AIMS Mathematics, 10(3): 5671–5695.
DOI: 10.3934/math.2025262
Received: 11 December 2024
Revised: 24 January 2025
Accepted: 20 February 2025
Published: 14 March 2025

http://www.aimspress.com/journal/Math

Research article

Batch generated strongly nonlinear S-Boxes using enhanced quadratic maps

Mohammad Mazyad Hazzazi[1], Farooq E Azam[2], Rashad Ali[3,*], Muhammad Kamran Jamil[4], Sameer Abdullah Nooh[5] and Fahad Alblehai[6]

[1] Department of Mathematics, College of Science, King Khalid University, Abha 61413, Saudi Arabia

[2] Department of Mathematics, Riphah International University, 50390 Lahore, Pakistan

[3] Department of Mathematics, University of Trento, 38122 Trento, Italy

[4] Department of Mathematics, Riphah International University, 54660 Lahore, Pakistan

[5] Faculty of Computing and Information Technology King AbdulAziz University Jeddah 80200, Saudi Arabia

[6] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudia Arabia

* **Correspondence:** Email: rashadwattu@gmail.com.

Abstract: One of the most crucial elements in the design of a block cipher is the substitution box or S-box. Its cipher strength directly impacts the cipher algorithm's security, and the block cipher algorithm requires a good S-box. According to the cryptanalysis result of the S-box construction in AES: (1) the number of irreducible polynomials can be increased to 30; (2) the affinity transformation constant c can be chosen from all elements if the existence of fixed points and reverse fixed points in an S-box is ignored; and (3) the S-box in AES is fixed, which poses possible security risks to the AES algorithm. The study above led us to build a non-degenerate 2D enhanced quadratic map (2D-EQM) with unpredictability and ergodicity. From there, we generated affine transformation constants and affine transformation matrices, which were then applied to seed S-boxes to create a batch of strongly nonlinear S-boxes. Finally, we assessed the performance of suggested S-boxes using six criteria. Security and statistical research showed that the suggested S-box batch generation procedure was practical and effective.

Keywords: 2D Enhanced Quadratic Map (2D-EQM); affine transformation; cryptographic algorithms; nonlinearity; S-Box design
Mathematics Subject Classification: 94A60, 68P25

1. Introduction

In terms of cybersecurity, protecting sensitive data and utilizing secure communication techniques are essential to preventing unauthorized access, data breaches, and cyber-attacks. Data security is a significant challenge for cryptographers given the rapid advancement of communication technologies. The main objective of cryptography is to develop methods that ensure secure network communication. The name "cryptography" comes from two Greek words: "graphein," which denotes the act of learning or writing, and "kryptos," which means something hidden or unrevealing. Various useful encryption methods and procedures have been established in engaging literary works to ensure the security of data transmission. Often, maintaining information security is thought to be the main goal of cryptography. Major contributions to the creation of modern cryptography have been made in fields including electrical engineering, physics, computer science, mathematics, and communication science. The nonlinear part of block cipher cryptosystems is called an S-box. The Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA), and the Data Encryption Standard (DES) are examples of cryptographic techniques that use the S-box. The S-box's security affects the security of the entire cryptosystem. Thus, It is well known that the S-box, a nonlinear component, is crucial to maintaining the security of cryptographic systems. The DES was introduced in 1977 by a well-known computer manufacturer, and further research resulted in major improvements to the cryptographic method.

Eventually, a group of college students broke through DES's protection. The most used encryption scheme is the Advanced Encryption Standard (AES), created by Daemen and Rijmen in 2002. The reliability of encryption is significantly influenced by the S-box. Using a subpar S-box when encrypting data is similar to exploiting the security of the encryption. Consequently, it is essential to evaluate an S-box's robustness before using it in a cryptosystem. The severe avalanche requirement, bit independence criteria, nonlinearity, linear approximation probability, and the probability of differential approximation are among the methods of strength measurement used for the S-box examination. An essential component of contemporary cryptographic approaches, symmetric key cryptography ensures the confidentiality, integrity, and validity of digital data. S-boxes, often called substitution boxes, are crucial elements in many symmetric key cryptography techniques since they add confusing things, and, being nonlinear there is an increase in security. In cryptography, chaotic maps are used to create difficult-to-predict pseudo-random sequences. Complex systems, even deterministic ones, can be studied and modeled using chaotic maps because of their irregularity and unpredictability. Bifurcation is used in chaos theory and dynamical systems to characterize a qualitative shift in a system's behavior when a parameter changes. The word "bifurcation" in S-boxes refers to cryptography, specifically the creation and examination of cryptographic methods. An S-box is an essential part of many symmetric key algorithms, including block ciphers. Its purpose is to create confusion by performing substitution operations, which obscures the connection between the ciphertext and the key. When examining an S-box's resistance to different types of assaults, bifurcation can be linked to how minor adjustments to the input or key impact the encryption procedure as a whole.

2. Related work

Performance and security level of the encryption system are directly determined by the quality of the S-box, which is the main nonlinear component in many block cipher algorithms. Consequently, the development of the S-box with superior performance has emerged as a significant area of study that has drawn interest from many academics. To increase the nonlinearity of the original S-boxes, [1] employed a Josephus circle problem. A nonlinearity of 110.75 is achieved by the suggested S-box. Ali *et al.* [2] proposed a new design that uses a direct product of cyclic groups and the Galois field to produce a robust S-box. Instead of a fractional transformation, they employed a highly nonlinear inversion map of the Galois field. [3] used Arnold's Cat map to generate dynamic S-boxes. The technique generated nonlinear and efficient S-boxes, however it does not ensure bijectivity for each S-box. On average, the proposed scheme's nonlinearity was 107. The authors in [4] created S-boxes using a novel chaotic system. A very nonlinear S-box was created in [5] using a newly constructed chaotic sine map. The S-box's nonlinearity was increased by the authors using an optimization model, however, the scheme's average nonlinearity remained at 110.25. A novel method for creating a sturdy S-box using a multi-layer perceptron architecture and linear fractional transformation was presented in [6]. S-boxes created using a combination of algebraic and chaotic processes outperform S-boxes constructed solely using algebraic operations or chaos in terms of cryptographic performance. Furthermore, combining algebraic and chaotic models yields a better trade-off between S-box execution and generating efficiency, and this strategy is starting to show promise for creating S-boxes. A growing number of study disciplines have recently focused on creating hyperchaotic maps with intricate dynamics. The authors designed a new two-dimensional exponential chaotic system (ECS) [7]. Because the 2D-ECS cascades exponential nonlinearity with bounded functions, it can produce a huge number of hyperchaotic maps. By using trigonometric functions to cascade the exponential nonlinearity, three hyperchaotic maps were produced to demonstrate the efficacy of the 2D-ECS. Using a variety of numerical measurements, the authors first constructed state-mapping networks with varying fixed-point arithmetic precisions in order to examine the dynamic features of the hyperchaotic maps in the digital realm. The developed hyperchaotic maps outperformed the current chaotic maps in terms of performance indicators, according to experimental data. As a universal system that may produce numerous 2-D chaotic maps with various exponent coefficient settings, [8] suggested a two-dimensional (2-D) parametric polynomial chaotic system (2D-PPCS). The 2D-PPCS first initialized two parametric polynomials before subjecting them to modular chaotification. By varying the control parameters, the 2D-PPCS was able to tailor its Lyapunov exponents to achieve the necessary complexity and robust chaos. The resilient chaotic behavior of the 2D-PPCS was shown via theoretical research. Two illustrated cases were presented and evaluated using numerical experiments to confirm the 2D-PPCS's efficacy. Additionally, a pseudorandom number generator based on chaos was created to demonstrate the uses of the 2D-PPCS. Based on the homogenized disturbed spatiotemporal chaotic system [9], the dynamic S-box generation method was developed. Various techniques are also used to create S-boxes, including heuristic, genetic, and genuine random methods. Researchers have created keyed S-boxes in response to the shortcomings of static S-boxes. Kazlauskas *et al.* [10] suggested ways to produce a substantial quantity of S-boxes based on keys.

The combination of chaos theory and algebraic techniques in the literature has led to innovative S-box designs that offer enhanced security against cryptanalytic attacks while maintaining efficiency for

encryption applications. Using the dynamic irreducible polynomial and the affine constant, a dynamic S-box was developed in [11]. The authors in [12] created and executed a novel AES block cipher variant that relies on an S-box cube that depends on the key. [13] developed a new computationally effective technique that uses key-dependent permutations over finite elliptic curves to create dynamic S-boxes. An extremely nonlinear S-box was constructed in [14] using a logistic chaotic map, symmetric group of permutation, and projective general linear group action. A genetic method was used in [15] to generate extremely nonlinear bijective S-boxes. S-boxes with an average nonlinearity of 110.75 were produced in [16] by putting forth a novel mixed chaotic system with favorable pseudo-random characteristics. A new approach to building an S-box using the chaotic system and the full Latin square was presented in [17]. A chaotic system first generates chaotic sequences that are used to create a complete Latin square. The full Latin square is then used to create an S-box. Performance analyses reveal that the S-box formed by the suggested method has a good performance and can withstand a wide range of security attacks, including the linear attack and differential attack. To demonstrate the efficacy of the S-box, this study used it to an image encryption application. [18] provided a crucial concept for creating symmetric rotating surfaces, and a generalized hybrid trigonometric Bézier curve is used to describe curves in engineering. The authors of [19] created a powerful S-box creation method based on EQM that combines and merges all rings with short periods into one maximum ring. The nonlinear confusion component was constructed in [20] using a straightforward and effective technique. The derived confusion component has a low nonlinearity of 105.5, making it resistant to differential and cryptographic attacks. The authors in [21] employed a watermarking-based method with chaotic fractional transformation properties to build the S-box. While the technique is intriguing and effective, the resulting S-box has a relatively low nonlinearity of 102.3. The quantum logistic map [22] was utilized to generate numerous nonlinear confusion components. However, by maximizing the parameters with the highest nonlinearity, choose just two confusion components. Although the produced S-boxes have extremely little non-linearity, this technique [23] is quite good. Strong S-boxes were constructed using three finite fields of order 256, an affine map, and an inversion map. The method used is easy to use and incredibly effective for creating robust S-boxes, nonetheless, we can only produce a certain amount of S-boxes with this arrival.

The combination of the chaotic systems has led to the development of a novel S-box generating technique in [24]. The authors in [25] used a method of image encryption using two keys. A linear congruential generator and a 2D logistic sine map produce the first key, whereas the Tent, Bernoulli, and KAA maps produce the second. For image security in cloud storage, [26] proposed a simplified picture encryption algorithm (SIEA) based on the Feistel cipher structure that utilized key generation and permutation. To encrypt digital images, [27] proposed ARHM (AES and Rossler hyperchaotic modelling), which combines the Rossler hyperchaotic system with AES with phantom transformation. The key space, key sensitivity, histogram, pixel correlation, entropy, and resistance to differential assaults are all simulated and examined using this model. It uses AES encryption speed and chaotic system randomization. Liu *et al.* [28] created an image encryption technique based on a non-degeneracy 3D chaotic map and a keyed strong S-box that can encrypt color images of all sizes. First, they created a non-degeneracy 3D discrete hyperchaotic map (3D-DHCM), which is then used to create a keyed strong S-box with no fixed point, reverse fixed point, or short period rings. The map is based on the discrete logarithm problem, which is the inverse function of the modular exponentiation procedure. Finally, the authors blurred the raw image before encryption, and then used permutation, confusion,

and diffusion algorithms to shuffle all pixels. The authors in [29] developed a technique for improving the security of medical photographs, and produced robust S-boxes via Mobius transformation on a Galois field. Quantum theory has been increasingly applied to image encryption in recent years. The DNA coding-based image encryption algorithm and quantum chaotic pap (QCMDC-IEA) has inherent security flaws; its DNA domain encryption is susceptible to attacks, such as the presence of an equivalent key generated by different chaos-based sequences. A proposed attack method exploits these shortcomings to provide complete decipherment and low complexity.

2.1. Literature gaps

A powerful S-box meets three conditions: no fixed points, no reverse fixed points, and no short iterating cycles. Nevertheless, the majority of S-boxes built with the previously discussed methods either lack many short cycles or have fixed or reverse fixed points, which means they do not meet these three requirements. These problems have a direct effect on the S-boxes' strength, opening up possible openings for attackers. A lot of these S-boxes also have low nonlinearity. In cryptography, nonlinearity is one of the most significant features of S-boxes since it is vital for increasing resistance to linear cryptanalysis [30]. However, it is important to consider the shortcomings of the popular 1D chaotic maps, like the Tent, Henon, sine, and logistic maps. One example is the short iteration periods, restricted chaotic range, weak randomness, and lack of ergodicity in 1D chaotic maps. The generated sequences could be attacked because of these flaws. The security of constructed S-boxes must therefore be guaranteed by building a multi-dimensional chaotic map.

2.2. Motivations and contribution

Our motivations are as follows:

1. Creating novel chaotic mappings for the creation of pseudo-random numbers.
2. Investigating how algebraic structures and transformations are affected by chaotic mappings.
3. Development of numerous S-boxes with robust cryptographic characteristics.

Our contributions are as follows:

1. S-box weakness analysis: The S-box structure has two flaws: short iteration cycles, which could be a cryptography exploit, and fixed point or reverse fixed point.
2. We created a 2D enhanced quadratic map (EQM) in order to get over the drawbacks of 1D chaotic maps, which include numerous bifurcations, narrow key space, dense periodic windows, and a brief iteration duration.
3. Security analyses show that the robust S-box construction approach works well for cryptography.

2.3. Organization of article

This article is organized as follows: Section 3 deals with the study of chaotic maps, their analysis, and construction of a new hybrid EQM. Section 4 consists of a construction algorithm for an S-box using affine matrices and Galois fields. The evaluation criteria are defined and discussed in Section 5 for dynamic S-boxes. Finally, Section 6 concludes the study.

3. Chaotic map

There are several characteristics of chaotic maps that make them very good options for encryption systems. Chaotic maps possess space in their system parameters, are pseudorandom, ergodic, and sensitive to initial conditions. They are divided into maps with low and large dimensions. The number of variables and parameters in low-dimensional maps is modest. Therefore, they are straightforward and simple to use. Because of their small chaotic range and parameter values, they are nevertheless easily predictable. However, a high-dimensional map's quantity of parameters and variables has a greater range of chaotic space since their height is higher. Nevertheless, their drawbacks make them difficult to apply in real-time due to their complexity and high processing overhead.

3.1. Logistic map

A particular kind of chaotic map, known as a logistic chaotic map, is more widely used than the others and is mostly used in picture encryption methods. A two degree polynomial mapping includes the logistic map. It is a famous example of a chaotic, complicated system with basic nonlinear dynamics. Many of the characteristics of this straightforward system are common to pseudorandom number generators (PRNGs) [3], and it can readily transition from order to chaos. Logistic map have the advantages of simplicity and ease of use due to their low variation, however they have several disadvantages, such as chaotic orbits and parameters and beginning values that may be used to define them being easily predictable. The logistic chaotic map may be computed mathematically. Figure 1 shows a logistic map Lyapunov exponent, and Figure 2 shows a logistic map bifurcation.

$$x_{n+1} = \lambda x_n(1 - x_n)$$

The number of iterations is denoted by n, while the chaotic parameter is represented by λ.

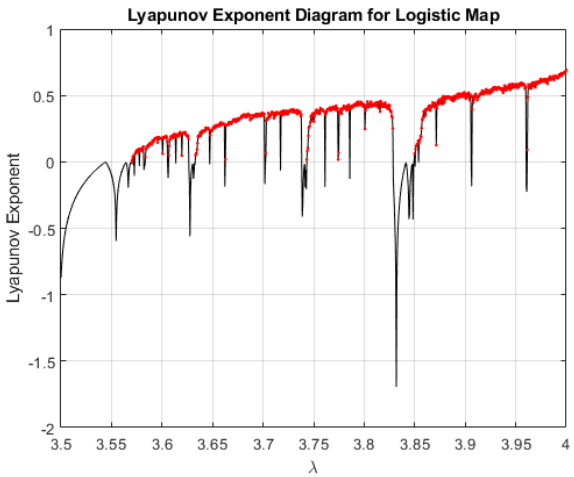

Figure 1. Lyapunov exponent diagram of a logistic map.

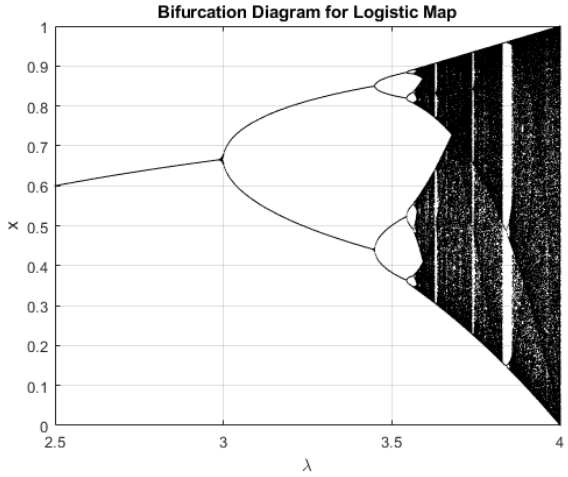

Figure 2. Bifurcation diagram of a logistic map.

3.2. Lyapunov exponent

One of the fundamental ideas in chaos theory is the Lyapunov exponent, which quantifies the speed at which neighboring paths in a dynamical system diverge or converge over time. It measures how sensitive a system is to starting conditions, which is a sign of chaotic behavior. The Lyapunov exponent is used to quantify how sensitive a chaotic system is to initial circumstances. Because it guarantees that even a small alteration to an initial key or state produces a radically different sequence, this sensitivity is desired in cryptography and adds to the system's unpredictability and security. The Lyapunov exponent formula is

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \log\left(\frac{\|\delta x_n\|}{\|\delta x_0\|}\right)$$

where: (1) the initial perturbation or difference in the input is denoted by δx_0; (2) the perturbation that occurs after n repetitions of the cryptographic function is δx_n and (3) an appropriate metric, such as the Euclidean norm, is shown by $\|\cdot\|$

3.3. Bifurcation

A term commonly used in the field of cryptography to describe bifurcation is taken from chaos theory and dynamical systems. When parameters are changed, cryptographic systems or functions exhibit behavior that can alter dramatically. This term is used to characterize such changes in behavior.

3.4. 2D-Hybrid hyper choatic map

We created a 2D-EQM with modular arithmetic based on the standard quadratic map 3.1 in order to address its shortcomings, including its limited key space that may communicate equations, lack of ergodicity, and poor unpredictability.

$$\begin{aligned} x_{i+1} &= \mathrm{mod}\left(a^{\pi+x_i} \cdot r\left(1 - y_i^2 + \exp(x_i) + \sinh(x_i)\right), 1\right) \\ y_{i+1} &= \mathrm{mod}\left(b^{\pi+y_i} \cdot r\left(1 - x_i^2 + \exp(y_i) + \sinh(y_i)\right), 1\right) \end{aligned} \tag{3.1}$$

where the state variables $x, y \in (0, 1)$ and the control parameter $r \in (0, 1800]$ are double precision floating point numbers in Eq 3.1, and $a \in (1, 10], b \in (1, 20]$. The phase diagram and bifurcation diagrams are shown in Figure 3, Figure 4, and Figure 5. The two positive Lyapunov exponents shown in Figure 6 demonstrate that, over a larger range of control settings, the 2D-EQM exhibits hyperchaotic behavior and is non-degenerate.

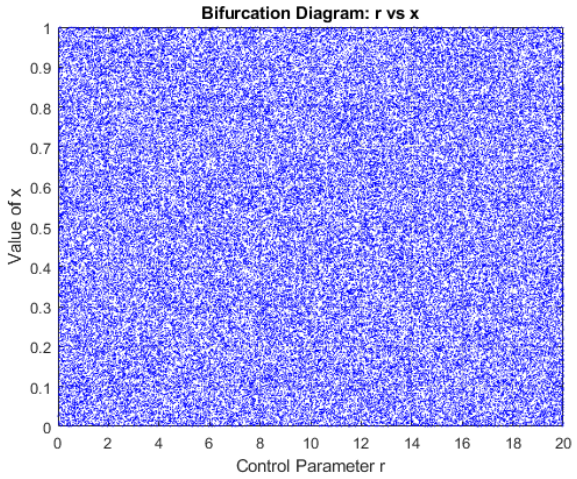

Figure 3. Bifurcation diagram of parameter r and state variable x.

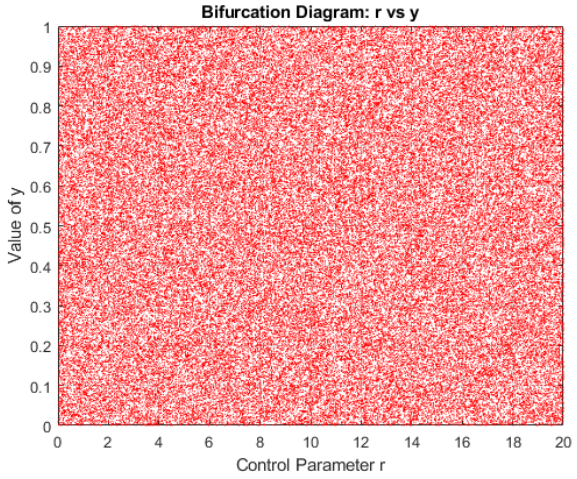

Figure 4. Bifurcation diagram of parameter r and state variable y.

The Lyapunov exponent of proposed map is shown in Figure 6. The values of the Lyapunov exponents for mapping (3.1) are 14.74813 and 18.235113 using the parameters $x_0 = 0.762853479752345, y_0 = 0.575685981383182, a = 5, b = 12$ and $r = 1000$.

4. Constructing an S-box with a powerful key

The construction of the affine transformation constant and matrix using 2D-EQM is described in this section. The number of S-boxes built was then determined by using them to create a keyed strong

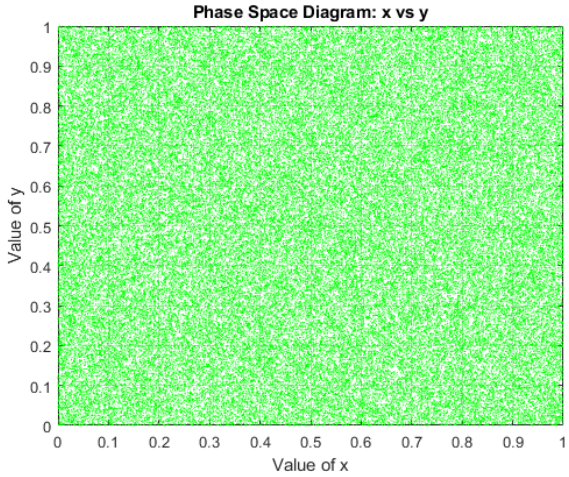

Figure 5. Phase diagram of state variables x and y.

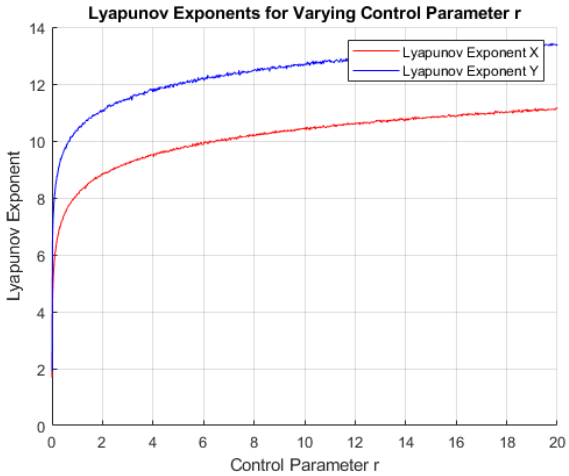

Figure 6. Lyapunov exponent for 2D-EQM (a = 9, b = 15).

S-box based on a seed S-box with high nonlinearity. Thirty irreducible polynomials in the order listed in the table make up this collection.

4.1. Description of algorithms

Input: The initial condition

$$(x_0, y_0, r_0)$$

in Eq (1) is the key of KEY. Output: An 8×8, S-box with high nonlinearity and a strong key. To construct a keyed strong S-box, follow these steps.

Step 1: Affine transformation through the construction of matrix B.

200 iterations of Eq 3.1 with (x_0, y_0, r_0) to eliminate the impact of the transitory process. After that, 64 iterations are needed to produce the two sequences X and Y, and Eq 4.1 yields an invertible matrix B:

$$B = \text{reshape}\left(\text{mod}\left(\left\lfloor (x_{201:264} + y_{201:264}) \cdot 10^{15} \right\rfloor, 2\right), 8, 8\right) \tag{4.1}$$

This equation contains the elements from index 201 to 264. In the given index range, the corresponding values of x and y are also added element-wise in the equation. The final values are multiplied by 10^{15} after the addition. To deal with tiny fractional portions, for example, if x and y are floats, this step greatly scales up the values and improves their precision. For every element, this operation applies modulo 2. Usually, this is done to change the values to binary. The rebuilt matrix will have eight rows and eight columns since the vector must have 64 elements in total. The matrix B, an 8×8 binary matrix, is the result. If $|B| = 0$, we can use Eq 4.2 to produce a new B by altering the control parameter r and the state variable values (x, y) from the previous iteration:

$$\begin{aligned} x &= x_0 + \frac{\sqrt[3]{p}}{10^8}, \\ y &= y_0 + \frac{\sqrt[3]{p}}{10^7}, \\ r &= r_0 + \frac{\sqrt[5]{p}}{10^6}. \end{aligned} \tag{4.2}$$

here, p is a prime number in the interval $(100, 1000)$.

Step 2: Selecting an irreducible polynomial

After being scaled by 10^{15} and reduced modulo 30, the sum of x is converted into an index, which is then increased by 1 in Eq 4.3:

$$i = \text{mod}\left(\text{floor}\left(\sum X \cdot 10^{15}\right), 30\right) + 1 \tag{4.3}$$

Step 3: An affine transformation vector C is created.

The sum of y, scaled by 10^{15}, scaled and reduced modulo 256, is used to compute C. After being transformed into a binary string, C is saved as c, a column vector of binary values in Eq 4.4

$$C = \text{mod}\left(\text{floor}\left(\sum y \cdot 10^{15}\right), 256\right) \tag{4.4}$$

Step 4: Constructing a potential S-box Sc.

Choose an element $z \in GF(2^8)$ generated by the irreducible polynomial. Convert z into binary form and consider the following transformation.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & b_{17} & b_{18} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} & b_{27} & b_{28} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} & b_{37} & b_{38} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} & b_{47} & b_{48} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} & b_{57} & b_{58} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} & b_{67} & b_{68} \\ b_{71} & b_{72} & b_{73} & b_{74} & b_{75} & b_{76} & b_{77} & b_{78} \\ b_{81} & b_{82} & b_{83} & b_{84} & b_{85} & b_{86} & b_{87} & b_{88} \end{bmatrix} \cdot \left(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \end{bmatrix} \right)^2 + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix}$$

After applying the transformation, convert

$$\left[a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\right]$$

to decimal form.

Algorithm 1 Keyed Strong S-Box Construction

Input: Initial condition (x_0, y_0, r_0)
Output: An 8×8 S-box
Step 1: Affine transformation through the construction of matrix B
Perform 200 iterations with (x_0, y_0, r_0) to remove transient effects
Perform 64 iterations to generate sequences X and Y
Compute binary matrix B using Equation (3.2)
Step 2: Selecting an irreducible polynomial
Scale the sum of x by 10^{15}, reduce modulo 30, and compute index i using Eq (3.4)
Step 3: Creating affine transformation vector C
Scale the sum of y by 10^{15}, reduce modulo 256, and compute C using Eq (3.6)
Step 4: Constructing potential S-box S_c
Choose an element $z \in GF(2^8)$ and apply the transformation
Convert the resulting vector to decimal form to complete S_c
Apply removal process to obtain strong keyed S-box

4.2. Determine and eliminate an S-box's weaknesses

Despite being widely utilized in several cryptosystems, S-box still has certain flaws that can make it vulnerable, like short cycles and fixed point or reverse fixed points. Invalid substitution may result from the fixed point or reverse fixed point. An attacker using an S-box can readily predict the fixed point or reverse fixed point of a different S-box, which can be a fingerprint. Repetitive iteration from any element cannot traverse all of the elements due to S-box's short cycles, which could result in a strong attack that is unusual. There are only 1108 S-boxes that can be built using 30 irreducible polynomials in AES, thus there are not many of them. They also do not depend on the key, and the majority of them have short cycles. The fixed point and reverse fixed-point detection are absent from the S-box in RC4. For the block cipher SM 4.0, there is still one fixed point and eight short periodic rings even if its S-box is built via nonlinear transformation. The elimination process was used by authors in [31] and

can be understood by Algorithm 2 and Algorithm 3. This method achieved 99.072 percent results for elimination.

Note: We can change the state variable values x, y of the previous iteration if there are still issues.

The final S-box will be available after all flaws have been fixed. It will be safe and robust for use in cryptography. A sample S-box is displayed in Table 1 with $x_0 = 0.830136384779407, y_0 = 0.207884140559460, a = 2, b = 3$ and $r = 59$.

Algorithm 2 Elimination of Fixed and Reverse Fixed Points in S-box with Cycle Detection

Input: Initial S-box
Output: Final S-box after elimination of fixed points, reverse fixed points, and cycle corrections.
Divide the S-box into 16 smaller 4×4 matrices $S_{i,j}$, where $i, j \in \{0, 1, 2, 3\}$
for each $S_{i,j}$ in the S-box **do**
Detect if there exists a fixed point $P_{i,j}(r, c)$ or a reverse fixed point $P_{i,j}(r, c)$, where $r \in [0, 3], c \in [0, 3]$
if a fixed point or reverse fixed point is found **then**
Apply Eq (6): $P_{i,j}(r, c) \leftrightarrow P_{i,j}(r, [c + 1] \mod 4)$ (swap with right neighbor)
Apply Eq (7): $P_{i,j}(r, c) \leftrightarrow P_{i,j}([r + 1] \mod 4, c)$ (swap with bottom neighbor)
end if
end for
Call: `Func_Cycles(S)`
Output: Final S-box after removing short iterating cycles

Table 1. Sample S-box.

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

Algorithm 3 Func_Cycles Function to Handle Cycles in S-box

```
Input: S-box S
Output: Updated S-box S after cycle elimination.
cycles = findCycles(S)    (Find all cycles in the S-box)
for each cycle in cycles do
    if length of the cycle equals the length of S then
        fprintf('No short cycles found.')
        return    (Exit function if no short cycles found)
    end if
end for
fprintf('Found d cycles:  ')
for i = 1 to length(cycles) do
    fprintf('Cycle i (length d):  ', i, length(cyclesi))
    disp(cyclesi)
end for
while length(cycles) ≥ 2 do
    Get the last element of the first cycle: lastElem = cycles1(end)
    Get the first element of the second cycle: firstElem = cycles2(1)
    Find positions of these elements in S:
    posLastElem = find(S == lastElem)
    posFirstElem = find(S == firstElem)
    Swap these elements:
    S([posLastElem, posFirstElem]) = S([posFirstElem, posLastElem])
    Recompute the cycles: cycles = findCycles(S)
end while
fprintf('S box after removing short iterating cycles:  ')
disp(S)
```

5. S-box security analysis

This section presents the results of security assessments that were carried out on the suggested S-boxes to ascertain their level of resistance to cryptographic assaults. The probability of linear approximation (LAP), bit independence criteria (BIC), nonlinearity, strict avalanche criteria (SAC), and differencing approximation (DAP) in action were the five tests used to evaluate the S-box (See Table 2).

Table 2. Comparison of cryptographic properties of S-boxes constructed from various mathematical structures.

Mathematical Structure	S-boxes	Nonlinearity	SAC	BIC Nonlinearity	BIC SAC	LAP	DAP
Hyper Chaotic map	Proposed	112	0.5066	112	0.5027	0.0625	0.0156
Optimization	[1]	110.5	0.5100	103	0.4998	-	0.0391
Cyclic groups	[2]	112	0.5034	112	0.5066	0.0625	0.0156
Chaos	[5]	110.25	0.5027	102.71	0.4936	0.1250	0.0469
ECC	[13]	107.75	0.5010	103.93	0.5038	0.1250	0.0391
Hyper Chaotic map	[24]	112	0.5017	111.64	0.5006	0.0156	0.0703
Hyper Chaotic map	[26]	103.75	0.501	103	-	0.141	0.039
$GF(2^8)$	[29]	112	0.4988	112	0.5008	0.0625	0.0156
Hyper Chaotic map	[31]	110.75	0.4976	110.07	0.5034	0.0859	0.0234
Hyper Chaotic map	[32]	107.25	0.4981	104.42	0.5008	-	-
Hyper chaotic map	[33]	112	0.4971	112	0.4997	0.0625	0.0156
Optimization	[34]	112.0	0.5031	112.00	0.51120	0.092610	0.0291800
Chaos	[35]	112.00	0.5061	111.28	0.5016	0.0703	1.5625
$GF(2^8)$	[36]	112	0.5032	112	0.5059	0.0625	0.0156
$GF(2^8)$	AES	112	0.5040	112	0.5046	0.0625	0.0156
$GF(2^8)$	[37]	112	0.4980	112	0.5017	0.0625	0.0156
transfer-function	[38]	105.4039	0.5024	105.3571	0.5063	0.1171	0.0390
Lu-Chen	[39]	105.75	0.4939	103.43	0.5032	0.1171	0.0390
random selection	[40]	102.75	0.4978	103.35	0.5007	0.1328	0.0468
Block Ciphers	[41]	106	0.5051	98	-	0.148	0.039
Block cipher	[42]	107.00	0.4970	-	0.5070	0.0148	0.0470
chaotic system	[43]	105.88	0.5084	103.18	0.5087	0.1288	0.0391
SEC	[44]	112	0.5010	112	0.5000	0.0625	0.0156
Chaos	[45]	109	0.5	-	-	-	-
$GF(2^8)$	[46]	112	0.5002	112	0.5054	0.0625	0.0156

5.1. Nonlinearity (NL)

Our goal is to have the nonlinearity value as high as feasible because it directly affects password security. By increasing nonlinearity, nonlinear attacks can be resisted. Our top S-boxes in Table 1 achieve the ideal nonlinearity value for 8-bit S-boxes, which is 112. In Figure 7, the nonlinearity for 1000 S-boxes is shown. Nonlinearity persists in the 109–112 range even after removing all fixed points, reverse fixed points, and short-period rings. Our function gives an average nonlinearity of 111.51051. There are 466 finest S-boxes with nonlinearity 112 out of the 1000 S-boxes produced by the function, and there are 778 S-boxes with nonlinearity greater than 111 overall. These findings are astounding and far superior to the current techniques without any weaknesses [31] and the nonlinearity of 1950 S-boxes is higher than the mean score of [31]. The scores of the other current methods that are flawless are significantly lower. The schemes [2, 4, 15, 29, 36] contain short cycles, fixed, and reverse fixed points, and its S-boxes of nonlinearity fall between 105 and 112. Figure 7 show that our proposed S-box has nonlinearity values in between 111.5 to 112.

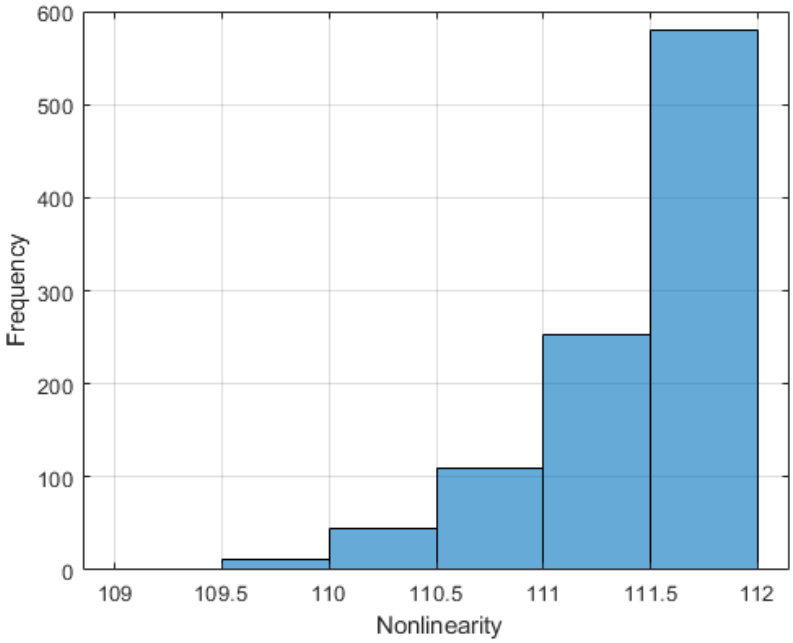

Figure 7. Nonlinearity distribution.

5.2. Strict Avalvanche Criteria (SAC)

A property of substitution boxes (S-boxes) called strict avalanche criteria (SAC) is used to assess the cryptographic strength of S-boxes in symmetric key algorithms. Using SAC quantifies the output such that a small change in the input results in significant changes in the production, such as how much an S-box's output bits change when a single bit in its input is changed. When each of the S-box's input bits is reversed, each output bit should change with a probability of 0.5. By doing this, it is ensured that the S-box does not favor any certain output value. At least $(k/2)$ output bits should ideally change if k input bits are altered. This feature makes sure that a slight change in the input results in a significant change in the output. If the function $f(x) \oplus f(x \oplus a)$ is balanced for each vector of hamming weight 1, then the boolean function f satisfies the SAC. Figure 8 shows the average values of the dependence matrices. We calculated for 1000 S-boxes to assess the strict avalanche requirements of S-boxes. The sample S-box average score was 0.5050, which is good when compared to sample S-boxes [1, 6, 31]. Our average score of SAC is 0.5050 for 1000 dynamic S-boxes. Our scores are better than compared to some other papers [4, 27, 35]. Figure 8 shows the score of strict avalanche criteria (SAC) of our proposed S-box.

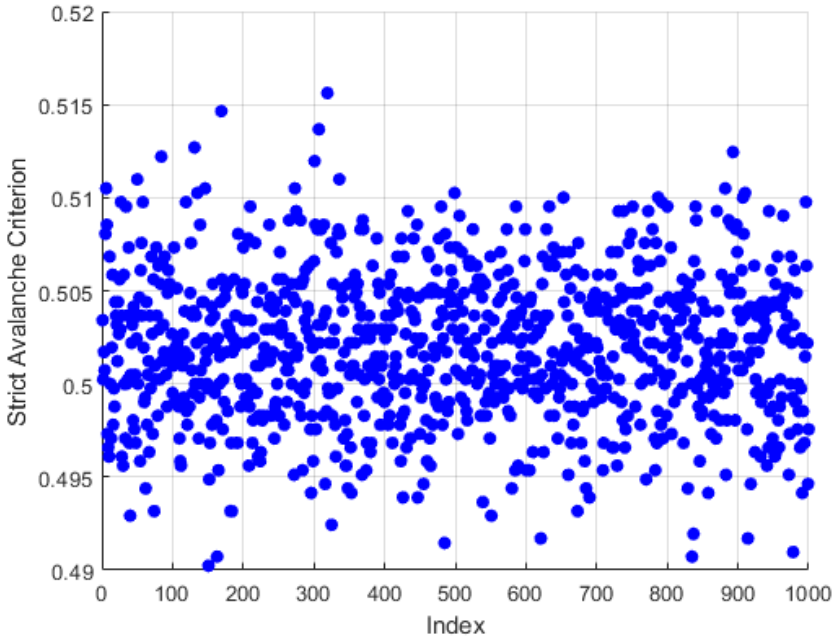

Figure 8. SAC score distribution.

5.3. Bit of Independence Criteria(BIC)

Let f_a and f_b be the S-box's two-bit outputs. When

$$f_a \oplus f_b \quad (a \neq b,\ 1 \leq a, b \leq n)$$

an S-box that meets the rigorous avalanche conditions and is extremely nonlinear is said to satisfy the **bit independence criterion (BIC)**. The term *bit independence criterion* describes a collection of characteristics that define the statistical independence of an S-box's input and output bits. The criteria that outline the conditions that must be fulfilled by the S-box in order to ensure that its output bits are statistically independent of its input bits. For an S-box's bit outputs f_i and f_j $(1 < i, j \leq n,\ i \neq j)$, if $f_i \oplus f_j$ is extremely nonlinear and meets the rigorous avalanche criterion, the S-box satisfies the BIC. The ideal BIC-SAC value is 0.5, and the assault resistance is increased by greater BIC nonlinearity values. The BIC The sample S-boxes in Figure 9 have a nonlinearity of 112, which is equivalent to the score of AES and sample S-boxes in [2, 4, 31, 35, 36]. The average BIC nonlinearity scores in [31] are 109.67 and 111.34 [35]. Thus, our average score is 111.49. The BIC nonlinearity scores of our suggested strategy are higher than [31, 35]. The BIC SAC scores of our sample S-boxes and 1000 randomly generated S-boxes are displayed in Figure 10. For the sample S-boxes, the mean scores are 0.5025.

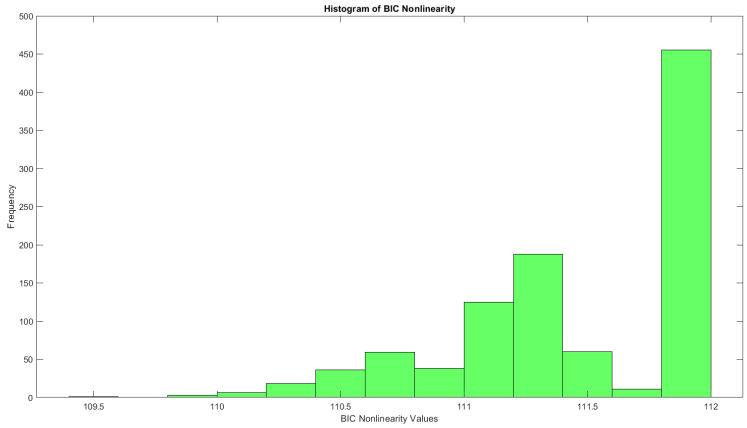

Figure 9. Bit independence nonlinearity scores.

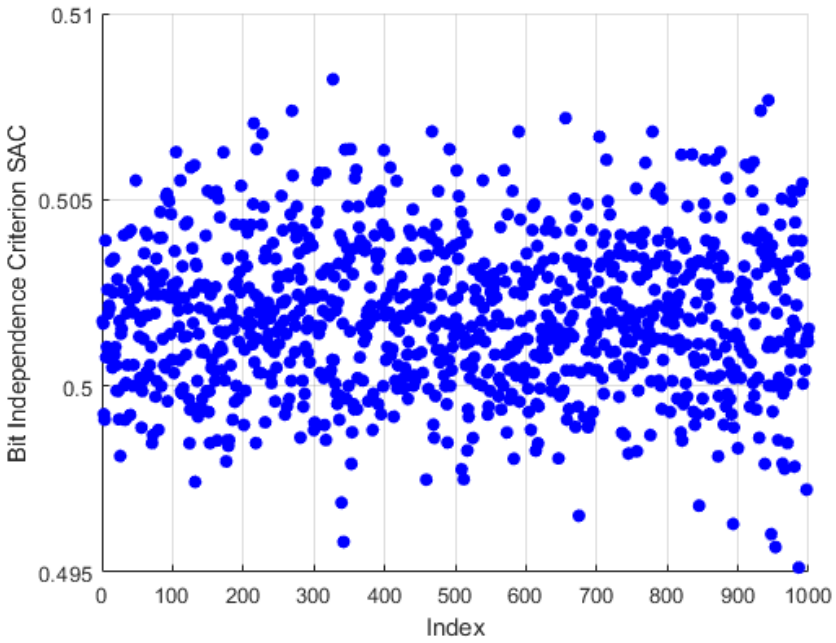

Figure 10. Bit independence SAC scores.

5.4. Linear Approximation probability (LAP)

The probability of linear approximation is the likelihood that, given a given number of input-output pairs, the inputs of an S-box will approach its outputs linearly. Due to its increased vulnerability to linear attacks, a weaker S-box would have a greater linear approximation probability. Conversely, a smaller linear approximation probability suggests a stronger S-box. The S-box shows improved resistance to linear attacks as a result. The linear approximation probability can be determined using the formula below. Linear Approximation probability (LAP) of dynamically generated 1000 S-boxes

is displayed in Figure 11.

$$\text{LPS} = \max_{\alpha,\beta\neq 0} |\{u \in \text{GF}(2^m) \mid \alpha \cdot S(u) = \beta \cdot S(v)\}| - \frac{2^{m-1}}{2^m}$$

assuming that the input and output masks are represented by u and v, respectively.

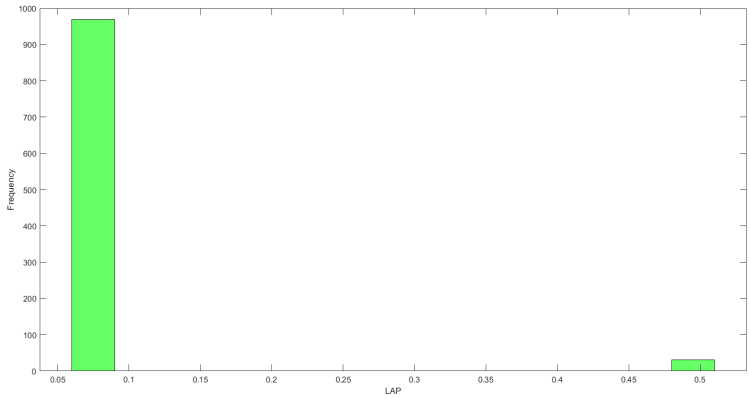

Figure 11. Linear approximation probability (LAP).

5.5. Differential approximation probability (DAP)

When considering a given number of rounds, the probability that a given input difference will result in a given output difference is estimated by the differential approximation probability for an S-box. The chance of a specific differential characteristic happening within the S-box is quantified. To calculate the differential approximation probability, a comprehensive search over all possible input and output differences across a predetermined number of rounds is often carried out. The occurrences of each difference are counted, and the total is then calculated. The ratio of input/output pairs examined to the number of occurrences of the desired difference is used to compute the likelihood. The S-box's resistance to differential cryptanalysis increases with decreasing differential approximation probability. A decreased likelihood suggests the lack of strong differentials displayed by the S-box makes, it is more difficult for an attacker to exploit the cipher's differential features. The attacker uses unique qualities to their advantage and cracks the cipher. Differential Approximation probability(DAP) of 1000 S-boxes can be observed in Figure 12.

$$\text{DP}(\Delta u, \Delta v) = \frac{|\{u \in \text{GF}(2^m) \mid S(u) \oplus S(u \oplus \Delta u) = \Delta v\}|}{2^m}$$

The differential between the input and output are denoted by Δu and Δv, respectively.

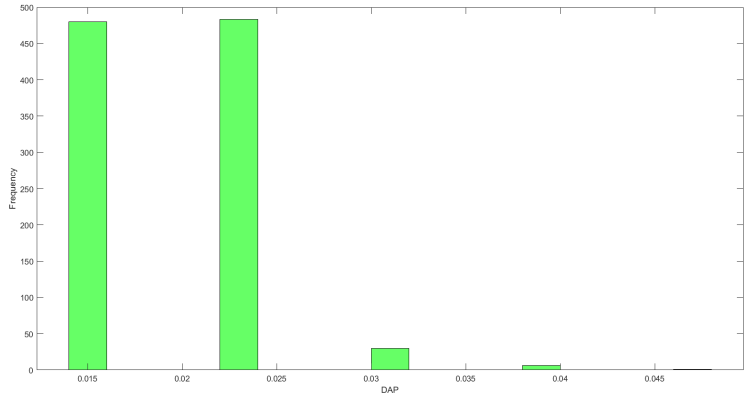

Figure 12. Differential approximation probability (DAP).

6. Conclusions

EQM was used to suggest a keyed strong S-box construction technique. In certain S-boxes, exploitable vulnerabilities related to fixed point, reverse fixed point, and short cycles were first revealed. In order to create a keyed S-box without any weaknesses, an EQM with ergodicity was suggested; this significantly increased the average cycle length and randomness when compared to the quadratic map. After EQM was used to develop a keyed strong S-box construction algorithm, all of the short periodic rings were combined into a maximized ring, and the fixed point or reverse fixed point was removed using a swapping technique. The efficacy and viability of the suggested S-box construction algorithm were confirmed by experimental data.

Author contributions

Mohammad Mazyad Hazzazi: Methodology, Software, Data curation. Farooq E Azam: Conceptualization, Validation, Writing-original draft. Rashad Ali: Conceptualization, Methodology, Software, Writing, reviewing & editing. Muhammad Kamran Jamil: Data curation, Formal analysis, Investigation, Supervision. Sameer Nooh: Methodology, Visualization, Writing- review & editing. Fahad Alblehai: Conceptualization, Formal Analysis, Software.
All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

All authors declare no conflicts of interest in this paper

Acknowledgments

The authors extend their gratitude to the deanship of scientific research of King Khalid University, for funding this work through a research project under grant R.G.P.2/34/45.

References

1. F. Artuger, Strong s-box construction approach based on Josephus's problem, *Soft Comput.*, (2024), 1–13. https://doi.org/10.1007/s00500-024-09751-7
2. R. Ali, M. K. Jamil, A. S. Alali, J. Ali, G. Afzal, A robust S box design using cyclic groups and image encryption, *IEEE Access*, **11** (2023), 135880–135890. https://doi.org/10.1109/ACCESS.2023.3337443
3. C. Luo, Y. Wang, Y. Fu, P. Zhou, M. Wang, Constructing dynamic S-boxes based on chaos and irreducible polynomials for image encryption, *Nonlinear Dynam.*, **11** (2024), 1–19. https://doi.org/10.1007/s11071-024-09353-w
4. F. Artuger, F. Ozkaynak, A new chaotic system and its practical applications in substitution box and random number generator, *Multimed. Tools Appl.*, **11** (2024), 1–15. https://doi.org/10.1007/s11042-024-19053-7
5. F. Artuger, A method for designing substitution boxes based on chaos with high nonlinearity, *Wireless Pers. Commun.*, **11** (2024), 1–16. https://doi.org/10.1007/s11277-024-11104-4
6. A. Waheed, F. Subhan, M. M. Su'ud, M. M. Alam, Molding robust S-box design based on linear fractional transformation and multilayer Perceptron: Applications to multimedia security, *Egypt. Inform. J.*, **26** (2024), 100480. https://doi.org/10.1016/j.eij.2024.100480
7. Y. Zhang, H. Bao, Z. Hua, H. Huang, Two-dimensional exponential chaotic system with hardware implementation, *IEEE T. Ind. Electron.*, **70** (2022), 9346–9356. 10.1109/TIE.2022.3206747
8. Z. Hua, Y. Chen, H. Bao, Y. Zhou, Two-dimensional parametric polynomial chaotic system, *IEEE T. Syst. Man, Cy-S.*, **52** (2021), 4402–4414. 10.1109/TSMC.2021.3096967
9. H. Ning, G. Zhao, Z. Li, S. Gao, Y. Ma, Y. Dong, A novel method for constructing dynamic S-boxes based on a high-performance spatiotemporal chaotic system, *Nonlinear Dynam.*, **112** (2024), 1487–1509. https://doi.org/10.1007/s11071-023-09125-y
10. K. Kazlauskas, R. Smaliukas, G. Vaicekauskas, A novel method to design S-boxes based on key-dependent permutation schemes and their quality analysis, *Int. J. Adv. Comput. Sc.*, **7** (2016), 93–99. https://epublications.vu.lt/object/elaba:16939446
11. P. Agarwal, A. Singh, A. Kilicman, Advanced encryption standard based on key-dependent S-Box cube, *Adv. Mech. Eng.*, **10** (2018), https://doi.org/10.1049/iet-ifs.2018.5043
12. A. Seghier, J. Li, D. Z. Sun, Development of key-dependent dynamic S-boxes with dynamic irreducible polynomial and affine constant, *IET Inform. Secur.*, **13** (2019), 552–558. https://doi.org/10.1177/1687814018781638
13. S. Ibrahim, A. M. Abbas, Efficient key-dependent dynamic S-boxes based on permutated elliptic curves, *Inform. Sciences*, **558** (2021), 246–264. https://doi.org/10.1016/j.ins.2021.01.014
14. I. Hussain, A. Anees, T. A. Al-Maadeed, M. T. Mustafa, Construction of s-box based on chaotic map and algebraic structures, *Symmetry*, **11** (2019), 351. https://doi.org/10.3390/sym11030351

15. Y. Wang, Z. Zhang, L. Y. Zhang, J. Feng, J. Gao, P. Lei, A genetic algorithm for constructing bijective substitution boxes with high nonlinearity, *Inform. Sciences*, **523** (2020), 152–166. https://doi.org/10.1016/j.ins.2020.03.025
16. D. Zhu, X. Tong, Z. Wang, M. Zhang, A novel lightweight block encryption algorithm based on the combined chaotic system, *J. Inf. Secur. Appl.*, **69** (2022), 103289. https://doi.org/10.1016/j.jisa.2022.103289
17. Z. Hua, J. Li, Y. Chen, Sh Yi, Design and application of an S-box using complete Latin square, *Nonlinear Dynam.*, **104** (2021), 807–825. https://doi.org/10.1007/s11071-021-06308-3
18. S. BiBi, M. Abbas, M. Y. Misro, G. Hu, A novel approach of hybrid trigonometric Bézier curve to the modeling of symmetric revolutionary curves and symmetric rotation surfaces, *IEEE Access*, **7** (2019), 165779–165792. https://doi.org/ 10.1109/ACCESS.2019.2953496
19. Y. Si, H. Liu, Y. Chen, Constructing keyed strong S-Box using an enhanced quadratic map, *Int. J. Bifurcat. Chaos*, **31** (2021), 2150146. https://doi.org/10.1142/S0218127421501467
20. A. Belazi, A. A. Abd El-Latif, A simple yet efficient S-box method based on a chaotic sine map, *Optik*, **130** (2017), 1438–1444. https://doi.org/10.1016/j.ijleo.2016.11.152
21. S. S. Jamal, M. U. Khan, T. Shah, A watermarking technique with chaotic fractional S-box transformation, *Wireless Pers. Commun.*, **90** (2016), 2033–2049. https://doi.org/10.1007/s11277-016-3436-0
22. F. Firdousi, S. I. Batool, M. Amin, A novel construction scheme for nonlinear components based on quantum map, *Int. J. Theor. Phys.*, **58** (2019), 3871–3898. https://doi.org/10.1007/s10773-019-04254-w
23. Alamsyah, A. Bejo, T. B. Adji, The replacement of irreducible polynomial and affine mapping for the construction of a strong S-box, *Nonlinear Dynam.*, **93** (2018), 2105–2118. https://doi.org/10.1007/s11071-018-4310-2
24. M. Zhao, H. Liu, Y. Niu, Batch generating keyed strong S-Boxes with high nonlinearity using the 2D hyper chaotic map, *Integration*, **92** (2023), 91–98. https://doi.org/10.1016/j.vlsi.2023.05.006
25. W. Alexan, M. Elkandoz, M. Mashaly, E. Azab, A. Aboshousha, Color image encryption through chaos and kaa map, *Symmetry*, **11** (2023), 11541–11554. https://doi.org/10.1109/ACCESS.2023.3242311
26. M. Lavanya, K. J. A. Sundar, S. Saravanan, Simplified Image Encryption Algorithm (SIEA) to enhance image security in cloud storage, *Multimed. Tools Appl.*, **83** (2024), 1–33. https://doi.org/10.1007/s11042-023-17969-0
27. G. Yi, Z. Cao, An algorithm of image encryption based on AES , Rossler hyperchaotic modeling, *Mobile Netw. Appl.*, (2023), 1–9. https://doi.org/10.1007/s11036-023-02216-5
28. Z. Lin, H. Liu, Constructing a non-degeneracy 3D hyperchaotic map and application in image encryption, *Multimed. Tools Appl.*, **83** (2024), 1–20. https://doi.org/10.1007/s11042-024-18741-8
29. J. Ali, M. K. Jamil, A. S. Alali, R. Ali, A medical image encryption scheme based on Mobius transformation and Galois field, *Heliyon*, **10** (2024), https://doi.org/10.1016/j.heliyon.2023.e23652
30. A. Kadeer, Y. Tuersun, H. Liu, Constructing keyed strong S-Box with optimized nonlinearity using nondegenerate 2D hyper chaotic map, *Phys. Scripta*, **99** (2024), 125281. 10.1088/1402-4896/ad91ed

31. R. Liu, H. Liu, M. Zhao, Cryptanalysis and construction of keyed strong S-Box based on random affine transformation matrix and 2D hyper chaotic map, *Expert Syst. Appl.*, **252** (2024), 124238. https://doi.org/10.1016/j.eswa.2024.124238
32. Y. Ma, Y. Tian, L. Zhang, P. Zuo, Two-dimensional hyperchaotic effect coupled mapping lattice and its application in dynamic S-box generation, *Nonlinear Dynam.*, **112** (2024), 17445–17476. https://doi.org/10.1007/s11071-024-09907-y
33. M. M. Hazzazi, Gulraiz, R. Ali, M. K. Jamil, S. A. Nooh, F. Alblehai, Cryptanalysis of hyperchaotic S-box generation and image encryption, *AIMS Math.*, **9** (2024), 36116–36139. 10.3934/math.20241714
34. N. Abughazalah, L. Said, M. Khan, Construction of optimum multivalued cryptographic Boolean function using artificial bee colony optimization and multi-criterion decision-making, *Soft Comput.*, **28** (2024), 5213–5223. https://doi.org/10.1007/s00500-023-09267-6
35. C. Luo, Y. Wang, Y. Fu, P. Zhou, M. Wang, Constructing dynamic S-boxes based on chaos and irreducible polynomials for image encryption, *Nonlinear Dynam.*, **112** (2024), 6695–6713. https://doi.org/10.1007/s11071-024-09353-w
36. A. S. Alali, R. Ali, M. K. Jamil, J. Ali, Gulraiz, Dynamic S-Box construction using Mordell Elliptic Curves over Galois Field and its applications in image encryption, *Mathematics*, **12** (2024), 587. https://doi.org/10.3390/math12040587
37. J. Ali, M. K. Jamil, R. Ali, Gulraiz, Extended fractional transformation based S-box and applications in medical image encryption, *Multimed. Tools Appl.*, (2025), 1–17. https://doi.org/10.1007/s11042-024-20575-3
38. M. Shadab, M. S. Jawed, M. Sajid, Substitution box construction using transfer-function assisted metaheuristic and booster algorithm: A hybrid approach, *Secur. Privacy*, **8** (2024), e462. https://doi.org/10.1002/spy2.462
39. M. A. Tootkaboni, M. B. Savadkouhi, S-Boxes design based on the Lu-Chen system and their application in image encryption, *Soft Comput.*, **28** (2024), 12119–12140. https://doi.org/10.1007/s00500-024-09912-8
40. F. Artuğer, F. Özkaynak, A method for generation of substitution box based on random selection, *Egypt. Inform. J.*, **23** (2022), 127–135. https://doi.org/10.1016/j.eij.2021.08.002
41. G. Murtaza, N. A. Azam, U. Hayat, Designing an efficient and highly dynamic substitution-box generator for block ciphers based on finite elliptic curves, *Secur. Commun. Netw.*, **2021** (2021), 3367521. https://doi.org/10.1155/2021/3367521
42. M. M. Hazzazi, M. Sajjad, Z. Bassfar, T. Shah, A. Albakri, Nonlinear components of a block cipher over eisenstein integers, *CMC-Comput. Mater. Con.*, **77** (2023), 3659–3675. https://doi.org/10.32604/cmc.2023.039013
43. B. Alabdullah, A. Banga, N. Iqbal, A. Ikram, H. Diab, Advancing cryptographic security with a new delannoy-derived chaotic S-box, *IEEE Access*, **12** (2024). https://doi.org/10.1109/ACCESS.2024.3410668
44. I. Martinez-Diaz, R. Ali, M. K. Jamil, On the search for supersingular elliptic curves and their applications, *Mathematics*, **13** (2025), 188. https://doi.org/10.3390/math13020188
45. R. S. Ali, O. Z. Akif, S. A. Jassim, A. K. Farhan, E. M. El-Kenawy, A. Ibrahim, et al., Enhancement of the CAST Block Algorithm based on novel S-Box for image encryption, *Sensors*, **22** (2022), 8527. https://doi.org/10.3390/s22218527

46. R. Ali, J. Ali, P. Ping, M. K. Jamil, A novel S-box generator using Frobenius automorphism and its applications in image encryption, *Nonlinear Dynam.*, **112** (2024), 19463–19486. https://doi.org/10.1007/s11071-024-10003-4

Appendix

Table A1. Sample 1.

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

Table A2. Sample 2.

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

Table A3. Sample 3.

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

Table A4. Sample 4.

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

Table A5. Sample 5.

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

Table A6. Comparison of cryptographic properties of sample S-boxes.

S-boxes	Nonlinearity	SAC	BIC Nonlinearity	BIC SAC	LAP	DAP
A1	112	0.5034	112	0.4986	0.0625	0.0156
A2	112	0.5042	112	0.5008	0.0625	0.0156
A3	112	0.4980	112	0.4995	0.0625	0.0156
A4	112	0.4978	112	0.4990	0.0625	0.0156
A5	112	0.5037	112	0.4983	0.0625	0.0156
AES	112	0.5040	112	0.5046	0.0625	0.0156