Nonlinear Dyn (2025) 113:3869–3890
https://doi.org/10.1007/s11071-024-10414-3

RESEARCH

AES S-box modification uses affine matrices exploration for increased S-box strength

Alamsyah · Abas Setiawan · Anggyi Trisnawan Putra · Kholiq Budiman · Much Aziz Muslim · Shahrul Nizam Salahudin · Budi Prasetiyo

Received: 7 August 2024 / Accepted: 24 September 2024 / Published online: 8 October 2024

Abstract The internet and technological advancements are rapidly growing in this era, potentially giving rise to new challenges, particularly in data security. One algorithm developed to secure data is the Advanced Encryption Standard (AES). The strength of AES lies in its S-box, which transforms input bits into output bits in a randomized manner. Therefore, the S-box must be highly resilient against various attacks, especially linear and differential attacks. This paper modifies the AES S-box algorithm. The modification begins with the irreducible polynomial in the AES S-box, $x^8 + x^4 + x^3 + x + 1$. It involves forming an inverse multiplicative matrix, exploring affine matrices, and using an 8-bit constant in the affine transformation. This process produces candidate S-boxes. The candidate S-boxes are subsequently evaluated to ensure they meet the criteria of balance and bijectivity. Once these criteria are satisfied, the candidate S-boxes are formalized into valid S-boxes. The resulting S-boxes are evaluated against several strength criteria, including nonlinearity, strict avalanche criterion, bit independence-nonlinearity criterion, bit independence-strict avalanche criterion, linear approximation probability, and differential approximation probability. The results indicate that the proposed S-box$_{44}$ has a nonlinearity value of 112, strict avalanche criterion of 0.50073, bit independence-nonlinearity criterion of 112, bit independence-strict avalanche criterion of 0.50237, linear approximation probability of 0.0625, and differential approximation probability of 0.015625. These results outperform the strength of the AES S-box and surpass previous studies.

Keywords AES · S-box · Irreducible polynomial · Affine matrices

Alamsyah (✉) · A. Setiawan · A. T. Putra · K. Budiman · M. A. Muslim · B. Prasetiyo
Department of Computer Science, Universitas Negeri Semarang, Semarang, Indonesia
e-mail: alamsyah@mail.unnes.ac.id

M. A. Muslim · S. N. Salahudin
Faculty of Technology Management and Business, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

1 Introduction

The rapid development of the internet and technology and the increasing number of internet users worldwide impact on the increase in cybercrime. Cybercrime can occur in various forms, including theft, piracy, and data misuse. Therefore, data security is essential for internet users. For this reason, it is necessary to implement a good algorithm to protect sensitive information.

One such algorithm is the Advanced Encryption Standard (AES) [1]. AES has been used since 2001 to replace the DES (Data Encryption Standard)

algorithm, which is vulnerable to linear attacks. AES is regarded as secure today. The primary strength of AES lies in the substitution box or S-box. The S-box plays a role in transforming input bits into output bits randomly during the encryption process. Encryption functions transform original data into coded data or encrypted data. Encrypted data must remain intact and secure.

AES is a block cipher that encrypts data in fixed-sized blocks using substitution techniques. The substitution technique will work well and safely if the built S-box construction produces a strong S-box. Currently, the S-box used in AES has proven to be very resilient to various attacks, especially linear and differential attacks.

The construction of the AES S-box [1] begins by selecting the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. This irreducible polynomial is used to generate the multiplicative inverse matrix, followed by applying an affine transformation. The affine transformation involves multiplying the affine matrix by the multiplicative inverse matrix, subsequently adding the result to an 8-bit constant.

In addition to the AES S-box, several other S-box construction methods have been explored, including various approaches. In his research, Artuğer [2] developed an S-box construction based on the Josephus problem. The algorithm applied is to iteratively update the S-box structure to maximize nonlinearity and eliminate fixed points, thereby producing a highly secure S-box for block cipher applications. Razaq et al. [3] introduced an innovative S-box construction method that utilizes a new algebraic subgroup, S*16, from the symmetric group S16 with an order of 1536. This method generates 2,359,296 S-boxes. Five S-boxes were selected as samples with a nonlinearity value of 112 and varying Strict Avalanche Criterion values, namely 0.4985, 0.5034, 0.5014, 0.5021, and 0.5024. Ma et al. [4] presented a method to generate dynamic S-boxes using a modified Two-Dimensional Hyperchaotic Effect Coupled Map Lattice (2D-HECML) system. This system combines a novel M-sequence and a spherical cavity hyperchaotic mapping. This system is also utilized to enhance its chaotic behavior and cryptographic properties. The resulting S-box is highly secure against various attacks. Ning et al. [5] introduced a method that generates dynamic S-boxes. This method uses a hybrid spatiotemporal chaotic system. The superior chaotic characteristics of the system make it highly suitable for cryptographic applications. The resulting S-box demonstrates robust security against various attacks. Zhao et al. [6] proposed a new scheme for constructing efficient S-boxes. The methods employed include chaotic maps and new permutation sequences. The analysis results show that the resulting S-box exhibits high nonlinearity and low differential uniformity and satisfies the SAC and BIC criteria, thereby enhancing the algorithm's ability to resist differential and linear cryptographic attacks.

In another study, Ali et al. [7] proposed a method that combines Frobenius automorphism and Möbius transformation over $GF(2^8)$. This method generates S-boxes with high nonlinearity values. The use of non-degenerate 3D hyperchaotic maps in S-box design was also investigated by Lin and Liu [8]. Similarly, a strong S-box construction method was proposed based on non-degenerate enhanced 3D quadratic maps (3D-IQM) [9]. This S-box satisfies cryptographic criteria and eliminates fixed points and short cycles. The introduction of a new 2D discrete hyperchaotic map for creating high-performance S-boxes is also highlighted [10]. This map generates eight dynamic S-boxes that improve encryption strength and resistance to security attacks.

Another approach combines 2D hyperchaotic maps with algebraic operations [11]. This approach is used to construct dynamic S-boxes. The results of this approach are improved encryption strength and increased resistance to various attacks. Pseudo-random S-boxes [12] were proposed with a focus on improving the Vigenère cipher. The generated S-boxes are derived from chaotic maps. These S-boxes improve encryption strength by enhancing the avalanche effect and resisting differential attacks. In addition, [13] proposes an S-box construction method using a two-dimensional discrete hyperchaotic map with enhanced chaotic properties. This method generates twelve S-boxes. These S-boxes are applied to the grouped data elements using bitwise and modular XOR operations, followed by chaotic randomization to generate the final cipher. A robust S-box design is proposed using the true physical randomness of phase noise combined with the SHA-256 hash algorithm [14]. This method significantly improves the strength of the S-boxes by improving their performance in nonlinearity, SAC, and BIC tests. The use of twisted Boolean functions [15] for the new S-box design is

also introduced. These functions are maximally nonlinear but inherently imbalanced, requiring imbalance mitigation to ensure bijectivity.

Another method introduced with a novel approach combines of chaos and nonlinear confusion components. In this method, S-boxes are developed by constructing them using double affine transformations, resulting in 40,320 S-boxes. Subsequently, utilizing the nonlinear Lorenz dynamic system, combined with the Chirikov discrete iterative map, the three best S-boxes with a nonlinearity value of 112 are selected [16]. This technique incorporates a dynamic optimization phase that increases the nonlinearity of the initial S-box configuration. The result is the generation of multiple robust S-boxes by slightly varying the input parameters. A different strategy was employed by [17] using Griewank function and sin mapping. Other researchers [18–21] have generated robust S-boxes. S-box construction was achieved using affine and quadratic polynomial transformation methods and a unique permutation approach [18], a new chaotic map [19], a hyperchaotic system two-dimension Chebyshev–Sine coupling map (2D-CSCM) [20], modified Piece Wise Linear Chaotic Maps [21]. The effectiveness of these approaches has been validated through simulations, demonstrating resistance to various attacks.

In other studies, various innovative approaches have been proposed to improve the design and implementation of S-boxes. One notable method uses particle swarm optimization (PSO) [22–24] combined with Rossler map [22], chaotic maps [23], and highly nonlinear [24]. The generated S-boxes have been effectively utilized in various applications. The Dynamic and Secure S-box (DS2B) [25], designed for IoT platforms, is a research result that has been proposed. This S-box has a high nonlinearity value and a low differential uniformity value. Additionally, this S-box was implemented on an FPGA showed high throughput and was proven highly to secure speech encryption during silent periods. A combination of cuckoo search algorithms paired with discrete space chaos map [26]. A chaos map-based block cipher scheme [27], chaotic map [28, 29], Latin square [30], quantum S-box [31], Linear Recurrences [32], 4D hyperchaotic system [33], and a novel chaotic-based S-box [34] have also produced S-box with high nonlinearity values. Furthermore, new S-box designs utilizing array indexing and chaotic S-boxes [35], a chaotic map based on the multiplication of integer numbers [36], elliptic curves and coupled map lattices [37], the Mobius Group and Finite Fields [38], and S-P boxes [39] were introduced. This approach successfully improved security with a high nonlinearity value. The use of chaotic permutation [40] was implemented to generate an optimal S-box. Further research introduced a 3D chaos map-based encryption algorithm for multiple images, with a critical module dedicated to S-box generation [41]. This approach, when combined with AES, significantly improved the encryption strength. Another study introduced a new S-box design that uses chaos maps and an improved artificial bee colony algorithm [42]. This method enhances cryptographic properties such as nonlinearity and SAC, outperforming recent S-box designs and offering a practical approach for creating strong S-boxes. However, vulnerabilities in chaos-based S-box encryption have been identified.

One study highlighted that the system can be compromised through chosen-plaintext and chosen-ciphertext attacks, enabling key retrieval with minimal effort [43]. This highlights the need for stronger cryptographic methods. In this paper, the recovery of a substitution box from a chosen-plaintext attack is presented. A secure image encryption algorithm combines chaos maps with 16 newly designed S-boxes derived from the projective general linear group and Galois field polynomials [44]. This S-box, integrated with the Lorenz chaos system, significantly improves the encryption strength, with its effectiveness confirmed through rigorous cryptographic analysis. In addition, a paper presents a three-layer optimization method for generating high-performance S-boxes using a new chaos map and artificial jellyfish optimization [45], pseudo-random coupling [46]. Furthermore, in [47], an S-box construction based on the Henon map has been designed. The result is key-dependent bijective S-boxes and pseudo-random sequences. These generated S-boxes are implemented in a color image encryption scheme that is resistant to various types of attacks, particularly differential attacks. A dynamic S-box has been constructed based on a piecewise map to achieve low automatic correlation [48]. The performance of the S-box is measured using nonlinearity, strict avalanche criterion, linear approximation probability, and differential approximation probability, with results close to the ideal criteria. Additionally, the generated S-box is

implemented in image encryption to withstand various attacks. Lastly, in [49], an S-box was introduced that was constructed using a key-dependent algorithm, which generates a set of Boolean functions $f1, f2, \ldots, fn$, such that the resulting S-box is bijective and nonlinear.

Overall, the AES S-box and the S-boxes produced from previous studies are robust against various attacks, especially linear attacks and differential attacks [2–49]. However, it is still possible that the strength of the resulting S-box can be improved to be better than that of previous studies.

The research aims to enhance the strength of the S-box by modifying the AES S-box. The approach used involves the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$, exploration of affine matrices, and 8-bit constants. The method presented in this paper builds upon the development of previous research [50]. The affine transformation process uses the best affine matrix selected from 128 candidate matrices. The resulting S-box is evaluated using established cryptographic strength testing standards, including nonlinearity (NL), strict avalanche criterion (SAC), bit-nonlinearity independence criterion (BIC-NL), bit-strict avalanche criterion independence criterion (BIC-SAC), linear approximation probability (LAP), and differential approximation probability (DAP).

The best S-box produced from previous studies has been implemented in the image encryption process [47, 48]. Some have even developed it to the level of a visual meaningful encryption and hiding algorithm for multiple images [51], as well as an image encryption algorithm designed for DNA storage that leverages the information processing mechanisms of molecular biology [52]. However, this study focuses on improving the S-box construction to produce encrypted data that is robust against various types of attacks, particularly linear and differential attacks.

The main contributions of the paper are detailed as follows:

(1) Modification of AES S-box Algorithm:

The paper introduces a modified S-box for the AES encryption algorithm. The modification begins with the irreducible polynomial in the AES S-box (specifically $x^8+x^4+x^3+x+1$). Although it uses the same irreducible polynomials as AES, it produces an S-box that is stronger than the AES S-box.

(2) Exploration of Affine Matrices:

The study explores a vast number (18,446,744,073,709,551,616) of possible affine matrices. After rigorous testing for balance and bijectiveness, only 128 affine matrices meet the necessary criteria for constructing an S-box.

(3) Formation of a New S-box ($S\text{-box}_{44}$):

Using the selected 128 affine matrices, the irreducible polynomial, and an 8-bit constant in the affine transformation, the study constructs a new S-box called $S\text{-box}_{44}$.

(4) Evaluation of Cryptographic Strength:

$S\text{-box}_{44}$ is thoroughly tested against several cryptographic strength metrics:

(a) Nonlinearity (NL): Achieved a value of 112.
(b) Strict Avalanche Criterion (SAC): Achieved a value of 0.50073.
(c) Bit Independence Criterion—Nonlinearity (BIC-NL): Achieved a value of 112.
(d) Bit Independence Criterion—Strict Avalanche Criterion (BIC-SAC): Achieved a value of 0.50237.
(e) Linear Approximation Probability (LAP): Achieved a value of 0.0625.
(f) Differential Approximation Probability (DAP): Achieved a value of 0.015625.

(5) Comparison with AES S-box:

The results demonstrate that $S\text{-box}_{44}$ is stronger than the original AES S-box, with the following improvements:

(a) SAC Improvement: 0.828790%
(b) BIC-SAC Improvement: 0.44390%
(c) Overall S-box Performance Improvement: 0.00268%

(6) Comparison with S-boxes from Previous Studies:

The proposed S-box, $S\text{-box}_{44}$, is also compared with the S-boxes from previous studies. $S\text{-box}_{44}$ outperforms those S-boxes with a strength value (SV) of 16.0031, indicating its superior cryptographic strength.

These contributions collectively demonstrate the enhanced security provided by the proposed $S\text{-box}_{44}$, making it a promising candidate for improving the robustness of encryption algorithms against various cryptographic attacks.

The rest of this paper is organized as follows. Section 2 provides an overview of the AES S-box.

This section details the construction of the AES S-box. Section 3 details the modifications made to the AES S-box. This section describes the stages of the AES S-box modification which include Affine Matrix Exploration, Candidate S-box Construction, S-box Candidate Testing, and Final AES S-box Modification. Section 4 discusses the robustness testing of the modified S-box. The tests performed in this section include NL, SAC, BIC-NL, BIC-SAC, DAP, and LAP. Section 5 introduces the proposed selected S-box. This section explains the selection process of the proposed S-box. Section 6 investigates the performance comparison of the proposed S-box, comparing its effectiveness with the original AES S-box and previous research S-boxes. Finally, Sect. 7 contains the conclusion summarizing the research findings and providing suggestions for future research.

2 AES S-box

The AES S-box construction uses the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. A multiplicative inverse matrix is formed using this polynomial, as shown in Table 1. This table is then used to construct the S-box by applying an affine transformation. The affine transformation involves the use of an affine matrix and an 8-bit constant. The affine matrix and 8-bit constant are represented by matrices K_{AES} and C_{AES}. The resulting AES S-box is displayed in Table 2.

$$K_{AES} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix}$$

$$C_{AES} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

3 AES S-box modification

The AES S-box modification involves several steps:

Step 1 Affine Matrix Exploration.
Step 2 Candidate S-box Construction.
Step 3 S-box Candidate Testing.
Step 4 Final AES S-box Modification.

Each step in the AES S-box modification process is detailed in the following sections.

Table 1 Multiplicative inverse matrix

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

Table 2 AES S-box

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

3.1 Affine matrix exploration

An affine matrix is an 8×8 matrix where each element in GF (2) is either 0 or 1. Exploring affine matrices begins by determining the number of rows and columns in the matrix. The affine matrix has eight rows and eight columns, resulting in a total of 64 elements in the matrix. Each element can be either 0 or 1. Exploration of affine matrices begins with a matrix.

$$\begin{bmatrix} 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix}$$ and ends with a matrix $$\begin{bmatrix} 11111111 \\ 11111111 \\ 11111111 \\ 11111111 \\ 11111111 \\ 11111111 \\ 11111111 \\ 11111111 \end{bmatrix}$$

Thus, 2^{64}, or 18,446,744,073,709,551,616 affine matrices can be explored. The sequences of affine matrices that can be explored are:

$$\begin{bmatrix} 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix}, \begin{bmatrix} 00000001 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix}, \begin{bmatrix} 00000010 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix},$$

$$\begin{bmatrix} 00000011 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix}, \begin{bmatrix} 00000100 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix}, \begin{bmatrix} 00000101 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix},$$

$$\begin{bmatrix} 00000110 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix}, \begin{bmatrix} 00000111 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix}, \begin{bmatrix} 00001000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{bmatrix},$$

$$\begin{bmatrix}00001001\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00001010\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00001011\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix},$$

$$\begin{bmatrix}00001100\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00001101\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00001110\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix},$$

$$\begin{bmatrix}00001111\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00010000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00010001\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}$$

$$\begin{bmatrix}00010010\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00010011\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}, \begin{bmatrix}00010100\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\\00000000\end{bmatrix}$$

$$\cdots \qquad \cdots \qquad \cdots$$

$$\begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11110111\end{bmatrix}, \begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111000\end{bmatrix}, \begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111001\end{bmatrix}$$

$$\begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111010\end{bmatrix}, \begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111011\end{bmatrix}, \begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111100\end{bmatrix},$$

$$\begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111101\end{bmatrix}, \begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111110\end{bmatrix}, \begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\end{bmatrix}$$

3.2 Candidate S-box construction

The key components for constructing an S-box candidate are the same as those used for constructing an S-box. This research utilizes one irreducible polynomial, 2^{64} affine matrices, and one 8-bit constant. Therefore, there are $1 \times 2^{64} \times 1$ candidate S-box constructions, resulting in 18,446,744,073,709,551,616 candidate S-boxes. These candidates include combinations of one irreducible polynomial, 2^{64} affine matrices, and one 8-bit constant.

Not all candidate S-boxes constructed with one irreducible polynomial, 2^{64} affine matrices, and one 8-bit constant qualify as valid S-boxes. For an S-box construction to be considered valid, it must satisfy two key conditions: balance and bijectivity. The selection of the affine matrix determines whether the resulting S-box satisfies the balance and bijectivity criteria. Fulfilling these criteria is crucial for producing a valid S-box. A detailed explanation of the balance and bijectivity criteria is provided in the following section.

3.3 S-box candidate testing

3.3.1 Balance

The balance criterion is satisfied if the number of output bits equal to 0 is the same as the number of output bits equal to 1. Mathematically, an S-box satisfies the balance criterion if it meets the condition in Eq. (1).

$$\#\{x|f(x) = 0\} = \#\{x|f(x) = 1\} \quad (1)$$

According to Eq. (1), the balance criterion is satisfied if the number of inputs x for which $f(x) = 0$ is equal to the number of inputs x for which $f(x) = 1$. For $GF(2^8)$, the value of each side in Eq. (1) is 128.

3.3.2 Bijective

The bijective criterion ensures that the input bits are randomized to become output bits and can be reverted to their original input state after decryption. In $GF(2^8)$, a candidate S-box satisfies the bijective criterion if each output value in the range [0, 255] is unique.

3.3.3 The S-box candidate becomes the S-box

For example, Table 2 presents a candidate S-box. It will be demonstrated that this candidate S-box can become a valid S-box after passing both the balance test and the bijective test.

3.3.3.1 Balance test The results of the balance test are presented in Table 3. According to Table 3, the calculation results from Eq. (1) are as follows:

$$\#\{x|f(0) = 0\} = 128 \quad \#\{x|f(0) = 1\} = 128$$
$$\#\{x|f(1) = 0\} = 128 \quad \#\{x|f(1) = 1\} = 128$$
$$\#\{x|f(2) = 0\} = 128 \quad \#\{x|f(2) = 1\} = 128$$
$$\#\{x|f(3) = 0\} = 128 \quad \#\{x|f(3) = 1\} = 128$$
$$\#\{x|f(4) = 0\} = 128 \quad \#\{x|f(4) = 1\} = 128$$
$$\#\{x|f(5) = 0\} = 128 \quad \#\{x|f(5) = 1\} = 128$$
$$\#\{x|f(6) = 0\} = 128 \quad \#\{x|f(6) = 1\} = 128$$
$$\#\{x|f(7) = 0\} = 128 \quad \#\{x|f(7) = 1\} = 128.$$

Thus, the balance criterion for the S-box candidate in Table 2 is satisfied.

3.3.3.2 Bijective test According to Table 2, the output values of the S-box candidate are all unique and fall within the range [0, 255]. Thus, the bijective criterion is satisfied for the S-box candidate in Table 2.

Based on the explanations in Sects. 3.3.3.1 and 3.3.3.2, the S-box candidate in Table 2 satisfies both the balance and bijective criteria. Therefore, it can be concluded that the candidate in Table 2 is a valid S-box.

3.4 Final AES S-box modification

The test results for the S-box candidate, as described in Sects. 3.3.3.1 and 3.3.3.2, indicate that the S-box candidate listed in Table 3 satisfies both the balance and bijective criteria. Therefore, this candidate S-box is designated as the AES S-box.

Out of 18,446,744,073,709,551,616 S-box candidates tested, 128 valid S-boxes were identified. These 128 S-boxes are constructed by combining one irreducible polynomial $x^8 + x^4 + x^3 + x + 1$, 128 affine matrices, and one 8-bit constant, C_{AES}. The 128 selected affine matrices used to construct the S-box are represented by matrices K_1, K_2, K_3, …, and K_{128}.

In this section, five S-boxes will be constructed using five different affine matrices. These matrices are selected based on four randomly chosen affine matrices and one specifically selected affine matrix. The five affine matrices used are K_4, K_{44}, K_{81}, K_{111}, and K_{128}.

$$K_1 = \begin{bmatrix} 00000001 \\ 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \\ 00000001 \end{bmatrix} \quad K_2 = \begin{bmatrix} 00000010 \\ 00000001 \\ 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \end{bmatrix}$$

Table 3 Results of the balance test

Decimal	Binary	$f(0)$	$f(1)$	$f(2)$	$f(3)$	$f(4)$	$f(5)$	$f(6)$	$f(7)$
99	01100011	0	1	1	0	0	0	1	1
124	01111100	0	1	1	1	1	1	0	0
119	01110111	0	1	1	1	0	1	1	1
123	01111011	0	1	1	1	1	0	1	1
242	11110010	1	1	1	1	0	0	1	0
107	01101011	0	1	1	0	1	0	1	1
111	01101111	0	1	1	0	1	1	1	1
197	11000101	1	1	0	0	0	1	0	1
48	00110000	0	0	1	1	0	0	0	0
1	00000001	0	0	0	0	0	0	0	1
103	01100111	0	1	1	0	0	1	1	1
43	00101011	0	0	1	0	1	0	1	1
254	11111110	1	1	1	1	1	1	1	0
215	11010111	1	1	0	1	0	1	1	1
171	10101011	1	0	1	0	1	0	1	1
118	01110110	0	1	1	1	1	1	1	0
202	11001010	1	1	0	0	1	0	1	0
130	10000010	1	0	0	0	0	0	1	0
201	11001001	1	1	0	0	1	0	0	1
125	01111101	0	1	1	1	1	1	0	1
250	11111010	1	1	1	1	1	1	1	0
89	01011001	0	1	0	1	1	0	0	1
71	01000111	0	1	0	0	0	1	1	1
240	11110000	1	1	1	1	0	0	0	0
…	…	…	…	…	…	…	…	…	…
153	10011001	1	0	0	1	1	0	0	1
45	00101101	0	0	1	0	1	1	0	1
15	00001111	0	0	0	0	1	1	1	1
176	10110000	1	0	1	1	0	0	0	0
84	01010100	0	1	0	1	0	1	0	0
187	10111011	1	0	1	1	1	1	1	1
22	00010110	0	0	0	1	0	1	1	0
$\#\{x \mid f(x) = 0\}$	128	128	128	128	128	128	128	128	
$\#\{x \mid f(x) = 1\}$	128	128	128	128	128	128	128	128	

$$K_3 = \begin{bmatrix} 00000100 \\ 00000010 \\ 00000001 \\ 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \end{bmatrix} \quad K_4 = \begin{bmatrix} 00000111 \\ 10000011 \\ 11000001 \\ 11100000 \\ 01110000 \\ 00111000 \\ 00011100 \\ 00001111 \end{bmatrix} \quad K_5 = \begin{bmatrix} 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \end{bmatrix} \quad K_6 = \begin{bmatrix} 00001011 \\ 10000101 \\ 11000010 \\ 01100001 \\ 10110000 \\ 01011000 \\ 00101100 \\ 00010110 \end{bmatrix}$$

$$K_7 = \begin{bmatrix} 00001101 \\ 10000110 \\ 01000011 \\ 10100001 \\ 11010000 \\ 01101000 \\ 00110100 \\ 00011010 \end{bmatrix} \quad K_8 = \begin{bmatrix} 00001110 \\ 00000111 \\ 10000011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix}$$

$$K_9 = \begin{bmatrix} 00010000 \\ 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 10000000 \\ 01000000 \\ 00100000 \end{bmatrix} \quad K_{10} = \begin{bmatrix} 00010011 \\ 10001001 \\ 11000100 \\ 01100010 \\ 00110001 \\ 10011000 \\ 01001100 \\ 00100110 \end{bmatrix}$$

$$K_{11} = \begin{bmatrix} 00010101 \\ 10001010 \\ 01000101 \\ 10100010 \\ 01010001 \\ 10101000 \\ 01010100 \\ 00101010 \end{bmatrix} \quad K_{12} = \begin{bmatrix} 00011001 \\ 10001100 \\ 01000110 \\ 00100011 \\ 10010001 \\ 11001000 \\ 01100100 \\ 00110010 \end{bmatrix}$$

$$K_{13} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix} \quad K_{14} = \begin{bmatrix} 00011010 \\ 00001101 \\ 10000110 \\ 01000011 \\ 10100001 \\ 11010000 \\ 01101000 \\ 00110100 \end{bmatrix}$$

$$K_{15} = \begin{bmatrix} 00011100 \\ 00001110 \\ 00000111 \\ 10000011 \\ 11000001 \\ 11100000 \\ 01110000 \\ 00111000 \end{bmatrix} \quad K_{16} = \begin{bmatrix} 00011111 \\ 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \end{bmatrix}$$

$$K_{17} = \begin{bmatrix} 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 10000000 \\ 01000000 \end{bmatrix} \quad K_{18} = \begin{bmatrix} 00100011 \\ 10010001 \\ 11001000 \\ 11100100 \\ 01110010 \\ 00111001 \\ 10011100 \\ 01001110 \end{bmatrix}$$

$$K_{19} = \begin{bmatrix} 00100101 \\ 10010010 \\ 01001001 \\ 10100100 \\ 01010010 \\ 00101001 \\ 10010100 \\ 01001010 \end{bmatrix} \quad K_{20} = \begin{bmatrix} 00100110 \\ 00010011 \\ 10001001 \\ 11000100 \\ 01100010 \\ 00110001 \\ 10011000 \\ 01001100 \end{bmatrix}$$

$$K_{21} = \begin{bmatrix} 00101001 \\ 10010100 \\ 01001010 \\ 00100101 \\ 10010010 \\ 01001001 \\ 10100100 \\ 01010010 \end{bmatrix} \quad K_{22} = \begin{bmatrix} 00101010 \\ 00010101 \\ 10001010 \\ 01000101 \\ 10100010 \\ 01010001 \\ 10101000 \\ 01010100 \end{bmatrix}$$

$$\cdots \qquad \cdots$$

$$K_{127} = \begin{bmatrix} 11111101 \\ 11111110 \\ 01111111 \\ 10111111 \\ 11011111 \\ 11101111 \\ 11110111 \\ 11111011 \end{bmatrix} \quad K_{128} = \begin{bmatrix} 11111110 \\ 01111111 \\ 10111111 \\ 11011111 \\ 11101111 \\ 11110111 \\ 11111011 \\ 11111101 \end{bmatrix}$$

The construction of these S-boxes involves modifying the AES S-box by using the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$, affine matrices K_4, K_{44}, K_{81}, K_{111}, and K_{128}, and an 8-bit constant C_{AES}.

This modification results in S-boxes: S-box$_4$, S-box$_{44}$, S-box$_{81}$, S-box$_{111}$, and S-box$_{128}$. These S-boxes are presented in Tables 4, 5, 6, 7, and 8.

Table 4 S-box$_4$

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

Table 5 S-box$_{44}$

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

$$K_4 = \begin{bmatrix} 00000111 \\ 10000011 \\ 11000001 \\ 11100000 \\ 01110000 \\ 00111000 \\ 00011100 \\ 00001111 \end{bmatrix}$$

$$K_{44} = \begin{bmatrix} 01010111 \\ 10101011 \\ 11010101 \\ 11101010 \\ 01110101 \\ 10111010 \\ 01011101 \\ 10101110 \end{bmatrix}$$

Table 6 S-box$_{81}$

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

Table 7 S-box$_{111}$

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

$$K_{81} = \begin{bmatrix} 10100001 \\ 11010000 \\ 01101000 \\ 00110100 \\ 00011010 \\ 00001101 \\ 10000110 \\ 01000011 \end{bmatrix} \qquad K_{111} = \begin{bmatrix} 11011100 \\ 01101110 \\ 00110111 \\ 10011011 \\ 11001101 \\ 11100110 \\ 01110011 \\ 10111001 \end{bmatrix}$$

Table 8 S-box$_{128}$

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

$$K_{128} = \begin{bmatrix} 11111110 \\ 01111111 \\ 10111111 \\ 11011111 \\ 11101111 \\ 11110111 \\ 11111011 \\ 11111101 \end{bmatrix}$$

A detailed explanation of the S-box$_{44}$ construction process, along with example calculations, is provided in the following section.

The S-box$_{44}$ construction process consists of several steps, namely:

Step 1 Determine the irreducible polynomial. The irreducible polynomial used in the S-box$_{44}$ construction is $x^8 + x^4 + x^3 + x + 1$.

Step 2 Determine the multiplicative inverse matrix. The multiplicative inverse matrix is obtained by calculating the inverse of all values in $GF\ (2^8)$ modulated by the irreducible polynomial $x^8 + x^{4-} + x^3 + x + 1$. The complete results of the multiplicative inverse matrix are shown in Table 1.

Step 3 Perform the affine transformation process. The affine transformation is described in Eq. (2).

$$B(X) = \left(K_{44}.X^{-1} + C_{\mathrm{AES}}\right)\mathrm{mod}2 \tag{2}$$

where:

$B(x)$ is a 1×8 matrix representing the S-box values as an 8-bit output.

K_{44} is an 8×8 matrix representing the optimal affine matrix.

X^{-1} is a 1×8 matrix representing the values of the multiplicative inverse.

C_{AES} is a 1×8 matrix representing the 8-bit constant values.

As an illustration, if the input S-box is in row 0 and column 15, as shown in Fig. 1, the 8-bit representation of the input S-box is X = 00001111. The value of X^{-1} is obtained from the multiplicative inverse matrix (Table 1) by taking the element from row 0 and column 15, yielding the value 199 (decimal) or 11000111 (binary). Subsequently, the affine transformation calculation is performed, as shown in Eq. (3).

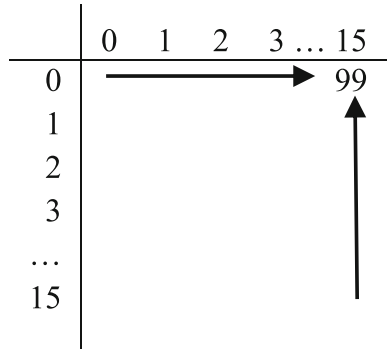

Fig. 1 Multiplicative Inverse Matrix with an Example Value at Row 0, Column 15

$$\left(\begin{bmatrix} 01010111 \\ 10101011 \\ 11010101 \\ 11101010 \\ 01110101 \\ 10111010 \\ 01011101 \\ 10101110 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right) \bmod 2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (3)$$

According to Eq. (3), the value of $B(X) = 11010110$ (binary) or 214 (decimal) is obtained as shown in Fig. 2.

Similarly, if the input S-box is located at row 15 and column 15, as shown in Fig. 3, the 8-bit representation of the input S-box is $X = 11111111$. The value of X^{-1} is obtained from the multiplication inverse matrix (Table 1) by taking the element from row 15 and column 15, yielding the value 28 (decimal) or 00011100 (binary). Subsequently, the affine transformation calculation is performed, as shown in Eq. (4).

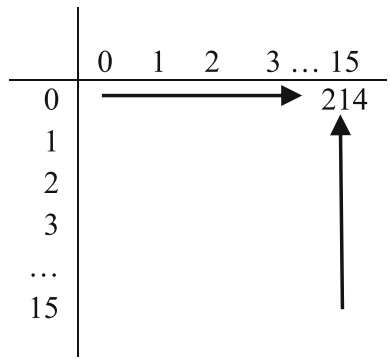

Fig. 2 S-box$_{44}$ with an Example Value at Row 0, Column 15

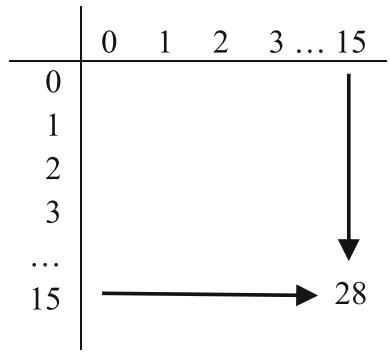

Fig. 3 Multiplicative Inverse Matrix with an Example Value at Row 15, Column 15

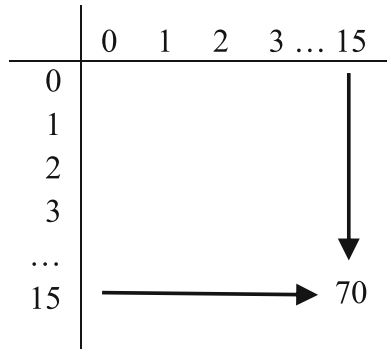

Fig. 4 S-box$_{44}$ with an Example Value at Row 15, Column 15

According to Eq. (4), the value of $B(X) = 01000110$ (binary) or 70 (decimal) is obtained as shown in Fig. 4.

$$\left(\begin{bmatrix} 01010111 \\ 10101011 \\ 11010101 \\ 11101010 \\ 01110101 \\ 10111010 \\ 01011101 \\ 10101110 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right) \bmod 2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (4)$$

The complete results of the S-box$_{44}$ construction are shown in Table 5.

4 S-box strength testing

This section evaluates S-box$_4$, S-box$_{44}$, S-box$_{81}$, S-box$_{111}$, and S-box$_{128}$ using standard S-box testing criteria, including NL, SAC, BIC-NL, BIC-SAC, LAP, and DAP. The following section presents a detailed explanation and the results of these tests.

4.1 Nonlinearity (NL)

NL testing utilizes the mathematical formulation presented in Eq. (5).

$$NL(f(x)) = \min d(f(x), g(x)) \quad (5)$$

where $d(f(x), g(x))$ represents the Hamming distance between the function $f(x)$ and all n-variable affine functions, and x is a variable taking values ranging from 0 to 7. The NL test results for S-box$_4$, S-box$_{44}$, S-box$_{81}$, S-box$_{111}$, and S-box$_{128}$ are listed in Table 9.

Table 9 NL values for the proposed S-boxes

S-box	NL
$S\text{-box}_4$	112
$S\text{-box}_{44}$	112
$S\text{-box}_{81}$	112
$S\text{-box}_{111}$	112
$S\text{-box}_{128}$	112

Table 9 shows that the NL value for each proposed S-box is 112.

4.2 Strict avalanche criterion (SAC)

SAC testing utilizes the mathematical formulation presented in Eq. (6), with the ideal SAC value being 0.5.

$$\left(S(x,i) = \frac{1}{2^n}\sum_{i=1}^{n} f(x) \oplus f\left(x \oplus c_i^n\right)\right) \tag{6}$$

where c_i^n is an n-dimensional vector with a Hamming weight of 1 at the ith position. The SAC test results for $S\text{-box}_4$, $S\text{-box}_{44}$, $S\text{-box}_{81}$, $S\text{-box}_{111}$, and $S\text{-box}_{128}$ are listed in Table 10.

According to Table 10, the best SAC value is 0.50073, with $S\text{-box}_{44}$ achieving a difference of 0.00073 from the ideal value.

4.3 Bit independence criterion nonlinearity (BIC-NL)

The BIC-NL value is calculated using the same mathematical formulation as NL, presented in Eq. (5). However, the difference lies in the output bit in the BIC-NL test. The BIC-NL test results for $S\text{-box}_4$, $S\text{-box}_{44}$, $S\text{-box}_{81}$, $S\text{-box}_{111}$, and $S\text{-box}_{128}$ are listed in Table 11.

Table 11 BIC-NL values for the proposed S-boxes

S-box	BIC-NL
$S\text{-box}_4$	112
$S\text{-box}_{44}$	112
$S\text{-box}_{81}$	112
$S\text{-box}_{111}$	112
$S\text{-box}_{128}$	112

Table 11 shows that the BIC-NL value for each proposed S-box is 112.

4.4 Bit independence criterion strict avalanche criterion (BIC-SAC)

The BIC-SAC value is calculated using the same mathematical formulation as SAC, presented in Eq. (6). However, the difference in the BIC-SAC test lies in the output bit. The BIC-SAC test results for $S\text{-box}_4$, $S\text{-box}_{44}$, $S\text{-box}_{81}$, $S\text{-box}_{111}$, and $S\text{-box}_{128}$ are listed in Table 12.

According to Table 12, the best BIC-SAC value is 0.50098, with a difference of 0.00098 from the ideal value achieved by $S\text{-box}_{81}$ and $S\text{-box}_{111}$.

4.5 Linear approximation probability (LAP)

LAP testing uses the mathematical formulation presented in Eq. (7).

$$LAP = \max_{\Delta x, \Delta y \neq 0} \left| \frac{\#\{x \in X | x.\Gamma x = S(x).\Gamma y\}}{2^n} - \frac{1}{2} \right| \tag{7}$$

where X is a set of all possible inputs, Γx represents all possible masking of input x, and Γy represents all possible masking of the output $S(x)$.

The LAP test results for $S\text{-box}_4$, $S\text{-box}_{44}$, $S\text{-box}_{81}$, $S\text{-box}_{111}$, and $S\text{-box}_{128}$ are listed in Table 13.

Table 10 SAC values for the proposed S-boxes

S-box	SAC	Abs (0.5-SAC)
$S\text{-box}_4$	0.50781	0.00781
$S\text{-box}_{44}$	0.50073	0.00073
$S\text{-box}_{81}$	0.50439	0.00439
$S\text{-box}_{111}$	0.50415	0.00415
$S\text{-box}_{128}$	0.49389	0.00610

Table 12 BIC-SAC values for the proposed S-boxes

S-box	BIC-SAC	Abs (0.5-BIC-SAC)
$S\text{-box}_4$	0.50572	0.00572
$S\text{-box}_{44}$	0.50237	0.00237
$S\text{-box}_{81}$	0.50098	0.00098
$S\text{-box}_{111}$	0.49902	0.00098
$S\text{-box}_{128}$	0.50572	0.00572

Table 13 LAP values for the proposed S-boxes

S-box	LAP
$S\text{-box}_4$	0.0625
$S\text{-box}_{44}$	0.0625
$S\text{-box}_{81}$	0.0625
$S\text{-box}_{111}$	0.0625
$S\text{-box}_{128}$	0.0625

Table 13 shows that the LAP value for each proposed S-box is 0.0625.

4.6 Differential approximation probability (DAP)

DAP testing uses the mathematical formulation presented in Eq. (8).

$$DAP = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X | f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right) \tag{8}$$

where X is the set of all possible inputs, the DAP test results for $S\text{-box}_4$, $S\text{-box}_{44}$, $S\text{-box}_{81}$, $S\text{-box}_{111}$, and $S\text{-box}_{128}$ are listed in Table 14.

As presented in Table 14, the DAP value for $S\text{-box}_4$, $S\text{-box}_{44}$, $S\text{-box}_{81}$, $S\text{-box}_{111}$, and $S\text{-box}_{128}$ is uniformly 0.01563. This uniformity indicates stable performance across all proposed S-boxes in terms of differential approximation probability, a critical measure of cryptographic strength. The identical DAP values suggest that these S-boxes offer a comparable level of resistance to differential cryptographic attacks, underscoring their robustness.

5 Proposed selected S-box

According to Tables 9, 11, 13, and 14, the strengths of $S\text{-box}_4$, $S\text{-box}_{44}$, $S\text{-box}_{81}$, $S\text{-box}_{111}$, and $S\text{-box}_{128}$ are identical. However, Tables 10 and 12 reveal differences in the SAC and BIC-SAC values. Therefore, a new mathematical formulation is introduced to determine the proposed S-box. The mathematical formulation is presented in Eq. (9).

$$S = \frac{abs(SAC - 0.5) + abs(BIC_{SAC} - 0.5)}{2} \tag{9}$$

S is the final value used to determine the proposed S-box, and the ideal S value is 0. The closer the S value is to 0, the stronger the resulting S-box. The S values for the proposed S-boxes are presented in Table 15.

According to Table 15, the sequence of S-boxes with the S value closest to 0 begins with $S\text{-box}_{44}$ with a value of 0.00155, followed by $S\text{-box}_{111}$ with a value of 0.00257, $S\text{-box}_{81}$ with a value of 0.00269, and $S\text{-box}_{128}$ with a value of 0.00591. On the other hand, the S-box with the value farthest from 0 is $S\text{-box}_4$ with a value of 0.00677. This sequence illustrates the variation in S values, reflecting the proximity of each S-box to the ideal reference value of 0. Thus, the best S value is found in $S\text{-box}_{44}$ with a value of 0.00155. Therefore, $S\text{-box}_{44}$ is proposed as the selected S-box. $S\text{-box}_{44}$ is detailed in Table 16. The distribution of SAC and BIC-SAC values is further described in Tables 17 and 18.

6 Performance analysis of the proposed S-box

This section compares the strength of the proposed $S\text{-box}_{44}$ with the AES S-box and S-boxes from previous studies.

6.1 AES S-box

The AES S-box is shown in Table 2. A detailed comparison of the strength between the AES S-box and $S\text{-box}_{44}$ is provided in Table 19. According to Table 19, the AES S-box and $S\text{-box}_{44}$ exhibit the same strength for the NL, BIC-NL, LAP, and DAP criteria, with values of 112, 112, 0.0625, and 0.01563,

Table 14 DAP values for the proposed S-boxes

S-box	DAP
$S\text{-box}_4$	0.01563
$S\text{-box}_{44}$	0.01563
$S\text{-box}_{81}$	0.01563
$S\text{-box}_{111}$	0.01563
$S\text{-box}_{128}$	0.01563

Table 15 S values for the proposed S-boxes

S-box	S
$S\text{-box}_4$	0.00677
$S\text{-box}_{44}$	0.00155
$S\text{-box}_{81}$	0.00269
$S\text{-box}_{111}$	0.00257
$S\text{-box}_{128}$	0.00591

Table 16 The proposed S-box$_{44}$

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

Table 17 The SAC matrix of the proposed S-box$_{44}$

0.48438	0.48438	0.48438	0.48438	0.45313	0.51563	0.56250	0.53125
0.53125	0.50000	0.51563	0.48438	0.48438	0.45313	0.51563	0.46875
0.46875	0.54688	0.56250	0.51563	0.48438	0.48438	0.45313	0.53125
0.53125	0.46875	0.48438	0.56250	0.51563	0.48438	0.48438	0.53125
0.53125	0.50000	0.50000	0.48438	0.56250	0.51563	0.48438	0.43750
0.43750	0.46875	0.484375	0.50000	0.48438	0.56250	0.51563	0.51563
0.51563	0.45313	0.50000	0.48438	0.50000	0.48438	0.56250	0.51563
0.51563	0.48438	0.54688	0.50000	0.484375	0.50000	0.48438	0.46875

Table 18 The BIC-SAC matrix of the proposed S-box$_{44}$

–	0.517578	0.517578	0.490234	0.527344	0.496094	0.482422	0.501954
0.517578	–	0.515625	0.505859	0.507813	0.507813	0.494141	0.490234
0.517578	0.515625	–	0.494141	0.523438	0.507813	0.492188	0.490234
0.490234	0.505859	0.494141	–	0.496094	0.513672	0.501953	0.513672
0.527343	0.507813	0.523438	0.496094	–	0.496094	0.503906	0.496094
0.496094	0.507813	0.507813	0.513672	0.496094	–	0.500000	0.492188
0.482422	0.494141	0.492188	0.501953	0.503906	0.500000	–	0.490234
0.501953	0.490234	0.490234	0.513672	0.496094	0.492188	0.490234	–

Table 19 Comparison of the Strength of AES S-box and S-box$_{44}$

S-boxes	NL	SAC	BIC-NL	BIC-SAC	LAP	DAP
AES [1]	112	0.50488	112	0.50460	0.0625	0.01563
Proposed S-box$_{44}$	**112**	**0.50073**	**112**	**0.50237**	**0.0625**	**0.01563**

Bold indicates the highest value for the criteria listed in the respective column

respectively. However, S-box$_{44}$ is stronger than the AES S-box in terms of the SAC criteria, with values of 0.50073 compared to 0.50488, and the BIC-SAC criteria, with values of 0.50237 compared to 0.50460.

We calculate the percentage increases in SAC, BIC-SAC, and overall S-box strength using Eqs. (10), (11), and (12) to quantify the improvements.

$$S_{SAC} = \frac{SAC_{AES} - SAC_p}{SAC_p} \text{X}100\% \tag{10}$$

$$S_{BIC\_SAC} = \frac{BIC\_SAC_{AES} - BIC\_SAC_p}{BIC\_SAC_p} \text{X}100\% \tag{11}$$

$$S_t = \frac{S_{t\_AES} - S_{t\_p}}{SAC_{t\_p}} \text{X}100\% \tag{12}$$

where:

S_{SAC} represents the percentage increase in the SAC value of S-box$_{44}$ compared to the AES S-box.

SAC_{AES} represents the SAC value of the AES S-box.

SAC_p represents the SAC value of S-box$_{44}$.

S_{BIC_SAC} represents the percentage increase in the BIC-SAC value of the S-box$_{44}$ compared to the AES S-box.

BIC_SAC_{AES} represents the BIC-SAC value of the AES S-box.

BIC_SAC_p represents the BIC-SAC value of S-box$_{44}$.

S_t represents the percentage increase in the overall strength of S-box$_{44}$ compared to the AES S-box.

S_{t_AES} represents the strength value of the AES S-box.

S_{t_p} represents the strength value of S-box$_{44}$.

According to Eqs. (10), (11), and (12), the calculations for the percentages increase in S_{SAC}, S_{BIC_SAC}, and S_t are presented in Eqs. (13), (14), and (19), respectively.

$$S_{SAC} = \frac{0.50488 - 0.50073}{0.50073} \times 100\% = 0,828790\% \tag{13}$$

$$S_{BIC\_SAC} = \frac{0.50460 - 0.50237}{0.50237} \times 100\% = 0,44390\% \tag{14}$$

The strength values of the AES S-box (S_{t_AES}) and S-box$_{44}$ (S_{t_p}) are calculated using Eqs. (15) and (16), respectively.

$$S_{t\_AES} = \frac{NL_{AES} + SAC_{AES} + BIC_{NL\_AES} + BIC_{SAC\_AES} + LAP_{AES} + DAP_{AES}}{6} \tag{15}$$

$$S_{t\_p} = \frac{NL_p + SAC_p + BIC_p + BIC_p + LAP_p + DAP_p}{6} \tag{16}$$

According to Eqs. (15) and (16), the results of calculating the S_{t_AES} and S_{t_p} values are shown in Eqs. (17) and (18), respectively.

$$S_{t\_AES} = \frac{112 + 0.50488 + 112 + 0.50460 + 0.0625 + 0.01563}{6} = 37.514602 \tag{17}$$

$$S_{t\_p} = \frac{112 + 0.50073 + 112 + 0.50273 + 0.0625 + 0.01563}{6} = 37.513598 \tag{18}$$

$$S_t = \frac{37.514602 - 37.513598}{37.513598} \times 100\% = 0.00268\% \tag{19}$$

According to Eqs. (13), (14), and (19), there is an increase in the strength of S-box$_{44}$ compared to the AES S-box for the SAC, BIC-SAC, and overall S-box performance criteria by 0.828790%, 0.44390%, and 0.00268%, respectively.

6.2 S-boxes from previous studies

In this study, the strength of the S-box is compared with those from previous studies using the mathematical formulation presented in Eq. (20).

$$SV = (120 - NL) + abs(0.5 - SAC) + (120 - BIC_{NL}) + abs(0.5 - BIC_{SAC}) \tag{20}$$

where SV represents the accumulated value for the four criteria. The value of 120 is based on the ideal NL and BIC-NL values, while 0.5 is based on the SAC and BIC-SAC values. The best SV value is the one closest to 0, as the ideal SV is 0. Table 20 presents the results of comparing the NL, SAC, BIC-NL, BIC-SAC, and SV values for S-boxes from previous studies.

Table 20 Performance Comparison of S-boxes

S-box	NL	SAC	BIC-NL	BIC-SAC	*SV*
In [2]	110.5	0.510010	103	0.496652	26.513358
In [3]	**112**	0.501465	**112**	0.502999	16.00446
In [4]	107.5	0.500732	104.428571	0.502441	28.324602
In [5]	107	0.504150	104	0.502720	29.006870
In [6]	110.25	0.492431	110.14286	0.500627	19.615339
In [7]	**112**	0.500488	**112**	0.505162	16.00565
In [8]	103.75	0.500488	103	0.503488	33.253976
In [9]	104.25	0.508789	103.928571	0.500698	31.830915
In [10]	107.25	0.500244	104.428571	0.500558	28.322231
In [11]	111.75	0.502930	111.571428	0.502651	16.684152
In [12]	110.75	0.498535	104	**0.500140**	25.394462
In [13]	105.25	0.507080	102.714286	0.503907	32.046701
In [14]	106.75	0.499512	105.071429	0.504255	28.183315
In [15]	109.75	0.508057	100.642857	0.492536	29.622663
In [16]	108.5	0.504639	108.642857	0.504046	22.865827
In [17]	107	0.500500	103.07	0.496200	29.9343
In [18]	111.5	0.509521	103.928571	0.498605	20.582345
In [19]	110	0.503418	103.5	0.504674	26.508092
In [22]	107	0.500500	103.07	0.496200	29.9343
In [23]	104.75	0.499023	103.285714	0.498326	31.966936
In [24]	111.5	0.502197	110.285714	0.506557	18.223040
In [25]	**112**	0.504883	**112**	0.504604	16.009487
In [26]	108.5	0.499512	103.857143	0.501604	27.644949
In [27]	103	0.498779	103.5	0.499023	33.502198
In [28]	102.75	0.499268	103.071429	0.502023	34.181326
In [29]	104	0.506104	104.0714	0.500279	31.934954
In [30]	105.25	0.535156	103.214286	0.508510	31.579381
In [31]	103.5	0.506836	103.4286	0.500349	33.078613
In [32]	106	0.49707	103.5714	0.504116	30.435617
In [33]	108.5	0.499756	103.2143	0.506417	28.292376
In [34]	104.5	**0.500000**	102.857143	0.498535	32.644322
In [35]	100.8	0.480469	102.5	0.493234	36.776297
In [36]	106.5	0.500977	104.0714	0.500488	29.430036
In [37]	106.8	0.502197	103.9286	0.496164	29.327462
In [38]	107.5	0.497070	105.7143	0.497140	26.791504
In [39]	107	0.508057	105.9286	0.500558	27.080043
In [40]	108	0.514404	102.8571	0.506278	29.163539
Proposed S-box$_{44}$	**112**	0.500730	**112**	0.502370	**16.003100**

Bold indicates the highest value for the criteria listed in the respective column

According to Table 20, the performance of each S-box is evaluated based on the NL, SAC, BIC-NL, BIC-SAC, and *SV* criteria.

The highest NL and BIC-NL values in Table 20 is 112, observed in the S-boxes designed by Razaq et al. [3], Ali et al. [7], Tolba et al. [25], and the proposed S-box$_{44}$. The highest SAC value in Table 20 is 0.5, the ideal value achieved by the S-box created by Khan et al. [34]. Unfortunately, this S-box has suboptimal performance in the NL, BIC-NL, and BIC-SAC criteria, with respective values of 104.5, 102.857143, and 0.498535, resulting in an *SV* value of 32.644322.

The highest BIC-SAC value, 0.50014, was achieved by Chemlal et al. [12], however, their S-box falls short in terms of the NL, SAC, and BIC-NL criteria, with respective values of 110.75, 0.498535, and 104. The overall *SV* obtained is 25.394462.

Finally, the proposed S-box$_{44}$ demonstrates the highest overall performance with an *SV* value of 16.0031, surpassing all other S-boxes listed in the table. This underscores the superior design of the proposed S-box$_{44}$ compared to those developed in previous research.

This analysis shows that while other S-boxes may excel in certain criteria, the proposed S-box$_{44}$ is overall the best across all key criteria, making it the most robust and optimal choice for cryptographic applications.

7 Conclusion

This study presents the construction of a modified S-box derived from the AES S-box. The emphasis of the modification lies in the exploration of affine matrices, which results in the generation of 18.446.744.073.709.551.616 possible affine matrices. Upon applying balance and bijectiveness tests, only 128 affine matrices meet the criteria for constructing an S-box. By employing the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$, these 128 affine matrices and an 8-bit constant, a new S-box is formed as a modification of the AES S-box.

The results indicate that S-box$_{44}$, the proposed S-box, has values of NL, SAC, BIC-NL, BIC-SAC, LAP, and DAP as 112, 0.50073, 112, 0.50237, 0.0625, and 0.015625, respectively. These values demonstrate that S-box$_{44}$ is stronger than the AES S-box. The strength improvement of S-box$_{44}$ compared to the AES S-box is 0.828790%, 0.44390%, and 0.00268% based on the SAC, BIC-SAC, and overall S-box performance criteria, respectively.

Additionally, compared to S-boxes from previous studies, S-box$_{44}$ outperforms them with an *SV* value of 16.0031, highlighting its superior strength.

In future research, further improvements in S-box design could be achieved by combining the exploration of affine matrices with all possible irreducible polynomials over *GF* (2^8).

Acknowledgements We are grateful to the Directorate of Research, Technology, and Community Service Directorate General of Higher Education, Research, and Technology Ministry of Education, Culture, Research, and Technology, Indonesia (Grant No 070/E5/PG.02.00.PL/2024).

Author contributions All authors reviewed the manuscript.

Funding The authors have not disclosed any funding.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declared no conflict of interest.

References

1. Daemen, J., Rijmen, V.: The Design of Rijndael. Springer-Verlag, Berlin Heidelberg New York (2002)
2. Artuğer, F.: Strong s-box construction approach based on Josephus problem. Soft. Comput. (2024). https://doi.org/10.1007/s00500-024-09751-7
3. Razaq, A., Ahmad, M., Yousaf, A., Alawida, M., Ullah, A., Shuaib, U.: A group theoretic construction of large number of AES-like substitution-boxes. Wirel. Pers. Commun. **122**, 2057–2080 (2022). https://doi.org/10.1007/s11277-021-08981-4
4. Ma, Y., Tian, Y., Zhang, L., Zuo, P.: Two-Dimensional Hyperchaotic Effect Coupled Mapping Lattice and its Application in Dynamic S-Box Generation. Springer, Netherlands (2024)
5. Ning, H., Zhao, G., Li, Z., Gao, S., Ma, Y., Dong, Y.: A novel method for constructing dynamic S-boxes based on a high-performance spatiotemporal chaotic system. Nonlinear Dyn. **112**, 1487–1509 (2024). https://doi.org/10.1007/s11071-023-09125-y
6. Zhao, M., Yuan, Z., Li, L., Chen, X.B.: A novel efficient S-box design algorithm based on a new chaotic map and permutation. Multimed. Tools Appl. **83**, 64899–64918 (2024). https://doi.org/10.1007/s11042-023-17720-9
7. Ali, R., Ali, J., Ping, P., Jamil, M.K.: A novel S-box generator using Frobenius automorphism and its applications in image encryption. Nonlinear Dyn. **112**, 19463–19486 (2024)
8. Lin, Z., Liu, H.: Constructing a non-degeneracy 3D hyperchaotic map and application in image encryption. Multimed. Tools Appl. (2024). https://doi.org/10.1007/s11042-024-18741-8
9. Liu, H., Liu, J., Ma, C.: Constructing dynamic strong S-box using 3D chaotic map and application to image encryption. Multimed. Tools Appl. (2022). https://doi.org/10.1007/s11042-022-12069-x
10. Wu, W.Q., Kong, L.S.: Image encryption algorithm based on a new 2D polynomial chaotic map and dynamic S-box. Signal Image Video Process. **18**, 3213–3228 (2024). https://doi.org/10.1007/s11760-023-02984-3

11. Si, Y., Liu, H., Zhao, M.: Constructing keyed strong S-box with higher nonlinearity based on 2D hyper chaotic map and algebraic operation. Integration **88**, 269–277 (2023)
12. Chemlal, A., Tabti, H., El Bourakkadi, H., Rrghout, H., Jarjar, A., Benazzi, A.: DNA-Level Action Accompanied by Vigenere Using Strong Pseudo Random S-Box for Color Image Encryption. Springer, US (2024)
13. Zhou, S., Qiu, Y., Wang, X., Zhang, Y.: Novel image cryptosystem based on new 2D hyperchaotic map and dynamical chaotic S-box. Nonlinear Dyn. **111**, 9571–9589 (2023). https://doi.org/10.1007/s11071-023-08312-1
14. Aydin, Y., Garipcan, A.M., Özkaynak, F.: A novel secure s-box design methodology based on FPGA and SHA-256 hash algorithm for block Cipher algorithms. Arab. J. Sci. Eng. (2024). https://doi.org/10.1007/s13369-024-09251-8
15. Arshad, S., Khan, M.: Construction of nonlinear component based on bent Boolean functions. Comput. Appl. Math. **43**, 1–18 (2024). https://doi.org/10.1007/s40314-023-02545-x
16. Khan, M., Jamal, S.S., Hazzazi, M.M., Ali, K.M., Hussain, I., Asif, M.: An efficient image encryption scheme based on double affine substitution box and chaotic system. Integration **81**, 108–122 (2021). https://doi.org/10.1016/j.vlsi.2021.05.007
17. Ding, C., Xue, R.: Signal-sensing dynamic S-box image encryption with 2D Griewank–sin map. Nonlinear Dyn. **111**, 22595–22620 (2023). https://doi.org/10.1007/s11071-023-08985-8
18. Zahid, A.H., Rashid, H., Shaban, M.M.U., Ahmad, S., Ahmed, E., Amjad, M.T., Baig, M.A.T., Arshad, M.J., Tariq, M.N., Tariq, M.W., Zafar, M.A., Basit, A.: Dynamic S-box design using a novel square polynomial transformation and permutation. IEEE Access **9**, 82390–82401 (2021). https://doi.org/10.1109/ACCESS.2021.30867177
19. Manzoor Manzoor, A., Zahid, A.H., Hassan, M.T.: A new dynamic substitution box for data security using an innovative chaotic map. IEEE Access **10**, 74164–74174 (2022). https://doi.org/10.1109/ACCESS.2022.3184012
20. Qiu, H., Xu, X., Jiang, Z., Sun, K., Xiao, C.: A color image encryption algorithm based on hyperchaotic map and Rubik's Cube scrambling. Nonlinear Dyn. **110**, 2869–2887 (2022). https://doi.org/10.1007/s11071-022-07756-1
21. Khan, J.S., Kayhan, S.K., Ahmed, S.S., Ahmad, J., Siddiqa, H.A., Ahmed, F., Ghaleb, B., Al Dubai, A.: Dynamic S-box and pwlcm-based robust watermarking scheme. Wirel. Pers. Commun. **125**, 513–530 (2022). https://doi.org/10.1007/s11277-022-09562-9
22. Khan, L.S., Hazzazi, M.M., Khan, M., Jamal, S.S.: A novel image encryption based on rossler map diffusion and particle swarm optimization generated highly non-linear substitution boxes. Chin. J. Phys. **72**, 558–574 (2021). https://doi.org/10.1016/j.cjph.2021.03.029
23. Hematpour, N., Ahadpour, S.: Execution examination of chaotic S-box dependent on improved PSO algorithm. Neural Comput. Appl. **33**, 5111–5133 (2021). https://doi.org/10.1007/s00521-020-05304-9
24. Ahmad, M., Khaja, I.A., Baz, A., Alhakami, H., Alhakami, W.: Particle swarm optimization based highly nonlinear substitution-boxes generation for security applications. IEEE Access **8**, 116132–116147 (2020). https://doi.org/10.1109/ACCESS.2020.3004449
25. Tolba, M.F., Saleh, H., Salami, Y.A., Al-Qutayri, M., Mohammad, B.: DS2B: dynamic and secure substitution box for efficient speech encryption engine. IEEE Access **9**, 93902–93915 (2021)
26. Alhadawi, H.S., Majid, M.A., Lambić, D., Ahmad, M.: A novel method of S-box design based on discrete chaotic maps and cuckoo search algorithm. Multimed. Tools Appl. **80**, 7333–7350 (2021). https://doi.org/10.1007/s11042-020-10048-8
27. Sani, R.H., Behnia, S., Ziaei, J.: Construction of S-box based on chaotic piecewise map: watermark application. Multimed. Tools Appl. (2022). https://doi.org/10.1007/s11042-022-13278-0
28. Ali, T.S., Ali, R.: A novel color image encryption scheme based on a new dynamic compound chaotic map and S-box. Multimed. Tools Appl. **81**, 20585–20609 (2022). https://doi.org/10.1007/s11042-022-12268-6
29. Zheng, J., Zeng, Q.: An image encryption algorithm using a dynamic S-box and chaotic maps. Appl. Intell. **52**, 15703–15717 (2022). https://doi.org/10.1007/s10489-022-03174-3
30. Hua, Z., Li, J., Chen, Y., Yi, S.: Design and application of an S-box using complete latin square. Nonlinear Dyn. **104**, 807–825 (2021). https://doi.org/10.1007/s11071-021-06308-3
31. Wang, J., Chen, J., Wang, F., Ni, R.: Optical image encryption scheme based on quantum s-box and meaningful ciphertext generation algorithm. Opt. Commun. **525**, 128834 (2022). https://doi.org/10.1016/j.optcom.2022.128834
32. Siddiqui, N., Iftikhar, W., Haq, M.E.U.: An effective s-box construction based on linear recurrences with constant coefficient. Wirel. Pers. Commun. **124**, 1109–1122 (2022). https://doi.org/10.1007/s11277-021-09396-x
33. Alkhayyat, A., Ahmad, M., Tsafack, N., Tanveer, M., Jiang, D., Abd El-Latif, A.A.: A Novel 4D hyperchaotic system assisted Josephus permutation for secure substitution-box generation. J. Signal Process. Syst. **94**, 315–328 (2022). https://doi.org/10.1007/s11265-022-01744-9
34. Khan, N.A., Altaf, M., Khan, F.A.: Selective encryption of JPEG images with chaotic based novel S-box. Multimed. Tools Appl. **80**, 9639–9656 (2021). https://doi.org/10.1007/s11042-020-10110-5
35. Podder, D., Deb, S., Banik, D., Kar, N., Sahu, A.K.: Robust medical and color image cryptosystem using array index and chaotic S-box. Cluster Comput. (2024). https://doi.org/10.1007/s10586-024-04584-3
36. Lambić, D.: A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design. Nonlinear Dyn. **100**, 699–711 (2020)
37. Azam, N.A., Murtaza, G., Hayat, U.: A novel image encryption scheme based on elliptic curves and coupled map lattices. Optik (Stuttg) **274**, 170517 (2023). https://doi.org/10.1016/j.ijleo.2023.170517
38. Arshad, B., Siddiqui, N., Hussain, Z., Ehatisham-ul-Haq, M.: A novel scheme for designing secure substitution boxes (s-boxes) based on mobius group and finite field. Wirel. Pers. Commun. **124**, 3527–3548 (2022). https://doi.org/10.1007/s11277-022-09524-1
39. Shah, D., Shah, T., Naseer, Y., Jamal, S.S., Hussain, S.: Cryptographically strong S-P boxes and their application in

steganography. J. Inf. Secur. Appl. **67**, 103174 (2022). https://doi.org/10.1016/j.jisa.2022.103174
40. Farah Farah, M.A.B., Guesmi, R., Kachouri, A., Samet, M.: A new design of cryptosystem based on S-box and chaotic permutation. Multimed. Tools Appl. (2020). https://doi.org/10.1007/s11042-020-08718-8
41. Tanveer, M., Shah, T., Rehman, A., Ali, A., Siddiqui, G.F., Saba, T., Tariq, U.: Multi-images encryption scheme based on 3D chaotic map and substitution box. IEEE Access **9**, 73924–73937 (2021). https://doi.org/10.1109/ACCESS.2021.3081362
42. Long, M., Wang, L.: S-box design based on discrete chaotic map and improved artificial bee colony algorithm. IEEE Access **9**, 86144–86154 (2021). https://doi.org/10.1109/ACCESS.2021.3069965
43. Alanazi, A.S., Munir, N., Khan, M., Asif, M., Hussain, I.: Cryptanalysis of novel image encryption scheme based on multiple chaotic substitution boxes. IEEE Access **9**, 93795–93802 (2021). https://doi.org/10.1109/ACCESS.2021.3092512
44. Al-Maadeed, T.A., Hussain, I., Anees, A., Mustafa, M.T.: A image encryption algorithm based on chaotic Lorenz system and novel primitive polynomial S-boxes. Multimed. Tools Appl. **80**, 24801–24822 (2021). https://doi.org/10.1007/s11042-021-10695-5
45. Su, Y., Tong, X., Zhang, M., Wang, Z.: A new S-box three-layer optimization method and its application. Nonlinear Dyn. **111**, 2841–2867 (2023). https://doi.org/10.1007/s11071-022-07956-9
46. Zhou, P., Du, J., Zhou, K., Wei, S.: 2D mixed pseudo-random coupling PS map lattice and its application in S-box generation. Nonlinear Dyn. **103**, 1151–1166 (2021). https://doi.org/10.1007/s11071-020-06098-0
47. Deb, S., Behera, P.K.: Design of key-dependent bijective S-boxes for color image cryptosystem. Optik (Stuttg) **253**, 168548 (2022). https://doi.org/10.1016/j.ijleo.2021.168548
48. Hoseini, R., Behnia, S., Sarmady, S., Fathizadeh, S.: Construction of dynamical S-boxes based on image encryption approach. Soft. Comput. **26**, 13985–13997 (2022). https://doi.org/10.1007/s00500-022-07443-8
49. Al-Dweik, A.Y., Hussain, I., Saleh, M., Mustafa, M.T.: A novel method to generate key-dependent s-boxes with identical algebraic properties. J. Inf. Secur. Appl. **64**, 103065 (2022). https://doi.org/10.1016/j.jisa.2021.103065
50. Alamsyah: Improving the quality of AES S-box by modifications irreducible polynomial and affine matrix. In: 2020 5th international conference on informatics and computing. ICIC 2020. (2020). https://doi.org/10.1109/ICIC50835.2020.9288567
51. Ye, G., Guo, L.: A visual meaningful encryption and hiding algorithm for multiple images. Nonlinear Dyn. **112**, 14593–14616 (2024). https://doi.org/10.1007/s11071-024-09790-7
52. Yao, X., Xie, R., Zan, X., Su, Y., Xu, P., Liu, W.: A novel image encryption scheme for DNA storage systems based on DNA hybridization and gene mutation. Interdiscip. Sci. **15**, 419–432 (2023). https://doi.org/10.1007/s12539-023-00565-z