Available online at http://jprm.sms.edu.pk/
Journal of Prime Research in Mathematics, 21(2) (2025), 120–139

Journal of Prime Research in Mathematics

Image Encryption Using Chaotic Logistic Map and Finite Field Inversion for Dynamic S-Box Design

Anam Rani[a], Muhammad Shams Ul Haq[a], Syed Ahtsham Ul Haq Bokhary[a], Usman Ali[a,*]

[a] *Centre for Advanced Studies in Pure and Applied Mathematics, Bahauddin Zakariya University, Multan, Pakistan.*

Abstract

We present a new method for constructing dynamic S-boxes that combines the chaotic behavior with Galois Field arithmetic over GF(2^8). The approach employs the logistic map $X_{n+1} = \mu X_n(1 - X_n)$, where $X_0 = 0.5$ and $3.57 \leq \mu \leq 3.99$ to generate values, which are then reduced modulo 256 and interpreted as field elements. Using finite field inversion with an irreducible polynomial, bijective S-boxes are obtained. The proposed S-boxes are applied in an image encryption scheme to enhance security. Experimental evaluation shows that our designs are highly resistant against cryptanalytic attacks and improves the quality of encrypted images, offering a reliable way to protect sensitive image data during transmission.

Keywords: S-box, chaotic map, logistic map, Galois field $GF(2^8)$, cryptography, nonlinearity, differential probability, linear probability, image encryption, dynamic S-box, lightweight cryptosystem.
2010 MSC: 47H10, 54H25.

1. Introduction

The security of image transmission has become a key research focus in modern cryptography, where S-boxes play a central role as nonlinear components. Image encryption, as a branch of cryptography, is inherently interdisciplinary, involving mathematics, information theory, computer science, and engineering. To resist statistical and differential attacks, researchers have increasingly employed chaotic maps and S-box transformations in image encryption schemes. Chaotic S-boxes introduce strong nonlinearity, unpredictability, and key sensitivity, which are essential for secure multimedia communication. In recent years, numerous image encryption algorithms based on chaotic S-boxes, trigonometric maps, polynomial transformations, and hybrid algebraic–chaotic structures have been proposed, showing significant improvements in entropy, histogram uniformity, correlation reduction, and robustness against cryptanalytic attacks.

Substitution boxes (S-boxes), first introduced by Claude Shannon in 1949, are a fundamental component in symmetric key cryptography. Their primary role is to introduce nonlinearity and confusion into cryptographic

*Corresponding author
Email addresses: anam.rani@bzu.edu.pk (Anam Rani), shamsulhaqsahu@gmail.com (Muhammad Shams Ul Haq), ahtsham@bzu.edu.pk (Syed Ahtsham Ul Haq Bokhary), uali@bzu.edu.pk (Usman Ali)

Received : 10 September 2025*; Accepted:* 13 October 2025*; Published Online:* 17 October 2025

algorithms, making it significantly more difficult for attackers to deduce the original message from the ciphertext. The importance of S-box design has become increasingly prominent since the selection of the Rijndael algorithm as the Advanced Encryption Standard (AES) by the National Institute of Standards and Technology (NIST) in 2000 [25]. Developed by Joan Daemen and Vincent Rijmen, AES exhibits strong resistance against well-known cryptanalytic techniques, such as differential cryptanalysis (DC) [24] and linear cryptanalysis (LC) [21].

Block ciphers, such as DES and AES, rely heavily on the strength and design of the S-box. These nonlinear substitution mechanisms are critical for increasing the cryptographic robustness of the entire encryption system. According to Ma et al. [23], the AES S-box has only 9 algebraic terms because it is structured over the Galois Field $GF(2^8)$, whereas its inverse has 255 terms. To address potential vulnerabilities associated with this simplicity, the Affine-Power-Affine (APA) structure was introduced, which increased the S-box's algebraic complexity from 9 to 253 terms while maintaining cryptographic performance. This enhancement significantly increases resistance to algebraic attacks [10].

A number of mathematical methods have been explored to generate S-boxes with better cryptographic characteristics. Chaotic maps, for example, have been extensively applied due to their sensitivity to initial value and control parameters, which guaranties unpredictability and sufficient confusion. The proposed a discrete-space integer multiplication-based chaotic map, resulting in S-boxes with maximum nonlinearity and minimal differential uniformity [18].

In a different direction, finite field theory has been utilized to ensure bijectivity and balanced mappings. [7] constructed an S-box based on the action of the modular group upon a projective line over a finite field, resulting in high nonlinearity and secure algebraic structure. Other related work by [5] employed linear fractional transformations with permutation functions, with further enhancements of nonlinearity and avalanche effect.

Other methods depend on polynomial and trigonometric transformations. [4] proposed a dynamic$S$-based S-box with a square polynomial transformation and permutation. [13] subsequently used linear trigonometric transformations to generate dynamic S-boxes efficiently. These schemes add further nonlinearity and periodicity and enhance resistance against correlation and statistical attacks.

In addition, multidimensional methods like coupled map lattices [12] have been tried to maximize entropy and randomness, rendering the output of S-box more random. Time-honored results such as those of Biham and Shamir also stress that resistance against differential cryptanalysis is important, leading to the designs with low differential uniformity and high nonlinearity. Graph Theory is also used to develop new cryptographic algorithms. Recently, Bokhary et al. [2] have presented three new graph-theoretic encryption and decryption techniques that use complete bipartite graphs and the Cartesian product of graphs to improve communication security.

Lastly, hybrid schemes involving chaos maps blended with algebraic or trigonometric maps have also revealed even stronger cryptographic features. By merging chao$S$-driven randomness with algebraic form, these schemes at the same time enjoy bijectivity, SAC, BIC, and high nonlinearity. Altogether, the above methods prove how mathematical tools directly enhance the security of S-boxes against popular attacks like linear and differential cryptanalysis.

1.0.1. Contributions

S-boxes are important in image cryptography because they provide confusion while also resisting cryptanalytic attacks. Chaotic maps are broadly used because they generate sequences with high sensitivity to initial conditions and parameters, which increases unpredictability. However, chaotic maps alone often suffer from issues such as limited key space, short cycle length, and non-uniform distribution when digitized. On the other hand, finite field inversion, especially over $\mathrm{GF}(2^8)$, is a well-established technique that guarantees bijectivity and strong algebraic properties, as seen in the AES S-box. By combining chaotic maps with finite field inversion, we can take advantage of both worlds: the randomness and sensitivity of chaos and the strong nonlinearity and bijection of finite fields. This integration ensures that the constructed S-boxes are not only secure against linear and differential attacks, but also dynamic in nature, making them better

suited to protecting images that are highly redundant and vulnerable to statistical attacks.

This paper presents a new method for creating S-boxes that combines chaotic behavior with finite field arithmetic over $GF(2^8)$. The primary contributions of this work are summarized as follows:

- A proposed S-box construction method combines the logistic map with finite field operations to increase nonlinearity and unpredictability.
- The logistic map's rational outputs are processed by finite field inversion and multiplication, which allows both numerators and denominators as input.
- A dynamic method avoids duplicate values, verifying that the final S-box is bijective.
- The S-box with ideal statistical values is used in an image encryption method and also tested using different statistical metrics to show its superiority and effectiveness over other methods.
- The process of decryption reverses pixel substitution using the inverse S-box and returns to the original image structure via transposition to recover the original grayscale image without loss.

1.0.2. Paper Organization

To explain the mathematical tools and theoretical concepts indicated our proposed S-box generation method, Section 2 reviews the logistic chaotic map and associated finite field arithmetic over the Galois Field $\mathrm{GF}(2^8)$. Section 3 analyses and compares the making of S-box to existing designs. An image encryption and decryption method based on the proposed S-box is shown in Section 3.6. Performance evaluation and comparisons follow.

2. Background

To explain the mathematical equipment and theoretical concepts underlying our proposed S-box generation method, the logistic chaotic map and relevant finite field arithmetic over $\mathrm{GF}(2^8)$ are first reviewed in the following section.

2.1. Logistic Chaotic Map

A map (an evolution function) that exhibits chaotic behavior is called a chaotic map. Such maps can be parameterized using discrete-time or continuou$S$-time variables. Discrete chaotic maps are often represented using iterated functions. One popular and well-known one-dimensional discrete chaotic map, introduced by May [11], is the logistic map, whose state evolves according to

$$X_{n+1} = \mu X_n(1 - X_n), \tag{2.1}$$

where X_n is the state variable at iteration n, determined by the initial condition X_0, and the control parameter satisfies $0 < \mu \leq 4$. For all $n \geq 0$, the outputs X_n remain within the interval $[0, 1]$ [14]. In the proposed method, the fractional-order parameters μ and X_0 are chosen as secret keys to construct the S-boxes.

2.2. Galois Field

The Galois Field $GF(2^8)$ is important in modern cryptography because it enables secure and efficient arithmetic computations on 8-bit data. Each element of $\mathrm{GF}(2^8)$ can be represented as an 8-bit binary number or a polynomial of degree at most 7 with coefficients in $\{0, 1\}$.

Modulo an irreducible polynomial, arithmetic operations, particularly multiplication and inversion, are performed, such as

$$f(x) = x^8 + x^4 + x^3 + x^2 + 1.$$

The field can thus be defined as

$$\mathrm{GF}(2^8) \cong \mathbb{Z}_2[x]/\langle f(x)\rangle,$$

where $\mathbb{Z}_2 = \{0, 1\}$ and $f(x)$ is a primitive irreducible polynomial [1]. It is ensured by this construction that a unique multiplicative inverse is had by each non-zero element, which is required for the formation of bijective mappings like S-boxes.

The foundation of cryptographic algorithms like AES, which have nonlinearity and cryptanalysis resistance added, is provided by the $\mathrm{GF}(2^8)$ for substitution operations. Bijectivity is maintained and cryptographic strength is ensured by our approach through the conversion of logistic map outputs into valid S-box values using GF arithmetic. The flow chart of encryption and decryption with the proposed S-box is illustrated in Figure 1.

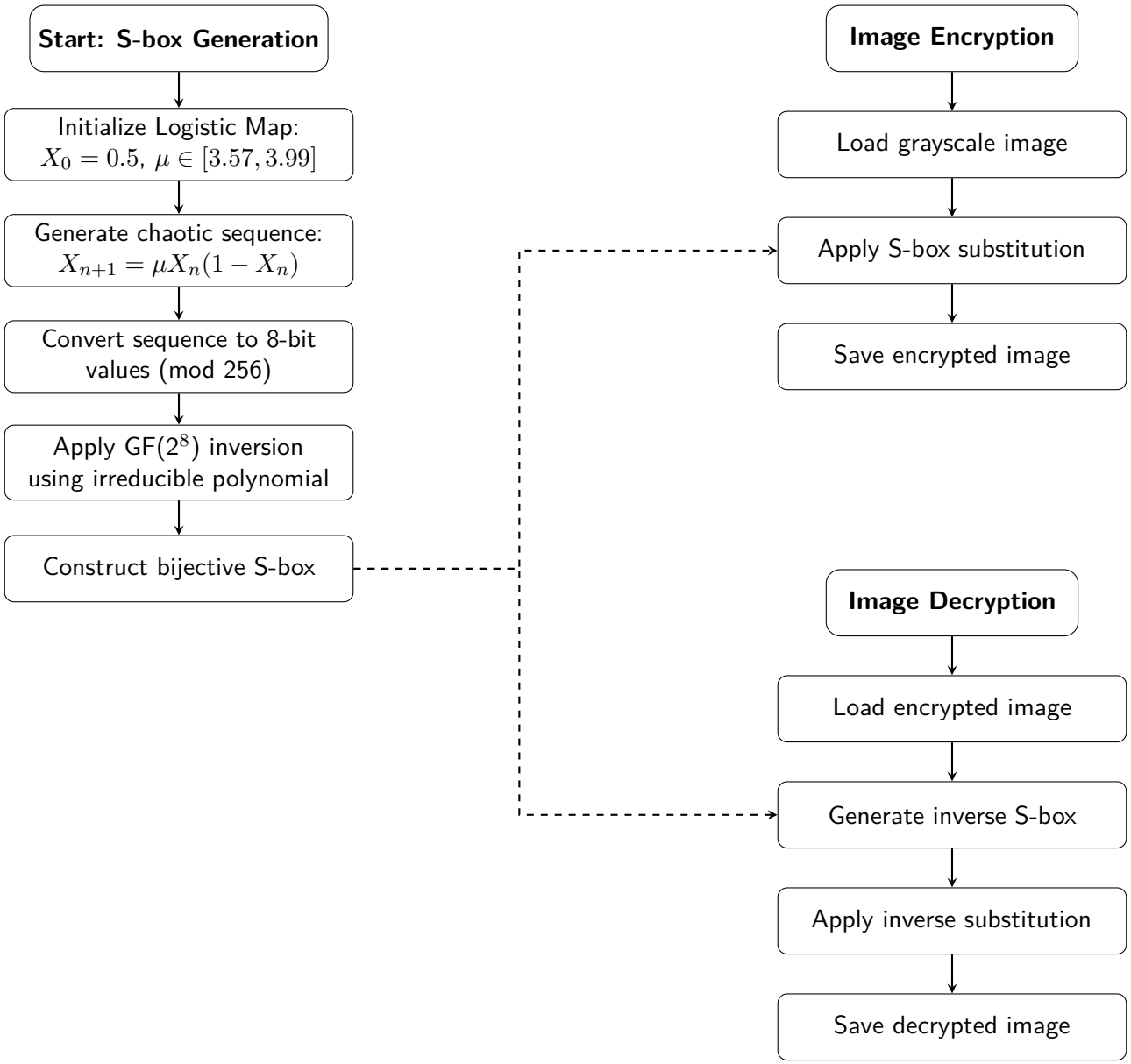


Figure 1: Flow Chart of construction of S-box

3. Results

3.1. Construction of S-boxes Using Logistic Map and Finite Field Operations

Several S-boxes are shown in this section, created with the proposed method, which combines the logistic map with operations over the Galois Field GF(2^8). The initial seed value is fixed at $X_0 = \frac{1}{2}$ in all cases, while the control parameter μ is varied to demonstrate the robustness and variability of the generated S-boxes.

3.1.1. Parameter Selection and Initialization

The chaotic logistic map is described as:

$$X_{n+1} = \mu X_n(1 - X_n)$$

For each S-box, we choose the value of μ as follows:

- Table 1: $\mu = 3.99$
- Table 2: $\mu = \frac{389}{100} = 3.89$
- Table 3: $\mu = \frac{379}{100} = 3.79$
- Table 4: $\mu = \frac{369}{100} = 3.69$
- Table 5: $\mu = \frac{359}{100} = 3.59$

All values are chosen within the chaotic regime ($\mu \geq 3.57$), in which randomness and confusion in the resulting S-boxes are ensured. We perform the following steps to obtain a field element in GF(2^8):

1. **Rational approximation:** compute a rational approximation
$$X_n^{(K)} = \frac{a_n}{b_n} \approx X_n,$$
with a fixed precision K (Implementation: `limit_denominator(2^16)`). Here $a_n, b_n \in \mathbb{Z}$ and $\gcd(a_n, b_n) = 1$.
2. **Reduction to 8 bits:** reduce numerator and denominator modulo 256,
$$\tilde{a}_n \equiv a_n \pmod{256}, \qquad \tilde{b}_n \equiv b_n \pmod{256},$$
producing integers $\tilde{a}_n, \tilde{b}_n \in \{0, \ldots, 255\}$. If $\tilde{b}_n = 0$, a defined fallback (we use $\tilde{b}_n \leftarrow 1$) is applied to avoid division by zero.
3. **Field interpretation and inversion:** interpret $\tilde{a}_n, \tilde{b}_n$ as elements of GF(2^8) under the irreducible polynomial $f(x) = x^8 + x^4 + x^3 + x^2 + 1$. Compute
$$s_n = [\tilde{a}_n] \cdot [\tilde{b}_n]^{-1} \in \mathrm{GF}(2^8),$$
where $[\tilde{b}_n]^{-1}$ is the multiplicative inverse in GF(2^8).
4. **Collision resolution (permutation enforcement):** To produce a bijective S-box (a permutation of $\{0, \ldots, 255\}$) we enforce uniqueness of outputs. In the manuscript we use the deterministic rule
$$\text{if } s_n \in \text{seen then } s_n \leftarrow (s_n + 1) \bmod 256$$

These steps explicitly map chaotic real outputs X_n to GF(2^8) elements and produce distinct S-box entries. Varying μ in the chaotic regime yields different sequences $\{X_n\}$ and different S-boxes.

3.1.2. Different Control Parameters for S-box Generation

S-boxes are created by applying GF(2^8) arithmetic operations to logistic map sequences, and then mapping them to a bijective 16×16 matrix.

3.2. Tabulated Results

The generated 8-bit S-boxes using different values of μ are displayed in the following tables. All S-boxes exhibit distinct permutations of integers in the range $[0, 255]$, validating their use in cryptographic applications.

Table 1: Generated S-box using logistic map and GF(2^8) operations with μ=3.99

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

Table 2: Generated S-box using logistic map and GF(2^8) operations with μ=3.89

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

Table 3: Generated S-box using logistic map and GF(2^8) operations with μ=3.79

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

Table 4: Generated S-box using logistic map and GF(2^8) operations with μ=3.69

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

Table 5: Generated S-box using logistic map and GF(2^8) operations with μ=3.59

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

3.3. Performance analysis of the proposed Sbox

This section presents a detailed performance evaluation of the proposed S-box using well-established cryptographic criteria. The S-box was implemented and tested in Python 3.8 using the PyCharm IDE on a machine equipped with an Intel Pentium 4417U processor and 4 GB RAM. For practical relevance, experiments were conducted on both grayscale and colored images.

3.3.1. Nonlinearity (NL)

The shape of the Boolean function is transformed by changing the bits in its truth table. The nonlinearity is the number of bits that must be changed in the truth table of the Boolean function to achieve the closest affine function.

The nonlinearity is bounded by:

$$N(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$$

for an S-box in GF(2^n). The results of the test for nonlinearity analysis are listed in Table 6. It can be seen that the proposed S-boxes exhibits nonlinear behavior with a maximum value of 108 and a minimum value of 106.

3.3.2. Strict Avalanche Criterion (SAC):

To evaluate the strict avalanche effect, which contains the basic concept of nonlinearity, the strict avalanche criterion (SAC) was put out [19]. It suggests that each output bit must change with a half-probability for each input bit change in order for a function to meet the strict avalanche criterion. By creating a correlation between the inputs and outputs, an encryption method can be readily broken in the absence of a strict avalanche effect.

First, the 8-bit vector X is entered into the S-box, and the corresponding to output 8-bit vector Y by substitution is taken. Second, a family of vectors $X_1, X_2, ..., X_8$ is formed such that X and X_j differ just in bit j. The resulting vectors $Y_1, Y_2, ..., Y_8$ can be determined by Y_j=S(X_j), with S($\cdot$) being a substitution operation with S-box.

An 8-bit binary avalanche vector family $X_1, X_2, ..., X_8$ can be computed by $V_j = Y \oplus Y_j$. An 8×8 dependence matrix is created by incorporating the value of bit i in V_j into element $a_{i,j}$. Do the above steps n times by randomly generating the vector X and the final dependence matrix is created by calculate the average elements.

The ideal value of the elements in dependent matrix is 0.5. An S-box can have high performance in SAC

when all elements of the matrix approach the ideal value. Table 3 presents the dependent matrix of the proposed S-box. From the table, it can be seen that the majority of the elements approach the ideal value 0.5. This suggests that the proposed S-box can get relatively ideal values.

3.3.3. Bits Independence Criterion (BIC)

Another important design criterion for reliable S-boxes is bit independence. A BIC test method was proposed by Adams and Tavares in [22]. Assume $v_1, v_2, \ldots, v_8$ be the component Boolean functions of an 8×8 S-box. The Boolean functions $V_j \oplus V_k$ (where $j \neq k$ and $1 \leq j, k \leq 8$) should be extremely nonlinear and meet the avalanche criterion (SAC) if the S-box satisfies BIC. Thus, for any 8×8 bijective S-box, BIC may be checked by computing the nonlinearity and SAC of all 56 functions $V_j \oplus V_k$. Table 7 displays the calculated potential nonlinearity scores and SAC of functions $V_j \oplus V_k$ for the suggested S-box. The average scores of BIC with respect to nonlinearities and SAC are found as 103.50 and 0.502, respectively. The obtained scores justify the satisfactory performance of the proposed S-box for the bits independence criterion.

3.3.4. Differential Approximation Probability (DP)

Differential Approximation Probability (DP) is an important criterion when evaluating an S-box, as it is a nonlinear component of a block cipher. In an ideal situation, an S-box should exhibit consistent behavior against differential attacks.

Let Δx denote the input differential and Δy represent the output differential. During differential cryptanalysis, the key observation is how likely a particular input difference Δx is to be mapped to an output difference Δy.

To evaluate this, we calculate the differential uniformity, which measures the maximum probability that a given input difference leads to a specific output difference. The DP of a specified S-box can be expressed as:

$$\mathrm{DP}(\Delta x, \Delta y) = \frac{|\{x \in \mathbb{X} \mid S(x) \oplus S(x \oplus \Delta x) = \Delta y\}|}{2^m}$$

where: S is the S-box, Δx is the input difference, Δy is the output difference, $\mathbb{X}$ is the set of all possible m-bit input values, and - 2^m denotes the total number of such inputs.

An S-box with low maximum DP values (i.e., low differential uniformity) is considered resistant to differential cryptanalysis.

3.3.5. Linear Approximation Probability (LP)

The LP criterion estimates how closely a linear combination of input bits can approximate a linear combination of output bits. It is defined as:

$$\mathrm{LP}(a, b) = \left| \Pr_{x \in \mathbb{F}_2^n} [a \cdot x = b \cdot S(x)] - \frac{1}{2} \right|$$

where $a, b \in \mathbb{F}_2^n$, and $\cdot$ denotes the dot product. Lower LP values indicate stronger resistance to linear attacks. The proposed S-box attained a maximum LP of 0.1406, as shown in Table 6, highlighting its security against linear approximations.

3.4. Comparison Among Proposed S-Boxes

In Table 6, we compare the proposed S-boxes based on their cryptographic properties, including Maximum Nonlinearity (Max. NL) for each S-boxe, Maximum and Minimum Strict Avalanche Criterion (SAC), BIC-SAC, BIC-NL, Maximum Differential Probability (DP), and Maximum Linear Probability (LP).

Table 6: Comparison of Cryptographic Properties of Proposed S-Boxes

S-Box	Max. NL	Max. SAC	Min. SAC	BIC-SAC	BIC-NL	Max. DP	Max. LP
S-box of Table 1	106	0.609	0.406	0.505	103.43	0.647	0.125
S-box of Table 2	106	0.625	0.375	0.505	103.21	0.047	0.1406
S-box of Table 3	108	0.578	0.375	0.502	103.50	0.037	0.1484
S-box of Table 4	108	0.594	0.406	0.499	102.64	0.039	0.1406
S-box of Table 5	108	0.625	0.375	0.497	103.86	0.647	0.1328

As shown in Table 6, the proposed S-box in Table 3 achieves the highest nonlinearity (Max. NL = 108) among all considered variants. It also demonstrates robust avalanche behavior with Max. SAC = 0.578 and Min. SAC = 0.375. Furthermore, it provides excellent differential uniformity (Max. DP = 0.037), which is among the lowest observed. The linear probability (Max. LP = 0.1484) remains within an acceptable range.

3.5. Comparison with Existing S-Boxes from Literature

In Table 7, we compare the proposed S-box with various existing S-boxes reported in the literature. The goal is to evaluate how well the proposed design performs relative to known S-box constructions across several critical cryptographic criteria.

Table 7: Comparison with Existing S-Boxes from the Literature

S-Box	Max. NL	Max. SAC	Min. SAC	BIC-NL	Max. DP	Max. LP
Proposed S-box	108	0.578	0.375	103.50	0.037	0.1406
M. Asif et al. [6]	107.3	0.610	0.570	101.3	0.060	0.150
B. Arshad et al. [7]	107.25	0.502	–	107.0	0.109	0.0234
A. A. Naveed et al. [8]	106	0.5233	0.0235	–	0.0391	0.138
A. K. Farhan et al. [15]	106	0.4978	0.002	103.92	0.0391	0.1402
G. Chen et al. [16]	106	0.6093	0.4218	103.1	–	–
M. Khan et al. [27]	106	0.4812	0.125	101.9	–	–
Hussain et al. [9]	104.75	0.594	0.391	100.0	0.039	0.125

As seen in Table 7, the proposed S-box outperforms several recent designs in terms of nonlinearity and differential uniformity. For example, M. Asif et al. achieved a Max. NL of 107.3, slightly lower than our result of 108. While A. A. Naveed et al. reported strong DP and LP values, their minimum SAC is extremely low (0.0235), suggesting weak avalanche characteristics. Similarly, other constructions exhibit deficiencies in one or more metrics. In contrast, the proposed S-box maintains a well-balanced performance across all major cryptographic indicators, making it a robust and reliable choice for secure system design.

Table 8: Comparison between AES S-box and proposed chaotic S-box

Feature	**AES S-box**	**Proposed Chaotic S-box**
Source of construction	Multiplicative inverse in $GF(2^8)$ + affine transform	Logistic map $X_{n+1} = \mu X_n(1-X_n)$ + rational approximation + $GF(2^8)$ arithmetic
Fixed or variable?	Fixed (same everywhere)	Variable (depends on μ, X_0)
Design motivation	Algebraic resistance to cryptanalysis	Chaos + algebraic inversion for unpredictability
Number of possible S-boxes	Exactly one	Many (different μ produce different boxes)
Affine transformation	Yes	No (uniqueness via collision-resolution)
Practical use	Standardized in AES	Dynamic cryptosystems, multimedia, IoT, experimental research

3.6. Image Encryption and Decryption with Proposed Sbox

3.6.1. Encryption Framework

In this encryption method (see Algorithm 1), a two-stage process is adopted to enhance image security through permutation and substitution. There is no doubt that this method can be used to color images. The method is essentially the same; the only difference is that the pixel representation is now multi-channel (such as RGB) instead of single-channel (grayscale). However, in order to effectively show the concept, we limited our trials in this paper to grayscale images. It is easy to extend to color photos, and this will be covered in later work.

- **Permutation:** The input grayscale image undergoes a row-column transformation via matrix transposition, effectively scrambling pixel positions to introduce *confusion*.
- **Substitution:** A pre-defined 8-bit S-box containing 256 unique values is then used to substitute each pixel value, achieving strong *diffusion*.

The S-box is imported from an external CSV file, ensuring a bijective mapping. After substitution, the encrypted image is saved and displayed. The original secret image (Lenna) [3] is shown in Fig. 2(a).

3.6.2. Algorithm for Image Encryption

Algorithm 1 Image Encryption with Row-Column Interchange and S-box Substitution

Require: Grayscale image I, 8-bit S-box table S, paths to image and S-box file
Ensure: Encrypted image I_E stored as PNG

function LOAD_SBOX($filepath$)
  $S \leftarrow$ Read CSV from $filepath$
  Flatten S to a 1D array of 256 elements
  return S
end function
function LOAD_IMAGE($image\_path$)
  $I \leftarrow$ Open image from $image\_path$ and convert to grayscale
  $I \leftarrow$ Convert image to 2D NumPy array
  return I
end function
function APPLY_SBOX(I, S)
  $F \leftarrow$ Flatten image array I to 1D
  for each pixel p in F **do**
    $p \leftarrow S[p]$
  end for
  Reshape F back to shape of I
  return encrypted image array I_E
end function
function SAVE_IMAGE($I_E, save\_path$)
  Create image object from I_E
  Save image to $save\_path$ and display it
end function
function ENCRYPT_IMAGE($image\_path, sbox\_path$)
  $S \leftarrow$ LOAD_SBOX($sbox\_path$)
  $I \leftarrow$ LOAD_IMAGE($image\_path$)
  $I_T \leftarrow$ Transpose(I) ▷ Row-column interchange
  $I_E \leftarrow$ APPLY_SBOX(I_T, S)
  SAVE_IMAGE(I_E, "$encrypted.png$")
end function

3.6.3. Visualization of Encryption Results

Our S-box performs well when encrypting and decrypting of two images using the same S-box, showing correct and secure operation. However, if the same S-box is used across many images, patterns in its construction may become visible to an attacker, weakening security. To prevent this, we can use multiple S-boxes with strong nonlinearity and good statistical properties, ensuring that each image is encrypted securely and reducing the risk of any information leakage.

(a) Original I

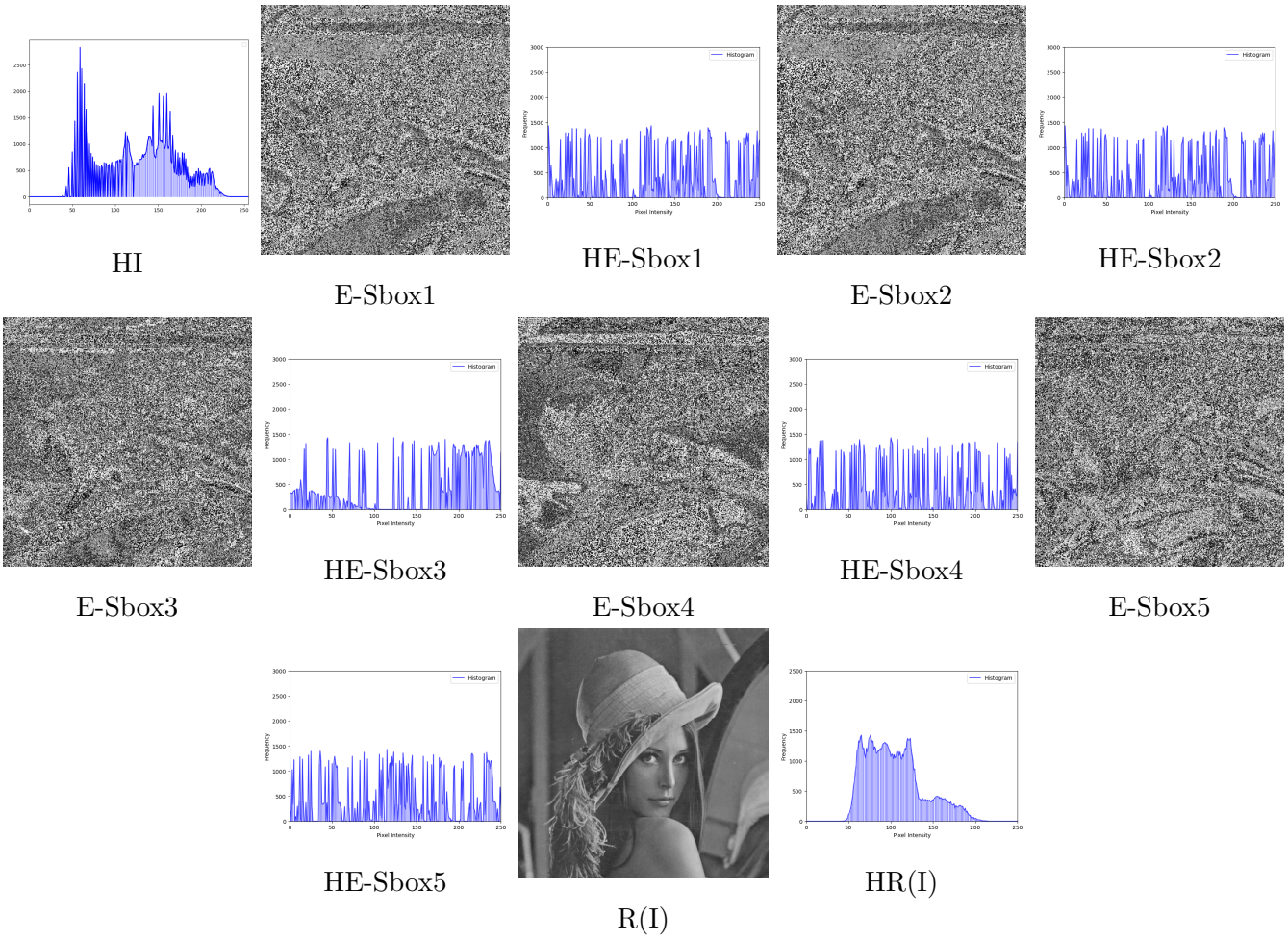


Figure 2: Encryption and decryption of grayscale images using different proposed S-boxes

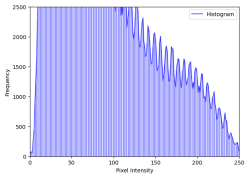
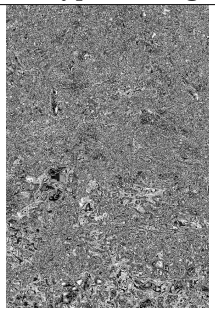
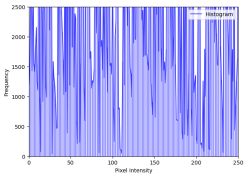
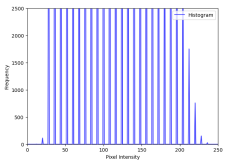
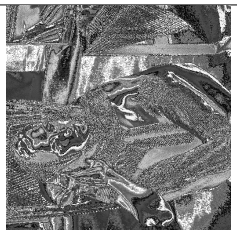
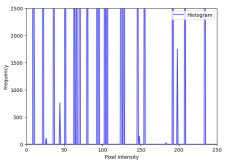
Original Image	Original Histogram	Encrypted Image	Encrypted Histogram	Decrypted Image
				
				

Table 9: Histogram analysis of original, encrypted, and decrypted images.

3.7. Texture-Based Security Analysis

To evaluate the strength of the proposed encryption scheme, texture-based statistical features are analyzed. These metrics assess the randomness and uniformity of the encrypted image.

3.7.1. Entropy

Entropy reflects the unpredictability in an image:

$$\text{Entropy} = -\sum_{i=0}^{255} p_i \log_2(p_i)$$

where p_i is the probability of gray level i.
Ideal value: Close to 8.

3.7.2. Contrast

Contrast quantifies pixel intensity differences:

$$\text{Contrast} = \sum_{i,j} (i-j)^2 \cdot P(i,j)$$

where $P(i,j)$ is the GLCM value.
Ideal value: High.

3.7.3. Correlation

Correlation assesses dependency between adjacent pixels:

$$\text{Correlation} = \frac{\sum_{i,j}(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j}$$

Ideal value: Close to 0.

3.7.4. Energy

Energy measures image uniformity:

$$\text{Energy} = \sum_{i,j} P(i,j)^2$$

Ideal value: Low (close to 0).

3.7.5. Homogeneity

Homogeneity indicates closeness of GLCM entries to diagonal:

$$\text{Homogeneity} = \sum_{i,j} \frac{P(i,j)}{1+|i-j|}$$

Ideal value: Low.

3.8. Comparison with Existing Schemes

Table 10: Comparison of Texture Features with Other Encryption Algorithms

Feature	**Proposed**	[20]	[17]	[9]	[26]
Entropy	7.6218	7.8523	7.6884	7.6596	7.6739
Contrast	9.0285	8.3301	7.7369	6.3684	6.8051
Correlation	0.0605	0.0201	0.2168	0.0996	0.1958
Energy	0.0079	0.0176	0.0229	0.0261	0.0261
Homogeneity	0.0199	0.4208	0.4863	0.4985	0.4951

As shown in Table 10, the proposed method achieves:

- High contrast (**9.0285**)—indicating strong pixel variation.
- Low correlation (**0.0605**)—demonstrating broken pixel relationships.
- Minimal energy and homogeneity—signifying high randomness and reduced pattern repetition.

These metrics validate the effectiveness of the proposed S-box in securing image data against statistical and visual analysis.

3.9. Decryption Framework

The decryption process is designed to reverse all operations performed during encryption and recover the original grayscale image without any data loss. This involves two primary steps: reversing the pixel substitution and restoring the original structure of the image.

First, we construct the inverse of the S-box used during encryption. This inverse S-box is applied to each pixel of the encrypted image, effectively reversing the substitution transformation. Next, we perform a transpose operation on the image to revert the row and column swapping carried out during encryption.

As a result, we obtain the fully reconstructed original image. Figure 2 shows the decrypted image denoted by $R(I)$, while Table 11 reports statistical metrics evaluating the fidelity of the reconstructed image relative to the original. The complete decryption process is outlined in Algorithm 2.

Algorithm 2 Image Decryption using Inverse S-box and Transpose

Input: Encrypted image I_E, 8-bit S-box S, file paths for encrypted image and S-box
Output: Decrypted image I_D

function INVERSE_SBOX(S)
    $S^{-1} \leftarrow$ array of same shape as S
    for each index i and value v in S **do**
        $S^{-1}[v] \leftarrow i$
    end for
    return S^{-1}
end function
function APPLY_INVERSE_SBOX(I_E, S^{-1})
    $F \leftarrow$ Flatten I_E to 1D array
    for each pixel p in F **do**
        $p \leftarrow S^{-1}[p]$
    end for
    Reshape F to original shape of I_E
    return Restored image I_R
end function
function DECRYPT_IMAGE($enc\_path, sbox\_path, save\_path$)
    $S \leftarrow$ LOAD_SBOX($sbox\_path$)
    $S^{-1} \leftarrow$ INVERSE_SBOX(S)
    $I_E \leftarrow$ LOAD_IMAGE(enc_path)
    $I_R \leftarrow$ APPLY_INVERSE_SBOX(I_E, S^{-1})
    $I_D \leftarrow$ Transpose(I_R) ▷ Restore original row-column alignment
    SAVE_IMAGE($I_D, save\_path$)
end function

Metric	Original vs. Reconstructed Image	Proposed Scheme
Correlation	High (≈ 1) (Perfect reconstruction)	1.00
RMSE	Low (≈ 0) (Minimal reconstruction error)	0.00
PSNR (dB)	High ($> 30 - 40$ dB) (High reconstruction quality)	∞
NPCR (%)	Low ($\approx 0\%$) (Minimal pixel variation)	0.00
UACI (%)	Low ($\approx 0\%$) (Minimal intensity variation)	0.00

Table 11: Statistical comparison between the original and reconstructed image

4. Discussion

We present a method for creating dynamic substitution boxes (S-boxes) that combines the sensitivity of chaotic systems with the algebraic structure of finite fields. The logistic map, which is the foundation of the design, is described as

$$X_{n+1} = \mu X_n(1 - X_n),$$

The control parameter is $\mu \in (3.5699, 4.0)$, while the initial condition is $X_0 \in (0, 1)$. These parameters are chosen so that the chaotic regime is operated in by the map, resulting in high entropy and randomness of the generated sequence.

When the control parameter is chosen close to the lower bound ($\mu \approx$3.57 or less), the logistic map does not show fully developed chaos or weakly chaotic behavior. As a result, the generated sequences are less unpredictable and random, which has a direct effect on the S-box's cryptographic strength. Nonlinearity decreased to 100 in our tests, SAC values were low (minimum 0.109, highest 0.719), and the encryption statistics significantly reduced. This occurs because increased correlations between values result from insufficient chaos, which weakens the defense of the substitution process against statistical and differential attacks.

To construct each S-box, 256 iterations of the logistic map are performed with rational approximations of X_n. At each step, a rational number is approximated as X_n, with the numerator and denominator reduced modulo 256. Elements of the Galois Field $GF(2^8)$ are shown by these values. For example, if $X_n = \frac{a_n}{b_n}$, the S-box entry is calculated as follows:

$$S[n] = a_n \cdot b_n^{-1} \mod P(x),$$

where the multiplicative inverse of b_n in GF(2^8) is denoted by b_n^{-1}., and arithmetic is performed modulo the irreducible polynomial

$$P(x) = x^8 + x^4 + x^3 + x^2 + 1.$$

To ensure that the resulting S-box is bijective, uniqueness is enforced by checking for duplicate outputs. The output value in GF(2^8) is continuously increased by us until a unique, unused value is obtained if a collision is discovered (i.e., an S-box value is already assigned).

S-boxes has 256 unique entries and are represented by a 16×16 matrix. The control parameter μ can be changed to result in multiple distinct S-boxes, each with unique internal structures. Key sensitivity is improved by this parametric dependence, and the creation of dynamic S-boxes based on specific cryptographic keys is enabled. The suggested structure is considered perfect for lightweight and adaptable cryptography systems since it is regarded as extremely resistant to static analysis and differential attacks.

Higher-dimensional chaotic maps, such as 2- D or 3-D systems, may offer improvements in the statistical properties of the generated S-box. Compared to 1-D maps, these systems often display richer dynamics and stronger mixing behavior, which could potentially lead to better values of nonlinearity, SAC, and entropy. Since our present work is focused only on a 1-D chaotic system, we cannot yet confirm their actual impact on cryptographic performance. Exploring this direction in the future—by comparing 1-D, 2-D, and 3-D chaotic maps—could provide valuable insights into whether such extensions deliver measurable gains in security metrics.

4.0.1. Discussion on Security Resistance

The proposed encryption scheme combines a permutation step (row–column interchange) with a nonlinear substitution step using an S-box. In this section, we briefly discuss its resistance against common types of cryptographic attacks.

1. **Statistical Attacks:** The substitution step introduces confusion by mapping each pixel value through an S-box, while the row–column interchange ensures diffusion by altering the pixel positions. Together, these operations reduce the correlation between adjacent pixels in the encrypted image. Histogram

analysis of the cipher image is expected to be uniform, which prevents an attacker from extracting meaningful statistical information.

2. **Differential Attacks:** A small change in the input image (such as modifying one pixel) spreads across multiple positions after the transpose and substitution operations. The nonlinearity of the S-box ensures that even a one-bit difference in the input leads to a significant difference in the output values, thereby providing resistance against differential cryptanalysis.
3. **Brute-force Attacks:** The security depends on the secrecy of the S-box and the applied permutation. If the S-box is generated securely (e.g., using a chaos-based or random construction), the attacker cannot feasibly guess the mapping. The search space for a full 8-bit S-box is 256!, which is computationally infeasible to exhaust.
4. **Known-plaintext and Chosen-plaintext Attacks:** Since the encryption uses both substitution and permutation, the relationship between the plaintext image and the ciphertext is highly nonlinear. Without access to the specific S-box used, recovering the plaintext from a known or chosen ciphertext pair is computationally impractical.
5. **Limitations:** The current scheme uses a single round of permutation and substitution. While this provides reasonable resistance for lightweight applications, security can be further improved by using multiple rounds of transpose and substitution, or by employing dynamic S-boxes that change with each encryption.
   Overall, the combination of permutation and S-box substitution provides confusion and diffusion, the two fundamental requirements of secure encryption as identified by Shannon. This makes the proposed scheme resistant to common cryptanalytic methods, while still remaining computationally efficient.

4.1. Complexity Analysis

The computational complexity of generating one S-box of size $N = 256$ is $O(N)$ in practice, with a rare worst-case of $O(N^2)$ due to collision resolution.Unlike the AES S-box, which is fixed and precomputed, our method regenerates S-boxes dynamically; however, since each iteration involves only constant-time GF(2^8) operations, the overall process remains lightweight and suitable for practical applications.

5. Conclusion

This work introduced a dynamic image encryption framework based on substitution boxes (S-boxes) generated from the chaotic logistic map and arithmetic over $GF(2^8)$. By varying the control parameter μ, multiple S-boxes can be produced, ensuring sensitivity to initial conditions and enhancing confusion. The proposed design achieved strong cryptographic performance, with a nonlinearity of 108, SAC values between 0.375 and 0.578, BIC-NL of 103.50, BIC-SAC of 0.502, a differential probability of 0.037, and a linear probability of 0.297, demonstrating resistance against differential and linear cryptanalysis and outperforming several existing S-box designs.

Validation through statistical and visual tests confirmed secure and reversible encryption, with decrypted images nearly identical to the originals (PSNR = ∞, RMSE = 0, and low NPCR/UACI). Since the current encryption method relies only on permutation and substitution, the design remains lightweight and efficient, making it particularly suitable for real-time multimedia, IoT, and biometric protection. Future extensions could explore higher-dimensional chaotic maps, integration with symmetric key ciphers such as AES, and robustness against emerging machine learning–based cryptanalysis.

6. Open Problems and Future Work

Even though good cryptographic characteristics and improved performance are seen by the proposed dynamic S-box based cryptography scheme compared to some of its existing competitors , there are open issues for future research:

- **Robustness against advanced attacks:** The resistance of the proposed scheme against algebraic, boomerang, and machine learning–based cryptanalysis has not yet been fully evaluated. Investigating these attack models remains an important open problem.
- **Extension to higher dimensions:**To increase nonlinearity and unpredictability, hybrid chaos-based generators or higher-dimensional chaotic maps could be used by future research.
- **Alternative finite fields:** Exploring different irreducible polynomials in $GF(2^8)$, or even extensions to other fields, may lead to more diverse and secure dynamic S-box constructions.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent for publication The author gave their consent for publication.

Ethical approval Not applicable.

Consent to participate Not applicable.

Code availability Code is available on request from the authors.

Author Contribution U. Ali conceived the presented idea and supervised. A. Rani, S. A. Bokhary and M.Shams performed the computations. M. Shams and S. A. Bokhary wrote the article and coding. S. A. Bokhary and U. Ali validate the findings. All authors discussed and reviewed the results and contributed to the final manuscript.

References

[1] J. Kim and R. C.-W. Phan, *Advanced differential-style cryptanalysis of the NSA's Skipjack block cipher*, Cryptologia, **33** (2009), 246–270. https://doi.org/10.1080/01611190903013936. 2.2

[2] S. A. U. H. Bokhary, A. Kharal, F. M. A. Samman, M. E. E. Dalam, and A. Gargouri, *Efficient Graph Algorithms in Securing Communication Networks*, Symmetry, **16**(10) (2024), 1269. https://doi.org/10.3390/sym16101269. 1

[3] D. C. Munson, *A Note on Lena*, IEEE Transactions on Image Processing, **5** (1996). https://doi.org/10.1109/83.481795. 3.6.1

[4] A. H. Zahid, H. Rashid, M. M. U. Shaban, S. Ahmad, E. Ahmed, M. T. Amjad, M. A. T. Baig, M. J. Arshad, M. N. Tariq, M. W. Tariq, M. A. Zafar, and A. Basit, *Dynamics-box Design Using a Novel Square Polynomial Transformation and Permutation*, IEEE Access, **9** (2021), 82390–82401. https://doi.org/10.1109/ACCESS.2021.3086759. 1

[5] R. Arshad and M. Jalil, "Comment on Nizam Chew, L.C.; Ismail, E.S. S-box Construction Based on Linear Fractional Transformation and Permutation Function," *Symmetry*, vol. 15, no. 5, Art. no. 1005, 2023. doi: 10.3390/sym15051005. Available: https://www.mdpi.com/2073-8994/15/5/1005. 1

[6] M. Asif, S. Mairaj, Z. Saeed, M. U. Ashraf, K. Jambi, and R. M. Zulqarnain, *A Novel Image Encryption Technique Based on Möbius Transformation*, Computational Intelligence and Neuroscience, (2021), 1–14. https://doi.org/10.1155/2021/1912859. 7

[7] B. Arshad, N. Siddiqui, Z. Hussain, and M. E. Haq, *A Novel Scheme for Designing Secure Substitution Boxes (S-boxes) Based on Möbius Group and Finite Field*, Wireless Personal Communications, **124** (2022). https://doi.org/10.1007/s11277-022-09609-y. 1, 7

[8] N. Azam, U. Hayat, and I. Ullah, *Efficient Construction of S-boxes Based on a Mordell Elliptic Curve Over a Finite Field*, Frontiers of Information Technology & Electronic Engineering, (2018), 1–18. https://doi.org/10.1631/FITEE.1800345. 7

[9] I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, and H. Mahmood, *A Group Theoretic Approach to Construct Cryptographically Strong Substitution Boxes*, Neural Computing and Applications, **23** (2013), 97–104. https://doi.org/10.1007/s00521-011-0749-2. 7, 10

[10] L. Cui and Y. Cao, *A new S-box structure named affine-power-affine*, *International Journal of Innovative Computing, Information and Control*, **3**(3) (2007), 751–759. 1

[11] R. M. May, *Simple Mathematical Models with Very Complicated Dynamics*, Nature, **261** (1976), 459–467. https://doi.org/10.1038/261459a0. 2.1

[12] P. Zhou, J. Du, K. Zhou, and S. Wei, *2D Mixed Pseudo-Random Coupling PS Map Lattice and Its Application in S-box Generation*, Nonlinear Dynamics, **103** (2021). https://doi.org/10.1007/s11071-020-06033-w. 1

[13] A. H. Zahid, L. Tawalbeh, M. Ahmad, A. Alkhayyat, M. T. Hassan, A. Manzoor, and A. K. Farhan, "Efficient Dynamic S-Box Generation Using Linear Trigonometric Transformation for Security Applications," *IEEE Access*, vol. 9, pp. 98460–98475, 2021. doi: 10.1109/ACCESS.2021.3095618. 1

[14] D. Lambic, *A Novel Method of S-box Design Based on Discrete Chaotic Map*, Nonlinear Dynamics, **87** (2017), 2407–2413. https://doi.org/10.1007/s11071-016-3170-7. 2.1

[15] A. K. Farhan, R. S. Ali, H. Natiq, and N. Al Saidi, *A New S-box Generation Algorithm Based on Multistability Behavior of a Plasma Perturbation Model*, IEEE Access, **7** (2019), 124914–124924. https://doi.org/10.1109/ACCESS.2019.2938767. 7

[16] 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. https://doi.org/10.1016/j.chaos.2006.03.030. 7

[17] D. Feng and W. Wu, *Design and Analysis of Block Ciphers*, Doctoral Dissertation, Ruhr University Bochum, Bochum, Germany, (2000). 10

[18] D. Lambic, *A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design*, *Nonlinear Dynamics*, **100**(1) (2020), 699–711. Springer Science+Business Media. DOI: 10.1007/s11071-020-05503-y. 1

[19] D. Feng and W. Wu, *Design and Analysis of Block Ciphers*, Tsinghua University Press, Beijing, (2000). 3.3.2

[20] A. Anees, A. M. Siddiqui, and F. Ahmed, *Chaotic Substitution for Highly Auto-correlated Data in Encryption Algorithm*, Communications in Nonlinear Science and Numerical Simulation, **19**(9) (2014), 3106–3118. https://doi.org/10.1016/j.cnsns.2013.12.031. 10

[21] M. Matsui, *Linear Cryptanalysis Method for DES Cipher*, In *Advances in Cryptology – EUROCRYPT'93*, Lecture Notes in Computer Science, Vol. 765, Springer, Berlin, Germany, (1994), 386–397. https://doi.org/10.1007/3-540-48285-7-33. 1

[22] M. Ibnkahla, *Signal Processing for Mobile Communications Handbook*, Chapter 27, CRC Press, Boca Raton, FL, USA, (2005). 3.3.3

[23] H. Ma and L. Liu, *Algebraic Expression for AES S-box and InvS-box*, Computer Engineering, **32**(18) (2006), 149–151. 1

[24] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Journal of Cryptology, **4**(1) (1991), 3–72. https://doi.org/10.1007/BF00630563. 1

[25] J. Daemen and V. Rijmen, *The Design of RIJNDAEL: AES—The Advanced Encryption Standard*, Springer-Verlag, Berlin, (2002). 1

[26] P. P. Mar and K. M. Latt, *New analysis methods on strict avalanche criterion of S-boxes*, World Academy of Science, Engineering and Technology, **48** (2008), 150–154. https://doi.org/10.5281/zenodo.1062500. 10

[27] M. Khan, T. Shah, and S. I. Batool, *Construction of S-box Based on Chaotic Boolean Functions and Its Application in Image Encryption*, Neural Computing and Applications, **27**(3) (2016), 677–685. 7