Received: 26 March 2019 | Revised: 2 September 2019 | Accepted: 31 October 2019

DOI: 10.4218/etrij.2019-0138

ORIGINAL ARTICLE

ETRI Journal WILEY

Highly dispersive substitution box (S-box) design using chaos

Zaid Bin Faheem[1] | **Asim Ali**[2] | **Muhamad Asif Khan**[1] | **Muhammad Ehatisham Ul-Haq**[1] | **Waqar Ahmad**[1]

[1]Department of Computer Engineering, University of Engineering and Technology, Taxila, Pakistan

[2]Department of Computer Science, University of Wah, Wah Cantonment, Pakistan

Correspondence
Muhamad Asif Khan, Department of Computer Engineering, University of Engineering and Technology, Taxila, Pakistan.
Email: masif.khan@uettaxila.edu.pk

Funding Information
This work was supported by the Department of Computer Engineering, University of Engineering and Technology Taxila, Pakistan.

Highly dispersive S-boxes are desirable in cryptosystems as nonlinear confusion sublayers for resisting modern attacks. For a near optimal cryptosystem resistant to modern cryptanalysis, a highly nonlinear and low differential probability (DP) value is required. We propose a method based on a piecewise linear chaotic map (PWLCM) with optimization conditions. Thus, the linear propagation of information in a cryptosystem appearing as a high DP during differential cryptanalysis of an S-box is minimized. While mapping from the chaotic trajectory to integer domain, a randomness test is performed that justifies the nonlinear behavior of the highly dispersive and nonlinear chaotic S-box. The proposed scheme is vetted using well-established cryptographic performance criteria. The proposed S-box meets the cryptographic performance criteria and further minimizes the differential propagation justified by the low DP value. The suitability of the proposed S-box is also tested using an image encryption algorithm. Results show that the proposed S-box as a confusion component entails a high level of security and improves resistance against all known attacks.

KEYWORDS

bit independence criterion, differential approximation probability, piecewise linear chaotic map, strict avalanche criterion, substitution box

1 | INTRODUCTION

Cryptography provides security services such as authenticity, integrity, and confidentiality to secure communication systems from adversaries. Modern block ciphers are designed iteratively and based on the notion of Shannon's principle of confusion and diffusion [1]. Confusion is introduced in a system using substitution-boxes (S-boxes). Typically, confusion is the only nonlinear component in a cryptosystem preventing an attacker from estimating the propagation of information from input to output. The known S-box structure includes the linear propagation of information using an attack known as differential cryptanalysis. The strength of an S-box is evaluated based on the cryptographic properties of bijection, nonlinearity [2], strict avalanche criterion (SAC) [3], bit independence criterion (BIC) [3], and linear and differential approximation probabilities [4–6]. An ideal or near optimal S-box acquires the upper bound of these given properties. An S-box with high nonlinearity and a low differential probability (DP) value is known as cryptographically strong.

Chaos, a nonlinear dynamic system that is favored in cryptography due to its simplicity in implementation, sensitivity in dependence on initial conditions, mixing capabilities, and ergodicity [7]. In the last decade, researchers exploited the chaotic phenomenon to generate S-boxes. Kocarev and others first explored the similarities between chaos and cryptography and proposed a simple method to generate a chaotic S-box [8]. Chaotic S-boxes are less complex, simpler to design, and easier to implement in hardware compared to algebraic S-boxes. Chaotic S-boxes are not optimal in terms of their cryptographic properties, but they are still considered to have good cryptographic properties. A number of methods have been proposed to generate chaotic S-boxes using chaotic trajectories of 1D and higher dimensional maps [9–12]. Moreover, S-boxes designed with a chaotic map can also be optimized using different optimization techniques to obtain highly nonlinear trajectories [13–15]. A list of recently proposed S-boxes with the design techniques used to generate these S-boxes and their nonlinearity and differential approximation probability cryptographic properties is presented in Table 1.

Continuous S-box design evolvement based on chaos has motivated researchers to utilize chaotic systems in combination with other nonlinear portents for image encryption, watermarking, and steganography [31–33]. As a result, chaos and other nonlinear phenomena have been utilized to encrypt images [34–37]. The authors in [18] proposed a novel method for image encryption using the Gingerbreadman chaotic map and S_8 permutation. Belazi and others [38] proposed a permutation–substitution-based cryptosystem for encryption. The authors in [39] proposed a novel method for the construction of chaotic S-boxes in captcha. The research studies [40] and [41] utilized multiple chaotic S-boxes and Fourier series for image encryption. The authors in [42] utilized a chaotic system and cyclic elliptic curve for image encryption. A few authors have also employed chaotic S-boxes for watermarking. Khan and others [43] utilized the classes of chain rings to design a novel S-box for image encryption and watermarking. Khan and Shah [44] utilized a nonlinear permutation and evaluated its quality metrics to present a novel scheme for image watermarking and copyright protection. In recent years, the authors in [45–47] utilized quantum spinning and rotation for image encryption and watermarking. Younas and Khan [48] presented a novel scheme for efficient image encryption based on a Lorenz chaotic system.

TABLE 1 Recent S-box design techniques with cryptographic properties

Study/Year	Technique	Properties: Nonlinearity	Properties: DP
[16]/2018	Chaotic quantum magnets and matrix Lorenz systems	108	0.03125
[17]/2018	1D discrete chaotic map	106.5	0.0390
[18]/2018	Gingerbreadman chaotic map and S_8 permutation	103.25	0.171
[19]/2017	Chaos and random number generator	106	0.0468
[20]/2017	Chen system	104.7	0.0390
[21]/2017	Zhongtang Chaotic system	106	0.0390
[15]/2017	Chaos and teaching-learning based optimization	106.5	0.0390
[22]/2017	Chaotic sine map	105.5	0.0468
[23]/2017	Logistic map and foraging optimization	107.5	0.0390
[24]/2016	Chaotic Boolean functions	100	0.0468
[25]/2015	Logistic map	108	0.0390
[26]/2014	The Chen, Rossler, Chua	105.5	0.0390
[27]/2013	Kuramoto equation and Galois field	108	0.0625
[28]/2013	Time delay chaotic model	105.1	0.0390
[29]/2012	Lorenz system	105.2	0.0312
[30]/2010	Nonlinear chaotic algorithm	105.2	0.0468

1.1 | Major contribution

In this paper, a well-structured methodology is presented for designing highly dispersive S-boxes based on chaos. Map selection is critical in designing chaotic S-boxes as it induces a high dispersion of initial values. For this reason, a piecewise linear chaotic map (PWLCM) is employed in this study. A random number generator (RNG) design is first proposed using the PWLCM. The random numbers generated using the PWLCM are cryptographically secure and statistically analyzed using the National Institute of Standards and Technology (NIST) criterion. Secondly, a new method for S-box generation based on the PWLCM is proposed, followed by a simple optimization technique for surplus nonlinear mapping between input and output entities, which is one of the core benefits of the proposed methodology that distinguishes it from all previously used optimization methods for S-box design. Due to the inherent mixing and ergodicity properties, the chaotic map is iterated using the given design conditions and can generate an S-box in a reasonable amount of time. The S-box is an auxiliary table of 256 fixed decimal positions. The proposed algorithm is linear, and the time complexity is $O(n)$, where n is the number of iterations required to generate the proposed S-box. This is also evident in the NIST test in Section 2.1. The designed S-box suitability is tested using an image encryption algorithm. Cryptosystems based on single chaotic maps are not insured against chosen and known plaintext attacks. Therefore, in this work, multiple chaotic maps are used to schedule the key. For key masking and mixing with plaintext, ciphertext feedback is used to make the proposed design resistant to known and chosen plaintext attacks.

The remainder of the paper is structured as follows. Section 2 presents the random number generator using a piecewise linear chaotic map. Section 3 details the proposed S-box design methodology. Section 4 provides a detailed performance analysis. Sections 5 and 6 detail the suitability of the proposed S-box based on the image encryption algorithm and performance analysis, respectively. The final section summarizes the conclusions of the study.

2 | PROPOSED METHOD FOR RNG DESIGN USING PWLCM

In recent years, many researchers have used chaotic maps when designing nonlinear dynamic systems. In this study, the PWLCM is used due to its simplicity in representation and sufficient dynamic nonlinear behavior with a positive lyapunov exponent [14]. A PWLCM with four intervals is selected in this study for S-box generation, which is represented in (1). The current input x is passed to the PWLCM to generate the next input x_{n+1}. Figure 1 shows the behavior and lyapunov exponent plot of the PWLCM. The positive lyapunov of the exponent indicates that it is chaotic in the regime for $p \in (0, 0.5)$.

$$x_{n+1} = \begin{cases} \frac{x_n}{p} & 0 \le x_n < p, \\ \frac{(x_n - p)}{(0.5 - p)} & p \le x_n < 0.5, \\ \frac{(1 - p - x_n)}{(0.5 - p)} & 0.5 < x_n < 1 - p, \\ \frac{(1 - x_n)}{p} & 1 - p < x_n < 1, \end{cases} \quad (1)$$

where $x_o \in [0, 1)$ is the initial value, and $p \in (0, 0.5)$ is the control factor.

The following section provides a detailed explanation of the proposed RNG design based on the PWLCM.

2.1 | RNG design using PWLCM

In this research, a PWLCM is used to generate random numbers. As the randomness of the numbers produced by the RNG

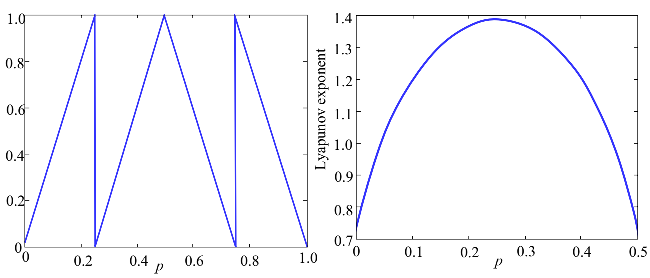

FIGURE 1 Piecewise Linear Chaotic Map (PWLCM) plot and Lyapunov exponent [14]

TABLE 2 NIST statistical tests and their results for the PWLCM-based RNG

NIST statistical test	*p*-value	Status
Frequency (monobit) test	0.315379	Passed
Block Frequency test	0.186620	Passed
Cumulative sum test	0.425888 (Forward) 0.202842 (Reverse)	Passed
Runs test	0.605161	Passed
Longest run test	0.954527	Passed
Rank test	0.287656	Passed
Discrete fourier transform test	0.679644	Passed
Non-overlapping template matchings test	0.509078	Passed
Overlapping template matchings test	0.045839	Passed
Universal statistical test	0.296564	Passed
Approximate entropy test	0.993287	Passed
Random excursions test	0.582411	Passed
Random excursions variant test	0.718984	Passed
Serial test	0.783850	Passed
Linear complexity test	0.697704	Passed

directly affects the security of encryption applications, they are crucial for information security. A good RNG consists of a uniform probability distribution of 1's and 0's, meaning the number of 1's and 0's in the bit stream should be equal or nearly equal. A PWLCM generates floating-point numbers with the range [0, 1). Therefore, there are infinite real number values in this range. To generate a random bit stream of 0's and 1's with a uniform probability distribution, a suitable threshold value is required to obtain continuous RNG output values. For this purpose, we chose the median value for the threshold, $\top = 0.5$, considering the range of RNG output values. The PWLCM is iterated one million times to generate a bit stream with a length of one million bits. The performance of the proposed RNG is evaluated with the NIST-800-22 statistical test suite [49] which includes 15 different tests. A bit stream one million bits in length is required for the NIST-800-22 statistical tests. For any random bit stream to be accepted as a successful and secure encryption key, it must pass all the tests. The random bit stream obtained from the proposed RNG using the PWLCM passed all the NIST tests, which are presented in Table 2 along with their *p*-values.

3 | PROPOSED S-BOX GENERATION METHODOLOGY

The proposed methodology to design an S-box using a PWLCM consists of four stages, as shown in Figure 2. In the first stage, the initial sequence is generated using the PWLCM. Real to

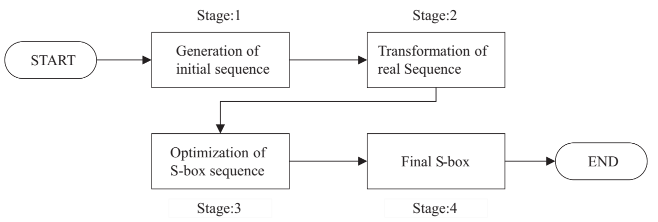

FIGURE 2 Proposed S-box design stages

integer domain transformation is performed during the second stage followed by integer S-box sequence optimization, which is the core part of the proposed S-box generation technique. In the last stage, the final S-box is produced. The details related to these stages are described in the following sections.

3.1 | Generation of initial sequence

The initial random sequence is generated using the PWLCM as described in Section 2. The randomness of the generated sequence is tested using the well-established NIST randomness test. The generated sequence is then transformed into an integer sequence, which is required for the auxiliary table. The following steps are involved in generating a random chaotic sequence with arbitrary input parameters.

- Set the initial conditions to $x_0 = [0, 1)$ and $p = (0, 0.5)$, which act as a key to iterate the chaotic map.
- Iterate (1) 50 times to remove the transient effect.
- Iterate (1) to generate the chaotic real values for the S-box sequence. This real value sequence is then used to generate the final S-box.

The steps involved in generating the real value chaotic sequence are illustrated Figure 3.

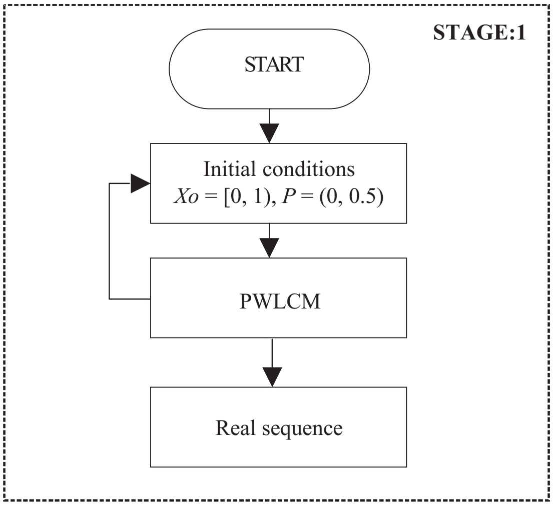

FIGURE 3 Generation of initial sequence for the proposed S-box

3.2 | Chaotic-cryptographic domain transformation

The chaotic real values obtained at the end of Stage 1 are transformed to a cryptographic integer domain that contains S-box positions within the range [1, 256] using (2).

$$X = \text{floor}\left(x' \times 256\right), \tag{2}$$

where X is the integer value within the range [1, 256], and x' is the real value range of [0,1].

The steps involved in real-integer domain transformation, as shown in Figure 4, are as follows:

- Using (2), convert the real values generated in the first stage, into integer values.
- Store the integer values in a row vector as S-box positions.
- When mapping the integer values to a row vector, check each generated integer value for repetition. If the value is repeated, ignore it and regenerate a new integer value.
- Continue the above steps until all the values are generated as S-box positions that follow the bijective property.

3.3 | Adaptive improvement of S-box positions

Chaotic maps are used to generate real value trajectories, and these real values are mapped to an integer domain as shown in the flow diagram in Figure 4. These trajectories are used to generate an S-box. For the real

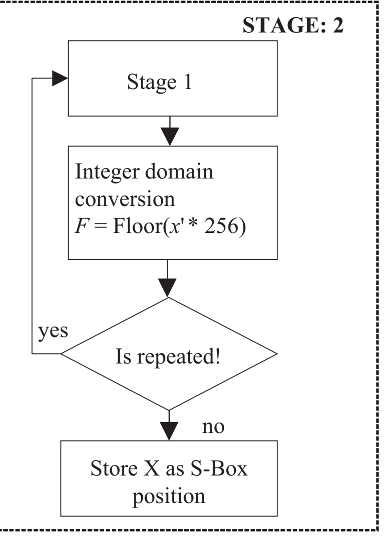

FIGURE 4 Real-integer domain transformation for the proposed S-box

domain, the Lyapunov exponent is used to measure the nonlinear behavior of chaotic trajectories. However, for the integer domain, the Hamming distance between elements defines the spread. In a typical chaos-based S-box design, the mapping from the real to integer domain is completely dependent on chaotic trajectories. Sometimes while mapping, the trajectory is in a position where the generated integer value falls very close to the previously mapped integer element, leading to a smaller spread value. Chaotic systems produce highly nonlinear trajectories with a positive Lyapunov exponent. However, during the transformation from the real to cryptographic domain, the inherent structure of chaotic trajectories often leads to a bad mapping or S-box position. These generated positions should be ignored to achieve near optimal cryptographic properties.

To adaptively improve the S-box positions, we propose a mapping criterion with a high spread of transformed S-box positions. This mapping criterion systematically ignores bad S-box positions that embody low nonlinearity and a high DP.

In this work, the Hamming distance between mapped integer values is kept greater than two. If any value is mapped at unity distance, it is regenerated to achieve the desired spread. Figure 5 shows the proposed mapping from the real (chaotic) domain to the integer (cryptographic) domain. The flow chart and procedure to ignore bad S-box positions are shown in Figure 6 and Figure 7, respectively.

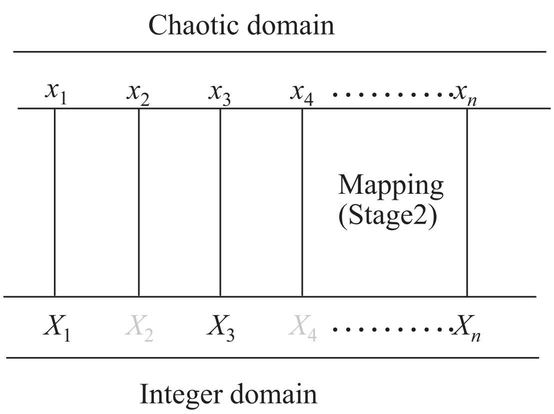

FIGURE 5 Mapping from the real domain to the integer domain

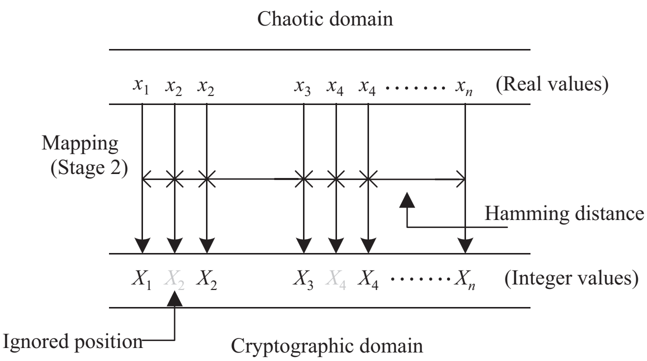

FIGURE 6 Mapping from the real domain to the integer domain using the proposed improvement criteria

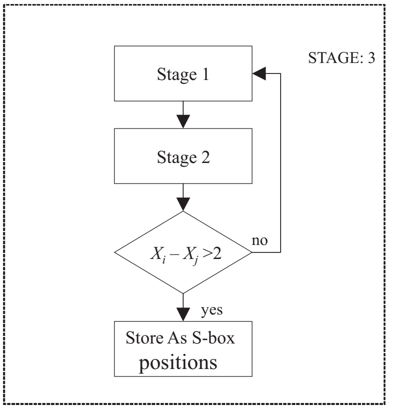

FIGURE 7 Optimization strategy for avoiding bad S-box positions

3.4 | Final S-box generation

In the last stage, the S-box sequence obtained from Stage 3 is rearranged in the form of a 16 × 16 table to produce the final S-box. The overall flow diagram of the proposed methodology is shown in Figure 8. The pseudo code of the proposed methodology is given in Table 3. The basic criteria for determining the performance of the proposed S-box are discussed in the next section.

4 | PERFORMANCE ANALYSIS OF PROPOSED S-BOX

It is important to measure the performance parameters of the proposed scheme to show its effectiveness in encryption algorithms. A cryptographically strong S-box has high resistance to a number of attacks, such as linear and differential cryptanalysis. In general, to achieve a strong S-box, a number of criteria should be satisfied such as bijection, nonlinearity, SAC, BIC, and linear and differential approximation probability. Due to the sensitive dependence on the initial conditions and system parameters, a slight change in initial values will generate an entirely different S-box. In the proposed model, the PWLCM with initial conditions $x_n = x_0 = 0.78$ and $p = .16$ is employed to generate the final S-box, which is shown in Table 4. The following sub-sections analyze the performance of the proposed S-box in detail.

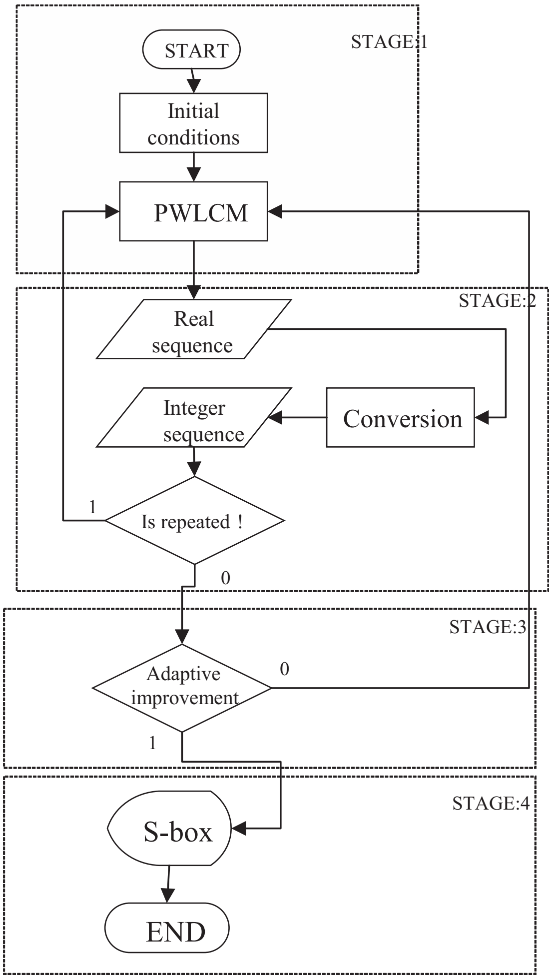

FIGURE 8 Proposed methodology for S-box design

TABLE 3 Pseudocode of proposed S-box design

Notations	
x_n	Initial real value
x_{n+1}	Next real value
X	Integer values
Max_iterations	Maximum number of iterations
S-box	Final S-box sequence
PWLCM	Piecewise linear chaotic map
Initialization Parameters	
$x_n = [0, 1)$	
$p = (0, 0.5)$	
$i = 1$	
Algorithm 1: Mainclass	
1: **Procedure** S-box	
2: **While** ($i <$ max_iterations) **Do**:	
3: iterate PWLCM with x_n	
4: set $x_{n+1} = x_n$	
5: X ← floor($x' \times 256$)	
6: **If** $X \notin$ S-box **then**	
7: S-box ← X	
8: $i = i + 1$;	
9: **Else**	
10: iterate PWLCM with x_n	
11: **End If**	
12: Call Algorithm Subclass	
13: **End While**	
14: Show S-box	
Algorithm 2: Subclass	
1: **Procedure** Optimization	
2: Call algorithm Mainclass	
3: Calculate: input_mutual_difference	
6: **If** input_mutual_difference > 2 **then**	
7: S-box ← X	
9: **Else**	
10: iterate PWLCM with x_n	
11: **End If**	

4.1 | Bijection

An S-box satisfies the bijection criteria if it has output values that are different from each other in the interval $[0, 2^{n-1}]$. Mathematically it is defined as:

$$wt\left(\sum_{i=1}^{n} a_i f_i\right) = 2^{n-1}, \tag{3}$$

where wt is the Hamming weight, f_i dentoes the Boolean functions, and $a_i \in \{0, 1\}$. 2^n is the total number of entries. The proposed S-box generates distinct output values in the interval [0, 255], and the Hamming weight of all Boolean functions in this proposed S-box is 128, which is bijective.

4.2 | Nonlinearity

The Walsh spectrum is used to measure the nonlinearity of the Boolean functions. For symmetric Boolean functions, the maximum nonlinearly achieved is 112, and nonlinear values above 98 are considered good. Mathematically, Boolean function nonlinearity is measured as follows:

$$N_f = 2^{m-1}\left(1 - 2^{-m}\max\left|S_{(g)}(w)\right|\right), \tag{4}$$

where $S_{(g)}(w)$ is the Walsh spectrum, which is defined as:

$$S_{(g)}(w) = \sum_{W \in GF(2^m)} (-1)^{g(x) \oplus x.w} \tag{5}$$

where $w \in GF(2^m)$, and $x.w$ is the dot product.

The nonlinearity values achieved for the proposed S-box and some well-known existing S-boxes are given in Table 5. The proposed S-box successfully achieves nonlinearity values comparable with those of the existing chaotic S-boxes.

4.3 | SAC

Webster and Tavares [47] introduced the SAC to check the strength of a cryptosystem. Satisfying the SAC means that when a single input bit is modified, approximately one-half of the output bits change. The method to measure the SAC is given in [50,51]. A dependence matrix is calculated to test the SAC. The expression used to calculate dependence matrix is given in (6).

$$p_{i,j}(f) = \frac{1}{2^n} \sum_{x \in b^n} f_j(x) \oplus f_j(x \oplus e_i). \tag{6}$$

The SAC value of the proposed S-box is 0.5160, and the ideal value of SAC is 0.5, which is very close to the proposed S-box value, demonstrating that the proposed S-box satisfies the strict avalanche criteria.

4.4 | BIC

The BIC, introduced by Webster and Tavares [3], is another cryptosystem property. It requires that output bits have no correlation with each other, and all input-output variables are pairwise independent for all avalanche vectors. These vectors are generated by inverting one plaintext bit at a time [20]. By using Boolean functions in the algorithm, we can assume that the S-box either satisfies the BIC or it does not. The BIC is calculated using the relation $f_i \bigoplus f_j$ $(i \neq j;\ 1 \leq i; j \leq n)$. The correlation strength between the input-output pair determines the level of independence between all avalanche pairs. The correlation can be represented as follows:

$$\rho\,(A, B) = \frac{(\mathrm{Cov}\,(A, B))}{(\sigma\,(A)\,\sigma\,(B))}, \tag{7}$$

where $\rho(A, B)$ is the correlation, and $\mathrm{Cov}(A, B)$ is the covariance coefficient of A and B. The average BIC-nonlinearity value of the proposed approach is 103.5 as shown in Table 6, which is comparable to the well-known chaotic S-boxes mentioned in Table 5.

TABLE 4 Proposed S-box

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

TABLE 5 Nonlinearity value comparison of the proposed S-boxes with well-known chaotic S-boxes

Study	0	1	2	3	4	5	6	7	Mean
[13]	104	100	106	102	104	102	104	104	103.2
[16]	106	108	108	106	106	106	106	106	106.5
[17]	108	100	106	108	106	104	102	104	104.7
[19]	108	110	102	102	100	106	106	106	105.5
[22]	104	108	106	106	106	104	106	104	105.5
Prop.	106	108	106	104	102	106	100	100	104.0

TABLE 6 BIC-nonlinearity matrix for the proposed S-box

0	1	2	3	4	5	6	7
0	106	102	102	100	106	96	100
106	0	104	104	104	106	100	104
102	104	0	108	104	106	106	106
102	104	108	0	102	102	106	104
100	104	104	102	0	104	102	102
106	106	106	102	104	0	102	104
96	100	106	106	102	102	0	106
100	104	106	104	102	104	106	0

4.5 | Differential approximation probability

Differential cryptanalysis searches for any structural weaknesses in a cryptosystem by finding the highest output difference whose input differences are in the range of [0, $n-1$]. DP is a measure of identical mapping for each input differenceΔx to output difference Δy[52]. A cryptographically strong S-box must have differential uniformity. The differential approximation probability is mathematically defined as:

$$DP(\Delta x \to \Delta y) = \left(\frac{\{x \in X | S(x) \oplus S(x \oplus \Delta x) = \Delta y\}}{2^n} \right), \quad (8)$$

where X is the set of possible input values, and 2^n is the total number of entries. The output differential distribution of the proposed S-box is shown in Figure 9. The maximum difference distribution value for the proposed S-box obtained after optimization is 10/256 = 0.0390. However, 99.9% of the values in the table are less than 0.0390, demonstrating that the proposed S-box is highly resistant to differential cryptanalysis.

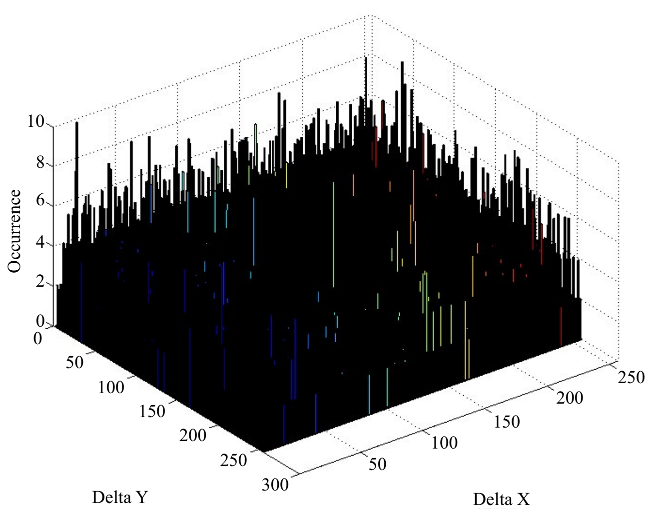

FIGURE 9 Differential distribution table of the proposed S-box (x-axis: input XORs, y-axis: output XORs, z-axis: number of occurrences) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 7 Comparison of DP values achieved for the proposed S-box

DP values	With optimization	Without optimization
2/256 = 0.0078	19 385	19 722
4/256 = 0.0156	5042	4961
6/256 = 0.0234	884	837
8/256 = 0.0312	121	105
10/256 = 0.0390	7	13

In Table 7, a comparison of the DP values achieved for the proposed S-box both with and without optimization is performed to show the effectiveness of the proposed optimization technique. In the 256 × 256 matrix of the difference distribution table, we calculated the DP values. With the optimization technique, the proposed S-box provides strong resistance against differential cryptanalysis. Table 8 compares the performance of the proposed S-box with some existing S-boxes that are based on chaos [17,20–22]. The table also compares the performance of the proposed S-boxes with algebraic S-boxes, such as the Advance Encryption Standard (AES), Affine-Power-Affine (APA) [53], Gray [54], Skipjack [55], Xyi [56], and Residue Prime [57] S-boxes. Based on the table, the performance of the proposed S-box is comparable to the existing chaos-based S-boxes.

5 | CHAOTIC CRYPTOSYSTEM DESIGN METHODOLOGY

In this scheme, each pixel value is first substituted using the proposed S-box. The achieved confusion and key effect generated for the R, G, and B channel using multiple PLCMs is then diffused over all other pixels. The steps involved in designing the proposed encryption algorithm, given in Figure 10, are described below.

1. The two-dimensional R, G, and B channel matrices are first converted into an ID stream. The pixel values of an image are read column-wise and sequentially, which is the established reading pattern for encryption and decryption.
2. A unique chaotic map initial condition is generated from the external secret key. The initial condition is used to generate masking keys for the R, G, and B channels. The external secret key is a 128-bit ASCII form denoted by

$$K = K_1 K_2 K_3 \cdots K_{16}. \quad (9)$$

Each K_i is an 8-bit block of secret key. The corresponding initial condition IC_i for all 16-bit secret key blocks is generated as follows.

$$IC_i = {K_i}/{256}. \quad (10)$$

For the PWLCM given in (1) with parameters IC_i and P_i, the unique initial condition is derived as follows:

$$R = \sum_{i=1}^{16} PWLCM_i^{K_i}(IC_i, p_i), \quad (11)$$

$$IC = R \bmod 1. \quad (12)$$

R is the real valued output for a given IC_i and P_i, and K_i is the number of iterations for the ith chaotic map with the given initial parameters of IC_i and P_i. The chosen P matrix to generate unique initial condition is given as:

$$\begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \\ p_{13} & p_{14} & p_{15} & p_{16} \end{bmatrix} = \begin{bmatrix} 0.15 & 0.1535 & 0.1555 & 0.1666 \\ 0.27 & 0.2745 & 0.2755 & 0.2777 \\ 0.43 & 0.4344 & 0.444 & 0.4377 \\ 0.45 & 0.4566 & 0.4576 & 0.4578 \end{bmatrix}. \quad (13)$$

3. The derived unique initial IC conditions with P_1, P_2, and P_3 are used as input parameters of the PWLCM to generate three chaotic key streams for masking the R, G, and B channels. The three PWLCMs are:

$$\begin{aligned} PWLCM_1 &= PWLCM_1(IC, p_1), \\ \text{PWLCM}_2 &= PWLCM_2(IC, p_2), \\ \text{PWLCM}_3 &= PWLCM_3(IC, p_3). \end{aligned} \quad (14)$$

The chaotic masking keys using the given chaotic maps are derived as follows:

$$K_k = \lfloor PWLCM_k * (10^n) \rfloor \ (\bmod 256) \text{ where } \quad k = 1, 2, 3. \quad (15)$$

The three key streams are generated using the given equations for the R, G, and B channels as K_1, K_2, and K_3, respectively.

TABLE 8 Performance comparison of existing S-boxes and the proposed S-box

Technique	NL	SAC	BIC-NL	DP
Lambic' et al. [17]	106.5	0.4978	104.2	0.0390
Özkaynak et al. [20]	104.7	0.4982	103.1	0.0390
Çavuşoğlu et al. [21]	106.0	0.5058	103.3	0.0390
Belazi et al. [22]	105.5	0.5000	103.7	0.0468
AES [58]	112.0	0.5058	112.0	0.0156
APA [53]	112.0	0.4987	112.0	0.0156
Gray [54]	112.0	0.5058	112.0	0.0468
Skipjack [55]	105.7	0.4980	104.1	0.0468
Xyi [56]	105.0	0.5048	103.7	0.2810
Residue Prime [57]	99.5	0.5012	101.7	0.0156
Proposed	104.0	0.5160	103.5	0.0390

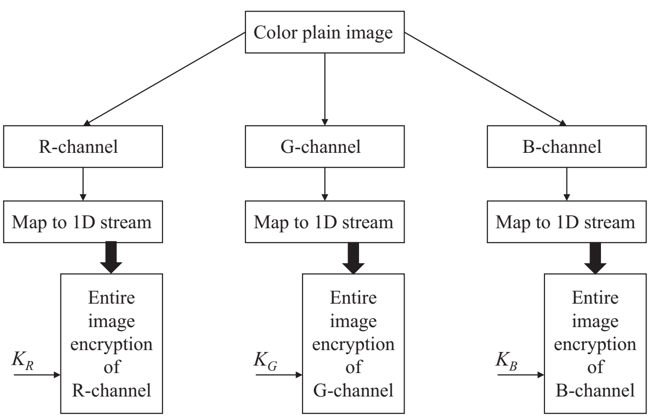

FIGURE 10 Conceptual flow diagram of the proposed cryptosystem

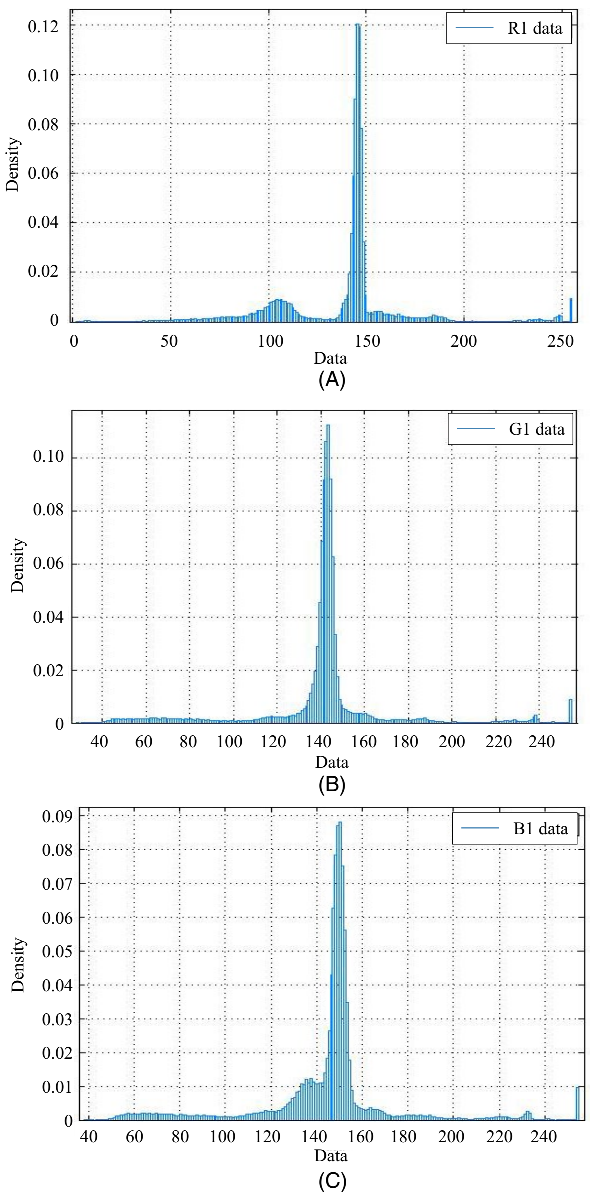

FIGURE 11 Histogram of plain image: (A) R-channel, (B) G-channel, and (C) B-channel [Colour figure can be viewed at wileyonlinelibrary.com]

4. The encrypted data are generated using substitution mixing and masking. Substitution is performed using the proposed S-box. Substituted data are masked using the generalized chaotic map with the generated key and pixel values as inputs. The generalized logistic map is

$$f(y_i) = \begin{cases} \left\lfloor \left\lfloor \frac{y_i(256-y_i)}{64} \right\rfloor \right\rfloor & \tilde{y}_i < 256, \\ 255 & \tilde{y}_i = 256. \end{cases} \tag{16}$$

5. Diffusion acts as the core resistance against statistical attacks by spreading the influence of a single bit over complete ciphertext. The generalized logistical map with ciphertext feedback introduces the diffusion, and the complete process of diffusion with key addition and mixing is given as follows:

$$\begin{aligned}
&\underbrace{sx_1 \oplus f(k_R(1) \oplus x'_N)}_{x'_1}, \underbrace{sx_2 \oplus f(K_R(2) \oplus sx_1)}_{x'_2}, \underbrace{sx_3 \oplus f(K_R(3) \oplus x'_2}_{x'_3}, \ldots\ldots, \\
&\underbrace{sx_N \oplus f(K_R((k \bmod L)+1) \oplus x'_{N-1})}_{x'_N}, \\
&\underbrace{sy_1 \oplus f(K_R(1) \oplus y'_N)}_{y'_1}, \underbrace{sy_2 \oplus f(K_G(2) \oplus sy_1)}_{y'_2}, \underbrace{sy_3 \oplus f(K_G(3) \oplus y'_2)}_{y'_3}, \ldots\ldots, \\
&\underbrace{sy_N \oplus f(K_G((k \bmod L)+1) \oplus y'_{N-1})}_{y'_N},
\end{aligned} \tag{17}$$

and

$$\begin{aligned}
&\underbrace{sz_1 \oplus (K_B(1) \oplus z'_N)}_{z'_1}, \underbrace{sz_2 \oplus f(k_G(2) \oplus sz_1)}_{z'_2}, \underbrace{sz_3 \oplus f(K_G(3) \oplus z'_2}_{z'_3}, \ldots., \\
&\underbrace{sz_N \oplus f(K_B(k \bmod L)+1) \oplus z'_{N-1}}_{z'_N}
\end{aligned}$$

$x'_i, y'_i,$ and z'_i are the updated pixel values, and L is the length of K_i.

6. The decryption process is the inverse of the encryption process. All S-box, chaotic mixing, and masking processes are reversible to recover the plaintext.

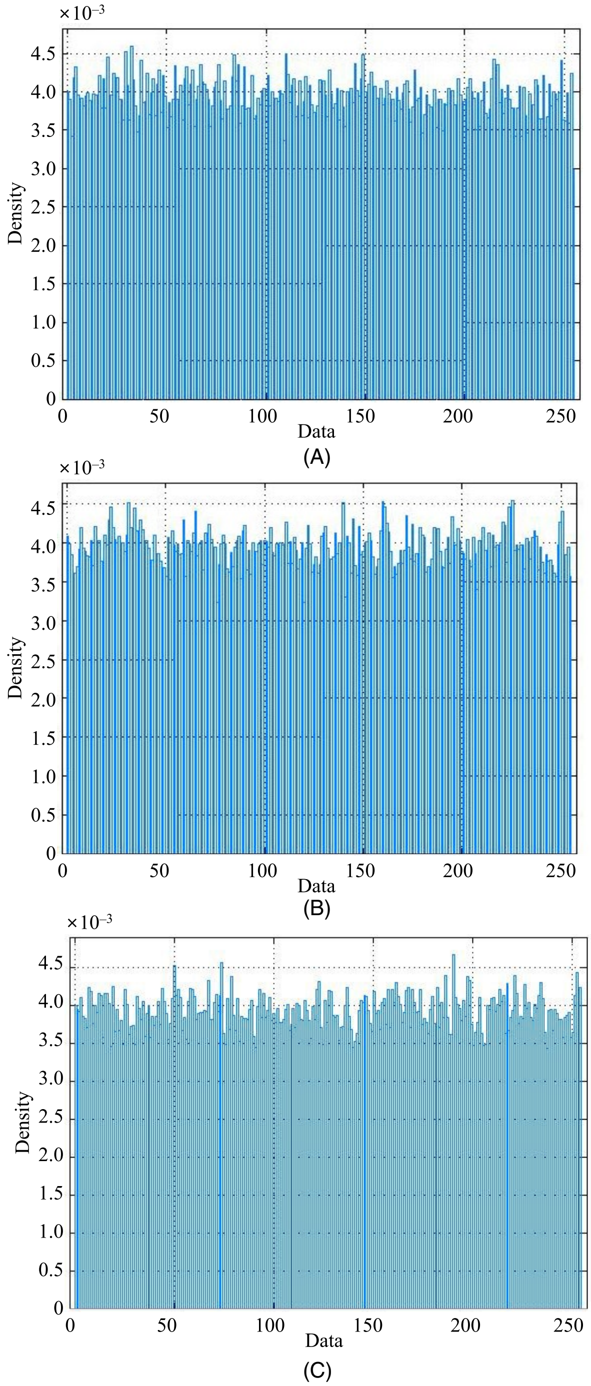

FIGURE 12 Histogram of the ciphered image: (A) R-channel, (B) G-channel, and (C) B-channel [Colour figure can be viewed at wileyonlinelibrary.com]

6 | SECURITY ANALYSIS OF CHAOTIC CRYPTOSYSTEM

This section covers the security analysis of the proposed encryption algorithm. The statistical analysis, key, and plaintext-related sensitivity analyses are presented in detail.

6.1 | Histogram analysis

Image histograms provide us with information of pixel distributions by plotting the pixel frequency for each colored band. The histograms of plain and ciphered images are compared for the analysis. The image histograms are taken from [http://sipi.usc.edu/database/] and shown in Figures 11A–C. Based on Figure 10, the proposed scheme effectively randomizes the plain image. The normalized mean square error (NMSE) is calculated as follows:

$$\mathrm{NMSE} = \frac{1}{N} \sum_k \left(\frac{X_k - \overline{X}}{\overline{X}} \right)^2, \tag{18}$$

where N is the number of bins, X_k is the frequency of occurrence of each bin, and $\overline{X}$ is the mean frequency of each bin.

The NMSE for the ciphered images in Figures 12A–C are 0.0041, 0.0038, and 0.0035, respectively, indicating the deviation from the uniform probability distribution. Given the provided figures and NMSE values, it is evident that statistical attacks on the proposed image encryption would be very difficult.

6.2 | Correlation coefficient analysis

An image correlation analysis between adjacent pixels is conducted with arbitrary large frames of gray scale values taken from the R, G, and B channels of both plain and encrypted images. The correlation coefficient between the adjacent pixels is calculated as:

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \overline{A}\right)^2\right)\left(\sum_m \sum_n \left((B_{mn} - \overline{B}\right)^2\right)}}, \quad (19)$$

where $\overline{A}$ and $\overline{B}$ correspond to the mean values, r corresponds to the normalized correlation between image A_{mn} & B_{mn} pixel by pixel. The plain image [4.1.01.tiff] is obtained from the database [http://sipi.usc.edu/database/]. The normalized correlation r, for the R, G, and B channels of the plain image are 0.9021, 0.9233, and 0.9059, respectively. The horizontal correlation of the R, G, and B channels of the ciphered image are 0.0021, –0.0045, and 0.0041, respectively. Based on the results, the pixels are uniformly distributed in the internal 0 to 255 and show a negligible correlation.

6.3 | Key sensitivity analysis

The proposed scheme is extremely sensitive to the changes in the key. Thus, single bit position changes in the encryption or decryption key result in a completely different ciphered image; hence, it is infeasible to decrypt the ciphered image. To test the key sensitivity, the plain image is encrypted with the key "abcdefghijklmnop" and decryption with a slightly wrong key "abcdefghijklmo**o**" is attempted. The decrypted image with the slightly wrong key along with its histogram is shown in Figure 13, showing that it is not feasible to decrypt the image with a slightly wrong key.

6.4 | Sensitivity to plain image

The number of pixel change rate (NPCR) is calculated to test the avalanche effect by measuring the influence of a one-pixel change on the ciphered image. For the test, two ciphered images C_1 and C_2 whose corresponding plain images are only one pixel different are considered. A bipolar array $D(i, i)$ is defined such that $D(i, j) = 0$, under condition $C_1(i, j) = C_2(i, j)$ and vice versa. The NPCR is defined as

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad (20)$$

where W and H are the width and height of C_1 and C_2. The comparison of NPCR with other algorithms is given in Table 9. The proposed scheme efficiently introduces diffusion and confusion to help resist differential cryptanalysis.

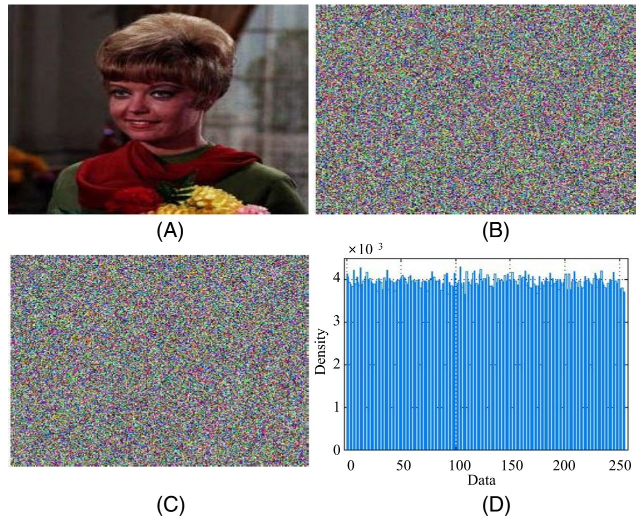

FIGURE 13 Key sensitivity analysis for a (A) plain image, (B) ciphered image encrypted using "abcdefghijklmnop", (C) decrypted image with slightly wrong key "abcdefghijklmno**o**", and (D) histogram of image decrypted with slightly wrong key [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 9 Comparison of NPCR with other algorithms

Schemes	NPCR (%)
Proposed	99.621
Ref [32]	99.617
Ref [33]	99.150
Ref [39]	99.630

7 | CONCLUSION

In this study, a simple method is proposed to design an S-box based on a piecewise linear chaotic map followed by an adaptive improvement technique. The proposed framework for chaos-based S-box generation minimizes the DP value. Furthermore, to design a robust S-box, we eliminate the trade-offs with other S-box parameters. Therefore, the performance assessment of the proposed S-box shows that it is resistant to linear and differential cryptanalysis. Additionally, the proposed S-box difference distribution table highlights the maximum differential approximation probability value compared to state-of-the-art encryption algorithm methods. The suitability of the proposed S-box is also tested by using it in an image encryption algorithm. Results show that the proposed method to design the S-box as a confusion component in the proposed encryption algorithm efficiently encrypts the plaintext, which provides a high level of security and resistance against all known attacks.

REFERENCES

1. C. E. Shannon, *Communication theory of secrecy systems*, Bell Syst. Tech. J. **28** (1949), 656–715.
2. W. Meier and O. Staffelbach, *Nonlinearity criteria for cryptographic functions*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in, Bioinformatics).1990, pp. 549–562.
3. A. F. Webster and S. E. Tavares, *On, the Design of S-Boxes*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1986, pp. 523–534.
4. M. Matsui, *New structure of block ciphers with provable security against differential and linear cryptanalysis*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1996, pp. 205–218.
5. S. Hong et al., *Provable security against differential and linear cryptanalysis for the SPN structure*, in Proc. Fast Softw. Encryption (New York, NY, USA), 2000, pp. 273–283.
6. H. M. Heys, *A tutorial on linear and differential cryptanalysis*, Cryptologia. **26** (2002), 189–221.
7. L. Kocarev, *Chaos-based cryptography: A brief overview*, IEEE Circuits Syst. Mag. **1** (2001), 6–21.
8. G. Jakimoski and L. Kocarev, *Chaos and cryptography: Block encryption ciphers based on chaotic maps, IEEE Trans*, Circuits Syst. I Fundam. Theory Appl. **48** (2001), 163–169.
9. Y. Wang et al., *A block cipher with dynamic S-boxes based on tent map*, Commun. Nonlinear Sci. Numer. Simul. **14** (2009), 3089–3099.
10. G. Tang, X. Liao, and Y. Chen, *A novel method for designing S-boxes based on chaotic maps*, Chaos, Solitons Fractals. **23** (2005), 413–419.
11. G. Chen, Y. Chen, and X. Liao, *An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps*, Chaos, Solitons Fractals. **31** (2007), 571–579.
12. M. Asim and V. Jeoti, *Efficient and simple method for designing chaotic S-boxes*, ETRI J. **30** (2008), 170–172.
13. F. Özkaynak and A. B. Özer, *A method for designing strong S-Boxes based on chaotic Lorenz system*, Phys. Lett. Sect. A Gen. At. Solid State Phys. **374** (2020), 3733–3738.
14. M. Ahmad et al., *Designing chaos based strong substitution box*, in Proc. Int. Conf. Contemporary Comput (Noida, India), 2015, pp. 97–100.
15. T. Farah, R. Rhouma, and S. Belghith, *A novel method for designing S-box based on chaotic map and Teaching–Learning-Based Optimization*, Nonlinear Dyn. **88** (2017), 1059–1074.
16. I. Hussain et al., *Construction of chaotic quantum magnets and matrix Lorenz systems S-boxes and their applications*, Chinese J. Phys. **56** (2018), 1609–1621.
17. D. Lambić, *S-box design method based on improved one-dimensional discrete chaotic map*, J. Inf. Telecommun. **2** (2018), 1–11.
18. M. Khan and Z. Asghar, *A novel construction of substitution box for image encryption applications with Gingerbreadman chaotic map and S8 permutation*, Neural Comput. Appl. **29** (2018), 993–999.
19. Ü. Çavuşoğlu et al., *Secure image encryption algorithm design using a novel chaos based S-Box*, Chaos, Solitons Fractals. **95** (2017), 92–101.
20. F. Özkaynak, V. Çelik, and A. B. Özer, *A new S-box construction method based on the fractional-order chaotic Chen system*, Signal, Image Video Process. **11** (2017), 659–664.
21. Ü. Çavuşoğlu et al., *A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system*, Nonlinear Dyn. **87** (2017), 1081–1094.
22. A. Belazi and A. A. A. El-Latif, *A simple yet efficient S-box method based on chaotic sine map*, Optik (Stuttg). **130** (2017), 1438–1444.
23. Y. Tianand and Z. Lu, *Chaotic S-Box: Intertwining logistic map and bacterial foraging optimization*, Math. Probl. Eng. **2017** (2017), 1–11.
24. M. Khan, T. Shah, and S. I. Batool, *Construction of S-box based on chaotic Boolean functions and its application in image encryption*, Neural Comput. Appl. **27** (2016), 677–685.
25. R. Guesmi et al., *Chaos-based designing of a highly nonlinear S-box using Boolean functions*, in Proc. Int. Multi-Conf. Syst., Signals Devices (Mahdia, Tunisia), 2015, pp. 1–5.
26. M. Ahmad and S. Alam, *A novel approach for efficient S-box design using multiple high-dimensional chaos*, in Proc. Int. Conf. Adv. Comput. Commun. Technol. (Rohtak, India), 2014, pp. 95–99.
27. M. Khan, T. Shah, and M. A. Gondal, *An efficient technique for the construction of substitution box with chaotic partial differential equation*, Nonlinear Dyn. **73** (2013), 1795–1801.
28. F. Özkaynak and S. Yavuz, *Designing chaotic S-boxes based on time-delay chaotic system*, Nonlinear Dyn. **74** (2013), 551–557.
29. M. Khan et al., *A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems*, Nonlinear Dyn. **70** (2012), 2303–2311.
30. T. Shah Hussain and M. A. Gondal, *A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm*, Nonlinear Dyn. **70** (2012), 1791–1794.
31. J. Peng et al., *Efficient chaotic nonlinear component for secure cryptosystems*, in Proc. Int. Conf. Ubiquitous Future Netw. (Milan, Italy), 2017, pp. 989–993.
32. A. Belazi et al., *Selective image encryption scheme based on DWT, AES S-box and chaotic permutation*, in Proc. Int. Wireless Commun. Mobile Comput. Conf. (Dubrovnik, Croatia), 2015, pp. 606–610.
33. A. A. A. El-Latif, B. Abd-El-Atty, and S. E. Venegas-Andraca, *A novel image steganography technique based on quantum substitution boxes*, Opt. Laser Technol. **116** (2019), 92–102.
34. N. Munir and M. Khan, *A generalization of algebraic expression for nonlinear component of symmetric key algorithms of any characteristic*, in Proc. Int. Conf. Appl. Eng. Math. (Tazila, Pakistan), 2018, pp. 48–52.
35. A. A. A. El-Latif et al., *A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces*, Signal Process. **93** (2013), 2986–3000.
36. A. A. A. Belazi, A. A. El-Latif, and S. Belghith, *A novel image encryption scheme based on substitution-permutation network and chaos*, Signal Process. **128** (2016), 155–170.
37. S. I. Batool, T. Shah, and M. Khan, *A color image watermarking scheme based on affine transformation and S4 permutation*, Neural Comput. Appl. **25** (2014), 2037–2045.

38. A. Belazi et al., *Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption*, Nonlinear Dyn. **87** (2017), 337–361.
39. M. Khan, T. Shah, and S. I. Batool, *A new implementation of chaotic S-boxes in CAPTCHA*, Signal, Image Video Process. **10** (2016), 293–300.
40. M. Khan, *A novel image encryption scheme based on multiple chaotic S-boxes*, Nonlinear Dyn. **82** (2015), 527–533.
41. M. Khan, *A novel image encryption using Fourier series*, Journal Vib. Contr. **21** (2015), 3450–3455.
42. A. A. A. El-Latif and X. Niu, *A hybrid chaotic system and cyclic elliptic curve for image encryption*, AEU Int. J. Electron. Commun. **67** (2013), 136–143.
43. M. Khan, T. Shah, and S. I. Batool, *A new approach for image encryption and watermarking based on substitution box over the classes of chain rings*, Multimed. Tools Appl. **76** (2017), 24027–24062.
44. M. Khan and T. Shah, *A copyright protection using watermarking scheme based on nonlinear permutation and its quality metrics*, Neural Comput. Appl. **26** (2015), 845–855.
45. H. M. Waseem and M. Khan, *A new approach to digital content privacy using quantum spin and finite-state machine*, Appl. Phys. B Lasers Opt. **125** (2019).
46. M. Khan and H. M. Waseem, *A novel image encryption scheme based on quantum dynamical spinning and rotations*, PLoS ONE **13** (2018), 1–23.
47. H. M. Waseem, M. Khan, and T. Shah, *Image privacy scheme using quantum spinning and rotation*, J. Electron. Imaging. **27** (2018), 1–13.
48. I. Younas and M. Khan, *A new efficient digital image encryption based on inverse left almost semi group and Lorenz chaotic system*, Entropy. **20** (2018), 1–22.
49. E. Bassham et al., *A statistical test suite for random and pseudo-random number generators for cryptographic applications*, NIST SP-800-22 Rev 1a, 2010.
50. Y. Su et al., *A Method for obtaining chaos-based S-Box via a PWLCM*, Adv. Mater. Res. **651** (2013), 885–890.
51. S. Li, G. Chen, and X. Mou, *On the dynamical degredation of digital piecewise linear chaotic maps*, Int. J. Bifurc. Chaos. **15** (2005), 3119–3151.
52. E. Biham and A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*, J. Cryptol. **4** (1991), 3–72.
53. L. Cui and Y. Cao, *A new S-box structure named affine-power-affine*, Int. J. Innov. Comput, Inf. Contr. **3** (2007), 751–759.
54. T. Tran, D. K. Bui, and A. D. Duong, *Gray S-box for Advanced Encryption Standard*, in Proc. Int. Conf. Comput. Intell. Security (Suzhou, China), 2008, pp. 253–258.
55. J. Kim and R. C. W. Phan, *A cryptanalytic view of the NSA's skipjack block cipher design*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2009, pp. 368–381.
56. X. Yi et al., *A method for obtaining cryptographically strong 8/spl times/8 S-boxes*, in Proc. IEEE Global Telecommun. Conf. (Phoenix, AZ, USA), 1997, pp. 689–693.
57. E. Abuelyman and A. A. S. Alsehibani, *An optimized implementation of the S-Box using residue of prime numbers*, Int J Comput Sci Network Security **8** (2008), 304-309.
58. J. Daemen and V. Rijmen, *The Design of Rijndael*, The Advanced Encryption Standard, 2002.

AUTHOR BIOGRAPHIES

Zaid Bin Faheem received his BSc and MS degree in computer engineering from the University of Engineering and Technology, Taxila, Pakistan in 2014 and 2018, respectively. He is currently working as a lecturer in the Computer Science Department of the government college the University of Sahiwal, Pakistan. His main research interests are chaos-based cryptography, watermarking, and information security.

Asim Ali is currently working as an assistant professor in the Computer Science Department of the University of Wah, Wah Cantonment, Pakistan. He received his BS degree in computer science from the Allama Iqbal Open University, Islamabad, Pakistan in 2001. He received his MSc and MS in computer science from the Comsats Institute of Information Technology, Wah Cantonment in 2004 and 2007, respectively. He is currently pursuing his PhD in computer science from the Comsats Institute of Information Technology, Wah Cantonment, Pakistan. His area of specialization is cryptography. His research interests are information security, software engineering, computational theory, and chaos-based cryptography.

Muhamad Asif Khan is currently working as an Assistant Professor in the Computer Engineering Department of the University of Engineering and Technology, Taxila, Pakistan. He received his BSc degree in computer engineering from the University of Engineering and Technology, Taxila, Pakistan in 2005. He completed his MS and PhD in Electrical and Electronic Engineering at the University Teknologi PETRONAS, Malaysia in 2009 and 2015, respectively. From 2005 to 2006, he was a lecturer for the University of Engineering and Technology, Taxila, Pakistan. From 2006 to 2013, he was a research assistant for the Electrical and Electronic Engineering Department, Universiti Teknologi PETRONAS. From 2013 to 2015, he worked as a research Scientist with PETRONAS SD BHD for the WiDUCT Project. His field of specialization is chaos-based cryptography and wireless PHY layer security. His research interests are blockchain, watermarking, multimedia security, and lightweight cryptography.

Muhammad Ehatisham Ul-Haq received his BSc and MSc degrees in computer engineering from the University of Engineering and Technology, Taxila, Pakistan in 2014 and 2017 and won the Gold Medal and Chancellor's Gold Medal during his BSc and MSc programs, respectively. He is currently pursuing his PhD in computer engineering at UET Taxila, Pakistan. His field of specialization is pervasive and ubiquitous computing. His research interests are signal, image, and video processing, biomedical signal processing, mobile sensing, machine learning, human activity and emotion recognition, and human behavior analysis.

Waqar Ahmad received the BSc and MSc degrees in Computer Engineering from COMSATS University Abbottabad and University of Engineering and Technology Taxila, Pakistan, respectively. He did his PhD degree from the Department of Electronics and Telecommunications, Politecnico di Torino, Italy. He was a Higher Education Commission (HEC) fully funded PhD scholar. He is currently an Assistant Professor with University of Engineering and Technology, Pakistan. His current research interests focus on developing machine learning algorithms and architectures for video coding and multi-camera networks. He serves as reviewer for Journal of Circuits, Systems and Computers.