each permuted component, respectively.

Step 4 Substitution phase

To ensure that the proposed encryption scheme is secure against known-/chosen-plaintext attacks, a substitution operation is necessary. Hence, we generated three S-boxes, denoted by SB_r^i, SB_g^i and SB_b^i , using the method described in Sect. 3, with $(\lambda_r^i, f_r^i), (\lambda_g^i, f_g^i)$ and (λ_b^i, f_b^i) pairs, respectively. Then, we substituted the matrixes R_p, G_p and B_p by SB_r^i, SB_g^i and SB_b^i , respectively, to obtain three substituted matrixes R_s, G_s and B_s .

The parameters $\left\{ \lambda_r^i, f_r^i, \lambda_g^i, f_g^i, \lambda_b^i, f_b^i \right\}_{i=1, \dots, N_r} = (\lambda_j^i, f_j^i)_{j=r, g, b, i=1, \dots, N_r}$ are given as follows:

$$\begin{aligned} \lambda_r^i &= 0.5 \times \lambda_r^{i-1} \\ &+ \bmod \left(\sum_{i=1}^M \sum_{j=1}^M R_p(i, j), 256 \right) / 256 \times 0.5 \\ f_r^i &= 0.5 \times f_r^{i-1} \end{aligned}$$

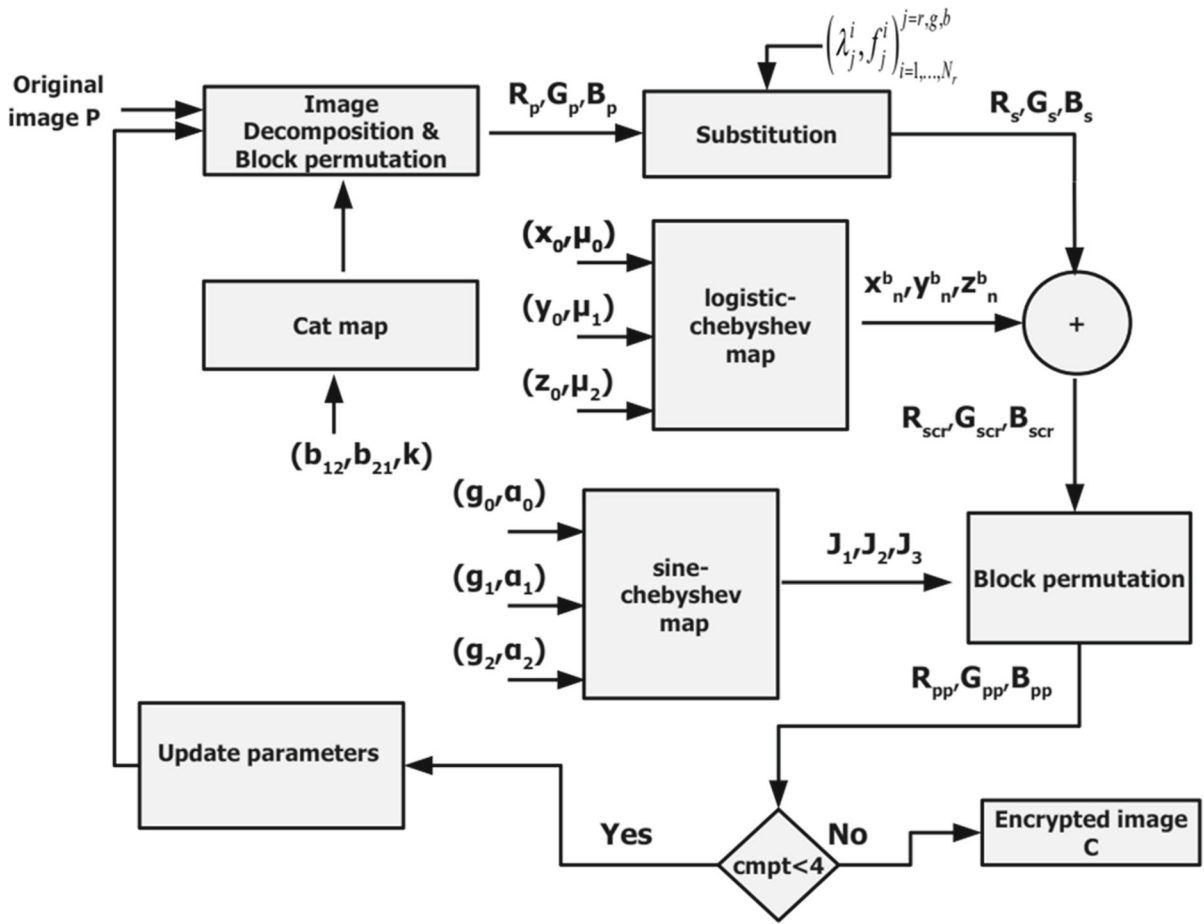


Fig. 6 Block diagram of the proposed encryption approach

$$\begin{aligned}
 & + \text{mod} \left(\sum_{i=1}^M \sum_{j=1}^M R_p(i, j), 256 \right) / 256 \times 0.5 \\
 \lambda_g^i &= 0.5 \times \lambda_g^{i-1} \\
 & + \text{mod} \left(\sum_{i=1}^M \sum_{j=1}^M G_p(i, j), 256 \right) / 256 \times 0.5 \\
 f_g^i &= 0.5 \times f_g^{i-1} \\
 & + \text{mod} \left(\sum_{i=1}^M \sum_{j=1}^M G_p(i, j), 256 \right) / 256 \times 0.5 \\
 \lambda_b^i &= 0.5 \times \lambda_b^{i-1} \\
 & + \text{mod} \left(\sum_{i=1}^M \sum_{j=1}^M B_p(i, j), 256 \right) / 256 \times 0.5
 \end{aligned}$$

$$\begin{aligned}
 f_b^i &= 0.5 \times f_b^{i-1} \\
 & + \text{mod} \left(\sum_{i=1}^M \sum_{j=1}^M B_p(i, j), 256 \right) / 256 \times 0.5
 \end{aligned}$$

where, $i = 1, \dots, N_r$, N_r is the number of the encryption scheme iteration.

Here, $\lambda_r^0 = \lambda_g^0 = \lambda_b^0 = \lambda_0$ and $f_r^0 = f_g^0 = f_b^0 = f_0$.

Step 5 Scrambling phase

Generate three chaotic matrixes x_n , y_n and z_n with size $(M \times M)$ using the logistic-Chebyshev map (Eq. 2) in the condition of initial values are x_0 , y_0 and z_0 , and the system parameters are μ_0 , μ_1 and μ_2 . Then, map each chaotic matrix from $[0, 1]$ to $\{0, 1, 2, \dots, 255\}$ using the following equations:

$$x_n^b = \text{mod} \left(x_n 10^{16}, 256 \right) \tag{20}$$

$$y_n^b = \text{mod} \left(y_n 10^{16}, 256 \right) \tag{21}$$

$$z_n^b = \text{mod} \left(z_n 10^{16}, 256 \right) \tag{22}$$

Then, calculate the scrambled components according to the following formulas:

$$R_{scr} = \text{bitxor} (R_s, x_n^b) \tag{23}$$

$$G_{scr} = \text{bitxor} (G_s, y_n^b) \tag{24}$$

$$B_{scr} = \text{bitxor} (B_s, z_n^b) \tag{25}$$

Step 6 Block division and permutation

Decompose the matrixes R_{scr} , G_{scr} and B_{scr} to blocks of size $(n \times n)$. The number of blocks is then $r = \frac{M \times M}{n \times n}$ and get three matrixes R_{dd} , G_{dd} and B_{dd} each created by these blocks with size $(\frac{M}{n}, \frac{M}{n})$. Then, permute each matrix R_{dd} , G_{dd} and B_{dd} by sequences J_1 , J_2 and J_3 , respectively to get three permuted matrixes R_{pp} , G_{pp} and B_{pp} . The sequences J_1 , J_2 and J_3 are obtained as follows:

Iterate the sine-Chebyshev map (Eq. 3) with the initial conditions g_0, g_1, g_2 and the control parameters α_0, α_1 and α_2 . Therefore, get three chaotic sequences I_1, I_2 and I_3 which will be mapped to $\{1, 2, \dots, r\}$ range to obtain three new sequences J_1, J_2 and J_3 .

Step 7 Set $cmpt = cmpt + 1$. If $cmpt \leq 3$ update the values of x_0, y_0, z_0, g_0, g_1 and g_2 according to Eqs. (26)–(31):

$$x_0 = 0.1 \times x_0 + \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M R_{pp}(i, j)^2, 256)}{256} \times 0.9 \tag{26}$$

$$y_0 = 0.1 \times y_0 + \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M G_{pp}(i, j)^2, 256)}{256} \times 0.9 \tag{27}$$

$$z_0 = 0.1 \times z_0 + \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M B_{pp}(i, j)^2, 256)}{256} \times 0.9 \tag{28}$$

$$g_0 = 0.1 \times g_0 + \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M R_{pp}(i, j)^2, 256)}{256} \times 0.9 \tag{29}$$

$$g_1 = 0.1 \times g_1$$

$$+ \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M G_{pp}(i, j)^2, 256)}{256} \times 0.9 \tag{30}$$

$$g_2 = 0.1 \times g_2 + \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M B_{pp}(i, j)^2, 256)}{256} \times 0.9 \tag{31}$$

Then, repeat steps (3)–(6).

Else if $cmpt > 3$ recover the RGB image C and this is the encrypted color image.

The original images and their encrypted images with the proposed cryptosystem are shown in Fig. 7.

5.2 Decryption process

The decryption process, illustrated in Fig. 8, consists of the following steps:

Note: Set $cmpt = 1$ and use the last values of x_0, y_0, z_0, g_0, g_1 and g_2 obtained from the encryption process.

Step 1 Split the encrypted color image C into red, green, and blue channels denoted by R_c, G_c and B_c , respectively.

Step 2 Decompose the matrixes R_c, G_c and B_c to blocks of size $(n \times n)$. Then, obtain three new matrixes R'_d, G'_d and B'_d each created by these blocks with size $(\frac{M}{n}, \frac{M}{n})$. Therefore, permute each matrix R'_d, G'_d and B'_d by sequences J_1^{-1}, J_2^{-1} and J_3^{-1} , respectively. Then, get three permuted matrixes R'_p, G'_p and B'_p . The sequence J_1^{-1}, J_2^{-1} and J_3^{-1} are the inverse sequences of J_1, J_2 and J_3 obtained in Step 6 of the encryption process.

Step 3 Make bitwise XOR operation between the matrixes x_n^b, y_n^b, z_n^b and the permuted matrixes R'_p, G'_p, B'_p , respectively.

$$R'_{scr} = \text{bitxor} (R'_p, x_n^b), \tag{32}$$

$$G'_{scr} = \text{bitxor} (G'_p, y_n^b), \tag{33}$$

$$B'_{scr} = \text{bitxor} (B'_p, z_n^b). \tag{34}$$

Here, the matrixes x_n^b, y_n^b, z_n^b are the three chaotic matrixes generated as in Step 5 of encryption process.

Step 4 Substitute R'_{scr}, G'_{scr} and B'_{scr} with the inverse S-boxes of SB_r^i, SB_g^i and SB_b^i (Here, $i = N_r, \dots, 1$), respectively. The outputs of this step are the matrixes R'_s, G'_s and B'_s .

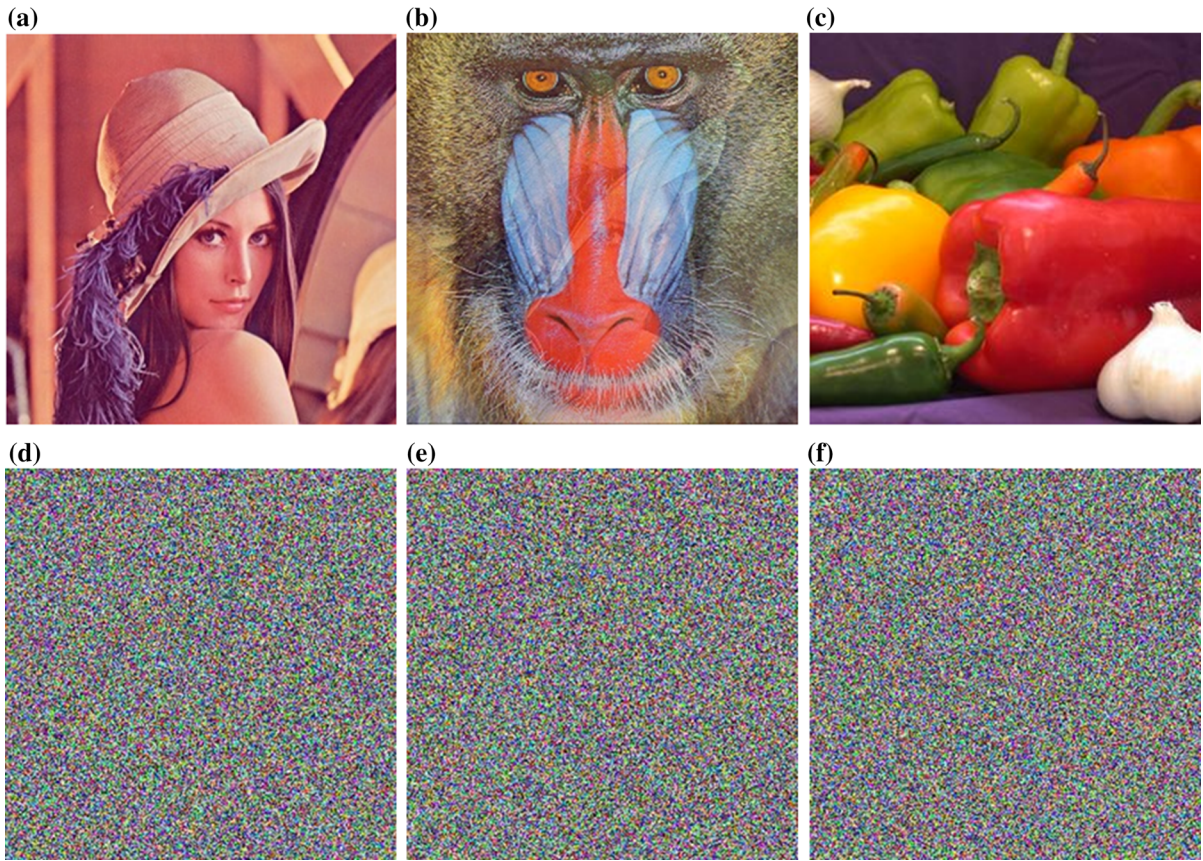


Fig. 7 a–c Plain images and their encrypted images d–f, respectively

Step 5 Decompose the matrixes R'_s, G'_s and B'_s to blocks of size $(m \times m)$. Then, obtain three new matrixes R'_{dd}, G'_{dd} and B'_{dd} each formed by these blocks with size $(\frac{M}{m}, \frac{M}{m})$. Therefore, permute the matrixes R'_{dd}, G'_{dd} and B'_{dd} by the inverse cat map. Then, get three permuted matrixes R'_{pp}, G'_{pp} and B'_{pp} . The permutation was carried out as follows:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{pmatrix} 1 & b_{12} \\ b_{21} & b_{12}b_{21} + 1 \end{pmatrix}^{T-k} \begin{bmatrix} i \\ j \end{bmatrix} \pmod{\frac{M}{m}} \quad (35)$$

where (i, j) pair is the block coordinates of the matrixes R'_{dd} and G'_{dd}, B'_{dd} and (i', j') is the block coordinates of the permuted matrixes R'_{pp}, G'_{pp} and B'_{pp} .

Step 6 Set $cmpt = cmpt + 1$ and update the values of x_0, y_0, z_0, g_0, g_1 and g_2 as in Eqs. (36)–(41):

$$x_0 = \frac{x_0}{0.1} - \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M R'_{pp}(i, j)^2, 256)}{256} \times 0.9 \quad (36)$$

$$y_0 = \frac{y_0}{0.1} - \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M G'_{pp}(i, j)^2, 256)}{256} \times 0.9 \quad (37)$$

$$z_0 = \frac{z_0}{0.1} - \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M B'_{pp}(i, j)^2, 256)}{256} \times 0.9 \quad (38)$$

$$g_0 = \frac{g_0}{0.1} - \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M R'_{pp}(i, j)^2, 256)}{256} \times 0.9 \quad (39)$$

$$g_1 = \frac{g_1}{0.1} - \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M G'_{pp}(i, j)^2, 256)}{256} \times 0.9 \quad (40)$$

$$g_2 = \frac{g_2}{0.1} - \frac{\text{mod}(\sum_{i=1}^M \sum_{j=1}^M B'_{pp}(i, j)^2, 256)}{256} \times 0.9 \quad (41)$$

Step 7 Repeat Steps (2-6) twice and get the decrypted channels $R_{decrypt}, G_{decrypt}$ and $B_{decrypt}$. Then, recover the RGB decrypted image D .

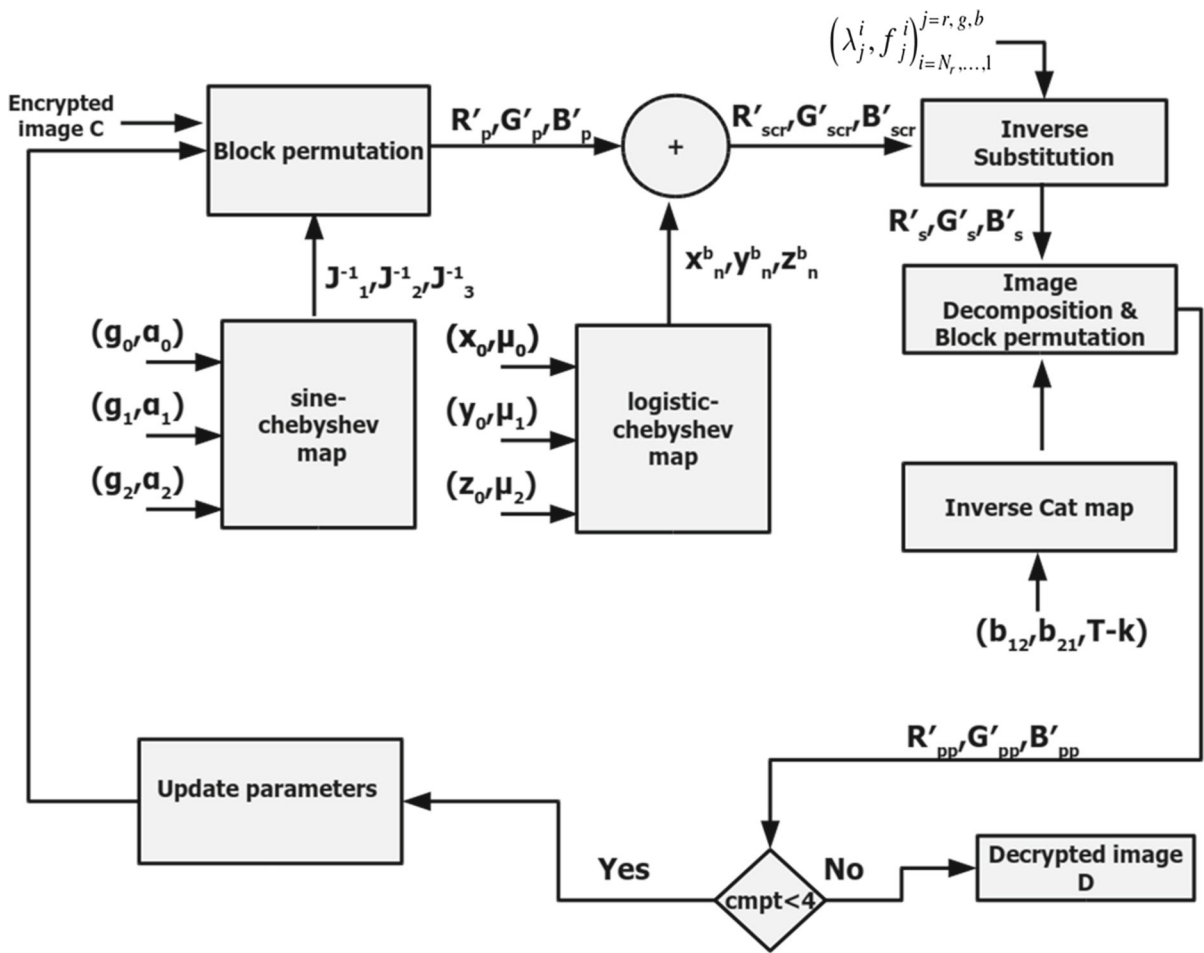


Fig. 8 Block diagram of the proposed decryption process

5.3 Security and performance analysis

Simulation results of the proposed scheme were performed using an Intel Core i3-3227U 1.9 CPU with 4 GB RAM running on Windows 7 and Matlab 7.9. The 256×256 true color JPEG images of ‘Lena’, ‘Baboon’ and ‘Peppers’ (Fig. 4a–c, respectively) are used as plain images. The initial conditions $(x_0, y_0, z_0, g_0, g_1, g_2)$ were fixed at $(0.99997, 0.99998, 0.99978, 0.12346, 0.11252, 0.17982)$ and the control parameters $(\mu_0, \mu_1, \mu_2, \alpha_0, \alpha_1, \alpha_2)$ were chosen as $(3.99978, 3.99978, 3.99978, 3.99799, 3.99999, 3.99999)$. The integers m and n were chosen as 4 and 8, respectively. The parameters of cat map were $b_{12} = 1, b_{21} = 1$ and $k = 90$.

5.3.1 Key space analysis

The key space is the total number of different keys that can be used in the encryption procedure. An effective encryption scheme should present a large key space to make the brute-force attacks impossible. In the proposed cryptosystem, the key space included:

1. The initial conditions $f_0, x_0, y_0, z_0, g_0, g_1$ and g_2 .
2. The control parameters $\lambda_0, \mu_0, \mu_1, \mu_2, \alpha_0, \alpha_1$ and α_2 .
3. The parameters of cat map b_{12}, b_{21} and k .
4. The integers m and n .

where $f_0, x_0, y_0, z_0, g_0, g_1, g_2 \in [0, 1]$ $\lambda_0, \mu_0, \mu_1, \mu_2, \alpha_0, \alpha_1, \alpha_2 \in [0, 4]$. The integers m and n were selected

Table 12 Avalanche effect for different algorithms

Image	Avalanche effect (%)								
	New scheme			Ref. [40]			Ref. [41]		
	R	G	B	R	G	B	R	G	B
Lena	49.91	50.00	49.98	49.57	49.69	49.82	49.45	49.39	49.42
Baboon	50.03	49.94	49.90	49.80	49.78	49.64	49.26	49.41	49.29
Peppers	50.00	49.99	49.95	49.86	50.00	49.80	48.97	48.98	49.02
Average	49.98	49.98	49.94	49.74	49.82	49.75	49.23	49.26	49.24

as follows: $(M \times M) \bmod (m \times m) \equiv 0$ and $(M \times M) \bmod (n \times n) \equiv 0$.

Assume that the initial conditions and the control parameters are double-precision numbers. If the computational precision of the double-precision numbers is 10^{-16} , the total number of different values f_0 which can be used as secret keys is more than 10^{16} , so are the numbers of $x_0, y_0, z_0, \lambda_0, \mu_0, \mu_1, \mu_2, g_0, g_1, g_2, \alpha_0, \alpha_1$ and α_2 . Also, assume that the number of possible triplets (b_{12}, b_{21}, k) is K and the number of possible values of m and n is L .

Therefore, the key space of the proposed image encryption scheme is:

$$KS(f_0, x_0, y_0, z_0, \lambda_0, \mu_0, \mu_1, \mu_2, g_0, g_1, g_2, \alpha_0, \alpha_1, \alpha_2, b_{12}, b_{21}, k, m, n) > K \times L \times 10^{224}$$

Accordingly, the key space of the encryption algorithm is adequate to resist all kinds of brute-force attacks.

5.3.2 Key sensitivity analysis

A good encipherment scheme should be sensitive to any bit flipping in the secret key to ensure security against

brute-force attacks. Avalanche effect is used to test the sensitivity with respect to a slight change in the key. Experimental results are reported in Table 12. As can be seen from this table, the avalanche effects for all encrypted images with the proposed algorithm are very close to 50% and are better than those of algorithms studied in [40,41]. The results prove a high key sensitivity performance of the proposed scheme with a tiny change in the key (s).

5.3.3 Histogram analysis

To examine the resistance of the suggested scheme against statistical attacks, we analyzed the histograms of different color cipher images. Figure 9 shows the histograms of the plain image of ‘Lena’ for red, green and blue channels, whereas their corresponding histograms of the encrypted image of ‘Lena’ are shown in Fig. 10. The histograms of the cipher images have an approximately uniform distribution and show a significant difference from those of the original images. Thus, the statistical attack is hard to apply on the proposed cryptosystem.

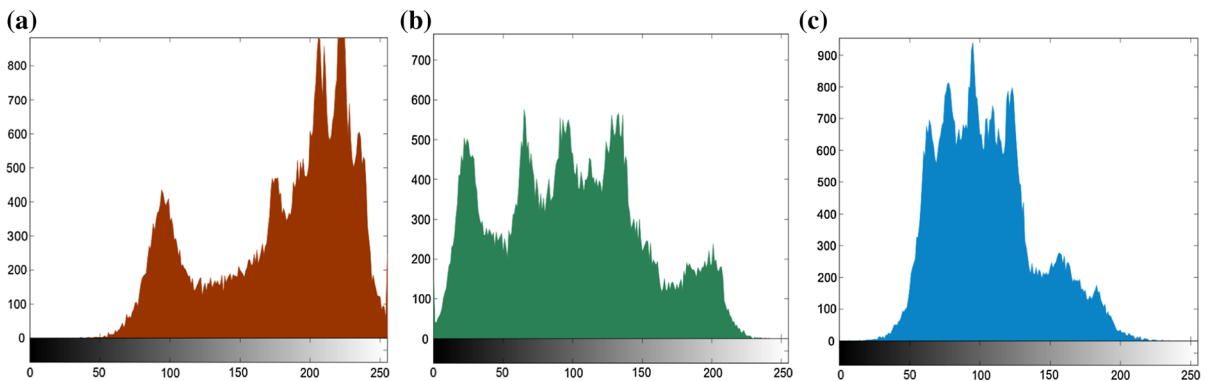


Fig. 9 Histograms of original images of ‘Lena’ a Red, b Green and c Blue

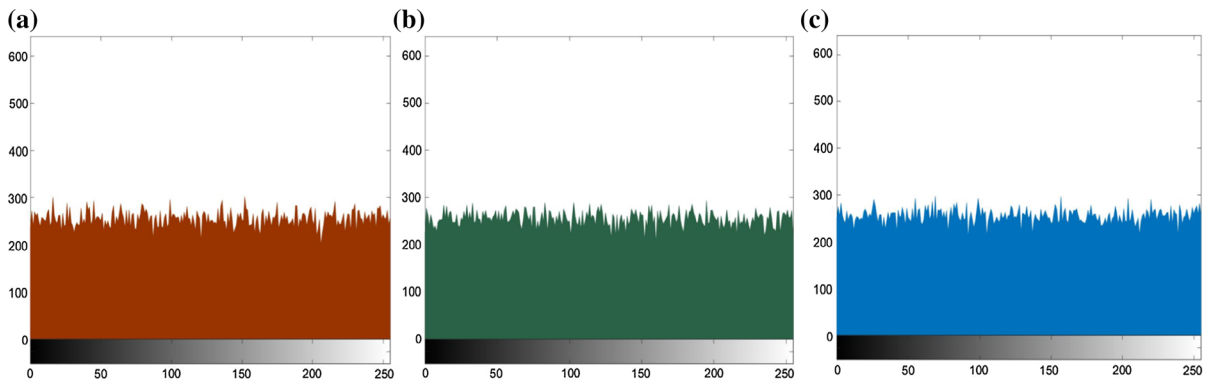


Fig. 10 Histograms of ciphered images of ‘Lena’ **a** Red, **b** Green and **c** Blue

Table 13 Correlation coefficients of two adjacent pixels in plain images and encrypted images

Images	Horizontal			Vertical			Diagonal		
	R	G	B	R	G	B	R	G	B
Original Lena	0.9307	0.9300	0.8810	0.9665	0.9638	0.9399	0.8915	0.9086	0.8448
Original Baboon	0.8266	0.7407	0.8444	0.7407	0.6830	0.8031	0.7386	0.6758	0.7732
Original Peppers	0.9934	0.9921	0.9885	0.9878	0.9844	0.9826	0.9839	0.9776	0.9741
Encrypted Lena	-0.0283	-0.0026	-0.0217	-0.0317	-0.0173	-0.0036	-0.0344	-0.0015	-0.0191
Encrypted Baboon	-0.0072	-0.0260	-0.0099	-0.0201	-0.0220	-0.0034	-0.0016	-0.0175	-0.0066
Encrypted Peppers	-0.0202	-0.0060	-0.0001	-0.0113	-0.0116	-0.0212	-0.0041	-0.0117	-0.0037

5.3.4 Correlation analysis

A robust encryption scheme should produce cipher image with low correlation between adjacent pixels [8,9]. The visual testing of the correlation of adjacent pixels can be conducted by plotting the distribution of the adjacent pixels in the plain image and its corresponding cipher image. The correlation coefficient between two adjacent pixels in an image is determined as:

$$r_{\alpha\beta} = \frac{cov(\alpha, \beta)}{\sqrt{\psi(\alpha)}\sqrt{\psi(\beta)}}, \tag{42}$$

where,

$$\psi(\alpha) = \frac{1}{N} \sum_{i=1}^N [\alpha_i - E(\alpha)]^2, \tag{43}$$

$$cov(\alpha, \beta) = \frac{1}{N} \sum_{i=1}^N [\alpha_i - E(\alpha)][\beta_i - E(\beta)], \tag{44}$$

where α_i and β_i denote two adjacent pixels (either vertical, horizontal or diagonal), N is the total num-

ber of duplets (α_i, β_i) obtained from the image; $E(\alpha)$ and $E(\beta)$ are the mean values of α_i and β_i , respectively. Table 13 shows the correlations of two adjacent pixels in the plain images mentioned above and their encrypted images. Moreover, Figs. 11, 12, and 13 show the correlation of adjacent pixels in the three directions of the original and ciphered images, respectively.

The strong correlations between adjacent pixels in plain images are greatly reduced in the encrypted images generated by the proposed encryption scheme (Table 13). Therefore, the proposed cryptosystem generates de-correlated adjacent pixels in the cipher image and thus satisfies the confusion and diffusion properties.

5.3.5 Sensitivity analysis

To measure the influence of one pixel change on the encrypted image quantitatively, two most common criteria, namely number of pixel change rate (NPCR) and unified average changing intensity (UACI) are used [4,29]. Let $I(i, j)$ and $J(i, j)$ be the (i, j) th pixel of

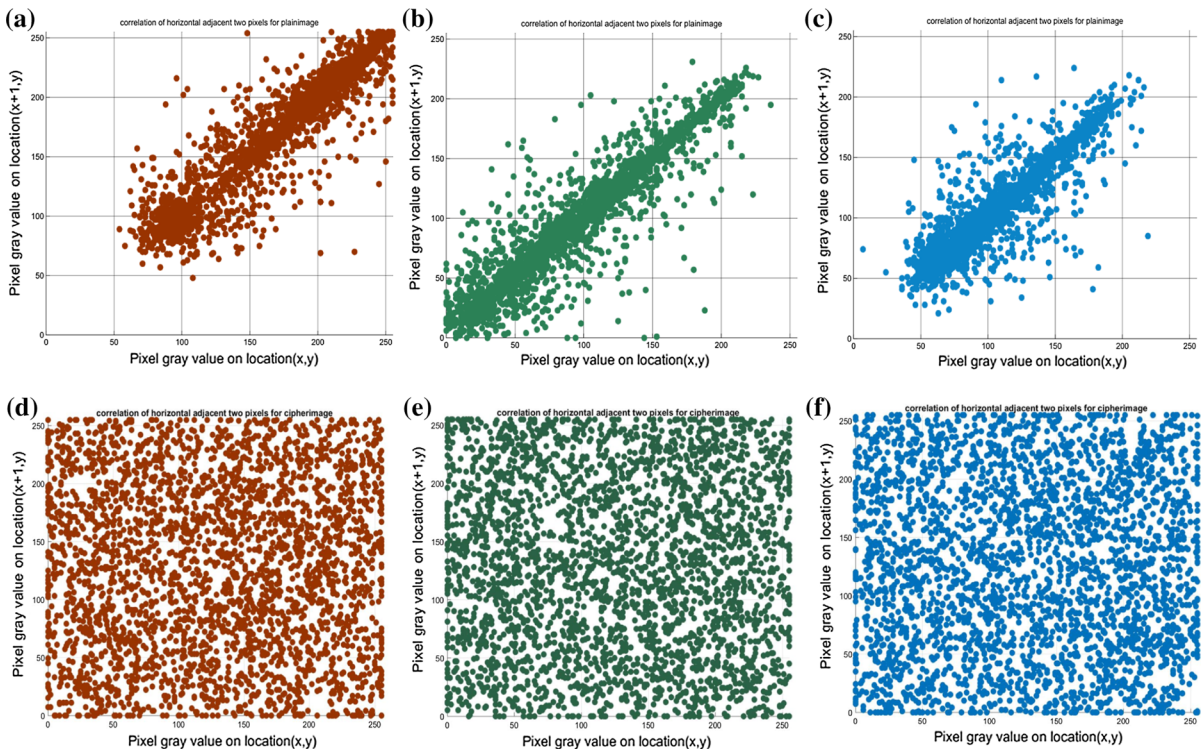


Fig. 11 Distributions of two horizontally adjacent pixels in plain and encrypted images of ‘Lena’: **a** R-plain image, **b** G-plain image, **c** B-plain image, **d** R-encrypted image, **e** G-encrypted image and **f** B-encrypted image

two images I and J , respectively. NPCR is defined as:

$$NPCR_{R,G,B} = \frac{\sum_{i,j} D(i, j)}{L} \times 100 \%, \tag{45}$$

where L is the total pixels in an image and $D(i, j)$ is described as

$$D(i, j) = \begin{cases} 0 & \text{if } I(i, j) = J(i, j), \\ 1 & \text{if } I(i, j) \neq J(i, j). \end{cases} \tag{46}$$

While, UACI, is defined as:

$$UACI_{R,G,B} = \frac{1}{L} \sum_{i,j} \frac{|I(i, j) - J(i, j)|}{2^N - 1} \times 100 \%, \tag{47}$$

where N is the number of bits used to represent the pixel value. The expected values of NPCR and UACI for an efficient cryptosystem scheme are defined by:

$$NPCR_{R,G,B}(Expected) = \left(1 - \frac{1}{2^{n_{R,G,B}}}\right) \times 100 \%, \tag{48}$$

$$UACI_{R,G,B}(Expected) = \frac{1}{2^{n_{R,G,B}}} \left(\frac{\sum_{p=1}^{2^{n_{R,G,B}}-1} p(p+1)}{2^{n_{R,G,B}} - 1} \right) \times 100 \%, \tag{49}$$

where, $2^{n_{R,G,B}}$ means the number of bits in one pixel of color channels(R, G and B) in a color image. The expected values of NPCR and UACI in a 24-bit true color image are 99.6094 and 33.4635 %, respectively.

A large number of plain images are evaluated using two measurements, i.e., NPCR and UACI, in order to test the influence of pixels change in the plain image on the cipher image. The results of NPCR and UACI of each component (red, green and blue) of ‘Lena’, ‘Baboon’ and ‘Peppers’ images for different algorithms are shown in Table 14. The proposed encryption scheme has little better NPCR and UACI perfor-

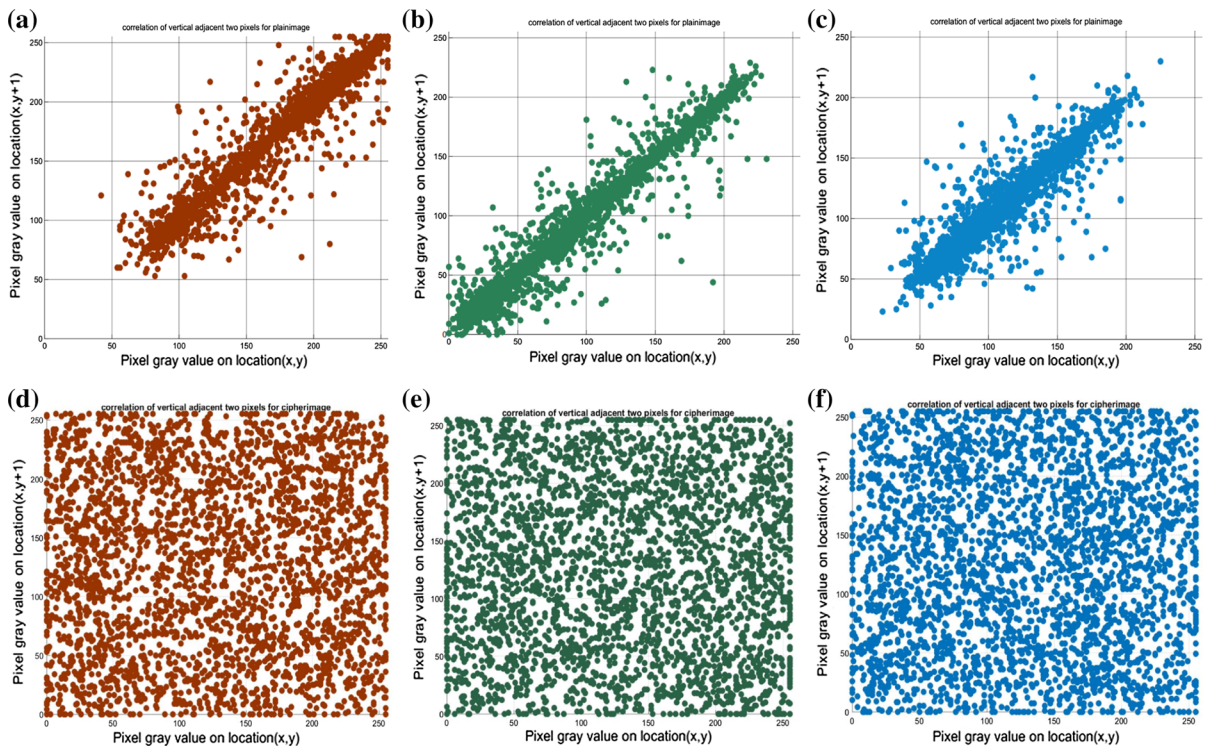


Fig. 12 Distributions of two vertically adjacent pixels in plain and encrypted images of 'Lena': **a** R-plain image, **b** G-plain image, **c** B-plain image, **d** R-encrypted image, **e** G-encrypted image and **f** B-encrypted image

mances compared to the other algorithms. Thus, the proposed cipher demonstrates resistance against differential attacks.

5.3.6 Information entropy analysis

Information entropy is a statistical measure of randomness. The formula to calculate information entropy can be found in [13]. The entropies of the ciphered images were measured to evaluate their uncertainties. For a random source emitting 256 symbols, the entropy is 8 bits and is obtained only if all symbols have the same probability. Consequently, the best entropy value (approximately 8) indicates the efficiency of the proposed encryption scheme. The entropies of the Red, Green and Blue channels of each encrypted image, using the proposed method, are very close to 8 (Table 15) and are better than those of the schemes used in [40, 41]. It is clear that the randomness is satisfactory and the probability of accidental information leakage is very little.

5.3.7 Encryption speed and computation complexity

5.3.7.1. Encryption speed The speed of an algorithm can be characterized by measuring the time required for the encryption process. We measured this parameter for the proposed algorithm and for the ones already available in the literature (Table 16). The average time required to encrypt the data using the proposed scheme was 166 kb/s; however, using the algorithms in [40, 41] the encryption took 214 and 125 kb/s, respectively. The result of our proposed algorithm is quite promising, compared to the already existing algorithms. Therefore, it is suitable for real time applications.

5.3.7.2. Computation complexity The multifaceted calculation of a cipher scheme is measuring the span of an occasion (the quantity of operations and steps required to fulfill the encryption/unscrambling process)—ignoring a few subtle elements, for example, the working framework, the programming dialect, the equipment the calculation keeps running on, and the pro-

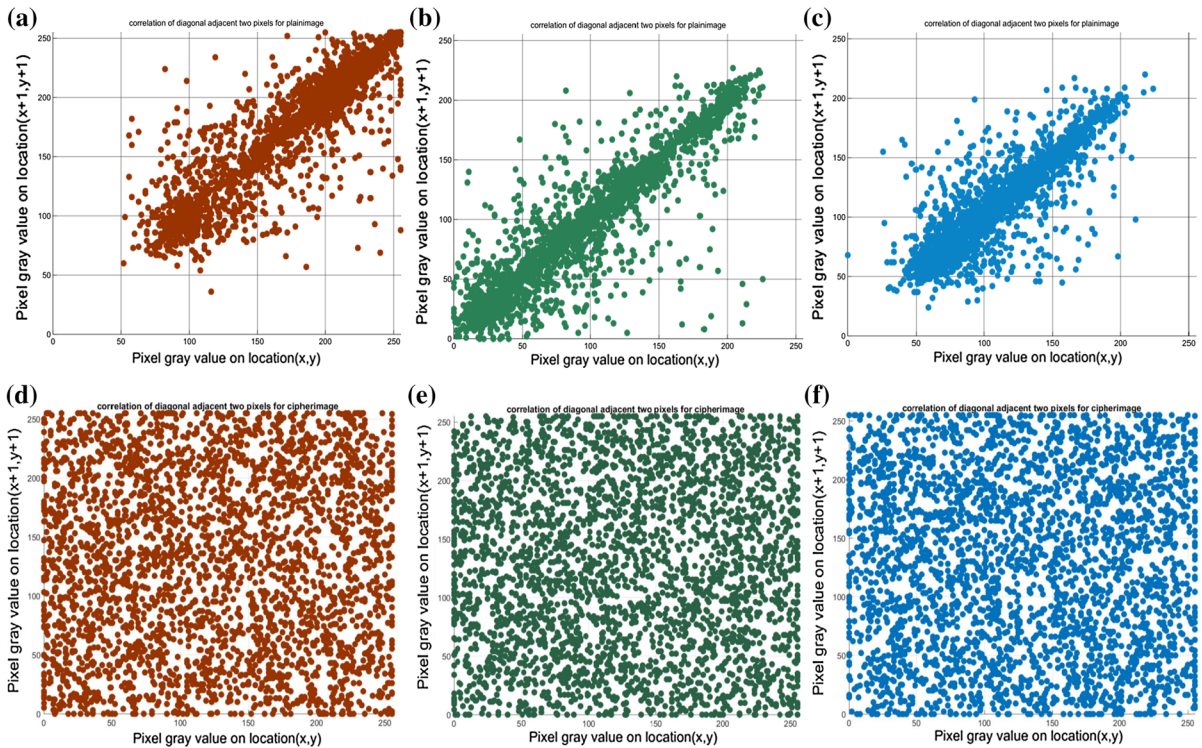


Fig. 13 Distributions of two diagonally adjacent pixels in plain and encrypted images of ‘Lena’: **a** R-plain image, **b** G-plain image, **c** B-plain image, **d** R-encrypted image, **e** G-encrypted image and **f** B-encrypted image

Table 14 The NPCR and UACI of encrypted images by changing their plain images one bit

Image	NPCR								
	New scheme			Ref. [40]			Ref. [41]		
	R	G	B	R	G	B	R	G	B
Lena	99.6317	99.6205	99.6211	99.6173	99.6233	99.6186	99.6085	99.6115	99.6184
Baboon	99.6199	99.6250	99.6273	99.6287	99.6290	99.6162	99.6091	99.6152	99.6058
Peppers	99.6202	99.6192	99.6224	99.6182	99.6177	99.6272	99.6204	99.6166	99.6182
Average	99.6239	99.6216	99.6236	99.6214	99.6233	99.6207	99.6127	99.6144	99.6141
Image	UACI								
	New scheme			Ref. [40]			Ref. [41]		
	R	G	B	R	G	B	R	G	B
Lena	33.6783	33.7999	33.6200	33.7075	33.6728	33.5293	33.4752	33.5174	33.5398
Baboon	33.6484	33.5908	33.6749	33.7491	33.6830	33.5976	33.5381	33.6053	33.6372
Peppers	33.6602	33.6575	33.7314	33.6813	33.5364	33.6506	33.6231	33.5570	33.6035
Average	33.6623	33.6827	33.6754	33.7126	33.6307	33.5925	33.5455	33.5599	33.5935

gramming aptitude [42]. Formally, the computational complexity of the proposed encryption plan can be computed as in Table 17. In this way, the most pes-

simistic scenario execution of the proposed encryption plan was $O(n)$, which is adequate for constant applications.

Table 15 Entropy results for encrypted images

Image	Entropy								
	New scheme			Ref. [40]			Ref. [41]		
	R	G	B	R	G	B	R	G	B
Lena	7.9993	7.9990	7.9989	7.9968	7.9975	7.9983	7.9969	7.9973	7.9977
Baboon	7.9987	7.9989	7.9990	7.9973	7.9969	7.9985	7.9981	7.9974	7.9977
Peppers	7.9992	7.9991	7.9988	7.9972	7.9977	7.9987	7.9986	7.9972	7.9968
Average	7.9991	7.9990	7.9989	7.9971	7.9974	7.9985	7.9979	7.9973	7.9974

Table 16 Encryption speeds (kb/s)

	Scheme		
	New scheme	Ref. [40]	Ref. [41]
Speed (kb/s)	167	214	125

Table 17 Computation complexity of the proposed algorithm

Operations	+	−	×	mod and /	XOR/bitxor	Compare
complexity	<19n	6n	<30n	<19n	3n	771n

where n is the number of pixels.

5.3.8 NIST SP 800-22 tests for the cipher

In order to test the cipher randomness, NIST SP 800-22 [43] tests are used. The role of these tests is to analyze the randomness of (arbitrary sequence) binary sequences produced by encryption systems. The NIST Tests consists of 16 tests which are used to detect any non-randomness that exists in a sequence. In this sense, the 16 tests were carried out for 150 sequences of ciphers with a length equal to 10^6 bits. The ciphered data were a colored image of size 256×256 .

The proposed algorithm went through all NIST SP 800-22 tests successfully (Table 18). Consequently, the ciphers generated by the proposed encryption scheme are absolutely stochastic.

5.3.9 Resistance to known-plaintext and chosen-plaintext attacks

The initial values of the logistic-sine, logistic-Chebyshev and sine-Chebyshev maps of the proposed algorithm are exchanged according to the cipher values after each round. Their state values depend on the plain

image. Since the parameters of diffusion and permutation, which are the important part of the keystream, i.e., $(SB_r^i, SB_g^i, SB_b^i)_{i=1, \dots, N_r}$, $x_n^b, y_n^b, z_n^b, J_1, J_2$ and J_3 are correlated to the state values of these maps, different images will have different $(SB_r^i, SB_g^i, SB_b^i)_{i=1, \dots, N_r}$, $x_n^b, y_n^b, z_n^b, J_1, J_2$ and J_3 . Therefore, the attacker cannot decrypt a particular cipher image using the parameters obtained from other images. As a result, the proposed algorithm can resist the known-plaintext and chosen-plaintext attacks properly [50–52].

6 Security and performance analysis of S-box-only chaotic image ciphers

6.1 Statistical analysis

The factual investigations give insight into the working of any cryptographic framework. So as to assess the execution of the proposed S-box, we conducted correlation analysis, entropy analysis, contrast analysis, homogeneity test and energy analysis. The aftereffects of correlation examination demonstrate the degree of likeness between the plain and encrypted information. In the event that there are any hints of correlation, there is a probability that cryptanalysis can

Table 18 SP800-22 tests suite for cipher image

Test name		<i>P</i> value	Result
Frequency		0.4479	Success
Block-frequency		0.8256	Success
Runs ($M = 10,000$)		0.3982	Success
Long runs of ones		0.7127	Success
Rank		0.2919	Success
Spectral DFT		0.2457	Success
No overlapping templates		0.9998	Success
Overlapping templates		0.8599	Success
Universal		0.9900	Success
Linear complexity		0.5183	Success
Serial	<i>P</i> value1	0.2337	Success
Serial	<i>P</i> value2	0.6419	Success
Approximate entropy		0.8991	Success
Cumulative sums forward		0.9995	Success
Cumulative sums reverse		0.2768	Success
Random excursions	$X = -4$	0.8368	Success
	$X = -3$	0.1535	Success
	$X = -2$	0.2574	Success
	$X = -1$	0.7510	Success
	$X = 1$	0.8833	Success
	$X = 2$	0.5999	Success
	$X = 3$	0.8006	Success
	$X = 4$	0.6037	Success
Random excursions variant	$X = -9$	0.7279	Success
	$X = -8$	0.8531	Success
	$X = -7$	0.8683	Success
	$X = -6$	0.6394	Success
	$X = -5$	0.4982	Success
	$X = -4$	0.4425	Success
	$X = -3$	0.9573	Success
	$X = -2$	0.5809	Success
	$X = -1$	0.5501	Success
	$X = 1$	0.6326	Success
	$X = 2$	0.8902	Success
	$X = 3$	0.2850	Success
	$X = 4$	0.1902	Success
	$X = 5$	0.9049	Success
	$X = 6$	0.8008	Success
	$X = 7$	0.8423	Success
	$X = 8$	0.7111	Success
	$X = 9$	0.5427	Success

decode the original information or might have the capacity to incompletely translate data. In the entropy examination, we measured irregularity presented in the plaintext. This measure is likewise valuable in picture encryption application where the visual type information may give extra data about the original information. The contrast analysis gives a review of the measure of dissemination presented in the plain image. This measure is particularly helpful in image encryption applications. A closeness of circulation among various arrangements of components is additionally seen to decide the homogeneity in the scrambled information. This measure decides the resistance to maintain a strategic distance from the cryptanalysis of the fundamental factual assaults. At long last, we carried out the energy analysis to decide the dispersion of energy before and after the encryption procedure. Despite these tests, the correlation of the whole picture was additionally assessed. We examined with interest the execution and investigation of the tests used to benchmark the execution of the proposed S-box. Experimental results of these analyses are reported in Tables 19, 20, 21.

Because of haphazardness, the estimations of contrast and entropy expand. This makes the encrypted picture hard to identify. The homogeneity, correlation and energy qualities are likewise distinctive in unique and encrypted pictures. The shaded parts of unique and scrambled pictures clarify the quality of disarray in relating layers as well. These qualities can be further improved by including some dispersion attributes in the proposed algorithm, however our principle is just to manage nonlinear component of block cipher, specifically the impacts of composition elements taking into account S-boxes. Expanding the estimations of contrast in encrypted pictures when contrasted with the original image is the impression of quality in outlined algorithm. It can be considered as a straight reliance of gray levels of neighboring pixels. In the event that the neighboring pixels are fundamentally the same as in their dark level values then the difference in the picture is low. If there should arise an occurrence of composition, the dark level variations demonstrate the variation of the surface itself. High contrast qualities are normal for heavy textures and low for smooth delicate surfaces. In the event of encrypted pictures the estimations of contrast increment however they decrease for the plain pictures.

Table 19 Second-order texture analyses for the proposed cryptosystem with one round of Lena image of size 256*256

	Plain image	Plain color components of image			Cipher image	Cipher color components of image		
		Red	Green	Blue		Red	Green	Blue
Contrast	0.4381	0.4415	0.4584	0.4412	5.2238	5.2083	5.3120	5.1731
Homogeneity	0.8689	0.8627	0.8616	0.8620	0.4643	0.4643	0.4623	0.4665
Entropy	7.7833	7.3116	7.5988	7.11581	7.9376	7.9506	7.9505	7.9393
Correlation	0.9055	0.9111	0.9188	0.8319	-0.0182	0.0524	0.0583	0.0592
Energy	0.1174	0.1349	0.0959	0.1589	0.0284	0.0269	0.0258	0.0268

Table 20 Second-order texture analyses for the proposed cryptosystem with one round of Baboon image of size 256*256

	Plain image	Plain color components of image			Cipher image	Cipher color components of image		
		Red	Green	Blue		Red	Green	Blue
Contrast	0.8167	0.8145	0.8309	0.8364	5.1644	5.1759	5.2536	5.1405
Homogeneity	0.7550	0.7541	0.7517	0.7497	0.4653	0.4655	0.4639	0.4661
Entropy	7.5311	7.4566	7.2492	7.4862	7.9252	7.9317	7.9381	7.9424
Correlation	0.6706	0.7914	0.7199	0.8135	-0.0108	0.0600	0.0637	0.0576
Energy	0.0977	0.0738	0.0835	0.0748	0.0285	0.0268	0.0259	0.0271

Table 21 Second-order texture analyses for the proposed cryptosystem with one round of Pepper image of size 256*256

	Plain image	Plain color components of image			Cipher image	Cipher color components of image		
		Red	Green	Blue		Red	Green	Blue
Contrast	0.1102	0.1251	0.1293	0.1112	5.1603	5.1631	5.2233	5.1205
Homogeneity	0.9511	0.9439	0.9431	0.9501	0.4648	0.4642	0.4628	0.4654
Entropy	7.7235	7.5327	7.6089	6.9293	7.9233	7.9191	7.9279	7.9117
Correlation	0.9794	0.9856	0.9814	0.9755	-0.0112	0.0558	0.0693	0.0587
Energy	0.1640	0.1292	0.1435	0.2591	0.0286	0.0270	0.0259	0.0271

The values of the original images homogeneity are high as their pixels concentrate along the corner to corner, implying that there are many pixels with the same or fundamentally the same gray level quality. The bigger the adjustments in gray qualities are, the lower the homogeneity is, which makes the contrast higher. The experimental results of our algorithm which depends on S-boxes show that there is a huge variation which plainly makes the homogeneity not equivalent to 1 that is just for pictures having no variations after encryption. Accordingly, high homogeneity alludes to surfaces that contain perfect redundant structures, while low homogeneity alludes to enormous variety in both, texture elements and their spatial arrangements as portrayed from the classified qualities introduced in Tables 19, 20, 21

individually. A totally arbitrary dispersion would have high entropy.

This component can be helpful in indicating whether the entropy is greater for scrambled pictures. This gives us an idea of which kind of algorithm can be factually considered more secure. The estimation of randomness would increment as we applied our recommended algorithm to plain pictures that produce encrypted pictures with high entropy estimations. Energy is a measure of local homogeneity and therefore it portrays the conflict of entropy. Mainly this component will inform us of how undeviating the surface is. The more propelled the energy estimation, the greater the homogeneity of the composition. Correlation is a measure of gray level straight reliance between the pixels at the predeter-

Table 22 Differential analysis of S-box-only-based encryption

Image	Component	NPCR	UACI
Lena	R	99.4125	29.9279
	G	99.6811	35.7653
	B	99.3912	34.5062
Baboon	R	99.6887	34.2894
	G	99.6521	34.3853
	B	99.5636	34.8038
Peppers	R	99.5651	36.3651
	G	99.5178	33.7755
	B	99.7040	37.6409
Average	R	99.5554	33.5275
	G	99.6170	34.6420
	B	99.5529	35.6503

mined positions in respect to each other. The estimation of the relationship reduces if there should be an occurrence of scrambled pictures and is close to 1 if there should be an occurrence of plain pictures. All these factual estimations guarantee the confirmation of our proposed criteria of building S-boxes and its appropriateness in picture encryption applications.

6.2 Differential attack analysis

In this section, we performed several tests to prove the effectiveness of the proposed S-box-only-based image encryption against differential attacks. In this sense, we tested the NPCR and UACI for some selected color images. Experimental results are given in Table 22.

Table 22 shows that the proposed S-box-only-based encryption has good performance on sensitivity of the plaintext and can resist differential attacks.

6.3 Resistance of S-box-only chaotic image ciphers against chosen-plaintext attacks (CPA)

In fact, substitution boxes play a good role in designing secure substitution architecture in most ciphers. However, most of the previous S-boxes designs are weak, especially if they are only used in image ciphers, against attacks such as CPA. In addition, the S-box analyses like nonlinearity and SAC cannot reveal the real efficiency against CPA attacks if it is used in image cryptosystem.

A good procedure to prove the resistance of S-box-only chaotic image cipher against CPA attacks was proposed by Zhang and Xiao in [53]. Therefore, to follow the rules raised in [53], we applied them to the proposed S-box-only chaotic image ciphers. After careful analysis, one can conclude that the proposed algorithm is secure enough according to the following rules:

Rule 1: The key generation of the S-box depends on the plain image

Rule 2: A well-designed cryptosystem must satisfy the confusion and diffusion [1]. In the proposed S-box, confusion is satisfied by a secure permutation method and diffusion is done by gray code combined with chaotic maps.

Rule 3: In the proposed cryptosystem, we used chaotic dynamical systems that are approximately uniform distribution to generate the pseudorandom number sequences.

Rule 4: The inner cipher rounds are not included as part of the keys, since its assessment can be obtained through an abstract analysis of the encipherment time.

In short, the proposed image cipher-based S-box in this paper is secure according to the above raised rules.

7 Conclusion

This paper proposes a simple and efficient S-box method based on logistic-sine map. To cope with the disadvantages of small key space and weak obscurity in the current chaotic encryption methods, an efficient chaos-based image encryption scheme in the form of permutation–substitution structure was proposed. The objective of using the permutation–substitution structure was to create confusion between the cipher image and the keystream and enhance the security by adding diffusion in the plain image. Detailed differential and statistical analyses were carried out to show the effectiveness of the proposed schemes. Results demonstrate that the proposed encryption scheme meets all the performance requirements of image encryption design criteria. It also has the advantages of large key space and is therefore adequate for and the practical implementation of encryption schemes.

Acknowledgements The authors would like to thank Mrs. Najet Yahia for her help in the proofreading of this work. This work is supported by Ministry of Higher Education and Scientific Research (Egypt-Tunisia Cooperation Program: 4-13 A1).

References

- Jakimoski, G., Kocarev, L.: Chaos and cryptography: block encryption ciphers. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **48**(2), 163–169 (2001)
- Baptista, M.S.: Cryptography with chaos. *Phys. Lett. A* **240**(1–2), 50–54 (1998)
- Schmitz, R.: Use of chaotic dynamical systems in cryptography. *J. Frankl. Inst.* **338**(4), 429–441 (2001)
- Chen, G., Mao, Y., Chui, C.K.: A symmetric image encryption scheme based on 3D chaotic standard maps. *Chaos Solitons Fractals* **21**(3), 749–761 (2004)
- Li, X., Wang, L., Yan, Y., Liu, P.: An improvement color image encryption algorithm based on DNA operations and real and complex chaotic systems. *Opt. Int. J. Light Electron Opt.* **127**(5), 2558–2565 (2016)
- Zhou, G., Zhang, D., Liu, Y., Yuan, Y., Liu, Q.: A novel image encryption algorithm based on chaos and Line map. *Neurocomputing* **169**, 150–157 (2015)
- Özkaynak, F., Yavuz, S.: Analysis and improvement of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Nonlinear Dyn.* **78**(2), 1311–1320 (2014)
- Diaconu, A.V.: Circular inter-intra pixels bit-level permutation and chaos-based image encryption. *Inf. Sci.* **355–365**, 314–327 (2016)
- Xu, L., Li, Z., Li, J., Hua, W.: A novel bit-level image encryption algorithm based on chaotic maps. *Opt. Lasers Eng.* **78**, 17–25 (2016)
- Hussain, I., Shah, T., Gondal, M.A.: A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm. *Nonlinear Dyn.* **70**(3), 1791–1794 (2012)
- Li, C., Zhang, L.Y., Ou, R., Wong, K.-W., Shu, S.: Breaking a novel colour image encryption algorithm based on chaos. *Nonlinear Dyn.* **70**(4), 2383–2388 (2012)
- Wang, X., Teng, L., Qin, X.: A novel colour image encryption algorithm based on chaos. *Signal Process.* **92**(4), 1101–1108 (2012)
- Liu, H., Wang, X.: Color image encryption using spatial bit-level permutation and high-dimension chaotic system. *Opt. Commun.* **284**(16–17), 3895–3903 (2011)
- Khan, M., Shah, T.: A novel image encryption technique based on Hénon chaotic map and S8 symmetric group. *Neural Comput. Appl.* **25**(7), 1717–1722 (2014)
- Zhang, Y., Xiao, D.: Self-adaptive permutation and combined global diffusion or chaotic color image encryption. *Int. J. Electron. Commun.* **68**(4), 361–368 (2014)
- Liu, H., Kadir, A., Gong, P.: A fast color image encryption scheme using one-time S-Boxes based on complex chaotic system and random noise. *Opt. Commun.* **338**, 340–347 (2015)
- Zhang, W., Yu, H., Zhao, Y., Zhu, Z.: Image encryption based on three-dimensional bit matrix permutation. *Signal Process.* **118**, 36–50 (2016)
- Dawson, M., Tavares, S.: An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks. In: *Advances in Cryptology: Proceedings of EURO-CRYPT'91*. Lecture Notes in Computer Science, pp. 352–367 (1991)
- Zhang, Y., Li, C., Li, Q., Zhang, D., Shu, S.: Breaking a chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn.* **69**(3), 1091–1096 (2012)
- Wang, Y., Xie, Q., Wu, Y., Du, B.: A software for S-box performance analysis and test. In: *2009 International Conference on Electronic Commerce and Business Intelligence (ECBI)*, pp. 125–128. IEEE (2009)
- Wang, X.Y., Yang, L., Liu, R., Kadir, A.: A chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn.* **62**(3), 615–621 (2010)
- Matsui, M.: Linear cryptanalysis method of DES cipher: advances in cryptology. In: *Proceedings of the Eurocrypt'93*, Lecture Notes in Computer Science, vol. 765, pp. 386–397 (1994)
- Tang, G., Liao, X., Chen, Y.: A novel method for designing S-boxes based on chaotic maps. *Chaos Solitons Fractals* **23**(2), 413–419 (2005)
- Chen, G., Chen, Y., Liao, X.: An extended method for obtaining S-boxes based on 3-dimensional chaotic baker maps. *Chaos Solitons Fractals* **31**(3), 571–579 (2007)
- Özkaynak, F., Özer, A.B.: A method for designing strong S-boxes based on chaotic Lorenz system. *Phys. Lett. A* **374**(36), 3733–3738 (2010)
- Khan, M., Shah, T., Mahmood, H., Gondal, M.A., Hussain, I.: A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems. *Nonlinear Dyn.* **70**(3), 2303–2311 (2012)
- Hussain, I., Shah, T., Gondal, M.A., Mahmood, H.: A projective general linear group based algorithm for the construction of substitution box for block ciphers. *Neural Comput. Appl.* **22**(6), 1085–1093 (2013)
- Khan, M., Shah, T., Mahmood, H., Gondal, M.A.: An efficient method for the construction of block cipher with multi-chaotic systems. *Nonlinear Dyn.* **71**(3), 489–492 (2013)
- Abd El-Latif, A.A., Li, L., Wang, N., Han, Q., Niu, X.: A new approach to chaotic image encryption based on quantum chaotic systems, exploiting color spaces. *Signal Process.* **93**(11), 2986–3000 (2013)
- Bao, J., Yang, Q.: Period of the discrete arnold cat map and general cat map. *Nonlinear Dyn.* **70**(2), 1365–1375 (2012)
- Khan, M., Shah, T., Batool, S.I.: Construction of S-box based on chaotic Boolean functions and its application in image encryption. *Neural Comput. Appl.* **27**(3), 677–685 (2016)
- Liu, Y., Wang, J., Fan, J., Gong, L.: Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences. *Multimed. Tools Appl.* **75**(8), 4363–4382 (2016)
- Liu, G., Yang, W., Liu, W., Dai, Y.: Designing S-boxes based on 3-D four-wing autonomous chaotic system. *Nonlinear Dyn.* **82**(4), 1867–1877 (2015)
- Belazi, A., Rhouma, R., Belghith, S.: A novel approach to construct S-box based on Rossler system. In: *2015 International Conference on Wireless Communications and Mobile Computing (IWCMC)*, pp. 611–615. IEEE (2015)
- Wang, Y., Lei, P., Wong, K.-W.: A method for constructing bijective S-Box with high nonlinearity based on chaos and optimization. *Int. J. Bifurc. Chaos* **25**(10), 1–15 (2015)
- Liu, Y., Tong, X., Ma, J.: Image encryption algorithm based on hyper-chaotic system and dynamic S-box. *Multimed. Tools Appl.* **75**(13), 7739–7759 (2016)

37. Zhou, Y., Bao, L., Philip Chen, C.L.: A new 1D chaotic system for image encryption. *Signal Process.* **97**, 172–182 (2014)
38. Webster, A.F., Tavares, S.: On the design of S-boxes. In: *Advances in Cryptology: Proceedings of CRYPTO'85. Lecture Notes in Computer Science*, pp. 523–534 (1986)
39. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol.* **4**(1), 3–72 (1991)
40. Hua, Z., Zhou, Y., Pun, C.-M., Philip Chen, C.L.: 2D Sine Logistic modulation map for image encryption. *Inf. Sci.* **297**, 80–94 (2015)
41. Wang, X., Liu, L., Zhang, Y.: A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Opt. Lasers Eng.* **66**, 10–18 (2015)
42. Zhang, Y.-Q., Wang, X.-Y.: A new image encryption algorithm based on non-adjacent coupled map lattices. *Appl. Soft Comput.* **26**, 10–20 (2015)
43. Pareschi, F., Rovatti, R., Setti, G.: On statistical tests for randomness included in the NIST SP800-22 test suite and based on the binomial distribution. *IEEE Trans. Inf. Forensics Secur.* **7**(2), 491–505 (2012)
44. Norouzi, B., Mirzakuchaki, S., Seyedzadeh, S.M., Mosavi, M.R.: A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion. *Multimed. Tools Appl.* **71**(3), 1469–1497 (2014)
45. Zhang, Y., Xiao, D., Wen, W., Li, M.: Breaking an image encryption algorithm based on hyper-chaotic system with only one round diffusion process. *Nonlinear Dyn.* **76**(3), 1645–1650 (2014)
46. SaberiKamarposhti, M., Mohammad, D., Rahim, M., Yaghobi, M.: Using 3-cell chaotic map for image encryption based on biological operations. *Nonlinear Dyn.* **75**(3), 407–416 (2014)
47. Zhang, Y., Li, Y., Wen, W., Wu, Y., Che, J.-x.: Deciphering an image cipher based on 3-cell chaotic map and biological operations. *Nonlinear Dyn.* **82**(4), 1831–1837 (2015)
48. Liu, Y., Zhang, L.Y., Wang, J., Zhang, Y., Wong, K.-W.: Chosen-plaintext attack of an image encryption scheme based on modified permutation-diffusion structure. *Nonlinear Dyn.* **84**(4), 2241–2250 (2016)
49. Zhu, H., Zhao, C., Zhang, X., Yang, L.: An image encryption scheme using generalized Arnold map and affine cipher. *Opt. Int. J. Light Electron Opt.* **125**(22), 6672–6677 (2014)
50. Wang, Y., Liao, X., Xiang, T., Wong, K.-W., Yang, D.: Cryptanalysis and improvement on a block cryptosystem based on iteration a chaotic map. *Phys. Lett. A* **363**(4), 277–281 (2007)
51. Wei, J., Liao, X., Wong, K.-W., Zhou, T.: Cryptanalysis of a cryptosystem using multiple one-dimensional chaotic maps. *Commun. Nonlinear Sci. Numer. Simul.* **12**(5), 814–822 (2007)
52. Wang, K., Pei, W., Zou, L., Song, A., He, Z.: On the security of 3D cat map based symmetric image encryption scheme. *Phys. Lett. A* **343**(6), 432–439 (2005)
53. Zhang, Y., Xiao, D.: Cryptanalysis of S-box-only chaotic image ciphers against chosen plaintext attack. *Nonlinear Dyn.* **72**(4), 751–756 (2013)