Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6969312, 11 pages
https://doi.org/10.1155/2017/6969312

Research Article

Chaotic S-Box: Intertwining Logistic Map and Bacterial Foraging Optimization

Ye Tian[1,2] and Zhimao Lu[1,3]

[1]*College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China*
[2]*Key Laboratory of Photonic and Electronic Bandgap Materials, Ministry of Education, School of Physics and Electronic Engineering, Harbin Normal University, Harbin 150025, China*
[3]*Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Ye Tian; hsdtianye@126.com

Received 16 June 2017; Revised 24 October 2017; Accepted 26 October 2017; Published 15 November 2017

Academic Editor: Maria L. Gandarias

As the unique nonlinear component of block ciphers, Substitution box (S-box) directly affects the safety of a cryptographic system. It is important and difficult to design strong S-box that simultaneously meets multiple cryptographic criteria such as bijection, nonlinearity, strict avalanche criterion (SAC), bit independence criterion (BIC), differential probability (DP), and linear probability (LP). Though many chaotic S-boxes have been proposed, the cryptographic performance of most of them needs to be further improved. A new chaotic S-box based on the intertwining logistic map and bacterial foraging optimization is designed in this paper. It firstly iterates the intertwining logistic map to construct many S-boxes and then applies a bacterial foraging optimization algorithm to find the optimal S-box. Moreover, bacterial foraging optimization algorithm considers the nonlinearity and differential uniformity as the fitness functions in the optimization process. We experiment that the proposed S-box can effectively resist multiple types of cryptanalysis attacks.

1. Introduction

The dynamic developments in the multimedia industry and the Internet lead to a considerable amount of worry regarding the security of information transmitted over open or stored channels [1–3]. How to protect information from being unauthorized handled is becoming extremely crucial. Modern cryptography technique, in which block cipher algorithm is an important research direction, is an effective way to guarantee the safety of information, since then many researchers have developed a lot of block cipher algorithms. In a block cipher algorithm, Substitution box (S-box) is the only one nonlinear component [4], providing the block cipher system with necessary confusing and scrambling effect against attacks. Moreover, its cryptography security features directly determine the safety of the entire cipher performance [1]. Mathematically, an $n \times n$ size of S-box is a nonlinear mapping $S: \{0,1\}^n \rightarrow \{0,1\}^n$, where $\{0,1\}^n$ represents the vector spaces of n elements from GF(2), and we set $n = 8$ in this paper.

Many papers on S-boxes have been published by scholars around the world over the past decades. In [5], Hussain and Gondal presented a design approach for $n \times n$ S-boxes, which was an exhaustive search method; nevertheless, the performance of this procedure would become rather difficult with the increase of n. Liu et al. [4] utilized near-bent Boolean functions of five variables to generate 5×5 S-boxes to resist the differential attack; however, their algorithm was useful only to create an S-box of odd input bit number. Therefore, most of these approaches were inefficient and were unable to construct S-boxes that could meet multiple assessment requirements simultaneously.

Chaotic systems that satisfy the major requirements of cryptography properties such as diffusion and confusion are differentiated on the basis of their reactiveness to ergodicity, pseudorandomness, unpredictability, control parameters, and initial conditions; this makes chaotic systems particularly catch heaps of heed for cryptology [1–5]. Due to this matter, chaotic S-boxes have proved to be superior for encrypting a message.

For example, literature [6] proposed a four-step method of generating chaotic S-box based on discrete logistic map. It turned out that very simple chaotic maps and discretization procedure generated secure S-boxes. Literature [7] improved the work in [6] using bit extraction and Baker map. Furthermore, literature [8] proposed an S-box design approach based on iteration discrete chaotic that had high immunity to the differential cryptanalysis. Using three-dimensional chaotic Baker map, literature [9] constructed an S-box that approximately fulfilled all the criteria for a cryptographically strong S-box.

However, a simple chaos system also has many defects; for example, the implementation of the chaos on a computer is affected by the limited precision; the time series outputted by the simple chaotic system generally cannot reach the theoretically complete random, resulting in the problem that the pseudorandom sequence appears periodicity [10].

To cope with these problems, many complex chaotic maps based S-boxes have been presented in recent years. For example, literature [11] indicated the pseudorandomness and complexity of binary sequences produced by the Lorenz system and Chebyshev map. Literature [12] designed a new pseudorandom number generator by mixing the couple map lattice technology and the chaos iteration technology. Literature [13] proposed a secure pseudorandom number generator three-mixer. Khan et al. presented a complex chaotic S-box construction method that could provide better security in terms of resistance against various attacks by deploying the 2D Henon chaotic map and skew tent map [14]. Ahmad et al. proposed a method for synthesizing cryptographically efficient chaotic S-box, which integrated four 1D chaotic systems, namely, logistic maps and cubic maps, to modulate the normal system trajectories of the other [15]. Peng et al. designed a novel approach for dynamically generating S-boxes using a spatiotemporal chaotic system, which mapped the key to system parameters and generated the hyperchaotic sequences to construct S-boxes [16].

By increasing the complexity of the chaotic system, these complex algorithms obtained S-boxes with the higher security level to some extent. Nevertheless, the performance gap between many of these chaotic S-boxes and classic ones still exists; for example, few chaos-based S-boxes can achieve the high performance like the one used in advanced encryption standard (AES) [17].

Compared with other intelligent optimization algorithms, bacterial foraging optimization algorithm (BFO) has a group of intelligence and can carry out parallel search. Besides, it may be easy to jump out of the local minimal solution; thus, it can find solutions of higher quality. Due to these advantages, it has been widely used in some research fields. For example, Abd-Elazim and Ali proposed an optimization algorithm BFOA for controlling the damping of the power system's electromechanical oscillations [18]. On the basis of literature [18], Ali and Abd-Elazim proposed a BFOA based Load Frequency Control (LFC) for the suppression of oscillations in power system [19]. In addition, Abd-Elazim and Ali developed an optimization algorithm BSO, which synergistically coupled the BFOA with the particle swarm optimization algorithm for the optimal design of the TCSC damping controller. Specifically, they transformed the controller design problem into an optimization problem, and the BSO was developed to find the optimal controller parameters [20]. Sur and Shukla also presented a discrete adaptive BFO algorithm, which could be applied to discrete search domains and various multidimensional problems [21]. Furthermore, to optimize a power network problem, Tripathy and Mishra proposed an improved BFO algorithm. In this work, the power network problem was formulated as a multiobjective multivariable problem, and the improved BFO was applied to solve this problem [22].

In this paper, a new scheme for designing an S-box is presented. Unlike other chaos-based algorithms that generate strong S-boxes by using the random distribution property of chaotic maps, we divide the process of designing an S-box into two steps. Firstly, we generate many S-boxes by iterating the chaotic map. Secondly, we apply a genetic algorithm, the evaluation function of which adopts the nonlinearity and differential uniformity to improve the performance of the generated S-box. We show via simulation that our scheme can generate stronger S-box. Abbreviations section shows some abbreviations of technical terms involved in this paper.

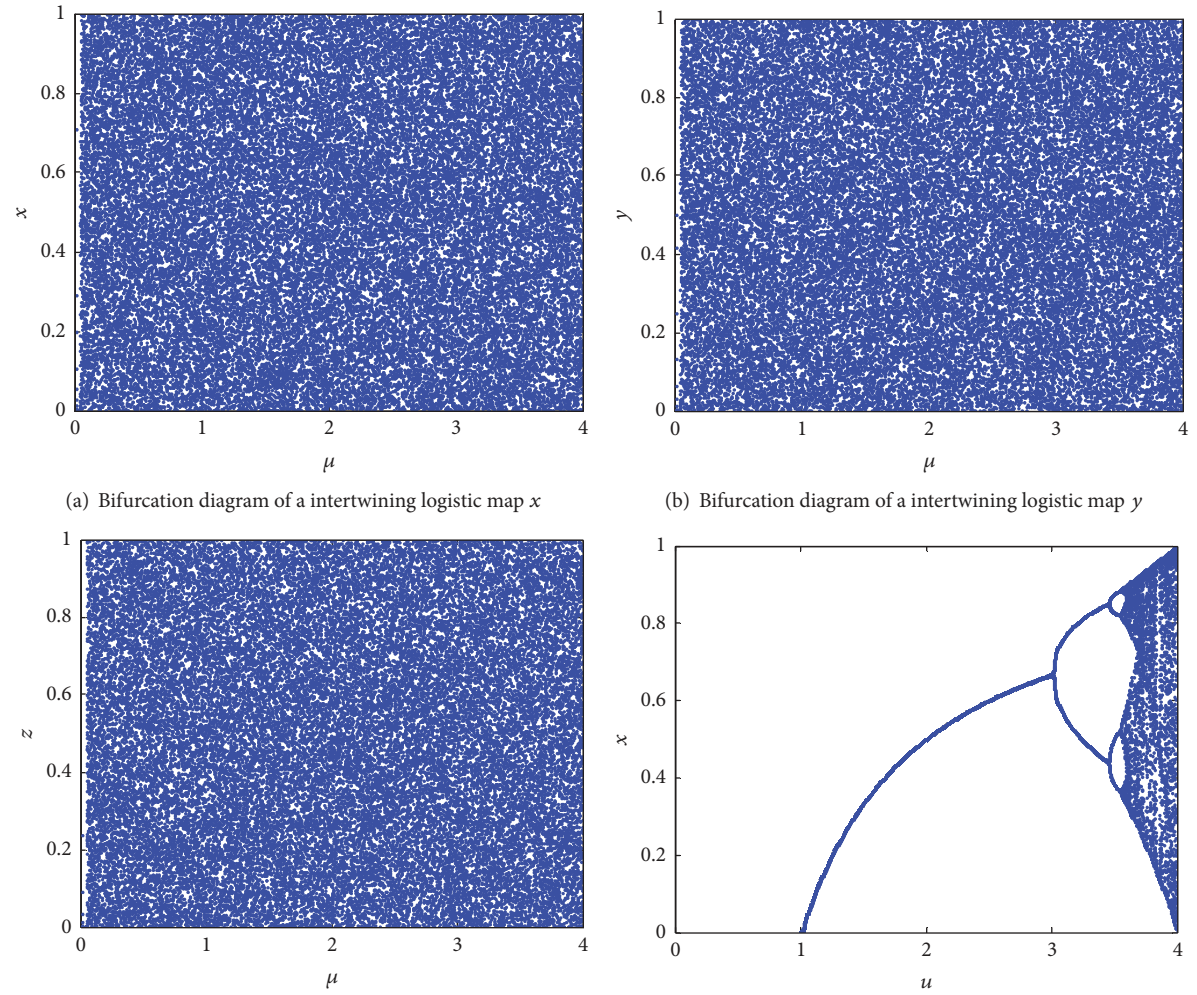
2. Preliminary Work

2.1. Intertwining Logistic Map. In this section, we will introduce a chaotic map, the intertwining logistic map [23], which is defined as follows:

$$
\begin{aligned}
x_{n+1} &= \left[\mu \times k_1 \times y_n \times \left(1 - x_n\right) + z_n\right] \bmod 1 \\
y_{n+1} &= \left[\frac{\mu \times k_2 \times y_n + z_n \times 1}{\left(1 + \left(x_{n+1}\right)^2\right)}\right] \bmod 1 \\
z_{n+1} &= \left[\mu \times \left(x_{n+1} + y_{n+1} + k_3\right) \times \sin\left(z_n\right)\right] \bmod 1,
\end{aligned}
\tag{1}
$$

where μ, k_i are the system parameters with the ranges $0 < \mu < 3.999$, $|k_1| > 33.5$, $|k_2| > 37.9$, $|k_3| > 35.7$. Figures 1(a), 1(b), and 1(c) show the chaotic bifurcation diagrams of the intertwining Logistic map when $k_1 = 39.7$, $k_2 = 40.2$, $k_3 = 38.5$. Figure 1(d) shows the chaotic bifurcation diagram of the logistic map. Figure 2(a) depicts the chaotic attractor diagram of the intertwining Logistic map when $k_1 = 39.7$, $k_2 = 40.2$, $k_3 = 38.5$, and $\mu = 1.5$. Figure 2(b) depicts the chaotic attractor diagram of the logistic map. In the intertwining logistic map system, the sequence distribution becomes more uniform, and, more importantly, empty windows are eliminated. Remarkably, comparing with a simple logistic map, the action of a intertwining logistic chaotic map is more complex, and the sequence distribution of it is more uniform.

2.2. Bacterial Foraging Optimization Algorithm. In 2002, Passino proposed a bacterial foraging optimization algorithm that imitated groups competition coordination mechanism of *Escherichia coli* (*E. coli*) in the process of searching for food in the human gut. In looking for the region of food sources, *E. coli* might determine whether they enter the region through a priori knowledge. Once entering foraging area, if a certain amount of food was consumed or foraging



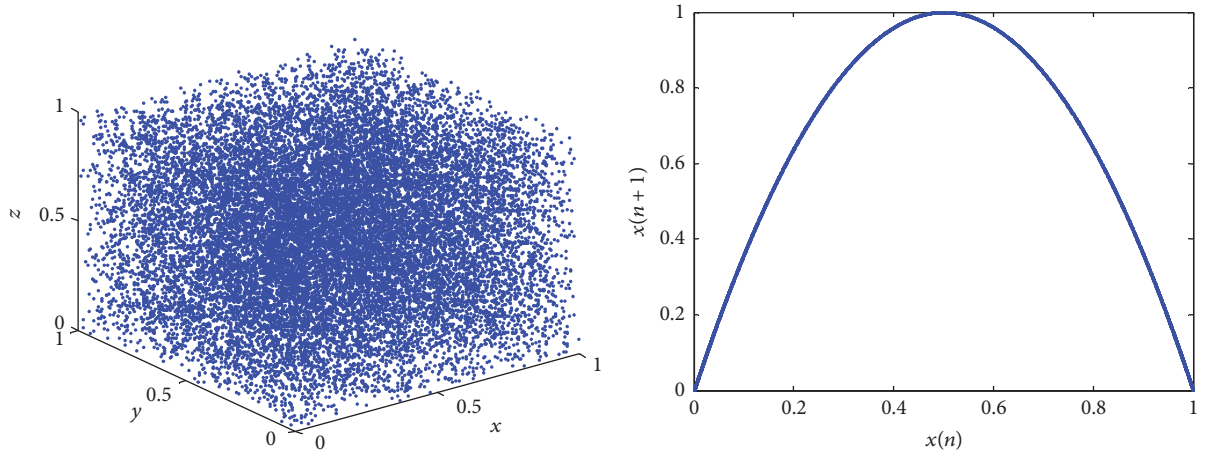
(a) Bifurcation diagram of a intertwining logistic map x

(b) Bifurcation diagram of a intertwining logistic map y

(c) Bifurcation diagram of a intertwining logistic map z

(d) Bifurcation diagram of a logistic map

Figure 1: Bifurcation diagrams comparisons of the logistic map and intertwining logistic map.



(a) Attractor of a intertwining logistic map

(b) Attractor of a logistic map

Figure 2: Attractor comparisons of the logistic map and intertwining logistic map.

region environment deteriorated, resulting in inhospitable conditions, a bacteria might die or move to another suitable region. Relying on its own unique behaviors, chemotaxis, reproduction, and elimination-dispersal, bacterial foraging algorithm proceeded to update bacteria individual position and searched the optimal locations of bacteria groups, so as to realize the evolution of the population. The main steps of bacterial foraging optimization algorithm can be described as follows [24].

Step 1 (parameters initialization). Ned is the number of elimination-dispersal events; Nre is the number of reproduction steps. Nc is the number chemotaxis steps. Ped is the basic probability of elimination-dispersal. SS is the bacteria scale. Ns is the maximum number of chemotactic steps.

Step 2. Initialize the bacteria locations; calculate the initialization fitness values of bacteria.

Step 3. Perform the elimination-dispersal loop $l = 1$: Ned, reproduction loop $k = 1$: Nre, and chemotaxis loop $j = 1$: Nc.

Step 4 (perform bacteria chemotaxis loop). Use $X_i(j,k,l)$ to express the space location vector of bacteria, where j represents the jth generation of chemotaxis loop, k represents the kth generation of reproduction loop, and l represents the lth generation of elimination-dispersal loop.

(1) Tumble. Update bacteria locations by

$$X_i(j+1,k,l) = X_i(j,k,l) + C(i) \times \phi(i,j)$$
$$\phi(i,j) = \frac{\Delta(i,j)}{\sqrt{\Delta^T(i,j)\,\Delta(i,j)}}, \quad (2)$$

where $C(i)$ is the chemotaxis step length of bacterium i. $\phi(i,j)$ is an normalized random direction vector of bacterium i when tumbling in the jth loop. $\Delta(i,j)$ is arbitrarily generated random direction vector with each element being a random number on $[-1,1]$.

(2) Move. If the fitness value of tumbling improves, it will not move according to the direction of tumbling until the fitness value no longer improves or reaches the maximum moves steps Ns.

Step 5 (reproduction loop). After completing the chemotaxis loop, accumulate the fitness values of each bacterium during its life cycle to get an energy value. Order bacteria by the energy values; eliminate half bacteria which are poor to obtain energy. Reproduce half bacteria that have strong ability of energy harvesting.

Step 6 (elimination-dispersal loop). After completing the reproduction operator, generate a random probability, and compare it with the fixed elimination-dispersal probability Ped. If the generated probability is lower than Ped, proceed with the bacteria elimination-dispersal, and random initialize in the definition domain of the solution space.

Step 7. Determine the loop end condition; if it meets the condition, end and output the results; otherwise, continue.

3. Scheme of Generating of an S-Box

3.1. Improved Bacteria Locations Update Equation. Bacteria foraging optimization algorithm was designed to solve the problem of continuous optimization. The map among different S-boxes, however, belongs to a discrete problem; thus we will firstly operate the BFO discretization. To realize it, we introduce a concept of commutator [42] in this paper.

The commutator denoted by $Z = \mathrm{Swap}(x,y)$, $x,y \in \{1,2,3,\ldots,N\}$, where N represents the number of elements of a food source location, represents the exchange between x and y in a food location. Based on one solution, exchange any two solutions several times to obtain another solution. The difference between these two solutions can be denoted by a swap sequence $\{Z_1, Z_2, \ldots, Z_m\}$,where m is the number of commutators.

Next, we should define an operator operation symbol: assume two different solutions X and Y; define $X - Y = \{Z_1, Z_2, \ldots, Z_m\}$; that is, the difference of X and Y determines the exchange sequence; the physical meaning of the operation is that the solution Y will be the same as X after operating m different commutators. The definition of the operation $X + \{Z_1, Z_2, \ldots, Z_m\}$ can be described as follows: exchange the solution X, $\{Z_1, Z_2, \ldots, Z_m\}$ times, and connect these two commutators sequences one after another, respectively. For example, $X_i(t) = [1,2,3,4,5,6,7,8]$, $X_k(t) = [8,6,5,1,4,7,2,3]$, $V(t) = X_k(t) - X_i(t) = \{(1,8)\ (2,6)\ (3,5)\ (4,1)\ (3,4)\ (2,7)\}$. Change (2) by $X_i(t+1) = X_i(t) + C \times \mathrm{rand}()(X_i(t) - X_k(t))$, where rand( ) is a random number on $[0,1]$, C is the speed adjustment parameter, and α is a constant ranging of $[0,1]$. Furthermore, if $C \times \mathrm{rand}()$ is lower than α, retain the commutator.

3.2. Fitness Function. In the BFO, fitness function described the quality of a feasible solution which refers to the richness of the food sources and directly relates to the optimization results. When utilizing chaos optimal search method to construct an S-box, since an S-box has more than one Boolean function characteristic, we should construct a weighted equation to weigh the different cryptography characteristics of the S-box.

As the linear and differential analyses are two effective ways of attack analysis, we will consider the nonlinearity and differential uniformity of an S-box when designing a fitness function that can take the following form:

$$f(s) = a_s f_s(N_s) + a_d f_d(\delta_s), \quad (3)$$

where the nonlinearity function is $f_s(N_s) = N_s$; the differential uniformity function is $f_d(\delta_s) = \delta_s$.

Fitness function should satisfy the monodromy and continuity. According to the experimental results, we order (N_s, δ_s) by the performance of the S-box, that is, $(108,6)$, $(108,8)$, $(106,6)$, $(106,8)$, $(108,10)$, and $(106,10)$ in this paper. Moreover, we assign a_s and a_d and obtain the fitness function as follows [43]:

$$f(s) = \begin{cases} N_s - \delta_s & N_s < 90,\ \delta_s > 18 \\ N_s - 2\delta_s & N_s < 90,\ \delta_s \le 18 \\ \frac{3}{2}N_s - \delta_s & N_s \ge 90,\ \delta_s > 18 \\ \frac{3}{2}N_s - 2\delta_s & N_s \ge 90,\ \delta_s \le 18. \end{cases} \quad (4)$$

3.3. The Main Steps of Generating an S-Box. According to the above description, we can summarize the detailed steps of generating an S-box as follows.

Step 1. Call the parameters initialization of the BFO, set a speed adjustment parameter C and constant α, substitute the initial values, x_0, y_0, z_0, into (1), iterate N_0 times to obtain x_1, y_1, z_1, and define an integer array **S** of length 256.

Step 2. Take x_1, y_1, z_1, as the initial values, record from $(N_0 + 1)$th value, and mark the following obtained real-value sequences, x_i, y_i, z_i $(i = 1, 2, \ldots)$.

Step 3. Convert these continuous real-value sequences x_i, y_i, z_i, to the corresponding integer sequences by (5), with $\{sx_i, sy_i, sz_i\} \in [0, 255]$, and sort them by (6).

$$\begin{aligned} sx_i &= \lfloor x_i \times 256 \rfloor \\ sy_i &= \lfloor y_i \times 256 \rfloor \\ sz_i &= \lfloor z_i \times 256 \rfloor, \end{aligned} \quad (5)$$

where $\lfloor x \rfloor$ returns the maximum integer less than or equal to x.

$$\begin{aligned} Q &= \{sx_i \ \ sy_i \ \ sz_i\} \\ R &= \{sy_i \ \ sz_i \ \ sx_i\} \\ T &= \{sz_i \ \ sx_i \ \ sy_i\}. \end{aligned} \quad (6)$$

Step 4. Iterate M_0 and record starting from $(M_0 + 1)$th value; if Q_i has appeared in array S_i, discard it; otherwise, save it in S_i. And the S_i box is obtained when the array is filled. So are R_i and T_i. Iterate M_0k and record from the $(M_0k + 1)$th value; if Q_i, has appeared in the array S_i, discard it; otherwise, save it in S_i until the array is filled. So are R_i and T_i. Similarly, iterate M_02k, record from $(M_02k + 1)$th value, and obtain Q_i, R_i, and T_i.

Step 5. Arrange the integer sequence S_i in a $2^{n/2} \times 2^{n/2}$ table, and construct $3k$ S-boxes.

Step 6. Calculate the fitness value of each S-box by (3).

Step 7. Call Steps 3–7 of BFO, except that we use (7) to update the bacteria locations in the tumbling stage.

$$X_i(t+1) = X_i(t) + C \times \text{rand}() * \left(X_i(t) - X_k(t)\right). \quad (7)$$

4. S-Box Evaluation Criteria

To obtain the S-box with desired cryptography properties, many evaluating criteria, such as bijectivity, Nonlinearity, SAC, BIC, DP, and LP, have been designed. We will also use these criteria to test the performance of the proposed S-box in this paper.

4.1. Bijectivity. Adams and Tavares pointed out that if the linear sum of the Boolean function f_i of each component of the designed $n \times n$ S-box was 2^{n-1}, f was then a bijection [44]. Specifically, the expression is as follows:

$$\text{wt}\left(\sum_{i=1}^{n} a_i f_i\right) = 2^{n-1}. \quad (8)$$

Here, $a_i \in \{0, 1\}$, $(a_1, a_2, \ldots, a_n) \neq (0, 0, \ldots, 0)$, wt( ) denotes the Hamming weight.

In fact, a reversible S-box is usually required; especially in a replacement network, the S-box must be bijective.

4.2. Nonlinearity

Definition 1. Let $f(x) : F_2^n \to F_2$ be an n Boolean function; the nonlinearity of $f(x)$ can take the following form:

$$H_f = \min_{l \in L_n} d_H(f, l), \quad (9)$$

where L_n is a set of the whole linear and affine functions and $d_H(f, l)$ denotes the Hamming distance between f and l.

The nonlinearity denoted by the Walsh spectrum can take the following form:

$$N_f = 2^{-n}\left(1 - \max_{\omega \in \text{GF}(2^n)} \left|S_{\langle f \rangle}(\omega)\right|\right). \quad (10)$$

The cyclic spectrum of the function $f(x)$ is

$$S_{\langle f \rangle}(\omega) = 2^{-n} \sum_{x \in \text{GF}(2^n)} (-1)^{f(x) \oplus x \cdot \omega}, \quad (11)$$

where $\omega \in \text{GF}(2^n)$ and $x \cdot \omega$ denotes the dot product of x and ω.

The larger the nonlinearity N_f of the function f, the stronger the ability of its resistance to the linear attacks, and vice versa.

4.3. Strict Avalanche Criterion. Strict avalanche criterion describes a fact that when one bit in the input of Boolean function changes, the changing probability of every bit in its output is 1/2. In practical application, a correlation matrix, the construction method of which refers to literature [45], is always constructed to test the SAC property of the Boolean function.

4.4. Bit Independent Criterion. Given that a Boolean function f_j, f_k $(j \neq k)$ is a two-bit output of an S-box, if $f_j \oplus f_k$ is highly nonlinear and meets the SAC, the correlation coefficient of each output bit pair may be close to 0 when one input bit is inversed. Thus, we can check the BIC of the S-box by verifying whether $f_j \oplus f_k$ $(j \neq k)$ of any two output bits of the S-box meets the nonlinearity and SAC [45].

Table 1: The generated S-box.

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

4.5. Differential Approximation Probability. The differential approximation probability DP_f can reflect the XOR distribution of the input and output of the Boolean function [6], that is, the maximum likelihood of outputting Δy, when the input is Δx,

$$\mathrm{DP}_f = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X \mid f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right), \quad (12)$$

where X denotes a set of all possible inputs and 2^n is the number of elements in the set.

The smaller the DP_f, the stronger the ability of the S-box for fighting against differential cryptanalysis attacks, and vice versa.

4.6. Linear Approximation Probability. Given two randomly selected masks Γx and Γy, we use Γx to calculate the mask of all possible values of an input x and use Γy to calculate the mask of the output values $S(x)$ of the corresponding S-box. After masking the input and the output values, the maximum number of the same results is called the maximum linear approximation that can be computed by the following [46]:

$$\mathrm{LP} = \max_{\Gamma x, \Gamma y \neq 0} \left| \frac{\#\{x \mid x \cdot \Gamma x = S(x) \cdot \Gamma y\}}{2^n} - \frac{1}{2} \right|, \quad (13)$$

where Γx and Γy are the mask values of the input and output, respectively, and $S(x)$ is a set of all possible input values of x, the elements of which are 2^n.

The smaller the LP, the stronger the ability of the S-box for fighting against linear cryptanalysis attacks, and vice versa.

5. Cryptographic Criteria Analyses of Proposed S-Box

To verify the generated S-box, we analyze its cryptographic criteria and compare it with many other S-boxes designed in [6–9, 25–41]. Moreover, our experimental environment is i3 CPU 540 3.07 GHz, memory 2.00 GHz Inter Core, Matlab2012b.

For initialization, we must choose the following parameters, $x_0 = 0.34$, $y_0 = 0.27$, $z_0 = 0.75$, $k_1 = 39.7$, $k_2 = 40.2$, $k_3 = 38.5$, $a = 3.735$, $b = 3.536$, $c = 3.828$, $M_0 = 100$, $k = 5$, $S = 15$, Ned = 3, Nre = 3, Nc = 8, Ns = 4, Ped = 0.25, $C = 0.5$, $\alpha = 0.2$. Calling the proposed algorithm with these parameters, we obtain the S-box that is depicted in Table 1.

5.1. Bijectivity Analysis. Use (8) to calculate the value of the objectivity of the S-box. Since the obtained bijectivity value of the proposed S-box is 128 that is the same as the desired value, we may conclude that our S-box satisfies the bijectivity.

5.2. Nonlinearity Analysis. Use (10) to calculate the nonlinearity of the S-box; Figure 3 depicts the statistical results of the nonlinearity of comparison algorithms. Table 6 shows the detailed nonlinearity results. From these results, we can see that the minimum nonlinearity value of eight Boolean functions of the proposed S-box is 106, and the average value 107.5 is greater than that of the other comparison algorithms. Thus, our S-box may have good nonlinearity and can resist the linear cryptanalysis.

5.3. Strict Avalanche Criterion Analysis. Table 2 depicts the dependence matrix of our S-box, and the SAC comparison results are shown in the third column of Table 6. From these two tables, we may conclude that our S-box satisfies the SAC. For our S-box, its obtained maximum SAC is 0.6094, the minimum is 0.3750, and its average value 0.5093 is close to the desired value 0.5. Moreover, compared with the algorithm

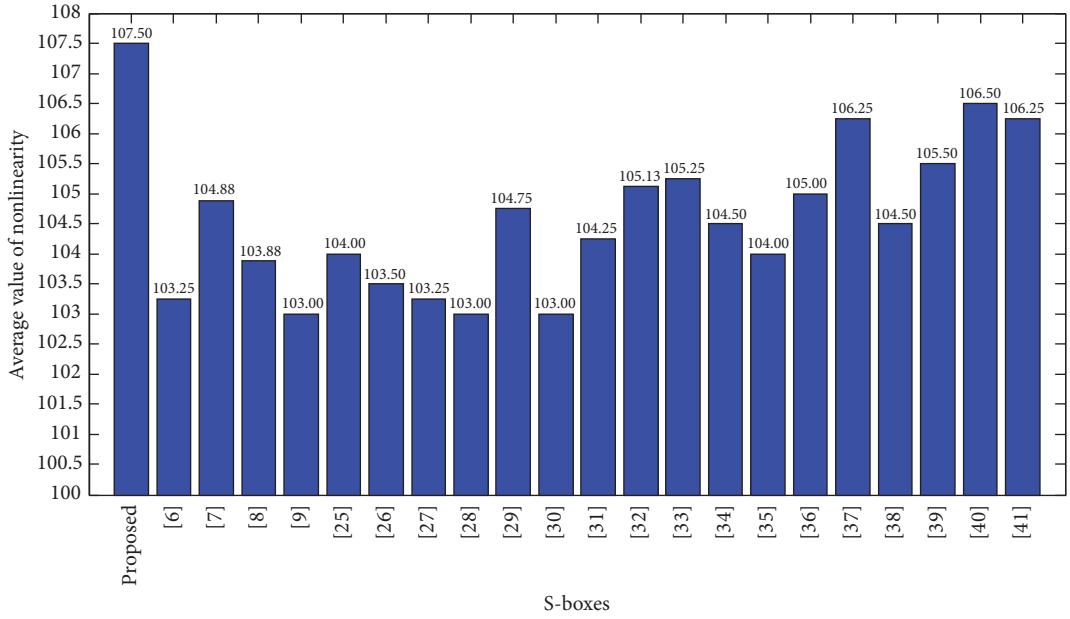


Figure 3: Statistical results of the nonlinearity.

Table 2: The dependence matrix of the generated S-box.

Number	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1)	0.5313	0.4844	0.4844	0.5156	0.4844	0.5469	0.5313	0.5000
(2)	0.4844	0.5469	0.5156	0.4688	0.4844	0.5156	0.5781	0.4844
(3)	0.6094	0.4375	0.4844	0.5625	0.5625	0.4688	0.5000	0.5000
(4)	0.4688	0.4844	0.5000	0.5469	0.5156	0.5156	0.5000	0.5625
(5)	0.4219	0.5469	0.5000	0.5156	0.5000	0.5938	0.5313	0.5000
(6)	0.4375	0.4688	0.5938	0.5000	0.5000	0.5156	0.5000	0.5000
(7)	0.5000	0.4375	0.5000	0.5313	0.5781	0.5156	0.5156	0.5313
(8)	0.5938	0.5000	0.3750	0.5313	0.4844	0.4844	0.5625	0.4531

proposed in [40], the average value of the dependence matrix of the generated S-box is closer to the idea value 0.5.

5.4. Bit Independent Criterion Analysis. Tables 3 and 4 describe the BIC related criterion results of our S-box. The fourth and fifth columns of Table 6 depict the corresponding comparison results of the experimental results. Additionally, the average BIC-nonlinearity value of the proposed S-box is larger than that of S-box generation algorithms in [7, 8, 28, 31]. The average BIC-SAC value of the proposed S-box is larger than that of algorithms in [6–8, 25, 31–33, 37–40]. For our S-box, its average nonlinearity of the $f_j \oplus f_k$ is 103.07; the average correlation matrix is 0.5025 that is close to the desired value 0.5. Therefore, we may obtain that our S-box has good bit independent criterion.

5.5. Differential Approximation Probability Analysis. Table 5 shows the differential probability result of our S-box calculated by (12), and the left of the right column in Table 6 depicts the DP comparison results of the experimental algorithms. Furthermore, Figure 4 depicts statistical results of the DP of comparison algorithms. These results show that the DP of the proposed S-box is not worse than that of the other comparison algorithms. It may indicate that our S-box can resist differential cryptanalysis because our S-box obtains a small differential probability, 0.0390.

5.6. Linear Approximation Probability Analysis. In this subsection, we use (13) to calculate the linear probability of our S-box and compare it with other comparison algorithms. The linear probability results are shown in the right column of Table 6, from which we argue that our S-box may have a good ability to resist linear cryptanalysis. Besides, Figure 5 depicts the statistical results of the LP of comparison algorithms. These results show that the LP of the proposed S-box is not worse than that of algorithms in [25, 26, 28, 30, 32, 34, 37, 38, 41].

Furthermore, Table 6 summarizes the above-mentioned evaluating criteria results of the experimental algorithms. From the table, we may make the following observations: (1) these chaotic based S-boxes have good cryptographic properties. (2) Comprehensively, our S-box, however, may have better encryption performance and can resist many modern cryptanalysis attacks to some extent.

Table 3: BIC-nonlinearity criterion result of the generated S-box.

Number	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1)	0	102	104	100	102	104	98	106
(2)	102	0	102	102	100	102	106	106
(3)	104	102	0	102	102	106	108	100
(4)	100	102	102	0	106	102	94	104
(5)	102	100	102	106	0	104	108	108
(6)	104	102	106	102	104	0	100	104
(7)	98	106	108	94	108	100	0	104
(8)	106	106	100	104	108	104	104	0

Table 4: BIC-SAC criterion result of the generate S-box.

Number	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1)	—	0.5117	0.4961	0.5176	0.4941	0.4922	0.5000	0.4961
(2)	0.5117	—	0.5117	0.4922	0.4941	0.4883	0.4863	0.5137
(3)	0.4961	0.5117	—	0.5020	0.5000	0.5371	0.5000	0.5039
(4)	0.5176	0.4922	0.5020	—	0.4922	0.4707	0.5156	0.4980
(5)	0.4941	0.4941	0.5000	0.4922	—	0.5020	0.5156	0.5059
(6)	0.4922	0.4883	0.5371	0.4707	0.5020	—	0.4941	0.5254
(7)	0.5000	0.4863	0.5000	0.5156	0.5156	0.4941	—	0.5137
(8)	0.4961	0.5137	0.5039	0.4980	0.5059	0.52540	0.5137	—

Table 5: Differential approach table for the generated S-box.

Number	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
(1)	6	6	6	6	6	8	6	8	8	8	6	6	6	6	6	6
(2)	6	6	6	6	8	8	8	6	6	8	6	6	6	6	8	10
(3)	6	8	6	6	8	6	6	6	6	6	6	6	8	8	6	8
(4)	8	6	6	10	4	6	6	4	8	6	6	6	8	6	6	6
(5)	6	8	6	6	6	8	6	6	10	6	6	6	6	6	6	8
(6)	6	6	6	8	8	8	8	8	8	10	6	6	6	10	6	10
(7)	8	6	8	8	8	6	6	6	8	8	6	6	4	6	8	8
(8)	8	10	8	8	4	6	6	6	8	6	6	6	10	8	6	6
(9)	6	6	6	6	8	8	8	8	6	8	6	6	6	8	8	8
(10)	8	8	6	8	6	6	6	6	6	6	6	6	8	6	6	4
(11)	6	6	8	6	6	6	8	4	6	8	6	6	6	6	6	8
(12)	6	6	6	6	6	8	8	6	6	4	6	6	6	6	6	6
(13)	8	8	6	8	6	8	8	6	6	8	10	6	6	8	6	6
(14)	6	8	6	8	8	8	6	6	6	6	6	8	6	6	6	6
(15)	8	6	6	6	6	6	6	6	6	6	8	6	6	6	6	6
(16)	8	8	6	6	6	6	6	8	6	6	8	6	6	6	8	—

6. Conclusion

This paper is concerned with developing a novel S-box generation algorithm, chaotic S-box based on the intertwining logistic map and bacterial foraging optimization. This algorithm firstly generates a set of intermediate S-boxes by iterating the chaotic map and then utilizes a bacterial foraging optimization algorithm to search for an S-box with good performance. During the search process, the nonlinearity and differential uniformity are involved to design the evaluation function. Experimental results investigate that the proposed S-box generation algorithm may generate an S-box with good cryptography characteristics. Since there are many other factors that affect the performance of S-boxes, for example, SAC, BIC, and LP, the investigation of applying them to design the evaluation function of the bacterial foraging optimization algorithm may be worth studying in the near future.

List of Abbreviations

S-box: Substitution box
BFO: Bacterial foraging optimization
SAC: Strict avalanche criterion

Table 6: Cryptanalysis comparison results of S-boxes.

S-boxes	Nonlinearity			SAC			BIC-nonlinearity	BIC-SAC	DP	LP
	Min	Max	Avg.	Min	Max	Avg.				
Proposed	106	110	107.5	0.3750	0.6094	0.5093	103.07	0.5025	0.0390	0.1406
Ref. [6]	100	108	103.250	0.3750	0.5938	0.5059	104.29	0.5031	0.0468	0.1250
Ref. [7]	103	109	104.875	0.3984	0.5703	0.4966	102.96	0.5044	0.0390	0.1328
Ref. [8]	101	108	103.875	0.3906	0.5781	0.5059	102.68	0.4958	0.0390	0.1328
Ref. [9]	100	106	103	0.4219	0.6094	0.5000	103.14	0.5024	0.0546	0.1328
Ref. [25]	102	106	104	0.3750	0.6094	0.4980	103.29	0.4971	0.0390	0.1484
Ref. [26]	96	108	103.500	0.3906	0.5859	0.4939	103.64	0.4992	0.0390	0.1523
Ref. [27]	100	106	103.250	0.4219	0.5938	0.5049	103.71	0.5010	0.0390	0.1328
Ref. [28]	96	106	103	0.3906	0.6250	0.5039	100.36	0.5010	0.0390	0.1484
Ref. [29]	102	108	104.750	0.3906	0.5938	0.5056	104.07	0.5022	0.0468	0.1250
Ref. [30]	98	108	103	0.4063	0.5938	0.5012	104.07	0.4989	0.0468	0.1484
Ref. [31]	98	108	104.25	0.2813	0.6094	0.4954	102.86	0.5048	0.0468	0.1406
Ref. [32]	103	109	105.125	0.4141	0.6094	0.5061	103.68	0.4983	0.0390	0.1562
Ref. [33]	102	108	105.250	0.4063	0.5781	0.5059	104.29	0.5029	0.0468	0.1250
Ref. [34]	100	108	104.5	0.4219	0.6094	0.4978	103.64	0.5010	0.0468	0.1406
Ref. [35]	100	106	104	0.3750	0.6250	0.4946	103.21	0.5019	0.0390	0.1328
Ref. [36]	98	108	105	0.4219	0.5781	0.4917	106.07	0.4997	0.1250	0.1171
Ref. [37]	104	110	106.25	0.4219	0.5938	0.5039	103.36	0.5059	0.0390	0.1406
Ref. [38]	101	107	104.5	0.4219	0.5781	0.4963	103.29	0.4938	0.0390	0.1406
Ref. [39]	100	110	105.50	0.4063	0.6094	0.5010	103.79	0.5036	0.0468	0.1328
Ref. [40]	104	110	106.5	0.4375	0.6406	0.5120	104.57	0.5042	0.0390	0.1328
Ref. [41]	104	108	106.25	0.3906	0.5625	0.4949	103.64	0.5007	0.0390	0.1406

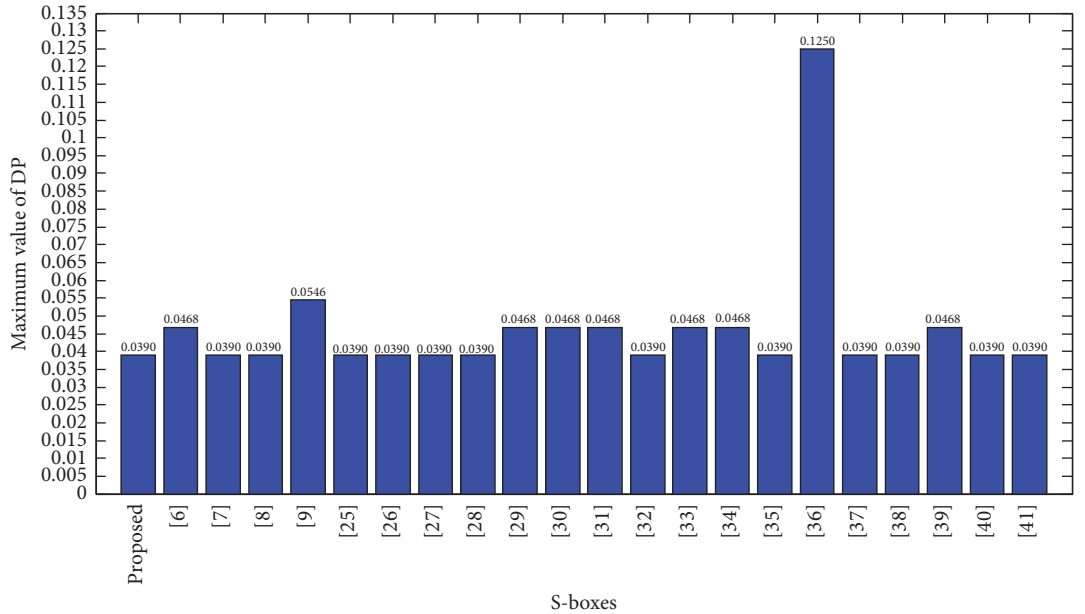


Figure 4: Statistical results of the DP.

BIC: Bit independent criterion
DP: Differential approximation probability
LP: Linear approximation probability.

Conflicts of Interest

The authors declare no conflicts of interest.

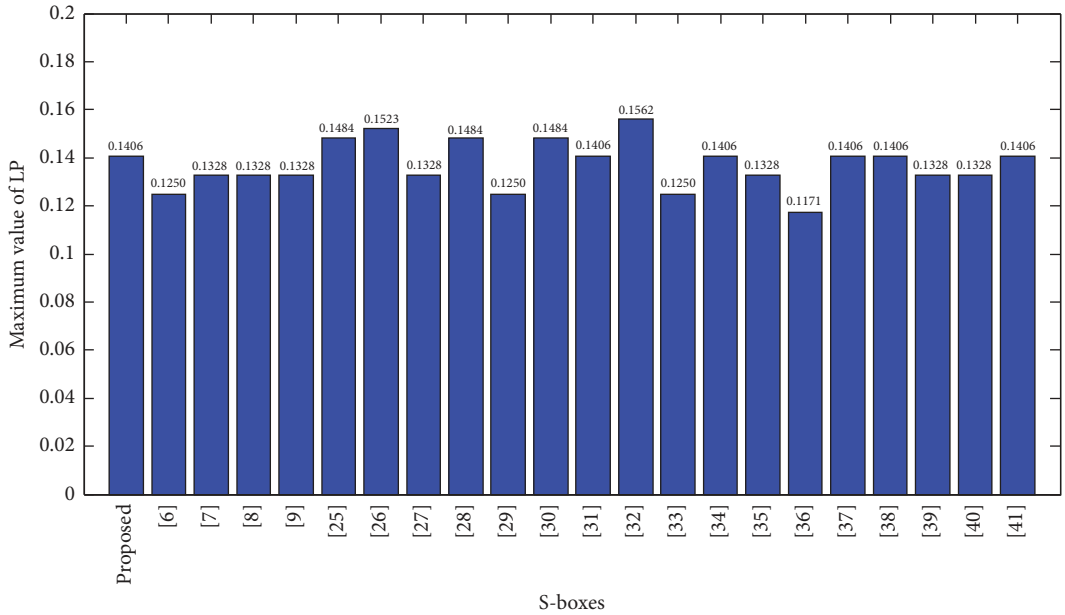


Figure 5: Statistical results of the LP.

Acknowledgments

The authors would like to acknowledge the financial supports by the National Natural Science Foundation of China (nos. 60603092, 60975042, and 51472066).

References

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

[2] I. Hussain, T. Shah, and M. A. Gondal, "Application of S-box and chaotic map for image encryption," *Mathematical and Computer Modelling*, vol. 57, no. 9-10, pp. 2576–2579, 2013.

[3] X. Zhang, Y. Mao, and Z. Zhao, "An efficient chaotic image encryption based on alternate circular S-boxes," *Nonlinear Dynamics*, vol. 78, no. 1, pp. 359–369, 2014.

[4] H. Liu, A. Kadir, and Y. Niu, "Chaos-based color image block encryption scheme using S-box," *AEÜ - International Journal of Electronics and Communications*, vol. 68, no. 7, pp. 676–686, 2014.

[5] I. Hussain and M. A. Gondal, "An extended image encryption using chaotic coupled map and S-box transformation," *Nonlinear Dynamics*, vol. 76, no. 2, pp. 1355–1363, 2014.

[6] G. Jakimoski and L. c. Kocarev, "Chaos and cryptography: block encryption ciphers based on chaotic maps," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 2, pp. 163–169, 2001.

[7] G. Tang, X. Liao, and Y. Chen, "A novel method for designing S-boxes based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 23, no. 2, pp. 413–419, 2005.

[8] G. Tang and X. Liao, "A method for designing dynamical S-boxes based on discretized chaotic map," *Chaos, Solitons & Fractals*, vol. 23, no. 5, pp. 1901–1909, 2005.

[9] G. Chen, Y. Chen, and X. Liao, "An extended method for obtaining S-boxes based on three-dimensional chaotic baker maps," *Chaos, Solitons & Fractals*, vol. 31, no. 3, pp. 571–579, 2007.

[10] S. Behnia, A. Akhshani, H. Mahmodi, and A. Akhavan, "A novel algorithm for image encryption based on mixture of chaotic maps," *Chaos, Solitons & Fractals*, vol. 35, no. 2, pp. 408–419, 2008.

[11] L. Nian-sheng, "Pseudo-randomness and complexity of binary sequences generated by the chaotic system," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 2, pp. 761–768, 2011.

[12] X.-Y. Wang and X. Qin, "A new pseudo-random number generator based on CML and chaotic iteration," *Nonlinear Dynamics*, vol. 70, no. 2, pp. 1589–1592, 2012.

[13] M. Francois, T. Grosges, D. Barchiesi, and R. Erra, "Pseudo-random number generator based on mixing of three chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 887–895, 2014.

[14] J. Khan, J. Ahmad, and S. O. Hwang, "An efficient image encryption scheme based on: Henon map, skew tent map and S-Box," in *Proceedings of the 6th International Conference on Modeling, Simulation, and Applied Optimization, ICMSAO 2015*, pp. 1–6, May 2015.

[15] M. Ahmad, H. Haleem, and P. M. Khan, "A new chaotic substitution box design for block ciphers," in *Proceedings of the 1st International Conference on Signal Processing and Integrated Networks, SPIN 2014*, pp. 255–258, February 2014.

[16] J. Peng, S. Jin, L. Lei, and X. Liao, "Construction and analysis of dynamic S-boxes based on spatiotemporal chaos," in *Proceedings of the IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pp. 274–278, 2012.

[17] Y. Wang, K.-W. Wong, C. Li, and Y. Li, "A novel method to design S-box based on chaotic map and genetic algorithm," *Physics Letters A*, vol. 376, no. 6-7, pp. 827–833, 2012.

[18] S. M. Abd-Elazim and E. S. Ali, "Power system stability enhancement via bacteria foraging optimization algorithm,"

Arabian Journal for Science and Engineering, vol. 38, no. 3, pp. 599–611, 2013.

[19] E. S. Ali and S. M. Abd-Elazim, "Bacteria foraging optimization algorithm based load frequency controller for interconnected power system," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 3, pp. 633–638, 2011.

[20] S. M. Abd-Elazim and E. S. Ali, "Synergy of particle swarm optimization and bacterial foraging for TCSC damping controller design," *WSEAS Transactions on Power Systems*, vol. 8, no. 2, pp. 74–84, 2013.

[21] C. Sur and A. Shukla, "Discrete bacteria foraging optimization algorithm for vehicle distribution optimization in graph based road network management," *Advances in Intelligent Systems and Computing*, vol. 235, pp. 351–358, 2014.

[22] M. Tripathy and S. Mishra, "Bacteria foraging-based solution to optimize both real power loss and voltage stability limit," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 240–248, 2007.

[23] I. S. Sam, P. Devaraj, and R. S. Bhuvaneswaran, "An intertwining chaotic maps based image encryption scheme," *Nonlinear Dynamics*, vol. 69, no. 4, pp. 1995–2007, 2012.

[24] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[25] G. Chen, "A novel heuristic method for obtaining S-boxes," *Chaos, Solitons & Fractals*, vol. 36, no. 4, pp. 1028–1036, 2008.

[26] M. Asim and V. Jeoti, "Efficient and simple method for designing chaotic S-boxes," *ETRI Journal*, vol. 30, no. 1, pp. 170–172, 2008.

[27] F. Özkaynak and A. B. Özer, "A method for designing strong S-Boxes based on chaotic Lorenz system," *Physics Letters A*, vol. 374, no. 36, pp. 3733–3738, 2010.

[28] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, and I. Hussain, "A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems," *Nonlinear Dynamics*, vol. 70, no. 3, pp. 2303–2311, 2012.

[29] I. Hussain, T. Shah, and M. A. Gondal, "A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm," *Nonlinear Dynamics*, vol. 70, no. 3, pp. 1791–1794, 2012.

[30] M. Khan, T. Shah, H. Mahmood, and M. A. Gondal, "An efficient method for the construction of block cipher with multi-chaotic systems," *Nonlinear Dynamics*, vol. 71, no. 3, pp. 489–492, 2013.

[31] M. Khan, T. Shah, and M. A. Gondal, "An efficient technique for the construction of substitution box with chaotic partial differential equation," *Nonlinear Dynamics*, vol. 73, no. 3, pp. 1795–1801, 2013.

[32] F. Özkaynak and S. Yavuz, "Designing chaotic S-boxes based on time-delay chaotic system," *Nonlinear Dynamics*, vol. 74, no. 3, pp. 551–557, 2013.

[33] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "A novel method for designing nonlinear component for block cipher based on TD-ERCS chaotic sequence," *Nonlinear Dynamics*, vol. 73, no. 1-2, pp. 633–637, 2013.

[34] M. Khan and T. Shah, "An efficient construction of substitution box with fractional chaotic system," *Signal, Image and Video Processing*, vol. 9, no. 6, pp. 1335–1338, 2015.

[35] M. Khan and T. Shah, "A construction of novel chaos base nonlinear component of block cipher," *Nonlinear Dynamics*, vol. 76, no. 1, pp. 377–382, 2014.

[36] M. Khan and T. Shah, "A novel construction of substitution box with Zaslavskii chaotic map and symmetric group," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 28, no. 4, pp. 1509–1517, 2015.

[37] 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 Dynamics*, vol. 87, no. 2, pp. 1081–1094, 2017.

[38] F. Özkaynak, V. Çelik, and A. B. Özer, "A new S-box construction method based on the fractional-order chaotic Chen system," *Signal, Image and Video Processing*, vol. 11, no. 4, pp. 659–664, 2017.

[39] A. Belazi and A. A. A. El-Latif, "A simple yet efficient S-box method based on chaotic sine map," *Optik - International Journal for Light and Electron Optics*, vol. 130, pp. 1438–1444, 2017.

[40] T. Farah, R. Rhouma, and S. Belghith, "A novel method for designing S-box based on chaotic map and Teaching–Learning-Based Optimization," *Nonlinear Dynamics*, vol. 88, no. 2, pp. 1059–1074, 2017.

[41] Y. Tian and Z. Lu, "S-box: L-L cascade chaotic map and line map," in *Proceedings of the International Conference on Image and Graphics*, pp. 297–309, Springer, Cham, Switzerland, 2015.

[42] X. J. Bi, L. Sheng, and J. Chen, "S-box optimization design based on improved particle swarm optimization algorithm," *Computer Engineering*, vol. 37, no. 23, pp. 149–151, 2011 (Chinese).

[43] B. X. Lu, *Research on Optimization and Improvement of S-box in Block Cipher*, Southwest Jiaotong University, Sichuan, China, 2013.

[44] C. Adams and S. Tavares, "Good S-boxes are easy to find," in *Advances in Cryptology—CRYPTO'89 Proceedings*, pp. 612–615, Springer, New York, NY, USA, 1990.

[45] A. F. Webster and S. E. Tavares, "On the design of S-boxes," in *Advances in Cryptology—CRYPTO'85 Proceedings*, pp. 523–534, Springer, Berlin, Germany, 1986.

[46] M. Matsui, "Linear cryptanalysis method of DES cipher," in *Advances in Cryptology, Proc. Eurocrypt'93, Proceedings*, pp. 386–397, Springer, Berlin, Germany, 1994.