Journal of Mathematical Modeling
Vol. 11, No. 2, 2023, pp. 277–300. Research Article

JMM

Application of S-Boxes based on the chaotic Hindmarsh-Rose system for image encryption

Meisam Bavand Savadkouhi[†], Hamid Haj Seyyed Javadi[‡], Mohammad Akbari Tootkaboni[§*]

[†]*Department of Mathematics, Shahed University, Tehran, Iran*
[‡] *Department of Mathematics and Computer Sciences, Shahed University, Tehran, Iran* [§]*Department of Pure Mathematics, Faculty of Mathematical Sciences, University of Guilan, Iran*
Email(s): meisam.bavand@shahed.ac.ir, h.s.javadi@shahed.ac.ir, tootkaboni.akbari@gmail.com

Abstract. The substitution box (S-Box) is a critical component in symmetric cipher algorithms. In this paper, we choose the Hindmarsh-Rose system to generate chaotic S-Boxes. We propose two S-Boxes based on the rotation algorithm relative to the rows (or columns) and the other based on the Zigzag transformation. The performance of the new S-Boxes is evaluated by bijective, nonlinearity, strict avalanche criterion (SAC), output bits independence criterion (BIC), differential approximation probability, linear approximation probability, and algebraic degree. The analysis results show that the presented S-Boxes have suitable cryptographic properties. Also, an image encryption algorithm based on two proposed S-Boxes, and a chaotic Hindmarsh-Rose system are presented. Experimental results show the recommended method has attained good security, and the suggested plan has potent resistance to different attacks.

Keywords: S-Box, chaotic Hindmarsh-Rose system, image encryption, image analysis.
AMS Subject Classification 2010: 34A34, 65L05.

1 Introduction

S-Box is an important nonlinear component in cryptography, which is defined as follows: Let the S-Box with an n-binary input into m-binary output be a nonlinear mapping, denoted by S. Therefore $S : GF(2)^n \to GF(2)^m$ is defined by $S(x) = y$ for each $x = (x_1, x_2, \ldots, x_n) \in GF(2)^n$ and for some $y = (y_1, y_2, \ldots, y_m) \in GF(2)^m$, where $GF(2) = \{0, 1\}$ is a Galois field. Clearly, S can be considered as a vectorial Boolean function consisting of m individual Boolean functions $f_1, f_2, \ldots, f_m$, where f_i :

*Corresponding author.
Received: 6 November 2022/ Revised: 24 January 2023 / Accepted: 7 February 2023
DOI: 10.22124/JMM.2023.23199.2068

© 2023 University of Guilan

$GF(2)^n \to GF(2)$ and $f_i(x) = y_i$ for $i \in \{1,2,\ldots,m\}$. These functions are called component Boolean functions of the S.

A significant part of the time to design or analyze the symmetric cipher algorithm is related to the S-Box. Since, except for the S-Box modular multiplication, and modular addition, the rest of the components of each symmetric cipher algorithm are linear, therefore any weakness in the design of the S-Box will lead to the vulnerability of the encryption system. The main purpose of the S-Box design is to create confusion between the ciphertext and the secret key. Since the ciphertext is related to the plaintext and the secret key in a complex way, we need suitable confusion in the design of the S-Box [28]. To prevent various attacks on symmetric cipher algorithms, the S-Box must-have good properties such as high algebraic degree, high nonlinearity, good resistance against differential and linear attacks, etc. Therefore, as it is known, the S-Box plays a fundamental role in cryptography.

S-Boxes are produced in two general ways, random and algebraic. Producing random S-Box can be done in different ways. In this paper, we will produce S-Box based on a chaotic system. The design of the S-Box based on the chaotic system is an example of the application of Chaos theory in cryptography. Random behavior, ergodic behavior, and sensitivity to initial conditions in Chaos theory overlap with the basic good properties of cryptography such as avalanche, diffusion, and confusion, and this relationship is used in the design of modern cryptographic systems [6]. Adam and Tavares proposed criteria for designing an S-Box of size $n \times n$ [2].

With the development of technology, the use of digital images in society increased widely. For this reason, the security of digital images in the transmission process has become one of the main topics in computer science. Due to the large amount of data in the image and the high correlation between adjacent pixels, it prevents the use of usual algorithms in image encryption. The interested reader could refer to [15], [26], and [27] to use these methods.

In this paper, we have used the continuous time chaotic Hindmarsh-Rose system in the S-Box design. For this task, by sampling the path, the system is discretized for certain time steps. After the discretization process by using two algorithms consisting of Zigzag transformation and an innovative method, two 8×8 S-Boxes are produced. By analyzing the outputs of the designed S-Boxes, we will show that the produced S-Boxes have good encryption properties compared to other chaotic S-Boxes.

The presented paper is organized as follows: In the second section, the mathematical structure of the Hindmarsh - Rose chaotic system is presented. In the third section, two different algorithms are presented that generate the S-Boxes using the chaotic system. The criteria used in the performance analysis of the designed S-Boxes are defined in section four and its comparison with other chaotic S-Boxes is presented in this section. In section five, the algorithm that is used to encrypt an image file is presented. It should be noted that in this algorithm two S-Boxes produced in the third section in the developed construction of Lai-Massey structure are used. In this section, the security analysis of the proposed encryption on an image file has been investigated, too. At the end of the paper, the conclusion is also discussed.

2 The chaotic Hindmarsh-Rose system

The Hindmarsh-Rose system [18] is a three-dimensional differential equation system, which is defined as follows

$$\frac{dx}{dt} = -x^3 + ax^2 + y - z + I, \qquad \frac{dy}{dt} = bx^2 - y + 1, \qquad \frac{dz}{dt} = r[s(x + 1.6) - z]. \tag{1}$$

In this formula, $a,b,r,s,I \in \mathbb{R}$ are the control parameters, and, $(x,y,z) \in \mathbb{R}^3$ are the state variables of the Hindmarsh-Rose system. In this paper, control parameters and initial conditions for the Hindmarsh-Rose system are chosen as follows: $a = 3$, $b = -5$, $r = 0.006$, $s = 4$, $I = 3.2$, $x_0 = 0$, $y_0 = 0.05$, and $z_0 = 0.02$, that satisfy the chaotic system. The attractor of the Hindmarsh-Rose is seen in Figure 1.

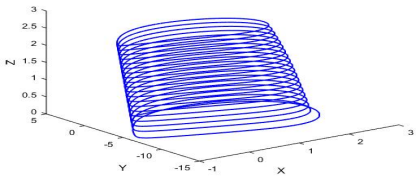

Figure 1: The Hindmarsh-Rose attractor ($a = 3$, $b = -5$, $r = 0.006$, $s = 4$, $I = 3.2$).

3 The chaos-based algorithms for designing S-Box

Algorithms 1 and 2 of the proposed method for constructing a chaos-based S-Box are explained below.

- **Algorithm 1:**
  - **Step A.1:** The Hindmarsh-Rose system trajectory is obtained by applying the selected initial conditions and chaotic parameter values. We calculate the unknowns of the system by the four-step Runge-Kutta method.
  - **Step A.2:** Iterate the chaotic Hindmarsh-Rose system, 10000 times to eliminate harmful effects of transient processes.
  - **Step A.3:** With Hindmarsh-Rose's chaotic system, we generate a sequence $X = \{x_i\}_{i=1}^{3\times 10^5}$, then we select an ordered set according to $mod(t+h,256)$ from the sequence X as follows $Y_h = \{(mod(t+h,256),x_i) : i = 4t^2+h, t \in \{0,1,,255\}\}$ which h is the stage number and $mod(x,256)$ returns the remainder of x divided by 256.
  - **Step A.4:** We sort each set of Y_h in ascending order concerning x_i, which results in the set of Z_h $Z_h = \{(j,m_j) : (j,m_j) \in Y_h\}$. Then we select the S-Box of the number h from Z_h $S-Box_h = \{(w,j) : w \in \{0,1,,255\}, Z_h(w) = j\}$.
  - **Step A.5:** After generating S-Box, we place the 256 elements in a 16×16 table. We number the rows of each table from top to bottom with zero to fifteen. Then we rotate each row to the left by the number of the row.
  - **Step A.6:** After the rows, we now number each table's columns from left to right with zero to fifteen. Then we rotate each column according to the number of that column from top to bottom.
- **Algorithm 2:**
  - We use Steps A.1, A.2, A.3, and A.4. Then, replace Steps A.5 and A.6 with Step A.7.
  - **Step A.7:** After generating S-Box, we put its 256 elements in a 16×16 table. Then, an operation is performed on this table in the dimensions of 16×16, as in Table 1. This operation is also called Zigzag transformation.

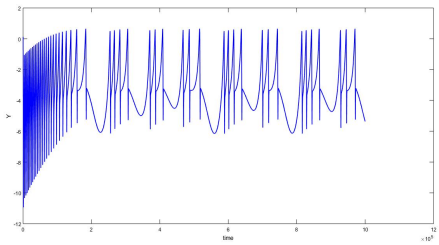

Figure 2: The change of Y data obtained from Eq. (2.1) with to time.

In this paper, the basic conditions for the S-Box design are $x = 0, y = 0.05$, and $z = 0.02$. The system trajectory is generated for 100,000 data. First, 10,000 data are discarded to die out the transient of the system. The data along the Y-axis is used to obtain the S-Box values. The change of system trajectory along the y-axis to time is given in Figure 2. Chaotic S-Boxes obtained from Algorithms 1 and 2 are presented in Tables 2 and 3. Until the end of the paper, we will denote the generated S-Boxes by S_1 and S_2, respectively.

4 The performance analysis of chaotic S-Box

Seven properties are selected to design cryptographically strong S-Boxes. These are bijective , nonlinearity, SAC, BIC, differential and linear approximation probability, and algebraic degree.

4.1 Bijective property: An $n \times n$ S-Box is called bijective if all its n individual Boolean functions have an equal number of 0 and 1. As a result, all 2^n output values of the S-Box are distinct and are in the range $[0, 2^n - 1]$. If the Hamming weight of the linear combination of all Boolean functions f_i of the designed $n \times n$ S-Box is equal to 2^{n-1}, then the S-Box is bijective. That mathematically, it is defined as $wt(a_1 f_1 \oplus a_2 f_2 \oplus \cdots \oplus a_n f_n) = 2^{n-1}$, so that $a_i \in \{0,1\}, (a_1, \ldots, a_n) \neq (0,,0)$. Each Boolean function f_i needs to have an equal number of 0 and 1 that leads to no information leakage to the attacker, [13]. According to the tests that have been done, the bijective property of S_1 and S_2 has been confirmed.

4.2 Nonlinearity criterion: As we know, it is important to design an S-Box that has a high nonlinearity property for a cryptographic system. Because the nonlinearity of S-Box causes uncertainty in the output, that in turn will cause good confusion and resistance of the cipher against linear attack. The bitwise dot product x and w is denoted by $\langle x, w \rangle$, i.e, $\langle x, w \rangle = x_1 w_1 \oplus \cdots \oplus x_n w_n$. Also, $S_{(f)}(w)$ is the Walsh spectrum of Boolean function f, described by $S_{(f)}(w) = \sum_{x \in GF(2)^n} (-1)^{f(x) \oplus \langle x, w \rangle}$, for each $w \in GF(2)^n$. Nonlinearity of f can be shown by the Walsh spectrum, $N_f = 2^{n-1}(1 - 2^{-n} \max_{w \in GF(2)^n} |S_{(f)}(w)|)$. Nonlinearity of 8 component Boolean functions of the S-Box S_1 can be shown as 106, 106, 104, 106, 108,

Table 1: The Zigzag Transform

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

$\xrightarrow{\text{Zigzag}}$

0	4	1	2
5	8	12	9
6	3	7	10
13	14	11	15

Table 2: The proposed chaotic S-Box by Algorithm 1.

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

Table 3: The proposed chaotic S-Box by Algorithm 2.

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

108, 104, 106. Nonlinearity of 8 component Boolean functions of the S-Box S_2 can be shown as 106, 106, 108, 106, 108, 108, 108, 106.

4.3 Strict avalanche criterion: The strict avalanche criterion was first published by Webster and Tavares [33]. If a function satisfies the strict avalanche criterion, it means that changing each bit of the input vector should change all the bits of the output vector with half probability. An efficient method is presented in [2] that checks whether an S-Box satisfies the strict avalanche criterion or not and computes a dependency matrix and a mean value for the S-Box. The mean values of the S-Boxes S_1 and S_2 are 0.497070 and 0.495850, respectively. Calculated mean values are very close to the ideal average value

of 0.5000. The dependence matrix for the S-Boxes S_1 and S_2 are given in Tables 4 and 5.

4.4 Output bits independence criterion: The bit independence is another criterion for the design of S-Boxes and security in the cryptosystem. This criterion was first presented by Adams and Tavares [2]. An S-Box f applies in the BIC condition if for each $i, j, k \in \{1, 2, \ldots, n\}$ with $j \neq k$ the inversion of the input bit i causes the output bits j and k can be changed independently. To measure the degree of independence between the couples of the avalanche variable, the correlation coefficient between the couples will calculate. Let f_i and f_j be two distinct Boolean functions which are the two-bits output of an S-Box. we can analyze the BIC of an S-Box by verifying whether $f_i \oplus f_j$ of both output bits has nonlinearity property and also the truth condition in SAC. Therefore, the BIC can be obtained by computing the nonlinearity and SAC of all 28 functions $f_i \oplus f_j$ for each 8×8 S-Box. The obtained results for S_1 are given in Tables 6 and 8, respectively. The average value of BIC compared to nonlinearity and SAC for S_1 is 103.3571 and 0.4993, respectively. Also, the obtained results for S_2 are given in Tables 7 and 9, respectively. The average BIC compared to nonlinearity and SAC for S_2 is 104.857 and 0.5022. The obtained values confirm the good behavior of S_1 and S_2 for the BIC criterion.

4.5 Differential approximation probability: The measure of differential uniformity is one of the indicators to check the resistance of S-Box against differential attack [9]. The differential approximation probability DP for an S-Box f is defined as follows

$$DP_f = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X : f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right),$$

where X is the set of all possible input values and 2^n is the number of its elements. The differential approach table for S_1 and S_2 is given in Tables 10 and 11. In these Tables the largest of its elements is 10.

4.6 Linear approximation probability: In 1993, linear cryptanalysis for breaking the DES block cipher was first proposed by Matsui. This analysis approximates the relationship between input, output, and secret key. Matsui's definition, the linear approximation probability LP of an S-Box is as follows

$$LP_f = \max_{\Gamma x, \Gamma y \neq 0} \left| \frac{\#\{x \in X : x \cdot \Gamma x = S(x) \cdot \Gamma y\}}{2^n} - \frac{1}{2} \right|,$$

where Γx and Γy are the input and output masks, respectively, and X is the set of all possible inputs x, whose size is 2^n for an $n \times n$ S-Box. For an S-Box to be resistant to attack, the value of linear approximation probability should be as low as possible. The linear approach table for S-Boxes S_1 and S_2 are given in Tables 12 and 13. It can be seen in the Tables, the largest of its elements is 30.

4.7 Algebraic degree: The concept of algebraic degree of Boolean function can be generalized for S-Boxes as well, and based on it, this is possible to analyze the strength of resistance of S-Boxes against differential attacks of algebraic attacks, and cubic attacks, [10, 11]. To determine the algebraic degree of Boolean function with n variables f, first the ANF of that Boolean function is obtained, in which the largest degree of monomial indicates the algebraic degree of the Boolean function. An $n \times n$ S-Box consists of n individual Boolean functions f_i, that $1 \leq i \leq n$. The algebraic degree of S-Box is the minimum degree of all individual Boolean functions f_i, which is denoted by $Deg(S)$ and is defined as follows: $Deg(f) = \min(deg(f_1), deg(f_2), \ldots, deg(f_n))$. The algebraic degree $n-1$ corresponds to the upper bound for the $n \times n$ bijective S-Box. In Table 14, $f_{10}, \ldots, f_{17}$ are the individual Boolean functions of S-Box S_1, and $f_{20}, \ldots, f_{27}$ are the individual Boolean functions of S-Box S_2. Using cryptographic properties, a performance comparison of S-Boxes S_1, S_2, and other S-Boxes are given in Table 15.

Table 4: The dependence matrix for S_1.

0.53125	0.421875	0.5	0.53125	0.453125	0.53125	0.484375	0.515625
0.546875	0.59375	0.5625	0.515625	0.5	0.40625	0.5	0.515625
0.4375	0.453125	0.53125	0.46875	0.484375	0.4375	0.484375	0.546875
0.546875	0.484375	0.5	0.546875	0.53125	0.453125	0.515625	0.53125
0.5	0.453125	0.421875	0.5	0.4375	0.53125	0.5	0.46875
0.4375	0.546875	0.390625	0.5	0.40625	0.484375	0.484375	0.5
0.5	0.515625	0.484375	0.453125	0.484375	0.5	0.546875	0.5625
0.546875	0.5	0.578125	0.484375	0.46875	0.53125	0.53125	0.5

Table 5: The dependence matrix for S_2

0.46875	0.453125	0.421875	0.453125	0.46875	0.515625	0.53125	0.546875
0.484375	0.53125	0.5	0.5	0.5	0.515625	0.4375	0.484375
0.546875	0.484375	0.453125	0.484375	0.46875	0.484375	0.4375	0.484375
0.5	0.578125	0.484375	0.5	0.5625	0.453125	0.4375	0.546875
0.546875	0.5	0.4375	0.53125	0.5	0.5	0.46875	0.46875
0.484375	0.46875	0.46875	0.484375	0.53125	0.5	0.484375	0.484375
0.546875	0.53125	0.5625	0.4375	0.515625	0.59375	0.46875	0.484375
0.625	0.515625	0.515625	0.484375	0.46875	0.46875	0.453125	0.484375

Table 6: BIC- nonlinearity criterion for S_1.

–	104	106	106	106	102	102	106
104	–	102	106	102	106	102	104
106	102	–	104	104	100	100	100
106	106	104	–	102	100	104	104
106	102	104	102	–	104	106	106
102	106	100	100	104	–	106	102
102	102	100	104	106	106	–	98
106	104	100	104	106	102	98	–

Table 7: BIC- nonlinearity criterion for S_2.

–	110	108	104	104	106	104	104
110	–	106	104	106	104	106	104
108	106	–	106	104	106	102	106
104	104	106	–	102	106	106	98
104	106	104	102	–	106	104	106
106	104	106	106	106	–	102	106
104	106	102	106	104	102	–	106
104	104	106	98	106	106	106	–

Table 8: BIC- SAC criterion for S_1.

–	0.501953	0.500000	0.527344	0.494141	0.490234	0.494141	0.513672
0.501953	–	0.501953	0.500000	0.490234	0.503906	0.486328	0.519531
0.500000	0.501953	–	0.492188	0.501953	0.513672	0.470703	0.500000
0.527344	0.500000	0.492188	–	0.492188	0.494141	0.507813	0.488281
0.494141	0.490234	0.501953	0.492188	–	0.482422	0.478516	0.511719
0.490234	0.503906	0.513672	0.494141	0.482422	–	0.527344	0.511719
0.494141	0.486328	0.470703	0.507813	0.478516	0.527344	–	0.484375
0.513672	0.519531	0.500000	0.488281	0.511719	0.511719	0.484375	–

Table 9: BIC- SAC criterion for S_2.

–	0.515625	0.486328	0.521484	0.515625	0.500000	0.507813	0.513672
0.515625	–	0.480469	0.503906	0.523438	0.503906	0.509766	0.521484
0.486328	0.480469	–	0.509766	0.511719	0.505859	0.476563	0.478516
0.521484	0.503906	0.509766	–	0.488281	0.521484	0.492188	0.498047
0.515625	0.523438	0.511719	0.488281	–	0.529297	0.482422	0.498047
0.500000	0.503906	0.505859	0.521484	0.529297	–	0.507813	0.482422
0.507813	0.509766	0.476563	0.492188	0.482422	0.507813	–	0.476563
0.513672	0.521484	0.478516	0.498047	0.498047	0.482422	0.476563	–

5 Design and analysis of Image Encryption Algorithm

In this section, two generated S-Boxes are selected for image encryption and then the performance and security analysis results of the image encryption will be presented.

5.1 Design of Image Encryption Algorithm

The Lai-Massey structure was first used in 1990 by Lai and Massey in the PSE cipher. Generating random values of the algorithm is performed by the Hindmarsh-Rose system as follows.

- **Step 1:** The Hindmarsh - Rose system trajectory is obtained by applying the selected initial conditions and chaotic parameter values, using Euler's method.
- **Step 2:** random values of $g(i)$ and $h(i)$ are obtained from the unknown y_i in Hindmarsh-Rose system trajectory for $1 \leq i \leq 65536$.

$$\begin{array}{ccc} g(i) = floor(|y_i| \times 10^4), & & y_i = |y_i| \times 10^4 - double(g(i)), \\ h(i) = floor(|y_i| \times 10^4), & g(i) = mod(g(i), 256), & h(i) = mod(h(i), 256), \end{array} \tag{2}$$

where the function $floor(x)$ returns the nearest integer less than or equal to x and $|x|$ returns the absolute value of the real number x. In the first iteration of the developed construction of Lai-Massey on the image

Table 10: The differential approach table for S_1.

$$\begin{bmatrix}
6 & 6 & 6 & 8 & 8 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 10 \\
6 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 8 & 8 & 8 & 8 & 8 & 8 & 10 & 8 \\
8 & 8 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 8 & 10 & 6 & 6 & 6 & 6 & 6 \\
6 & 6 & 6 & 6 & 10 & 6 & 8 & 8 & 6 & 6 & 6 & 8 & 6 & 8 & 6 & 6 \\
6 & 8 & 6 & 6 & 6 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 8 \\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 8 & 6 & 10 \\
6 & 6 & 8 & 8 & 6 & 6 & 10 & 10 & 6 & 8 & 6 & 8 & 6 & 6 & 6 & 6 \\
6 & 8 & 6 & 6 & 8 & 8 & 4 & 8 & 6 & 4 & 6 & 6 & 6 & 8 & 6 & 8 \\
6 & 8 & 8 & 10 & 6 & 8 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 8 & 8 \\
8 & 6 & 6 & 8 & 6 & 6 & 6 & 8 & 8 & 8 & 6 & 8 & 8 & 6 & 8 & 8 \\
6 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 8 & 8 & 6 & 6 & 6 & 8 & 6 \\
8 & 8 & 6 & 8 & 10 & 8 & 6 & 8 & 10 & 4 & 8 & 6 & 6 & 6 & 6 & 8 \\
6 & 6 & 8 & 8 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 6 & 8 & 8 & 8 & 4 \\
8 & 6 & 6 & 8 & 6 & 8 & 8 & 6 & 8 & 6 & 6 & 8 & 6 & 6 & 10 & 6 \\
6 & 6 & 6 & 6 & 8 & 8 & 8 & 6 & 6 & 6 & 6 & 8 & 6 & 6 & 10 & 4 \\
8 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 6 & 10 & 8 & 8 & 8 & 6 & - \\
\end{bmatrix}$$

Table 11: The differential approach table for S_2

$$\begin{bmatrix}
8 & 8 & 8 & 6 & 6 & 8 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 8 & 6 & 6 \\
8 & 6 & 6 & 6 & 6 & 6 & 6 & 8 & 6 & 8 & 6 & 6 & 8 & 8 & 4 & 8 \\
8 & 8 & 6 & 6 & 6 & 8 & 6 & 8 & 8 & 8 & 6 & 6 & 8 & 8 & 6 & 6 \\
6 & 6 & 6 & 6 & 6 & 6 & 8 & 8 & 8 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
6 & 8 & 6 & 6 & 8 & 6 & 8 & 6 & 6 & 6 & 8 & 8 & 10 & 6 & 6 & 8 \\
6 & 8 & 6 & 8 & 8 & 6 & 6 & 8 & 8 & 6 & 8 & 6 & 6 & 6 & 8 & 8 \\
6 & 8 & 8 & 6 & 8 & 8 & 8 & 8 & 6 & 6 & 8 & 10 & 6 & 6 & 6 & 6 \\
6 & 8 & 6 & 8 & 8 & 6 & 6 & 6 & 6 & 6 & 8 & 8 & 6 & 8 & 8 & 6 \\
8 & 6 & 8 & 6 & 6 & 8 & 10 & 6 & 4 & 10 & 6 & 6 & 6 & 8 & 10 & 8 \\
8 & 6 & 8 & 6 & 8 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 10 & 6 & 8 & 6 \\
6 & 6 & 6 & 6 & 6 & 10 & 8 & 6 & 8 & 6 & 6 & 8 & 6 & 6 & 6 & 6 \\
8 & 8 & 6 & 6 & 6 & 6 & 6 & 8 & 6 & 4 & 6 & 8 & 8 & 6 & 6 & 6 \\
8 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 7 & 8 & 6 & 6 & 6 & 6 & 6 & 8 \\
6 & 6 & 6 & 8 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 6 & 6 & 8 & 6 & 10 \\
6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 8 & 6 & 6 & 6 & 6 & 8 \\
8 & 8 & 6 & 6 & 8 & 6 & 6 & 6 & 8 & 6 & 6 & 6 & 8 & 8 & 6 & - \\
\end{bmatrix}$$

file needs up to 32768 random values of $g(i)$ and $h(i)$, and also in the second iteration of the developed construction of Lai-Massey needs up to 32768 other random values of $g(i)$ and $h(i)$. So we need a total of 65536 randomly generated values. Now, the two S-Boxes designed in the previous section are used to encrypt the image in the developed construction of Lai-Massey. A general and detailed overview of the image encryption algorithm is shown in Figure 3 and Figure 4, respectively. The image encryption algorithm consists of two steps: Suppose we have the original image in 256×256 grayscale. The original image file corresponds to a 256×256 matrix, where each element of the matrix is equal to one pixel of the original image. In the first iteration step of the developed construction of Lai-Massey on the image

Table 12: The linear approach table for S_1

$$\begin{bmatrix}
0 & 22 & 22 & 26 & 30 & 24 & 22 & 24 & 22 & 22 & 24 & 26 & 24 & 22 & 26 & 28 \\
26 & 24 & 22 & 24 & 26 & 30 & 20 & 22 & 26 & 24 & 26 & 24 & 26 & 28 & 30 & 20 \\
24 & 28 & 22 & 24 & 24 & 22 & 28 & 24 & 22 & 24 & 24 & 24 & 28 & 24 & 22 & 28 \\
30 & 24 & 20 & 24 & 28 & 24 & 28 & 26 & 22 & 22 & 24 & 26 & 24 & 20 & 24 & 22 \\
22 & 28 & 22 & 22 & 24 & 24 & 22 & 24 & 28 & 20 & 22 & 24 & 20 & 22 & 22 & 26 \\
24 & 22 & 24 & 28 & 24 & 22 & 20 & 26 & 28 & 22 & 22 & 20 & 22 & 22 & 24 & 20 \\
22 & 22 & 26 & 26 & 22 & 22 & 24 & 20 & 24 & 20 & 28 & 24 & 20 & 22 & 24 & 24 \\
20 & 22 & 26 & 22 & 26 & 22 & 22 & 26 & 22 & 28 & 22 & 22 & 22 & 22 & 30 & 26 \\
26 & 28 & 22 & 30 & 22 & 22 & 30 & 26 & 24 & 24 & 24 & 26 & 28 & 20 & 22 & 24 \\
26 & 28 & 20 & 28 & 24 & 24 & 24 & 24 & 26 & 26 & 22 & 24 & 20 & 22 & 26 & 24 \\
26 & 22 & 26 & 20 & 24 & 24 & 24 & 22 & 22 & 24 & 26 & 26 & 24 & 26 & 22 & 30 \\
30 & 24 & 26 & 22 & 28 & 24 & 28 & 20 & 24 & 22 & 24 & 20 & 28 & 22 & 24 & 24 \\
22 & 30 & 24 & 24 & 24 & 26 & 30 & 28 & 26 & 24 & 22 & 24 & 24 & 30 & 26 & 26 \\
22 & 24 & 26 & 24 & 26 & 24 & 22 & 22 & 24 & 28 & 20 & 20 & 20 & 24 & 24 & 26 \\
26 & 22 & 24 & 24 & 24 & 22 & 28 & 24 & 22 & 24 & 22 & 24 & 22 & 24 & 26 & 26 \\
22 & 20 & 26 & 28 & 28 & 20 & 22 & 26 & 28 & 24 & 24 & 24 & 20 & 24 & 22 & 28
\end{bmatrix}$$

file, pixels $X1 = (1,256)$ and $Y1 = (256,1)$ are selected from the image file. Then, they are done "XOR" with the random numbers $g(1)$ and $h(1)$ generated based on the Hindmarsh-Rose chaotic system. The result enters the developed construction of Lai-Massey. In addition to replacing the input pixels in the original image, the output of the developed construction of Lai-Massey is also used as an input with two pixels $X2 = (1,255)$ and $Y2 = (256,2)$ from the original image file and with the random values, $g(2)$ and $h(2)$ generated based on Hindmarsh-Rose chaotic system is done "XOR". The result is then entered into the developed construction of Lai-Massey. In the same way, all the pixels of the image pass through the developed construction of Lai-Massey, and the temporary encrypted image file is obtained. Now, the temporary encrypted image file is used as input in the second iteration of the developed construction of Lai-Massey. In the second step, from the developed construction of Lai-Massey on the temporary encrypted image file, two pixels $C'1 = (1,1)$ and $D'1 = (129,1)$ from the input, with the random values $g(32769)$ and $h(32769)$ generated based on Hindmarsh-Rose chaotic system, is done "XOR". Then the result enters the developed construction of Lai-Massey. In addition to replacing the pixels in the temporary encrypted image file, the output of the developed construction of Lai-Massey in this step is also used as an input with two pixels $C'2 = (1,2)$ and $D'2 = (129,2)$ from the temporary encrypted image file and with the random values $g(32770)$ and $h(32770)$ generated based on Hindmarsh-Rose chaotic system is done "XOR". The result enters the developed construction of Lai-Massey. Likewise, all the pixels pass through the developed construction of Lai-Massey. The final output is the final encrypted image file. By applying the reverse of these operations on the encrypted image file, the original image will be obtained.

5.2 Image encryption and security analysis

In this section and its related subsections, first, by using the image encryption algorithm, encrypting and decrypting processes on the Cameraman image is performed, which can be seen in Figure 5. When

Table 13: The linear approach table for S_2.

$$\begin{bmatrix}
0 & 22 & 22 & 22 & 24 & 26 & 24 & 26 & 24 & 28 & 22 & 22 & 20 & 22 & 26 & 22 \\
20 & 22 & 20 & 20 & 22 & 26 & 22 & 22 & 22 & 24 & 20 & 26 & 30 & 28 & 26 & 22 \\
24 & 26 & 20 & 26 & 24 & 24 & 28 & 30 & 26 & 26 & 20 & 26 & 26 & 26 & 22 & 22 \\
24 & 24 & 24 & 22 & 22 & 22 & 26 & 20 & 26 & 24 & 22 & 22 & 24 & 24 & 26 & 22 \\
22 & 24 & 30 & 30 & 22 & 22 & 26 & 26 & 24 & 22 & 20 & 24 & 22 & 22 & 22 & 22 \\
24 & 24 & 22 & 30 & 18 & 26 & 24 & 22 & 18 & 22 & 28 & 20 & 24 & 24 & 28 & 30 \\
30 & 20 & 22 & 26 & 26 & 22 & 24 & 20 & 28 & 28 & 26 & 28 & 22 & 22 & 26 & 22 \\
24 & 22 & 24 & 22 & 20 & 22 & 26 & 24 & 28 & 28 & 22 & 24 & 20 & 22 & 22 & 22 \\
26 & 26 & 24 & 24 & 26 & 24 & 22 & 24 & 28 & 28 & 20 & 26 & 26 & 30 & 22 & 28 \\
26 & 20 & 30 & 26 & 24 & 28 & 24 & 28 & 24 & 24 & 24 & 26 & 26 & 26 & 22 & 24 \\
28 & 28 & 28 & 26 & 28 & 28 & 28 & 24 & 28 & 22 & 26 & 22 & 24 & 22 & 24 & 22 \\
22 & 26 & 22 & 30 & 24 & 22 & 24 & 24 & 26 & 22 & 24 & 22 & 22 & 22 & 26 & 30 \\
26 & 22 & 26 & 26 & 26 & 26 & 24 & 26 & 20 & 26 & 22 & 28 & 22 & 22 & 22 & 22 \\
24 & 24 & 24 & 28 & 28 & 22 & 28 & 24 & 22 & 22 & 28 & 22 & 22 & 22 & 22 & 24 \\
24 & 22 & 22 & 22 & 26 & 22 & 20 & 20 & 24 & 22 & 22 & 22 & 22 & 24 & 24 & 24 \\
26 & 24 & 22 & 22 & 20 & 24 & 26 & 28 & 28 & 22 & 22 & 24 & 26 & 24 & 26 & 26
\end{bmatrix}$$

Table 14: The algebraic degree of S_1 and S_2.

S-Boxe	Number of monomial degrees of							
	0	1	2	3	4	5	6	7
f_{10}	0	4	12	27	34	27	9	4
f_{11}	1	7	12	30	31	22	15	2
f_{12}	1	4	13	28	37	27	16	1
f_{13}	0	5	15	34	32	28	15	2
f_{14}	1	4	15	28	40	21	14	2
f_{15}	0	4	12	31	35	31	13	5
f_{16}	0	4	15	30	39	31	13	3
f_{17}	0	4	14	32	38	27	14	3
f_{20}	0	5	12	27	38	33	19	7
f_{21}	0	4	17	32	37	28	12	2
f_{22}	0	3	15	28	38	28	17	5
f_{23}	0	3	11	27	31	27	16	4
f_{24}	0	5	16	32	28	26	17	8
f_{25}	0	2	15	26	40	22	17	2
f_{26}	1	5	12	33	22	23	14	6
f_{27}	1	4	10	25	29	22	11	3

the original and decrypted image files are compared, the encryption process appears successful. Now the image encrypting process, to determine the quality of the process, is subjected to performance and

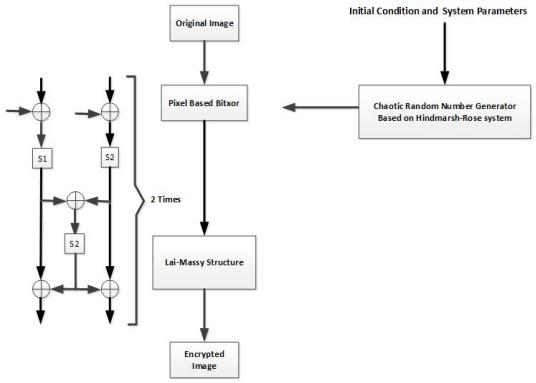

Figure 3: The general overview of the image encryption algorithm.

security analysis.

Table 15: Performance comparison of the proposed S-Box

S-Box	Nonlinearity			SAC			BIC-NL	BIC-SAC	DP	LP
	Min	Max	Avg.	Min	Max	Avg.				
[1]	98	108	105.25	0.4062	0.6094	0.5073	103.86	0.4986	0.0391	-
[3]	106	110	108.5	0.4063	0.5781	0.4995	103.86	0.5016	0.0391	0.1328
[4]	108	112	110	- - - -	- - - -	0.4995	103.14	-	0.0390	0.1328
[5]	102	106	105	0.4750	0.6093	0.5046	103.6	0.5004	0.0391	-
[7]	100	110	105.5	0.4063	0.6094	0.5010	103.79	0.5036	0.0468	0.1328
[8]	104	108	106.25	0.3594	0.6094	0.5002	103.64	0.4993	0.0391	0.1328
[12]	104	110	106.25	0.4219	0.5938	0.5039	103.36	0.5059	0.0390	0.1406
[14]	104	110	106.5	0.4375	0.6406	0.5120	104.57	0.5042	0.0390	0.1328
[16]	102	110	106.5	0.4063	0.5938	0.5010	103.43	0.4980	0.0391	0.1328
[17]	104	108	106.5	0.4063	0.6406	- - - -	102.86	0.4939	0.0469	0.1406
[20]	106	108	106.5	0.4219	0.6094	- - - -	104.07	0.4968	0.0391	0.1328
[21]	104	108	106.25	0.3906	0.6250	- - - -	103.21	0.5004	0.0469	0.1406
[25]	101	107	104.5	0.4219	0.5781	0.4963	103.29	0.4938	0.0390	0.1406
[24]	102	108	105.25	0.4375	0.5625	0.5037	102.6	0.4994	0.0391	-
[30]	106	110	107.5	0.3750	0.6094	0.5093	103.07	0.5025	0.0390	0.1406
[31]	102	108	105.5	0.4219	0.5781	0.5061	103	0.5009	0.0391	0.1406
[32]	104	110	106.5	- - - -	- - - -	0.495	103.8	0.4980	0.039	0.141
[35]	104	108	106.8	- - - -	- - - -	0.507	103.9	0.5070	0.054	0.140
[37]	104	110	107	0.4219	0.5938	0.4993	103.29	0.5051	0.0391	0.1328
S_1	104	108	106.0	0.3906	0.5781	0.4971	103.357	0.4993	0.0390	0.1171
S_2	106	108	107.0	0.4375	0.6250	0.4958	104.857	0.5022	0.0390	0.1171

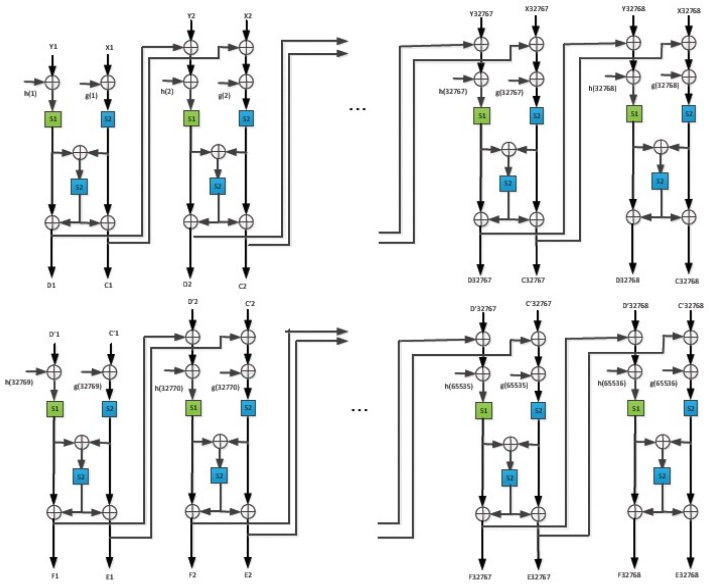

Figure 4: The detailed overview of the image encryption algorithm.

5.2.1 Histogram analysis and Chi-square test: The distribution of image pixels from the perspective of the gray level is examined by histogram distribution. In the histogram distribution, the more balanced pixels of the encrypted image is shown the more successful encryption. The histogram of the original image and the encrypted image is shown in Figure 6. As shown in Figure 6, the histogram of the encrypted image shows a fine gray-level distribution. Now we use variance for performance analysis which is defined as follows. $Var(H) = \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\frac{(c_i-c_j)^2}{2}$, where $H = \{c_0, c_1, \ldots, c_{255}\}$ is an array of histogram values, and c_i and c_j are the numbers of pixels whose grey values are equal to i and j, respectively. In general, the histogram variance of the encrypted image is smaller than the histogram variance of the original image. The smaller the histogram value of the encrypted image, the higher the uniformity

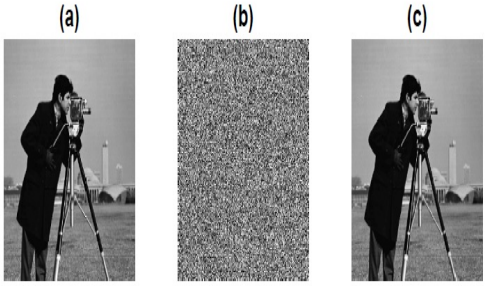

Figure 5: The result of image encryption. (a) original image, (b) encrypted image, (c) decrypted image.

Table 16: The Variance comparison result.

Algorithm	Variance of original image	Variance of encrypted image	χ^2 Test	Testing Result $\chi^2_{0.05}(255) = 293.248$
Proposed	138863.38	265.1016	265.1	pass
[34]	5.129×10^6	5347.24	-	-
[22]	110973.30	251.85	251.85	pass
[23]	-	268.28	268.28	pass
[36]	-	215.1563	215.1563	pass
[38]	-	281.1953	281.195	pass

of its gray value is. This means that the proposed scheme passes the histogram test. We calculated the histogram variance of the original image and the encrypted image of the Cameraman with the formula $Var(H)$, and showed the calculated results in Table 16. The χ^2 test is used to determine whether the encrypted image pixels are evenly distributed or not. The mathematical expression of χ^2 distribution is as follows $\chi^2 = \sum_{i=0}^{n-1} \frac{(x_i - m)^2}{m}$, where x_i is the frequency of pixel i in the image. Assume the image size is $M \times N$ and $m = \frac{M \times N}{256}$; when we consider the confidence level to be 0.05, $\chi^2_{0.05}(255) = 293.248$, in this case, if the value of the χ^2 test for the encrypted image is less than 293.248, it means that this test has successfully passed. Table 16 shows the χ^2 test results for the Cameraman encrypted image, the test results for this image are less than 293.248. Therefore, it can be concluded that the encrypted image by this algorithm is close to a uniform distribution.

5.2.2 Correlation analysis: In an original image, there is a high correlation between each pixel and its horizontal, vertical, and diagonal adjacent pixels. But, in an efficient encryption system, the correlation between adjacent pixels of the cipher image in the horizontal, vertical, and diagonal directions should be low enough. The correlation coefficient between two adjacent pixels of the image is denoted by c_{xy} and is calculated as follows

$$c_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 (y_i - \bar{y})^2}},$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$.

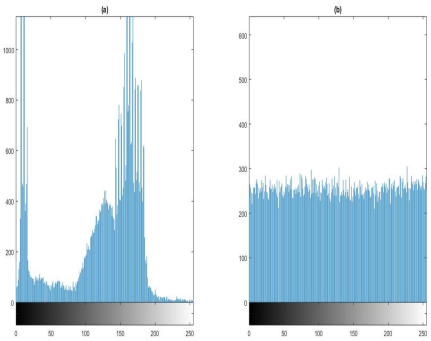

Figure 6: The result of Histogram. (a) histogram of image, (b) histogram encrypted image.

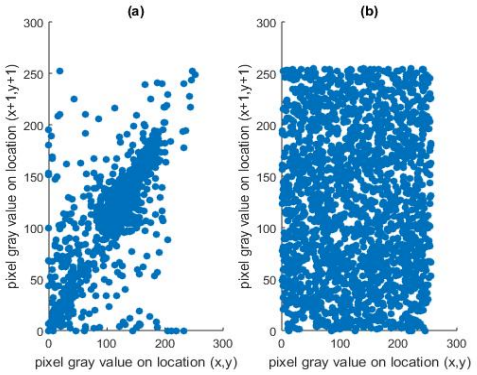

Figure 7: Correlation of horizontal adjacent two pixels. (a) Cameraman image (b) encrypted image.

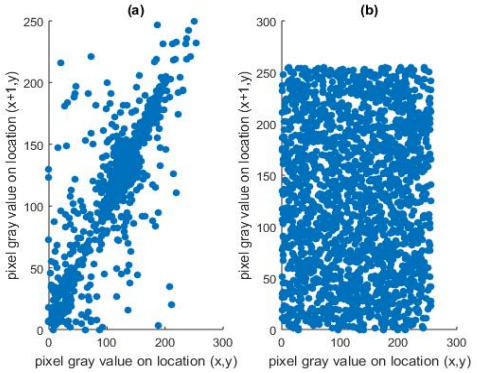

Figure 8: Correlation of vertical adjacent two pixels. (a) Cameraman image (b) encrypted image.

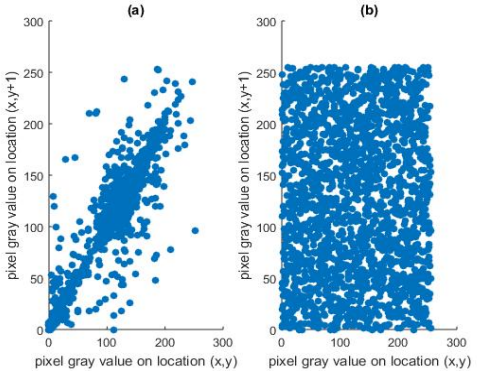

Figure 9: Correlation of diagonal adjacent two pixels. (a) Cameraman image (b) encrypted image.

To test the effectiveness of the encrypted algorithm on the Cameraman image, the correlation between the adjacent pixels of the original image and the encrypted image (in horizontal, vertical, and diagonal directions) is tested. For this purpose, we have randomly selected 2000 adjacent pixel pairs in the horizontal, vertical, and diagonal directions from the original image and the encrypted image, and then calculated the correlation coefficient of the original image and the encrypted image in those direc-

Table 17: Correlation coefficients of two adjacent pixels and comparisons.

Image	Horizontal	Vertical	Diagonal
Cameraman	0.9348	0.9601	0.9083
The proposed algorithm	-0.0024	0.0029	-0.0040
[22]	-0.000282	0.016975	0.011328
[23]	-0.002259	-0.000313	-0.000009
[29]	0.0456	-0.0568	-0.0202
[34]	-0.009	0.014	-0.006
[36]	-0.0046	0.0011	0.0159

Table 18: Comparison of information entropy.

Image	Information entropy
encrypted Cameraman	7.9971
[19]	7.9970
[22]	7.997538
[23]	7.9971
[34]	7.9993
[36]	7.9976
[38]	7.99689

tions. The correlation distribution test of the original image and encrypted image for two adjacent pixels in the horizontal, vertical, and diagonal direction, which is produced by the image encrypting algorithm, is shown in Figures 7, 8 and 9, respectively. The results of correlation coefficients for adjacent pixels in the horizontal, vertical, and diagonal directions for the original image and the encrypted image for the Cameraman image are given in Table 17. In the table 17, we can see that the correlation coefficient of two adjacent pixels in the horizontal, vertical, and diagonal directions for the original image is close to 1, On the contrary, after encrypting the image, the correlation coefficient between two adjacent pixels in different directions is close to 0. These results show the good effect of the image encrypting algorithm on the Cameraman's image.

5.2.3 Information entropy analysis: To measure the amount of randomness and complexity of the image, the information entropy on the encrypted image is used. Encrypted images should have high complexity and also have not given any information about the original image to the analyst or attacker. Entropy; $H(S)$ of a source S is calculated as follows: $H(S) = -\sum_{i=0}^{2^N-1} P(s_i)\log_2 P(s_i)$, Where N is the number of bits of symbol $s_i \in S$ and $P(s_i)$ represents the probability of symbol s_i. The optimal value of the information entropy for a grayscale image whose value of each pixel is in the interval $[0,255]$, is eight. After applying the image encryption algorithm to the Cameraman's image, the information entropy value is 7.9971, which is very close to the value of eight. This means that the encrypted image of the Cameraman according to the proposed encryption algorithm has good random distribution and high complexity. Table 18 shows the information entropy value on the Cameraman encrypted image for the proposed image encryption algorithm and other algorithms.

5.2.4 Differential analysis: Differential analysis of the image encrypting process is accomplished by calculating the number of pixel change rates ($NPCR$) and the uniform average changing intensity ($UACI$). For this purpose, these criteria are used to determine the resistance of the image encryption process against differential attacks and to detect the effect of small changes in the original image on the encrypted images. The first criterion, $NPCR$, measures the percentage change of pixels between two images. To calculate $NPCR$, we first encrypt two original images with a difference of one pixel with the same secret key. Suppose $C1$ and $C2$ are two encrypted images, equivalent to encrypting with the same secret key two original images that differ in one pixel. In this case, $C1(i,j)$ and $C2(i,j)$ are pixels (i,j) of the two encrypted images $C1$ and $C2$, respectively. $D(i,j)$ is defined as if $C1(i,j) = C2(i,j)$ then $D(i,j) = 0$ and if $C1(i,j) \neq C2(i,j)$ in this case $D(i,j) = 1$. Now, mathematically, $NPCR$ is defined as:

$$NPCR(C_1, C_2) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\%,$$

where M and N are the width and height of C_1 or C_2. The value of $NPCR$ for two encrypted images is close to 99.609%. That means the amount of change of one pixel in the original image causes a significant change in the encryption image.
The second criterion, $UACI$, is used to measure the average intensity of differences between two images, and its mathematical relationship is defined as follows.

$$UACI(C_1, C_2) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |C_1(i,j) - C_2(i,j)|}{M \times N \times 255} \times 100\%.$$

For a Cameraman's original image, the expected $UACI$ value is close to 33.464%.
In this test, we randomly change only one pixel of the original image and then analyze two encrypted images using $NPCR$ and $UACI$, whose results are shown in Table 19. The results show that a tiny change in the original image will lead to a significant change in the encryption image. Therefore, the proposed encryption algorithm shows good resistance and ability against differential attacks.

5.2.5 Key space analysis: An image encryption algorithm will be effective when the key space is large enough. The proposed image encrypting plan including parameters a, b, r, s, and I and initial values x_0, y_0, and z_0 is. These parameters and initial are used to generate random values by the chaotic Hindmarsh-Rose system as secret key in the proposed image encryption algorithm. So we can estimate the key space as 10^{8r} and set the largest accuracy to $r = 14$ to compare the results with other image encryption algorithms. The key space is approximately equal to 2^{372}. As shown in Table 20, such a key space result means that the proposed encryption algorithm has an ability to resist brute force attacks.

5.2.6 Encryption key sensitivity analysis: In the Key sensitivity analysis, with a tiny change in the secret key, which derives from the parameters and initial values of the chaotic system, the amount of change in the encrypted image is analyzed. First, the encryption algorithm runs on the input image P with the key K_1, and the encrypted image C_1 is acquired. Then, the encryption algorithm performs on the same input image with another key K_2, and the encrypted image C_2 is acquired. Employing these two encrypted images, the difference images $|C_1 - C_2|$ are obtained. Figure 10 shows the encryption of the Cameraman image using the keys derived from the parameters and initial values of the chaotic system. An encrypted algorithm that is sensitive to the key is ideal and ensures security against brute force attacks. During the process of the encryption algorithm, a tiny change in the parameters or initial values causes a considerable change in the encrypted image, which shows the sensitivity to the secret

Table 19: NPCR and UACI values and comparisons.

Algorithm	NPCR	UACI
Proposed	99.6216	33.4327
[19]	99.61	33.43
[22]	99.6017	33.4313
[23]	99.6131	33.4615
[29]	97.7814	17.5430
[34]	99.62	33.46
[36]	99.6109	33.3735
[38]	99.5986	33.4562

Table 20: Key Space and comparisons.

Algorithm	Key Space
Proposed	2^{372}
[19]	10^{112}
[23]	2^{512}
[34]	10^{165}

Table 21: Key sensitivity by NPCR and UACI.

Figure	NPCR	UACI
(a_1) and (b_1)	99.5972	33.5365
(a_1) and (b_2)	99.6384	33.4860
(a_1) and (b_3)	99.5819	33.5102
(a_1) and (b_4)	99.6231	33.5757
(a_1) and (b_5)	99.6368	33.5175
(a_1) and (b_6)	99.6017	33.5314
(a_1) and (b_7)	99.6109	33.3463
(a_1) and (b_8)	99.6307	33.6345

key. Usually, *NPCR* and *UACI* are used to measure the sensitivity to the secret key in the encryption algorithm. Considering the Cameraman image, when the parameters and initial values change as keys, the *NPCR* and *UACI* values between the two encrypted images show in Table 21.

5.2.7 Decryption key sensitivity analysis: Key sensitivity is crucial in the process of decryption. When the secret key of the decryption algorithm has a tiny change compared to the key in the encryption algorithm, the decrypted image should be significantly different from the original image. Figure 11 shows the decryption of the Cameraman image using the keys derived from the parameters and initial values of the chaotic system. When the secret key of the decryption algorithm change slightly, the Cameraman

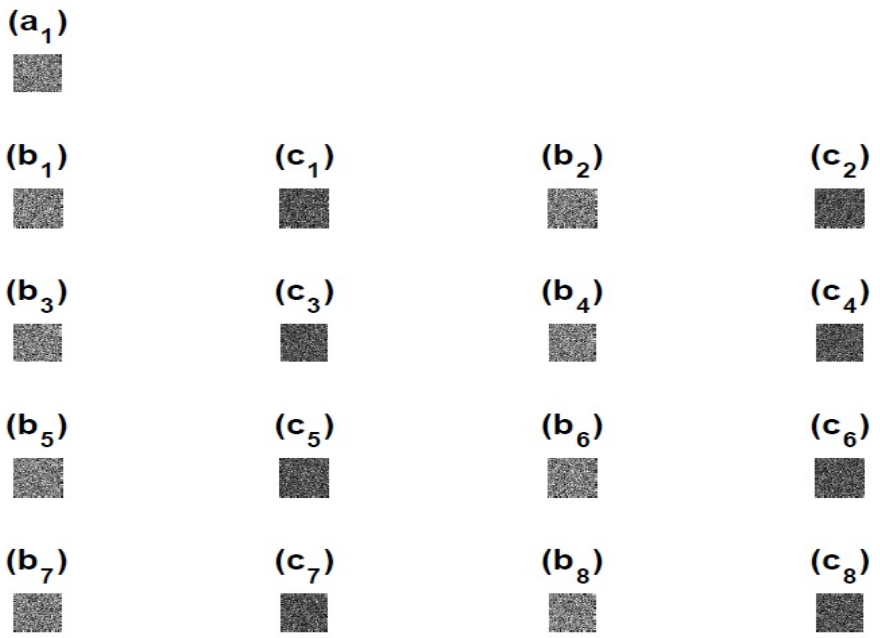

Figure 10: The encryption key sensitivity. (a_1) encrypted image using secret key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$, (b_1) encrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3 + 10^{-13}, b = -5, r = 0.006, s = 4, I = 3.2)$, (c_1) differential image between (a_1) and (b_1), (b_2) encrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -(5 + 10^{-13}), r = 0.006, s = 4, I = 3.2)$, (c_2) differential image between (a_1) and (b_2), (b_3) encrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006 + 10^{-12}, s = 4, I = 3.2)$, (c_3) differential image between (a_1) and (b_3), (b_4) encrypted image using key$(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4 + 10^{-12}, I = 3.2)$, (c_4) differential image between (a_1) and (b_4), (b_5) encrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2 + 10^{-12})$, (c_5) differential image between (a_1) and (b_5), (b_6) encrypted image using key $(x_0 = 10^{-12}, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$, (c_6) differential image between (a_1) and (b_6), (b_7) encrypted image using key $(x_0 = 0, y_0 = 0.05 + 10^{-9}, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$, (c_7) differential image between (a_1) and (b_7), (b_8) encrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02 + 10^{-11}, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$, (c_8) differential image between (a_1) and (b_8).

image with its decrypted image is different. A decrypted algorithm that is sensitive to the key is ideal and ensures security against brute force attacks.

5.2.8 Chosen - Plaintext Attack and Chosen - Ciphertext Attack: According to Kerckhoffs principle, when analyzing an encryption algorithm, the assumption is that the attacker knows precisely the design and workings of the cryptosystem. Namely, the attacker knows everything about the cryptosystem except key. There are numerous techniques to implement cryptanalysis. In this scenario of attack, the attacker has an encrypted image, but the encryption key is unknown. However, the attacker has a plain image P_0 of all-zero (or all-one) and its corresponding encrypted image C_0 obtained with the same unknown key. The attacker constructs the following subkey extraction for pixel encryption.

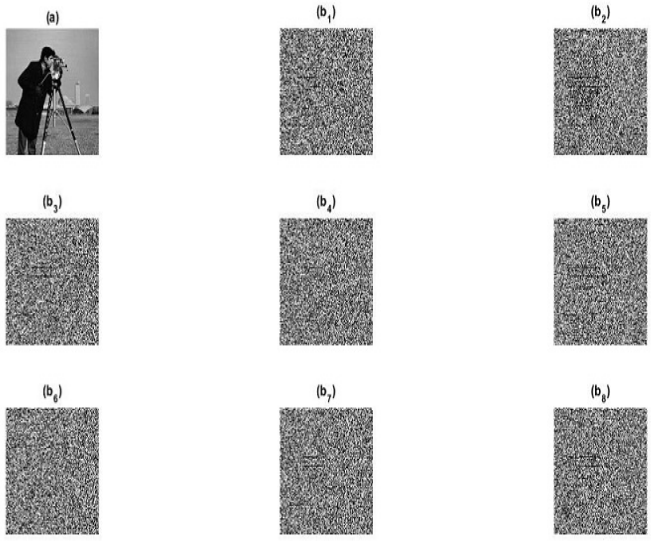

Figure 11: The decryption key sensitivity. (a) original image using secret key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$, (b_1) decrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3 + 10^{-9}, b = -5, r = 0.006, s = 4, I = 3.2)$, (b_2) decrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -(5 + 10^{-9}), r = 0.006, s = 4, I = 3.2)$, (b_3) decrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006 + 10^{-8}, s = 4, I = 3.2)$, (b_4) decrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4 + 10^{-8}, I = 3.2)$, (b_5) decrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2 + 10^{-8})$, (b_6) decrypted image using key $(x_0 = 10^{-7}, y_0 = 0.05, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$, (b_7) decrypted image using key $(x_0 = 0, y_0 = 0.05 + 10^{-4}, z_0 = 0.02, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$, (b_8) decrypted image using key $(x_0 = 0, y_0 = 0.05, z_0 = 0.02 + 10^{-7}, a = 3, b = -5, r = 0.006, s = 4, I = 3.2)$.

$K_0(i, j) = C_0(i, j) \oplus P_0(i, j)$, where P_0 is a null image in terms of grey values, C_0 is its corresponding encrypted image, and (i, j) is the two - dimensional pixel position. Above equation provides a key stream K_0. In trying to get the plain image $P(i, j)$ of the encrypted one $C(i, j)$, the attacker makes use of the key stream K_0 as $P(i, j) = C(i, j) \oplus K_0(i, j)$. In Figure 12(a), it can be seen that the chosen - plaintext attack on the Cameraman encrypted image using a null image has failed. The corresponding histogram is given in Figure 12(b). The reason for this failure is the pixel confusion phase and the developed construction of Lai-Massey which are sensitive to the tiny changes of a grey value. Therefore, the proposed technique demonstrates resistance to the chosen - plaintext attack.

Chosen - ciphertext attack is another type of attack having no information about the secret key. Knowing an encrypted image C_0 of all - one (or all - zero), and its corresponding decrypted image P_0, the attacker tries to determine the key stream K_0 using above Equation. Then, the plain image $P(i, j)$ would be obtained by above Equation. In Figure 12(c) and (d), it can be seen that the chosen - ciphertext attack on the Cameraman encrypted image using a null image has failed. The reason for this failure is the pixel confusion phase and the developed construction of Lai-Massey which are sensitive to the tiny changes of

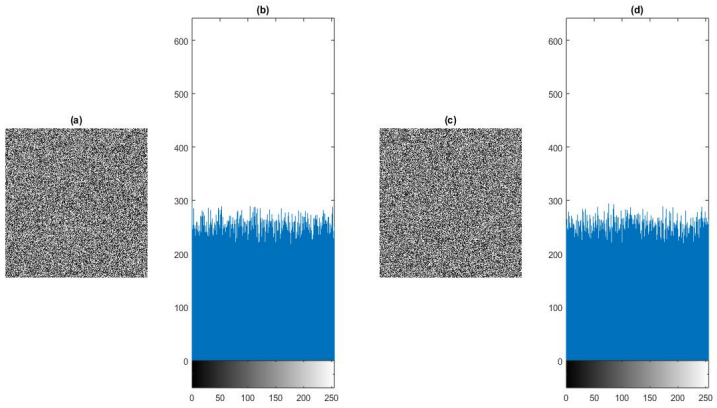

Figure 12: Cryptanalysis. (a) Chosen - plaintext attack, (b) Corresponding image histogram, (c) Chosen - ciphertext attack, (d) Corresponding image histogram.

a grey value. Therefore the proposed technique demonstrates resistance to the chosen - ciphertext attack.

6 Conclusion

In this paper, the design of two S-Boxes based on the Hindmarsh-Rose chaotic system was presented. The experimental results show that the proposed S-Boxes have good cryptographic properties compared to other chaotic-based S-Boxes. Also, an algorithm based on two proposed S-Boxes was designed to encrypt an image in the developed construction of Lai-Massey, and the main results of it include the following: In the Chi-square test, we have reached 265.1, which means that this test has been passed, the correlation coefficient test in three horizontal, vertical, and diagonal modes is equal to -0.0024, 0.0029, and -0.004, respectively. The entropy value is equal to 7.9971. The sensitivity to the key in the differential analysis is $NPCR = 99.6216$, and $UACI = 33.4327$. These results show that the proposed algorithm has proper performance in various tests and also confirm that the proposed plan has achieved high sensitivity to the key, information entropy, low correlation coefficients, large key space, and good resistance to attacks.

References

[1] M.Ş. Açikkapi, F. Özkaynak, *A method to determine the most suitable initial conditions of chaotic map in statistical randomness applications*, IEEE Access. **9** (2021) 1482–1494.

[2] C.M. Adams, S.E. Tavares, *A Note on the generation and counting of bent sequences*, Tech. Rep., Department of Electrical Engineering, Queenss University, 1989.

[3] H.S. Alhadawi, M.A. Majid, D. Lambić, M. Ahmad, *A novel method of S-Box design based on discrete chaotic maps and cuckoo search algorithm*, Multimed. Tools Appl. **80** (2021) 7333–7350.

[4] F. Artuğer, F. Özkaynak, *An effective method to improve nonlinearity value of substitution boxes based on random selection*, Inf. Sci. **576** (2021) 577–588.

[5] F. Artuğer, F. Özkaynak, *A novel method for performance improvement of chaos-based substitution boxes*, Symmetry **12** (2020).

[6] M.S. Baptista, *Cryptography with chaos*, Phys. Lett. A. **240** (1998) 50–54.

[7] A. Belazi, A.A.A. El-Latif, *A simple yet efficient S-Box method based on chaotic sine map*, Optik. **130** (2017) 1438–1444.

[8] M.A. Ben Farah, A. Farah, T. Farah, *An image encryption scheme based on a new hybrid chaotic map and optimized substitution box*, Nonlinear Dyn. **99** (2020) 3041–3064.

[9] E. Biham, A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology. **4** (1991) 3–72.

[10] C. Boura, A. Canteaut, *On influence algebraic degree of on the algebraic degree of F^{-1} on the algebraic degree of GoF*, IEEE Trans. Inf. Theory **59** (2013) 691–702.

[11] C. Boura, A. Canteaut, C. De Canniere, *Higher-order differential properties of Keccak and Luffa*, Springer, Berlin-Heidelberg, in International Workshop on Fast Software Encryption, (2011) 252-269.

[12] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, S. Kaçar, *A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system*, Nonlinear Dyn. **87** (2017) 1081–1094.

[13] T. Cusick, P. Stanica, *Cryptographic Boolean Functions and Applications*, Amsterdam, The Netherlands: Elsevier, 2009.

[14] T. Farah, R. Rhouma, S. Belghith, *A novel method for designing S-box based on chaotic map and TeachingLearning- Based Optimization*, Nonlinear Dyn. **88** (2017) 1059–1074.

[15] J. Fridrich, *Symmetric ciphers based on two-dimensional chaotic maps*, Int. J. Bifurcat. Chaos **8** (1998) 1259–1284.

[16] N. Hematpour, S. Ahadpour, I.G. Sourkhani, R.H. Sani, *A new steganographic algorithm based on coupled chaotic maps and a new chaotic S-box* Multimed. Tools Appl. **14** (2022).

[17] N. Hematpour, S. Ahadpour, *Execution examination of chaotic S-box dependent on improved PSO algorithm*, Neural Comput. Appl. **33** (2021) 5111–5133.

[18] J.L. Hindmarsh, R.M. Rose, *A model for neuronal bursting using three coupled differential equations*, Proc. R. Soc. Lond. B. Biol. Sci. **221** (1984) 87–102.

[19] C. Lakshmi, K. Thenmozhi, J.B.B. Rayappan, R. Amirtharajan, *Hopfield attractor-trusted neural network: an attack-resistant image encryption*, Neural. Comput. Appl. **32** (2020) 11477–11489.

[20] D. Lambić, *A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design*, Nonlinear Dyn. **100** (2020) 699711.

[21] H. Liu, J. Liu, C. Ma, *Constructing dynamic strong S-Box using 3D chaotic map and application to image encryption*, Multimed. Tools Appl. **81** (2022).

[22] H. Liu, B. Zhao, L. Huang, *Quantum image encryption scheme using Arnold transform and S-Box scrambling*, Entropy. **21** (2019).

[23] B. Norouzi, S.M. Seyedzadeh, S. Mirzakuchaki, M.R. Mosavi, *A novel image encryption based on hash function with only two-round diffusion process*, Multimed. Syst. **20** (2013).

[24] F. Özkaynak, *On the effect of chaotic system in performance characteristics of chaos based S-box designs*, Phys. A, Stat. Mech. Appl. **550** (2020).

[25] F. Özkaynak, V. Çelik, A.B. Özer, *A new S-box construction method based on the fractional-order chaotic Chen system*, Signal Image Video P. **11** (2017) 659–664.

[26] W.C. Qiu, S.J. Yan, *An image encryption algorithm based on the combination of low-dimensional chaos and high-dimensional chaos*, 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), (2019) 684–687 .

[27] H.R. Shakir, *An image encryption method based on selective AES coding of wavelet transform and chaotic pixel shuffling*, Multimed. Tools Appl. **78** (2019) 26073–26087.

[28] C.E. Shannon, *Communication theory of secrecy systems*, Bell Syst. Tech. J. **28** (1949) 656–715.

[29] S. Somaraj, M.A. Hussain, *Performance and security analysis for image encryption using key image*, Indian J. Sci. Technol. **8** (2015) 1–4.

[30] Y. Tian, Z. Lu, *Chaotic S-Box: Intertwining logistic map and bacterial foraging optimization*, Math. Probl. Eng. **11** (2017).

[31] X. Tong, X. Liu, J. Liu, M. Zhang, Z. Wang, *A novel lightweight block encryption algorithm based on combined chaotic S-Box*, Int. J. Bifurc. Chaos. **31** (2021).

[32] X. Wang, Ü. Çavuşoğlu, S. Kacar, A. Akgul, V. Pham, S. Jafari, F. Alsaadi, X.Q. Nguyen, *S-Box based image encryption application using a chaotic system without equilibrium*, Appl. Sci. **9** (2019).

[33] A.F. Webster, S.E. Tavares, *On the design of S-boxes*, Advances in Cryptology- Crypto85, Santa Barbara. Lecture Notes in Computer Science. **218** (1970) 523–534.

[34] L.M.H. Yepdia, A. Tiedeu, G. Kom, *A robust and fast image encryption scheme based on a mixing technique*, Secur. Commun. Netw. **2021** (2021).

[35] A.H. Zahid, M.J. Arshad, *An innovative design of substitution-boxes using cubic polynomial mapping*, Symmetry **11** (2019).

[36] X. Zhang, L. Wang, Y. Wang, Y. Niu, Y. Li, *Image encryption algorithm based on hyperchaotic system and variable-step Josephus problem*, Int. J. Opt. **2020** (2020).

[37] P. Zhou, J. Du, K. Zhou, S. Wei, *2D mixed pseudo-random coupling PS map lattice and its application in S-box generation*, Nonlinear Dyn. **103** (2021), 1151–1166.

[38] D. Zou, M. Li, J. Li, Z. Li, *An image encryption algorithm based on a new hybrid power exponent chaotic system*, Secur. Commun. Netw. **2021** (2021).