IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 29, 2020, accepted February 6, 2020, date of publication February 13, 2020, date of current version February 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973679

Generation of Highly Nonlinear and Dynamic AES Substitution-Boxes (S-Boxes) Using Chaos-Based Rotational Matrices

MUHAMMAD SARMAD MAHMOOD MALIK[1], MUHAMMAD ASIM ALI[2], MUHAMMAD ASIF KHAN[1], MUHAMMAD EHATISHAM-UL-HAQ[1], SYED NASIR MEHMOOD SHAH[3], MOBASHAR REHMAN[4], AND WAQAR AHMAD[1]

[1]Department of Computer Engineering, University of Engineering and Technology (UET), Taxila 47050, Pakistan
[2]Department of Computer Science, University of Wah, Wah Cantt 47040, Pakistan
[3]Department of Computer Science KICSIT, Institute of Space Technology, Islamabad 44000, Pakistan
[4]Department of Information Systems, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar 31900, Malaysia

Corresponding author: Mobashar Rehman (mobashar@utar.edu.my)

This work was supported by the Universiti Tunku Abdul Rahman.

ABSTRACT This work reports a novel chaos-based affine transformation generation method, which is based on rotational matrices to design strong key-based S-boxes. Chaotic logistic map's nonlinear trajectories are used to generate rotational matrices under given design conditions. Thus, the inherent logic is to generate key-based S-boxes, as strong as AES S-box, in terms of cryptographic properties using chaos in affine transformation. The randomness of chaotic sequences is tested using the National Institute of Standard and Technology (NIST) Statistical Test Suit (STS) 800-22 that validates the generated sequences for S-box design. The results show that methodology adapted to design proposed key-based dynamic S-boxes entails near-optimal cryptographic properties so that proposed S-boxes are as stronger as AES S-box.

INDEX TERMS Affine transformation, chaotic logistic map, S-box, NIST test.

I. INTRODUCTION

Cryptography plays a critical role in secure transmission of information. With the increasing demand for communication systems, the role of encryption becoming more critical and cryptographers continue to work on new algorithms to ensure secure transmission of confidential information. Likewise, cryptanalysts are hard at work to find new ways of breaking those algorithms. Advanced Encryption Standard (AES) is the standard algorithm approved by the National Institute of Standards and Technology (NIST). Although no attack still exist that can break AES, there are some attacks like side-channel attack [1], which exploits the incomplete diffusion feature in AES [2] and SQUARE [3]. The meet-in-the-middle attack exploits the weakness in the key scheduling [4] in AES. The advanced capabilities of cryptanalysis demand cryptographers to modify cryptographic algorithms. According to Shannon, modern block ciphers are based on confusion and diffusion components. Confusion is achieved in encryption algorithms by substitution-box (S-box). The primary function of S-box is to change the position of the input block of data x with a new position of data block named as y, in a nonlinear fashion. There are many applications of S-box in image encryption [5], [6], low profile mobile applications [7], multimedia encryption [8], watermarking [9], and steganography [10]. In encryption, S-box is commonly used in modern block ciphers like Data Encryption Standard (DES) and AES. It is responsible for providing confusion property in algorithms. The strength of the algorithm depends on the strength of the S-box [11], [12]. NIST defined the criteria for finding the strength of an S-box. These include bit independence criterion (BIC), nonlinearity (NL), strict avalanche criterion (SAC), linear probability (LP), and differential probability (DP). Generally, S-box with low DP value and high non-linearity is desirable.

The non-linearity of an S-box causes uncertainty in the output, which offers resistance against linear and differential cryptanalysis attacks [13]. AES S-box is considered highly

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran Tariq.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

strong as it is based on finite field and $GF(2^8)$ inverse operations that play critical role to made S-box highly non-linear, thus resulting in higher security.

Researchers have proposed different methods to design S-box to achieve high nonlinearity and low DP value. Chaos, due to strong similarities with cryptography, is considered as a good candidate in designing S-box. Chaos, with inherent properties like sensitive dependence on initial condition, mixing, and ergodicity, has attracted different researchers [14]. The authors proposed S-boxes using one dimensional [15], [16], two dimensional [17], and three dimensional [18]–[20] chaotic maps. Tian and Lu [21] proposed an S-box design based on a six-dimensional hyperchaotic map and an artificial bee colony algorithm. Tanyildizi and Özkaynak [22] utilized seven (07) different optimization algorithms to determine the most suitable initial condition and control parameter values corresponding to four chaotic algorithms. In addition, they proposed a new method for S-box generation design based on the optimized chaotic maps. However, their proposed S-box is not as strong as AES S-box. The authors in [23] examined the properties of a three-dimensional plasma system from a cryptographic perspective and proposed a new S-box generation algorithm based on a 3D plasma chaotic system. In [24], the authors designed a new S-box for wireless sensor networks using Linear Congruence Generator (LCG) in combination with compound chaotic map, Baker's map, and sinusoidal chaotic map. Khan *et al.* [25] proposed a new cryptographic method for substitution-permutation network design, where the results of gold sequences and chaotic logistic tent maps are used in linear fraction transformation to generate a new S-box. Lu *et al.* [26] proposed a new compound technique for S-box design using tent-logistic chaotic map, which involves a novel linear mapping scheme for the generation of the initial S-box. In [6], authors proposed a chaotic Jaya optimization algorithm for image encryption, the S-box achieved good cryptographic properties. A few researchers also incorporate other methods [5], [7] with chaos to make S-box more robust. For instance, combining chaos with algebra makes S-box more suitable for cryptography.

Elliptic curve [27], cubic fractional transformation [28], and algebraic properties like group and rings [29]–[31] are supposed to improve the DP value of S-box, but those do not give significant results. Most of these S-boxes give DP value of 10/256. In [32], an S-box is designed using Gaussian distribution and linear fractional transformation by applying the Box-Muller transform, polarization decision, and central limit algorithm. The research work in [33] proposed a way of generating highly non-linear $n \times n$ S-boxes for $3 \leq n \leq 7$. For each iteration, the best S-box candidate is chosen based on heuristic optimization. This approach is good for non-linear key-dependent S-boxes. In [34], the authors proposed a technique for designing S-box via single expression algebra instead of using matrix algebra that reduced the computational complexity for S-Box construction.

Researchers have also worked on AES in terms of performance and security, both in hardware and software. Tillich and Herbst [35], Rupanagudi *et al.* [36], Moh'd [37], Talha [38], and Shreedhar *et al.* [39] proposed a method for improving the performance of AES in hardware in terms of area and clock speed. Intel's corporation proposed new AES instruction set for improving its performance and security for their processors [40]. Sahoo *et al.* [41] proposed new affine transformation for improving performance of AES in software. Pachori *et al.* [42] proposed java-based AES parallel processing for improving processing speed. In [41], the author modified the S-box of AES by modifying the affine transformation to minimize the time complexity of AES.

Considerable research has been done to improve AES security in software. De Los Reyes *et al.* [43], [44] worked on AES key scheduling to increase diffusion and confusion rate and enhance the security of AES. They also proposed different versions of AES to increase its key length and security [45]. The authors in [46] proposed two different S-boxes instead of one AES S-box by modifying the affine transformation matrices that are used alternatively in the algorithm. In [47], [48] the authors proposed a variable mapping S-boxes for AES. Using secret key as well as using different irreducible polynomials produce dynamic S-boxes. A different S-box is designed in each round in byte substitution layer only, while all other layers of AES (shift row, mix column and key addition layer) remain the same. This enhances the security of AES but generates only limited number of S-boxes. The research works in [49]–[56] proposed dynamic S-boxes based on secret keys for enhancing the security of AES.

S-boxes are designed to resist differential attacks. The attacker tries to investigate the differential properties of S-box to analyze the ciphertext. Stronger the S-box cryptographically, higher will be the possibility to resist attacks. If there are different S-boxes in every round of algorithm, it will be more difficult for an attacker to investigate each S-box. This will provide an extra security layer. In this paper, authors proposed a novel key-based dynamic S-box design technique that is as strong as static AES S-box, with DP value of 4/256. S-boxes are designed using $GF(2^8)$ inverse and modified affine transformation. In affine transformation, using chaos modifies rotational matrix. Rotational matrix selection is based on specific parameters. A change in 8×8 rotation matrix can give rise to a different S-box that is as strong as AES S-box.

The rest of the paper is organized as follows. In Section II, an overview of the AES S-box generation method is described. In Section III, NIST tests are applied to the chosen chaotic map as a random number generator, and the obtained results are discussed. Section IV provides the proposed S-box design technique. Section V presents the detailed results regarding the proposed S-boxes. In Section VI, the proposed S-boxes are compared with state of the art S-boxes, and it is verified that proposed dynamic S-boxes are as strong as AES S-box. Finally, the conclusions are provided in Section VII.

II. PRELIMINARIES

This section briefly describes the standard AES S-box generation method, chaotic logistic map (CLM), and its NIST test that shows the effectiveness of the CLM in proposed S-box generation.

A. THE AES S-BOX

The AES S-Box is the only non-linear element in AES encryption, which bijectively maps every input element to exactly one output element so that the reverse process stays possible during decryption. The S-box is generated by calculating the multiplicative inverse of a given number in Rijndael's finite field, i.e. $GF(2^8)$, using a fixed generator polynomial $G(x) = x^8 + x^4 + x^3 + x + 1$. No inverse exists for the input element zero in Rijndael's finite field; hence, zero is mapped to itself. In all other cases, the multiplicative inverse A_i^{-1} of an input element A_i belongs to the elements of $GF(2^8)$ with a fixed irreducible polynomial $G(x) = x^8 + x^4 + x^3 + x + 1$. Table 1 provides the multiplicative inverse for all 8-bit numbers in $GF(2^8)$. Once the multiplicative inverse is computed, it is transformed to S_i using the affine transformation given in (1) as follows:

$$S_i \equiv R.A_i^{-1} + C \bmod 2 \tag{1}$$

where, R is a rotational matrix, C is an additive constant, and S_i is the output of the AES S-box that corresponds to input A_i. Equation (1) can be expanded by placing the values of R and C as in (2), where, $S_i = [s_7, \ldots, s_0]$ represents the 8-bit output of the AES S-box. Table 2 presents the final AES S-box generated by (2).

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \begin{pmatrix} a'_0 \\ a'_1 \\ a'_2 \\ a'_3 \\ a'_4 \\ a'_5 \\ a'_6 \\ a'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \bmod 2 \tag{2}$$

B. CHAOTIC LOGISTIC MAP (CLM)

The logistic equation is proposed by Pierre Francois Verhulst [57] in 1838 and introduced as a chaotic map first time by May [58]. The mathematical equation can be written as:

$$q_{i+1} = uq_i\,(1 - q_i), \quad \text{where } 0 < u < 4,\ q_i \in [0, 1] \tag{3}$$

The behavior of (3) is chaotic for the interval $3.57 \leq u \leq 4$. Before (0, 3.57), its behavior is deterministic and non-chaotic. Figure 1(a) shows the bifurcation diagram that elaborates its behavior, whereas Figure 1(b) shows the Lyapunov exponent of a chaotic logistic map.

In this research work, we use chaotic logistic map as random number generator (RNG) to obtain a random key value in the range (0, 255]. The logistic map is easy to implement and computationally feasible. The random key value obtained from the logistic map is used for producing certain affine transformations based on rotational matrices. Any rotational matrix can be generated with this behavior, and there is an equal proportion in the selection of any rotational matrix. Hence, any random S-box can be generated under the defined settings, based on the affine transformation matrix with strong cryptographic properties.

III. NIST STS 800-22 TESTS ON LOGISTIC MAP

To validate the randomness of a bitstream generated by any RNG, NIST statistical test suite (STS) 800-22 [59] is used. It entails fifteen (15) different tests that investigate the security of an RNG design. All tests must be passed by any RNG to ensure the security of the random number sequence produced. As discussed before, chaotic logistic map was used in the proposed scheme to generate a random key value for designing an S-box. Hence, chaotic logistic map (given in (3)) was initially tested as RNG. In this aspect, the chaotic map was iterated for 10^6 times (with initial seed value of $q_0 = 0.33$ and control parameter $u = 4$), which produced 1M bits using a threshold value of 0.5. This bitstream was then passed to NIST STS 800-22 tool for validating the randomness of the binary sequence. The results (Table 3) showed that all the tests were passed. Hence, the use of chaotic logistic map as RNG for the proposed scheme is validated as fulfills the NIST criteria. It is essential to understand here that NIST STS for RNG validation does not have any direct relation with the S-box generation and its cryptographic properties. However, it provides help in ensuring the randomness of the key value obtained using the chaotic logistic map. The key is used to generate a random rotational matrix for affine transformation, which in turn produces an S-box. Hence, in the case of a random key value, it becomes unpredictable for the attackers to find out which S-box is going to be used in the next round of AES. In this way, NIST STS for RNG helps in selecting a random key for generating a random S-box, thus increasing the strength of AES with undetectable effect in the performance of algorithm.

IV. PROPOSED METHOD FOR S-BOX GENERATION

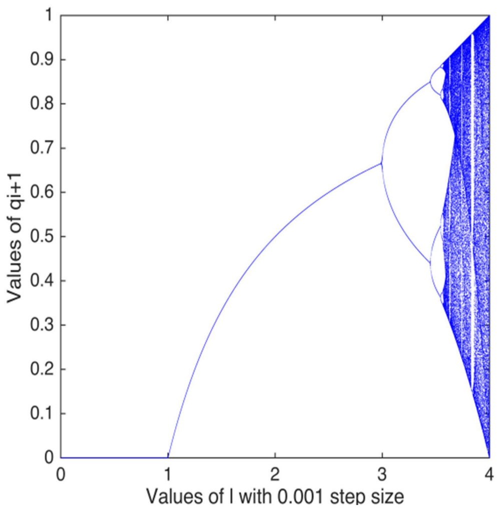
The proposed methodology for S-box generation is comprised of four stages, as shown in Figure 2. These stages include computation of the $GF(2^8)$ inverse, key generation using chaotic logistic map, computation of affine transformation rotational matrix R_k, and finally generation of a new S-box based on affine transformation operation. In the first stage, the same methodology is used as in AES S-box to generate the multiplicative inverse of an input byte A_i.

TABLE 1. Multiplicative inverse of input bytes XY in $GF(2^8)$.

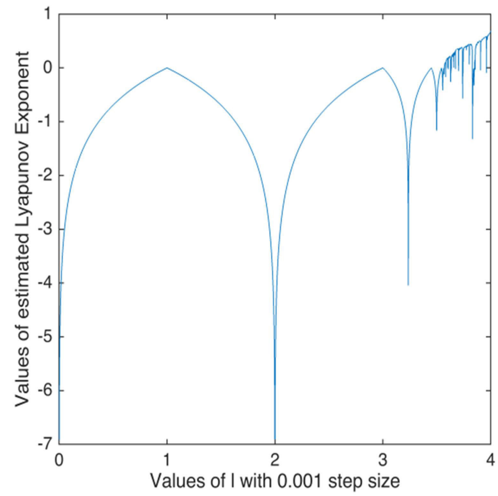
		Y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
	0	0	1	141	246	203	82	123	209	232	79	41	192	176	225	229	199
	1	116	180	170	75	153	43	96	95	88	63	253	204	255	64	238	178
	2	58	110	90	241	85	77	168	201	193	10	152	21	48	68	162	194
	3	44	69	146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

TABLE 2. The AES S-box.

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



(a)



(b)

FIGURE 1. a) Bifurcation diagram and b) Lyapunov exponent for chaotic logistic map.

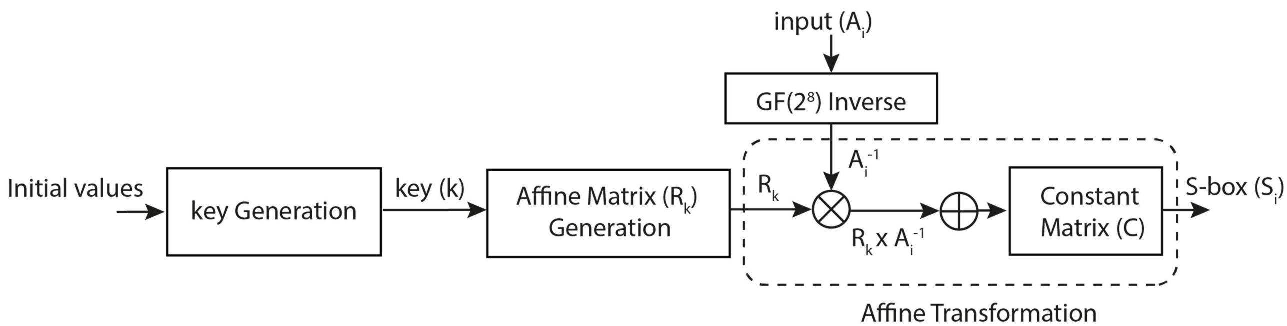


FIGURE 2. Block-level view of the proposed S-box generation method.

TABLE 3. NIST STS 800-22 tests results for logistic map.

NIST Statistical Test	P-value	Status
Approximate Entropy	0.935081	SUCCESS
Block Frequency	0.881134	SUCCESS
Cumulative Sums	0.577018 (FORWARD) 0.622087 (REVERSE)	SUCCESS
Fast Fourier Transform (FFT)	0.365257	SUCCESS
Frequency Test	0.508261	SUCCESS
Linear Complexity	0.861830	SUCCESS
Longest Run	0.014259	SUCCESS
Non-Overlapping Template	0.56962	SUCCESS
Overlapping Template	0.245325	SUCCESS
Random Excursions	0.579367	SUCCESS
Random Excursions Variant	0.44762	SUCCESS
Rank test	0.673806	SUCCESS
Runs test	0.644401	SUCCESS
Serial test	0.762361 0.214860	SUCCESS
Universal test	0.034041	SUCCESS

In the second stage, four chaotic logistic maps are combined to generate a random key based on the proposed key scheme. This key is considered as the first row of the affine transformation matrix, which is used to generate the entire rotational matrix. Finally, the output of the first stage, i.e. A_i^{-1}, is multiplied by a variable rotational matrix R_k generated under specific settings, followed by the addition of an 8-bit constant C to obtain the final output.

Equation (4) shows the entire process of generating the S-box output using the proposed S-box.

$$S_i \equiv R_k.A_i^{-1} + C \bmod 2 \tag{4}$$

where, R_k represents a 8×8 variable rotational matrix used for affine mapping and k is the matric key that actually represents the dynamicity of the rotational matrix as compared the AES S-box that utilizes a static matrix R. The rotational matrix R_k is entirely generated based on key value k. Hence, a large number of dynamic S-boxes can be produced by generating random keys. Figure 3 shows the detailed view of proposed S-box generation, and the following sections explain in detail the proposed algorithms for generating affine transformation matrix and the S-box.

A. KEY GENERATION

The proposed algorithm for generating the key for the affine transformation matrix consists of the following steps:

Step-1: Choose initial seed values for four different chaotic maps i.e., w_0, x_0, y_0, z_0, with a range between [0 1].

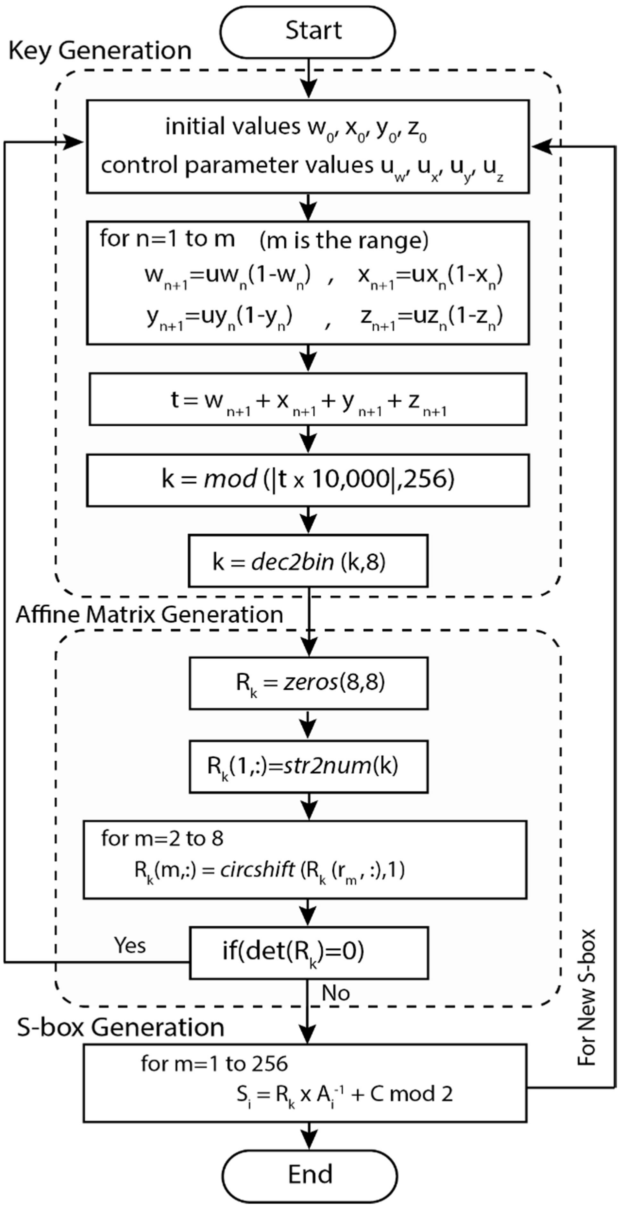


FIGURE 3. Detailed methodology for proposed S-box generation design.

Step-2: As the logistic map is chaotic for the region [3.57 4], hence, take four different parameter values (i.e., u_w, u_x, u_y, u_z) corresponding to four chaotic maps with range [3.57 4]. Herein, we choose the control parameter $u = 4$ to iterate the chaotic logistic map.

Step-3: Apply four chaotic maps (following (3)) shown to these four initial parameter pairs, i.e., (w_0, u_w), (x_0, u_x), (y_0, u_y), (z_0, u_z), and perform n iterations to get $n + 1$ output sequences for each pair between [0, 1].

Step-4: Add the last value from all four chaotic map sequences together to obtain t.

Step-5: Use equation $k = \text{mod}(|t \times 10000|, 256)$ and obtain a decimal number k within (0, 255].

Step-6: Convert this decimal number k into an 8-bit binary number that represents the binary key value for the affine matrix generation.

The pseudo-code for the key generation algorithm is given in Algorithm 1.

Algorithm 1 Generation of Key

Output: Key k for the affine transformation matrix R_k
Procedure:
% initial seed value $w_0, x_0, y_0, z_0 \forall \in [0, 1]$
1. Choose w_0, x_0, y_0, z_0
% logistic map is chaotic between 3.57 to 4
2. Choose u_w, u_x, u_y, u_z
3. **for n** 1 : m **do** % m is the range
4. $w_{n+1} = uw_n(1 - w_n)$
5. $x_{n+1} = ux_n(1 - x_n)$
6. $y_{n+1} = uy_n(1 - y_n)$
7. $z_{n+1} = uz_n(1 - z_n)$
8. **end for**
9. $t = w_{n+1} + x_{n+1} + y_{n+1} + z_{n+1}$
10. $k = \text{mod}(|t^*10, 000|, 256)$ % convert real number into decimal
11. $k = dec2bin(k, 8)$% convert decimal into binary string
12. $k = str2num(k)$% convert string to binary number

B. AFFINE TRANSFORMATION MATRIX GENERATION

The proposed algorithm for generating the affine transformation matrix consists of the following steps:

Step-1: Take a bit matrix R_k of size 8×8 and initialize it with all zero bits.

Step-2: Initialize the first row (r_1) of the matrix R_k with the key k as an 8-bit binary integer obtained in step-6 of the 1st algorithm as first row (r_1) of 8×8 the rotational matrix.

Step-3: Rotate r_1 one-bit right to generate the second row (r_2) for the matrix R_k. Likewise, rotate (r_2) one-bit right again to generate an 8-bit binary number as the third row (r_3).

Step-4: Repeat the step-3 for all eight rows and generate a 8×8 rotational matrix.

Step-5: Find the determinant $det(R_k)$ of the rotational bit matrix R_k.

Step-6: D iscard the matrix R_k and go to the step-1 key generation of 1st algorithm if its determinant-modulo 2 is equal to zero, i.e. $det(R_k) \equiv 0 \; mod 2$. Otherwise, go to next step.

Step-7: Take the bit matrix R_k that passes the step-6 as the affine transformation matrix and uses (4) to generate the S-box.

The pseudo-code for the proposed affine matrix algorithm is given in Algorithm 2.

C. PROPOSED S-BOX GENERATION

Algorithm 3 presents the proposed S-box generation steps. After the affine transformation matrix R_k is generated using Algorithm 2, the S-box is created in the following way:3

Algorithm 2 Generation of Affine Transformation Matrix

Output: Affine Transformation Matrix R_k

Procedure:

% 8 × 8 matrix with all elements equal to zero

1. $R_k = \text{zeros}(8,8)$

% make generated key 1st row of the rotational matrix

2. $R_k(l, ;) = k$
3. **for** $m = 2{:}8$ **do** % iterate for all rows in R_k
4. $R_k(m, :) = circshift(R_k(r_m, :), 1)$
5. **end for**

% *det* represents the determinant of a matrix

6. $D = det(R_k)$

% *mod* represents the modulus operator

7. **if** $D\ mod 2 \neq 0$ **then**
8. Set as an affine transformation matrix
9. **else**
10. Go to line 1 of the 1st algorithm
11. **endif**

Algorithm 3 S-Box Generation Based on Affine Transformation Matrix

Output: S-Box S_i

Procedure:

1: **for** $A_i = 0 : 255$ **do**

% Multiplicative inverse of A_i in $GF(2^8)$

2: Find A_i^{-1}

% S_i is the required S-box output for input A_i

3: $S_i = R_k \cdot A_i^{-1} + C\ mod 2$

4: **end for**

Step-1: Take an input byte A_i and compute its multiplicative inverse A_i^{-1} in $GF(2^8)$ using Table 1.

Step-2: Apply affine mapping on A_i^{-1} using (4) and compute the S-box output S_i. Use R_k obtained from Algorithm 2 as the affine transformation matrix in (4).

Step-3: Repeat step-1 and step-2 for all possible inputs A_i, i.e., *0 to 255* keeping the bit matrix R_k constant and generate the final S-box.

Table 4 shows four different rotational matrices of size 8 × 8, and Table 5 to Table 8 specifies the corresponding S-boxes generated by these matrices using the proposed methodology. R_{11}, R_{88}, R_{145}, and R_{203} are produced when the value of key generated is to be 11, 88, 145, and 203, respectively. The value of the subscript index k with the matrix R_k represents the key used to produce the rotational matrix. The sample rotational matrices R_{11}, R_{88}, R_{145}, and R_{203} fulfill the specified criteria in Algorithm 2.

V. RESULTS

S-boxes designed using the proposed methodology have good cryptographic properties like the AES S-box. This paper presents four different S-boxes (Table 5 - Table 8) generated corresponding to the matrices R_{11}, R_{88}, R_{145}, R_{203} and validates their cryptographic properties.

TABLE 4. Rotational matrices generated with the proposed S-box design.

Matrix index	Matrices
R_{11}	$\begin{bmatrix} 0&1&1&0&1&0&0&0\\ 0&0&1&1&0&1&0&0\\ 0&0&0&1&1&0&1&0\\ 0&0&0&0&1&1&0&1\\ 1&0&0&0&0&1&1&0\\ 0&1&0&0&0&0&1&1\\ 1&0&1&0&0&0&0&1\\ 1&1&0&1&0&0&0&0 \end{bmatrix}$
R_{88}	$\begin{bmatrix} 0&1&0&1&1&0&0&0\\ 0&0&1&0&1&1&0&0\\ 0&0&0&1&0&1&1&0\\ 0&0&0&0&1&0&1&1\\ 1&0&0&0&0&1&0&1\\ 1&1&0&0&0&0&1&0\\ 0&1&1&0&0&0&0&1\\ 1&0&1&1&0&0&0&0 \end{bmatrix}$
R_{145}	$\begin{bmatrix} 1&0&0&1&0&0&0&1\\ 1&1&0&0&1&0&0&0\\ 0&1&1&0&0&1&0&0\\ 0&0&1&1&0&0&1&0\\ 0&0&0&1&1&0&0&1\\ 1&0&0&0&1&1&0&0\\ 0&1&0&0&0&1&1&0\\ 0&0&1&0&0&0&1&1 \end{bmatrix}$
R_{203}	$\begin{bmatrix} 1&1&0&0&1&0&1&1\\ 1&1&1&0&0&1&0&1\\ 1&1&1&1&0&0&1&0\\ 0&1&1&1&1&0&0&1\\ 1&0&1&1&1&1&0&0\\ 0&1&0&1&1&1&1&0\\ 0&0&1&0&1&1&1&1\\ 1&0&0&1&0&1&1&1 \end{bmatrix}$

A. S-BOX TESTING CRITERIA IN CRYPTOGRAPHY

The effectiveness of the proposed S-boxes is validated by analyzing the results of conventional S-box tests, i.e., NL, SAC, BIC, DP, and LP.

TABLE 5. S-box designed with rotational matrix R_{11}.

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

TABLE 6. S-box designed with rotational matrix *R88*.

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

TABLE 7. S-box designed with rotational matrix R_{145}.

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

TABLE 8. S-box designed with rotational matrix R_{203}.

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

TABLE 9. Cryptographic properties of proposed S-boxes.

Initial value	NL	SAC	BIC	DP	LP
11	112	0.500977	112	4/256	0.0625
88	112	0.500977	112	4/256	0.0625
145	112	0.49585	112	4/256	0.0625
203	112	0.495605	112	4/256	0.0625

1) NON-LINEARITY (NL)

The importance of non-linearity in the cryptosystem was first introduced by Staffelbach and Meier in 1980s and later by Nyberg in early 1990s after the discovery of linear and differential cryptanalysis [60]. S-box is the only non-linear component in any cryptographic algorithm. The non-linearity of an S-box is its ability to resist against linear and differential cryptanalysis, which is measured using Walsh's spectrum [8]. Higher non-linearity value means strong resistance against linear and differential attacks [61]. Mathematically, it can be defined as:

$$S_f(w) = \sum_{S_f(w)} (-1)^{f(x) \oplus x.w} \tag{5}$$

The non-linearity of an n-bit Boolean function f is calculated as:

$$NL(f) = 2^{n-1} - \frac{1}{2}\left(\max_{z\in\{0,1\}^n} \left|W_f(z)\right|\right) \tag{6}$$

Higher the value of non-linearity, higher will be the security of S-box. The highest value of non-linearity that is achieved with AES S-box is 112. It is the standard value till now for all other S-boxes and the S-boxes with this non-linearity value are generally considered as secure. The proposed S-boxes of this study obtained the highest value of nonlinearity, i.e., 112, which has been achieved till today, as shown in Table 9.

2) STRICT AVALANCHE CRITERIA (SAC)

Tavares and Webster introduced strict avalanche criteria (SAC) [62] for the first time. According to the SAC, a single bit change in input will affect half of the output bits and the value of the SAC should be near 0.5. The proposed S-boxes fulfill the SAC satisfactory, as presented in Table 9.

3) BIT INDEPENDENCE CRITERION (BIC)

Bit independence criterion says that for the given change in the input bits there will be the change in the output independently. If the S-box satisfies the BIC, all the functions will be non-linear and fulfill the SAC also [62]. In our case, this criterion is also fulfilled by the proposed S-boxes, as presented in Table 9.

4) DIFFERENTIAL PROBABILITY (DP)

Differential probability is one of the most common methods to decrypt the ciphertext. It provides the difference in the original and the cipher message [28]. Mathematically it can be defined as an input difference Δx_i that should map to an output difference y_i [63]. It is defined in (7) as:

$$DP^s(\Delta x \to \Delta y) = \left(\frac{\#\{x \in X | S(x) \oplus S(x \oplus \Delta x) = \Delta y\}}{2^m}\right) \tag{7}$$

where, X is the set of all possible values of input and 2^m is the number of elements. The maximum achievable value of DP is 4/256, and our S-boxes fulfilling this criterion also, as shown in Table 9.

TABLE 10. Key values for S-box design with NL=112 and DP = 4/256.

Initial keys	DP Value
1, 2, 4, 7, 8, 11, 13, 14, 16, 19, 21, 22, 25, 26, 28, 31, 32, 35, 37, 38, 41, 42, 44, 47, 49, 50, 52, 55, 56, 59, 61, 62, 64, 67, 69, 70, 73, 74, 76, 79, 81, 82, 84, 87, 88, 91, 93, 94, 97, 98, 100, 103, 104, 107, 109, 110, 112, 115, 117, 118, 121, 122, 124, 127, 128, 131, 133, 134, 137, 138, 140, 143, 145, 146, 148, 151, 152, 155, 157, 158, 161, 162, 164, 167, 168, 171, 173, 174, 176, 179, 181, 182, 185, 186, 188, 191, 193, 194, 196, 199, 200, 203, 205, 206, 208, 211, 213, 214, 217, 218, 220, 223, 224, 227, 229, 230, 233, 234, 236, 239, 241, 242, 244, 247, 248, 251, 253, 254	4/256

5) LINEAR PROBABILITY (LP)

If an S-box has a small linear probability, it is considered as very strong against linear cryptanalysis [28]. Linear probability (LP) [64] of an S-box can be defined as:

$$LP = \max_{A_x, B_x \neq 0} \left| \frac{\#\{x \in Z | x.A_x = S(x).B_x\}}{2^n} - \frac{1}{2} \right| \tag{8}$$

where, A_x and B_x are the input and output values respectively while Z = {1,2,3…255}. As shown in Table 9, the LP value achieved with our proposed S-boxes is 0.0625, which is the same as the standard AES S-box.

B. CRYPTOGRAPHIC PROPERTIES OF THE PROPOSED S-BOXES

As the research on S-box design and construction is growing and becoming increasingly vital in the field of cryptography, numerous researchers have designed tools for testing and verifying the performance of an S-box [63], [65], [66]. These tools are based on the NIST criteria for S-box performance analysis and entail common S-box tests as defined by the NIST (including NL, SAC, BIC, DP, and LP as given in Eq. (5) – (8)). The purpose of designing these tools is to simplify the research process and provide an ease to the researchers in testing and verifying the S-box performance. For verifying the proposed S-boxes in this study, authors used the same S-box testing tool as presented in [63]. Table 9 shows the cryptographic properties of the proposed S-boxes, which are generated as a result of matrices R_{11} R_{88}, R_{145} and R_{203}. It can be analyzed from table 9 that all S-boxes have excellent cryptographic properties and are as stronger as the AES S-box. Table 10 presents all the key values that successfully generate the corresponding rotational

TABLE 11. Performance comparison with other states of the art S-boxes.

Study	Proposed Techniques to design S-box	NL	SAC	BIC	DP	LP
[16]	S-box based on 1-D chaotic map and beta hill climbing	110.25	0.500	104	10/256	0.125
[17]	S-box design based on proposed 2-D bakers chaotic map	104	0.4965	102.9	10/256	0.1289
[20]	Construction of S-boxes based on TD ERC sequence	104	0.507	102.9	12/256	0.086
[21]	S-box based on hyperchaotic map and bee colony algorithm	108	0.5073	104	10/256	0.1523
[22]	Construction of S-box based on optimized chaotic maps	106.75	0.5015	104.07	10/256	0.1367
[23]	S-box generation based on 3-D plasma chaotic system	106	0.4978	103.92	10/256	0.1298
[24]	S-box using LCG and combination of different chaotic maps	107.75	0.4976	105.07	10/256	0.1250
[25]	S-box based on gold sequence and chaotic logistic tent system	112	0.5065	109	6/256	0.1090
[6]	S-box design for image encryption based on hybrid chaotic map	106.2	0.5009	103.6	10/256	0.132
[69]	Effect of chaotic system in S-box performance characteristic	105.25	0.5037	102.64	10/256	0.125
[27]	Substitution box based on elliptic curves	100	0.5007	104.1	10/256	0.1250
[28]	S-box design using cubic fractional transformation	107	0.497	103.5	10/256	0.156
[29]	S-box generation using projective general linear group	105	0.5021	106	10/256	0.1250
[31]	Group theoretic approach to construct S-box	104.8	0.493	105.1	10/256	0.125
[32]	S-box designing using Gaussian distribution method	111	0.5036	110	6/256	0.0781
[48]	Design S-boxes using $GF(2^8)$ different polynomial	105.5	0.507	106	6/256	0.140
[53]	Dynamic AES-128 with key-dependent S-box	104.3	0.497	103.4	12/256	0.133
AES	S-box designed using AES	112	0.504	112	4/256	0.0625
S_1	Proposed S-box with first row of rotational matrix as 1	112	0.503	112	4/256	0.0625
S_{88}	Proposed S-box with first row of rotational matrix as 88	112	0.501	112	4/256	0.0625
S_{145}	Proposed S-box with first row of rotational matrix as 145	112	0.496	112	4/256	0.0625
S_{203}	Proposed S-box with first row of rotational matrix as 203	112	0.495	112	4/256	0.0625

matrices for affine transformations. These rotational matrices produce the S-boxes having NL and DP values 112 and 4/256, respectively.

VI. COMPARISON AND DISCUSSION

The performance of the proposed S-box compared to other state-of-the-art S-boxes, based on their cryptographic properties, is given in Table 11. The proposed S-boxes are cryptographically more effective than some of the existing S-boxes. When comparing the proposed S-boxes with the standard AES S-box, it can be seen that their cryptographic values are the same as the AES S-box, which means that these S-boxes are as strong as the AES S-box.

In [67], [68], the authors have investigated the effect of different types of practical attacks on the AES algorithm, and validated the efficacy of AES as compared to other encryption algorithms. However, AES produces only a static S-box that can be vulnerable to cryptanalyst study to crack the ciphertext. The advantage of the proposed method over AES algorithm is that the former generates a large number of dynamic S-boxes. Furthermore, the proposed S-boxes are generated based on the equivalence classes of AES, hence they provide the same security and equally strong cryptographic properties as the AES S-box. Any of the proposed S-box can be used in each round of AES encryption. The selection of the S-box is based on the key scheduling algorithm that uses a chaotic logistic map. Chaos is generally important to produce entropy in the sequence but cannot directly generate the S-boxes with high NL and low DP values [69]. For generating a cryptographically strong S-box using chaos, some additional non-linear phenomenon is incorporated in the system, which can be observed from the existing studies reported in Table 11. In our proposed scheme, we used affine transformation in addition to chaotic logistic map for generating the strong S-boxes. The main reason of using chaos in the proposed scheme is to obtain a pseudorandom sequence that is further used to generate an unpredictable and secure key for creating a new S-box (based on the rotational matrix) in each round of encryption. As a result, it becomes challenging for an attacker to analyze which S-box has been used in each round of the algorithm. Hence, it will be near impossible to decrypt the ciphertext. Thus, the addition of dynamic S-boxes provides an additional layer of security using the proposed scheme, which tremendously enhances the security of a cryptosystem.

In comparison to AES S-box, our proposed method entails two additional steps, which include key generation and affine matrix generation. A chaotic logistic map is used for generating the 8-bit random key value, which is computationally very cheap and fast [14]. Moreover, the generation of the affine transformation matrix is only based on rotating the obtained key value. Hence, the additional transformation cost tends to be very insignificant. Table 12 shows the overall computational time required for generating a large number of random key values and the corresponding affine matrices. The computational time is computed in MATLAB using a core-i7 Intel processor (3.2 GHz, 8GB RAM). It can be analyzed that the average additional time required for these

TABLE 12. Additional cost for keys and affine matrices generation.

No. of Keys and Affine Matrices	Computational Time (sec)
1	0.006722
2	0.007955
4	0.008723
8	0.012374
16	0.012773
32	0.013795
64	0.004586
128	0.014601
256	0.014706

transformations in generating a single S-box is very trivial (i.e., 6.7 milliseconds) and hence negligible. As a result, the proposed S-box generation design is computationally viable and comparable to AES S-box as well.

VII. CONCLUSION

In this study, dynamic AES S-boxes are generated using proposed chaos based rotation matrices. The standard AES S-box is adequate for AES security, but with the enhancement in computing resources and cryptanalysis techniques, there is a need to modify the AES algorithm. As a result, we proposed a significant modification in affine transformation of the AES S-box by proposing chaos based rotation metrics that generates strong AES S-boxes. Dynamic S-boxes nullifies the effects of algebraic attacks. The selection of S-box is based on a chaotic map-based key scheduling algorithm that makes it undetermined which S-box will be the targeted one. In this way, the security of a cryptosystem is greatly enhanced. The proposed S-boxes are tested based on the state of the art S-box tests, which demonstrate the effectiveness of the proposed S-box design.

ACKNOWLEDGMENT

The authors acknowledge the support from the Department of Information Systems, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar, Perak, Malaysia to carry out this research.

REFERENCES

[1] S. Guo, X. Zhao, F. Zhang, T. Wang, Z. J. Shi, F.-X. Standaert, and C. Ma, "Exploiting the incomplete diffusion feature: A specialized analytical side-channel attack against the AES and its application to microcontroller implementations," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 6, pp. 999–1014, Jun. 2014.

[2] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. New York, NY, USA: Springer, 2013.

[3] J. Nakahara, Jr., P. S. L. Barreto, B. Preneel, J. Vandewalle, and H. Y. Kim, "SQUARE attacks on reduced-round PES and IDEA block ciphers," in *Proc. 23rd Symp. Inf. Theory Benelux Date*, 2001, p. 68.

[4] J. Huang, H. Yan, and X. Lai, "Transposition of AES key schedule," in *Information Security and Cryptology* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2017.

[5] 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.*, vol. 29, no. 4, pp. 993–999, Feb. 2018.

[6] M. A. B. Farah, A. Farah, and T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box," *Nonlinear Dyn.*, pp. 1–24, Dec. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s11071-019-05413-8

[7] I. Hussain, A. Anees, M. Aslam, R. Ahmed, and N. Siddiqui, "A noise resistant symmetric key cryptosystem based on S8 S-boxes and chaotic maps," *Eur. Phys. J. Plus*, vol. 133, no. 4, Apr. 2018.

[8] M. A. Khan, A. Ali, V. Jeoti, and S. Manzoor, "A chaos-based substitution box (S-box) design with improved differential approximation probability (DP)," *Iranian J. Sci. Technol., Trans. Elect. Eng.*, vol. 42, no. 2, pp. 219–238, Jun. 2018.

[9] S. S. Jamal, M. U. Khan, and T. Shah, "A watermarking technique with chaotic fractional S-box transformation," *Wireless Pers. Commun.*, vol. 90, no. 4, pp. 2033–2049, Oct. 2016.

[10] A. A. Abd EL-Latif, B. Abd-El-Atty, and S. E. Venegas-Andraca, "A novel image steganography technique based on quantum substitution boxes," *Opt. Laser Technol.*, vol. 116, pp. 92–102, Aug. 2019.

[11] A. ALabaichi, R. Mahmod, and F. Ahmad, "Analysis of some security criteria for S-boxes in blowfish algorithm," *Int. J. Digit. Content Technol. its Appl.*, vol. 7, no. 12, p. 8, 2013.

[12] F. H. B. T. H. J. M. Ali and F. Hani, "A new 128-bit block cipher," Univ. Putra Malaysia, Seri Kembangan, Malaysia, Tech. Rep. FSKTM 2009 5, 2009.

[13] H. M. Heys, "A tutorial on linear and differential cryptanalysis," *Cryptologia*, vol. 26, no. 3, pp. 189–221, Jul. 2002.

[14] L. Kocarev, "Chaos-based cryptography: A brief overview," *IEEE Circuits Syst. Mag.*, vol. 1, no. 3, pp. 6–21, Mar. 2001.

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

[16] A. A. Alzaidi, M. Ahmad, M. N. Doja, E. Al Solami, and M. M. S. Beg, "A new 1D chaotic map and β-Hill climbing for generating substitution-boxes," *IEEE Access*, vol. 6, pp. 55405–55418, 2018.

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

[18] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "Efficient method for designing chaotic S-boxes based on generalized Baker's map and TDERC chaotic sequence," *Nonlinear Dyn.*, vol. 74, nos. 1–2, pp. 271–275, Oct. 2013.

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

[20] A. Hussain Alkhaldi, I. Hussain, and M. A. Gondal, "A novel design for the construction of safe S-boxes based on TDERC sequence," *Alexandria Eng. J.*, vol. 54, no. 1, pp. 65–69, Mar. 2015.

[21] Y. Tian and Z. Lu, "S-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm," *J. Syst. Eng. Electron.*, vol. 27, no. 1, pp. 232–241, 2016.

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

[23] A. K. Farhan, R. S. Ali, H. Natiq, and N. M. G. Al-Saidi, "A new S-box generation algorithm based on multistability behavior of a plasma perturbation model," *IEEE Access*, vol. 7, pp. 124914–124924, 2019.

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

[25] M. F. Khan, A. Ahmed, K. Saleem, and T. Shah, "A novel design of cryptographic SP-network based on gold sequences and chaotic logistic tent system," *IEEE Access*, vol. 7, pp. 84980–84991, 2019.

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

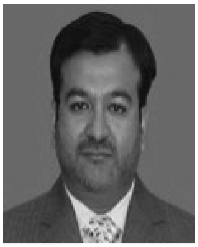
[27] U. Hayat, N. A. Azam, and M. Asif, "A method of generating 8×8 substitution boxes based on elliptic curves," *Wireless Pers. Commun.*, vol. 101, no. 1, pp. 439–451, Jul. 2018.

[28] A. Zahid, M. Arshad, and M. Ahmad, "A novel construction of efficient substitution-boxes using cubic fractional transformation," *Entropy*, vol. 21, no. 3, p. 245, Mar. 2019.

[29] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal, "A projective general linear group based algorithm for the construction of substitution box for block ciphers," *Neural Comput. Appl.*, vol. 22, no. 6, pp. 1085–1093, May 2013.

[30] S. Farwa, T. Shah, and L. Idrees, "A highly nonlinear S-box based on a fractional linear transformation," *SpringerPlus*, vol. 5, no. 1, p. 1658, Dec. 2016.
[31] I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, and H. Mahmood, "A group theoretic approach to construct cryptographically strong substitution boxes," *Neural Comput. Appl.*, vol. 23, no. 1, pp. 97–104, Jul. 2013.
[32] M. F. Khan, A. Ahmed, and K. Saleem, "A novel cryptographic substitution box design using Gaussian distribution," *IEEE Access*, vol. 7, pp. 15999–16007, 2019.
[33] M. Ahmad, M. Alauddin, and H. D. AlSharari, "Heuristic approach for nonlinear $n \times n$ ($3 \leq n \leq 7$) substitution-boxes," in *Proc. Adv. Intell. Syst. Comput. Data Eng. Intell. Comput.*, 2018, pp. 347–355.
[34] N. Munir and M. Khan, "A generalization of algebraic expression for nonlinear component of symmetric key algorithms of any characteristic P," in *Proc. Int. Conf. Appl. Eng. Math. (ICAEM)*, pp. 48–52, Sep. 2018.
[35] S. Tillich and C. Herbst, "Boosting AES performance on a tiny processor core," in *Topics in Cryptology—CT-RSA* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2008.
[36] S. Rao Rupanagudi, V. Vidya J, V. G. Bhat, P. Padmavathi, G. Darshan, S. K. Gurikar, S. Darshan, and N. Sindhu, "A further optimized mix column architecture design for the advanced encryption standard," in *Proc. 11th Int. Conf. Knowl. Smart Technol. (KST)*, Jan. 2019, pp. 181–185.
[37] A. Moh'd, Y. Jararweh, and L. Tawalbeh, "AES-512: 512-bit advanced encryption standard algorithm design and evaluation," in *Proc. 7th Int. Conf. Inf. Assurance Secur. (IAS)*, Dec. 2011, pp. 292–297.
[38] S. M. U. Talha, M. Asif, H. Hussain, A. Asghar, and H. Ameen, "Efficient advance encryption standard (AES) implementation on FPGA using Xilinx system generator," in *Proc. 6th Int. Conf. Intell. Adv. Syst. (ICIAS)*, Aug. 2016, pp. 1–6.
[39] A. Shreedhar, K.-S. Chong, N. K. Z. Lwin, N. A. Kyaw, L. Nalangilli, W. Shu, J. S. Chang, and B.-H. Gwee, "Low gate-count ultra-small area nano advanced encryption standard (AES) design," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
[40] S. Gueron, "Intel's new AES instructions for enhanced performance and security," in *Fast Software Encryption* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, 2009. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-03317-9_4
[41] O. B. Sahoo, D. K. Kole, and H. Rahaman, "An optimized S-box for advanced encryption standard (AES) design," in *Proc. Int. Conf. Adv. Computing Commun. (ICACC)*, 2012, pp. 154–157.
[42] V. Pachori, G. Ansari, and N. Chaudhary, "Improved performance of advance encryption standard using parallel computing," *Int. J. Eng. Res. Appl.*, vol. 2, no. 1, pp. 967–971, 2012.
[43] E. M. De Los Reyes, A. M. Sison, and R. Medina, "Modified AES cipher round and key schedule," *Indonesian J. Electr. Eng. Informat.*, vol. 7, no. 1, pp. 29–36, 2019.
[44] H. M. Hussien, Z. Muda, and S. M. Yasin, "Enhance the robustness of secure Rijndael key expansion function based on increment confusion," in *Proc. 6th Int. Conf. Comput. Inform.*, 2017.
[45] A. M. Sagheer, S. S. Al-Rawi, and O. A. Dawood, "Proposing of developed advance encryption standard," in *Proc. Develop. E-Syst. Eng.*, Dec. 2011, p. 1.
[46] N. Tiwari and A. Kumar, "Security effect on AES in terms of avalanche effect by using alternate S-box," in *Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things*, 2018, pp. 1–14.
[47] F. Mohammad, A. E. Rohiem, and A. Elbayoumy, "A novel S-box of AES algorithm using variable mapping technique," in *Proc. Int. Conf. Aerosp. Sci. Aviation Technol.*, vol. 13, May 2009, pp. 1–10.
[48] S. Mahmood, S. Farwa, M. Rafiq, S. M. J. Riaz, T. Shah, and S. S. Jamal, "To study the effect of the generating polynomial on the quality of nonlinear components in block ciphers," *Secur. Commun. Netw.*, vol. 2018, pp. 1–8, Apr. 2018.
[49] A. Alabaichi and A. I. Salih, "Enhance security of advance encryption standard algorithm based on key-dependent S-box," in *Proc. 5th Int. Conf. Digit. Inf. Process. Commun. (ICDIPC)*, Oct. 2015, pp. 44–53.
[50] F. J. D'Souza and D. Panchal, "Advanced encryption standard (AES) security enhancement using hybrid approach," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 647–652.
[51] J. Juremi, R. Mahmod, and S. Sulaiman, "A proposal for improving AES S-box with rotation and key-dependent," in *Proc. Int. Conf. Cyber Secur., Cyber Warfare Digit. Forensic (CyberSec)*, Jun. 2012, pp. 38–42.
[52] V. Ramaswamy, "Making AES stronger?: AES with key dependent S-box," *Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 9, pp. 388–398, 2008.
[53] E. M. Mahmoud, "Dynamic AES-128 with key-dependent S-box," *Int. J. Eng. Res. Appl.*, to be published.
[54] N. Stoianov, "One approach of using key-dependent S-BOXes in AES," in *Proc. Int. Conf. Multimedia Commun., Services Secur.*, 2011, pp. 317–323.
[55] G. Jacob, A. Murugan, and I. Viola, "Towards the generation of a dynamic key-dependent S-box to enhance security," *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 92, Feb. 2015.
[56] K. Kazlauskas and J. Kazlauskas, "Key-dependent S-box generation in AES block cipher system," *Informatica*, vol. 20, no. 1, pp. 23–24, 2009.
[57] N. Bacaër and N. Bacaër, "Verhulst and the logistic equation (1838)," in *A Short History of Mathematical Population Dynamics*. London, U.K.: Springer, 2011.
[58] R. M. May, "Simple mathematical models with very complicated dynamics," in *Universality in Chaos: A Reprint Selection*, 2nd ed. Nature, 2017. [Online]. Available: https://www.nature.com/articles/261459a0
[59] A. Rukhin, J. Soto, and J. Nechvatal, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," in *Proc. NIST*, 2010, pp. 1–164.
[60] C. Blondeau and K. Nyberg, "Perfect nonlinear functions and cryptography," *Finite Fields Their Appl.*, vol. 32, pp. 120–147, Mar. 2015.
[61] S. Gao, W. Ma, and J. Zhu, "Nonlinearity profile test for an S-box," in *Future Wireless Networks and Information Systems* (Lecture Notes in Electrical Engineering). Berlin, Germany: Springer, 2012.
[62] A. F. Webster and S. E. Tavares, "On the design of S-boxes," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 1986.
[63] Y. Wang, Q. Xie, Y. Wu, and B. Du, "A software for S-box performance analysis and test," in *Proc. Int. Conf. Electron. Commerce Bus. Intell.*, Jun. 2009, pp. 125–128.
[64] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *J. Cryptol.*, vol. 4, no. 1, pp. 3–72, 1991.
[65] F. Özkaynak, "An analysis and generation toolbox for chaotic substitution boxes: A case study based on chaotic labyrinth rene thomas system," *Iranian J. Sci. Technol., Trans. Elect. Eng.*, to be published.
[66] S. Picek, L. Batina, D. Jakobović, B. Ege, and M. Golub, "S-box, SET, match: A toolbox for S-box analysis," in *Information Security Theory and Practice. Securing the Internet of Things* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, 2014. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-43826-8_10
[67] A. Biryukov and J. Großschädl, "Cryptanalysis of the full AES using GPU-like special-purpose hardware," *Fundam. Informaticae*, vol. 114, nos. 3–4, pp. 221–237, 2012.
[68] S. Ali, X. Guo, R. Karri, and D. Mukhopadhyay, "Fault attacks on AES and their countermeasures," in *Secure System Design and Trustable Computing*. Cham, Switzerland: Springer, 2015, pp. 163–208.
[69] F. Özkaynak, "On the effect of chaotic system in performance characteristics of chaos based S-box designs," *Phys. A, Stat. Mech. Appl.*, Jan. 2020, Art. no. 124072. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437119322514

MUHAMMAD SARMAD MAHMOOD MALIK received the B.Sc. degree in computer engineering from Bahauddin Zakariya University Multan, Pakistan, in 2017. He is currently pursuing the M.Sc. degree in computer engineering from the University of Engineering and Technology, Taxila, Pakistan. His field of research is cryptography and information security.



MUHAMMAD ASIM ALI received the B.Sc. degree in computer science from Allama Iqbal Open University, Islamabad, Pakistan, in 2001, and the M.Sc. and M.S. degrees in computer science from the Comsats Institute of Information Technology, Wah Cantt, Pakistan, in 2004 and 2007, respectively, where he is currently pursuing the Ph.D. degree in computer science. He is currently working as an Assistant Professor with the Computer Science Department, University of Wah, Wah Cantt. His Area of specialization is cryptography. His research interests are within the field of information security, software engineering, computational theory, and chaos-based cryptography.

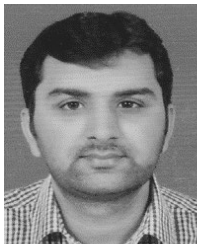
MUHAMMAD ASIF KHAN received the B.Sc. degree in computer engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2005, and the M.S. and Ph.D. degrees in electrical and electronic engineering from University Teknologi PETRONAS, Malaysia, in 2009 and 2015, respectively.

From 2005 to 2006, he was a Lecturer with the University of Engineering and Technology. From 2006 to 2013, he was a Research Assistant with the Electrical and Electronic Engineering Department, Universiti Teknologi PETRONAS. From 2013 to 2015, he worked as Research Scientist with PETRONAS SD BHD for the Project WiDUCT. He is currently working as an Assistant Professor with the Computer Engineering Department, University of Engineering and Technology. His field of specialization is chaos-based cryptography and wireless PHY layer security. His research interests are within the field of blockchain, watermarking, multimedia security, and lightweight cryptography.

MUHAMMAD EHATISHAM-UL-HAQ received the B.Sc. and M.Sc. degrees in computer engineering from the University of Engineering and Technology (UET), Taxila, Pakistan, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in computer engineering.

His field of specialization is pervasive and ubiquitous computing. His research interests are within the areas of signal, image, and video processing, biomedical signal processing, mobile sensing, machine learning, human activity and emotion recognition, and human behavior analysis. He was a recipient of the Gold Medal for his B.Sc. degree and the Chancellor's Gold Medal for his M.Sc. degree.

SYED NASIR MEHMOOD SHAH received the Ph.D. degree in information technology from Universiti Teknologi PETRONAS, Malaysia, in 2012. He is currently serving as the Head for the Department of Computer Science and the Dean for the Sciences with the Dr. A. Q. Khan Institute of Computer Sciences and Information Technology (KICSIT), Kahuta, Pakistan. His research interests include cyber security, cloud computing, grid computing, and information systems. He has published around 40 articles in peer-reviewed conferences, quality journals, and book chapters.



MOBASHAR REHMAN received the B.S. degree (Hons.) in computer science from the University of Arid Agriculture, Rawalpindi, Pakistan, the M.Sc. degree in information technology from UTP, the M.B.A. degree from COMSATS, Islamabad, Pakistan, and the Ph.D. degree in information technology (knowledge management) from Universiti Teknologi Petronas, Malaysia. He is currently working as an Assistant Professor with the Department of Information Systems, Universiti Tunku Abdul Rahman, Kampar, Malaysia. He has published his work in various reputable international journals. His research interests include knowledge management, knowledge sharing, human factors in software engineering, machine learning, cyber-psychology, and organization behavior.

WAQAR AHMAD received the B.Sc. degree in computer engineering from COMSATS University Abbottabad and the M.Sc. degree in computer engineering from the University of Engineering and Technology Taxila, Pakistan, respectively, the Ph.D. degree from the Department of Electronics and Telecommunications, Politecnico di Torino, Italy. He was a Higher Education Commission (HEC) fully funded Ph.D. Scholar. He is currently an Assistant Professor with the University of Engineering and Technology. His current research interests focus on developing machine learning algorithms and architectures for video coding and multicamera networks. He has serves as a Reviewer for the *Journal of Circuits, Systems, and Computers*.