AIMS Mathematics

AIMS Mathematics, 9(12): 35377–35400.
DOI: 10.3934/math.20241681
Received: 01 September 2024
Revised: 30 November 2024
Accepted: 03 December 2024
Published: 18 December 2024

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

Research article

Enhancing the robustness of block ciphers through a graphical S-box evolution scheme for secure multimedia applications

Abdul Razaq[1,*], **Muhammad Mahboob Ahsan**[2], **Hanan Alolaiyan**[3], **Musheer Ahmad**[4] **and Qin Xin**[5]

[1] Department of Mathematics, Division of Science and Technology, University of Education, Lahore 54770, Pakistan, abdul.razaq@ue.edu.pk
[2] Department of Mathematics, Division of Science and Technology, University of Education, Lahore 54770, Pakistan, ahsanmahboob1983@gmail.com
[3] Department of Mathematics, College of Science, King Saud University, Riyadh, Saudi Arabia, holayan@ksu.edu.sa
[4] Department of Computer Engineering, Jamia Millia Islamia, New Delhi 110025, India, musheer.cse@gmail.com
[5] Faculty of Science and Technology, University of the Faroe Islands, Vestara Bryggja 15, Faroe Islands, Denmark, qinx@setur.fo

* **Correspondence:** Email: abdul.razaq@ue.edu.pk, holayan@ksu.edu.sa.

Abstract: Block ciphers are essential for the secure exchange of data and communication, as they are one of the primary components of network security systems. Modern-day block ciphers are most significantly reliant on substitution-boxes (S-boxes). In essence, the security of these cryptosystems is contingent upon the quality of the S-box that is implemented. Robustness and assurance of the security competency necessary to block ciphers are provided by the cryptographically strong S-boxes. A novel coset graph-based algebraic method was proposed to evolve a robust and efficient S-box in order to address the challenges of strong S-box generation. To begin, the vertices of coset graphs for two Galois fields and a bijective function were employed to generate an initial S-box of sufficient cryptographic strength. Afterwards, a permutation group of large order enhances the robustness of the initial S-box, ensuring its resistance against various cryptanalytic attacks. The proposed method's efficacy was verified by comparing the attributes of our S-box with those of S-boxes that have been recently investigated. Furthermore, the proposed S-box was used for image encryption. The outcome of the majority logic criterion (MLC) criteria, differential analysis, and histogram test demonstrates the suitability of the proposed S-box for secure multimedia applications in the results.

Keywords: substitution-box; Galois fields; coset graphs; block ciphers; image encryption
Mathematics Subject Classification: 20D35, 94A60

1. Introduction

Data security includes methods that prevent unauthorized entry, disclosure, alteration, tampering, and interruption of sensitive information [1,2]. In cryptography, we implement procedures to safeguard data, referred to as cryptosystems or ciphers. There are two principal categories of cryptography: Asymmetric cryptography and symmetric cryptography. Asymmetric cryptography utilizes a pair of keys for its operations, whereas symmetric cryptography employs just one key for both encryption and decryption. We categorize symmetric ciphers into two branches: block ciphers and stream ciphers [3]. Sun et al. [4] proposed a full mesh aggregation approach, emphasizing the robustness required for securing cryptographic operations.

The stream cipher operates on a byte-by-byte basis to transform plaintext into ciphertext. In the block cipher, a cryptographic key and algorithm encrypt the secret data into blocks to produce ciphertext. The block cipher's substitution box (S-box) is a fundamental and distinctive feature that is essential for the purpose of concealing the connections between the ciphertext and the key. S-boxes are used in conventional block ciphers, such as the advanced encryption standard (AES) [5] and data encryption standard (DES) [6], to provide cryptosystems with the obfuscation property described by Shannon [7]. In general, the protection of a cryptosystem is contingent upon the grade of the S-box that is employed. The competency of block ciphers is guaranteed by the robust S-box. The significance of the S-box in encryption techniques can be illustrated. By investigating its capacity to induce confusion and perplexity in plaintext and by conducting a variety of analyses, we can illustrate its importance.

The S-box in DES is compromised and should no longer be utilized in critical systems. Consequently, inadequate S-boxes diminish the reliability of cryptosystems, making robust S-boxes essential for the creation of dependable cryptosystems. This prompted the cryptographer to embark on a deeper study to create cryptographically secure S-boxes. The AES block cipher has efficiently employed the 8-bit S-box. As a result of the advantageous implementation of the 8-bit S-box, cryptographers globally concentrated on developing resilient S-boxes of the dimensions 8×8. Siddiqui et al. [8] employed an elliptic curve to form a secure S-box. In [9], the authors introduced a novel method based on the chaotic function firefly algorithm to construct a reliable S-box. In [10], DNA-based computing was employed alongside a chaotic dynamical framework to generate a robust S-box. In [11], a certain type of algebraic structure was employed to compose more than 2 million copies of AES-like S-boxes. In [12], the authors provided an innovative S-box design method. To construct an S-box with elevated nonlinearity, the authors used graphs for a certain triangle group and a permutation group of large order. In [13], the authors introduced a resilient S-box utilizing the stochastic fractal search technique. Artuger and Ozkaynak [14] proposed a novel method for constructing safe S-boxes utilizing chaos theory and genetic algorithms. Fadhil et al. [15] employed a one-dimensional logistic chaotic map to generate an S-box with satisfactory cryptographic attributes. In [16], we present an efficient S-box construction methodology utilizing algebraic rings and symmetric groups. Ullah et al. devised a systematic methodology for the creation of extremely nonlinear S-boxes [17]. The authors utilized the concept of Mordell elliptic curves in the proposed strategy. In [18], a robust S-box design was introduced, incorporating a chaotic sequence and a complete Latin square. In [19], the suggested S-box was developed using a chaotic dynamical oscillator. In [20], researchers developed a novel mechanism of S-box formation that makes use of coset diagrams and a newly defined matrix operation. Khan et al. introduced an S-box derived from a chaotic map exhibiting minimal differential uniformity [21]. Artuger and Özkaynak [22] offered an innovative method to improve the quality of chaos-based substitution boxes. The technique was successfully evaluated on many S-boxes. In [23], a novel external parameter-independent cost mapping was utilized in the development of robust S-boxes.

Recent advances in data security include DNA storage encryption methods, hybridization techniques, and chaotic semi-tensor product theory applications [24–26]. Video encryption innovations leverage temporal action segmentation and 2D memristive cubic maps for enhanced security [27,28]. Innovative approaches to data security include the construction of non-degenerate hyperchaotic systems [29], video encryption algorithms leveraging 2D extended Schaffer function maps and neural networks [30], and the application of 2D-HELS hyperchaotic maps for secure image encryption through RNA operations and dynamic confusion [31].

The growth of algebraic frameworks and their automorphism features are essential concerns. Furthermore, AES proved to be susceptible to several linear cryptanalysis attacks, and numerous loops were deciphered. These issues necessitate the development of new, complex, and robust methods for constructing secure S-boxes. The goal of this research is to develop a novel S-box design scheme based on the combination of two group theoretic graphs whose sum of vertices is 256. We generate a sequence with the vertices of these graphs that possess enough randomness, which is necessary for a reliable S-box. We organize the remaining content of this study as follows: In Section 2, we present some basic knowledge about the Galois field of prime power order and coset graphs. Section 3 provides the construction scheme of the proposed S-box. Section 4 focuses on evaluating the generated S-box's performance using various algebraic analyses. We conduct several statistical analyses in Section 5 to assess the suitability of the generated S-box for image encryption applications through MLC. This section also contains differential analysis, and histogram test. Section 6 presents the conclusion of this research.

2. Mathematical preliminaries

Prior to detailing the suggested strategy, it is essential to explain certain details regarding the modular group $\boldsymbol{M}$, Galois fields $\boldsymbol{GF(2^n)}$, and the associated coset graphs.

2.1. Coset graphs of the modular group over Galois fields

Until 1830, algebraists believed that a finite field has always prime order. For each prime p and $n \in \mathbb{N}$, Évariste Galois constructed a field having p^n number of elements. It is referred to as Galois field, symbolized by $GF(p^n)$. Galois proved that $GF(p^n) = {}^{\mathbb{Z}_p[X]}\big/_{f(\Upsilon)} = \{\Upsilon, \Upsilon^2, \Upsilon^3, \dots, \Upsilon^{n-1} = 1\}$, where $\mathbb{Z}_p[X]$ represents the field extension of $\mathbb{Z}_p$ and $f(\Upsilon)$ is an *nth* degree irreducible polynomial over $\mathbb{Z}_p$ [32]. In other words, a Galois field $GF(p^n)$ can be built using an irreducible polynomial of degree n over $\mathbb{Z}_p$. It is essential to note that, for fixed values of p and n, many irreducible polynomials of degree n exist over $\mathbb{Z}_p$; hence, several Galois fields $GF(p^n)$ of a given order p^n can be generated.

The modular group M [33] is generated by $x\colon \gamma \longrightarrow \frac{-1}{\gamma}$ and $y\colon \gamma \longrightarrow \frac{\gamma-1}{\gamma}$, with the finite presentation $\langle x, y\colon x^2 = y^3 = 1\rangle$. The coset graphs for M were developed in 1983 by Q. Mushtaq [34]. These graphs result from the action of M on $GF(p^n) \cup \{\infty\}$. The generators $x\colon \gamma \longrightarrow \frac{-1}{\gamma}$ and $y\colon \gamma \longrightarrow \frac{\gamma-1}{\gamma}$ of M are applied to each element of $GF(p^n)$. Consequently, we obtain permutation representations of x and y. Since x has order two, then, the permutation representation of x is the product of disjoint transpositions. Similarly, the permutation representation of y is the product of disjoint cycles of length three. In a coset graph, each cycle (a, b, c) of y gives rise to a triangle $\Delta\, abc$ and each cycle (r, s) of x represents a line joining r and s called the x-edge. Note that if r and s are in different cycles of y, then (r, s) is a line joining vertices of two different triangles; otherwise, (r, s) is a loop, that is, a line joining two vertices of the same triangle. The elements of $GF(p^n) \cup \{\infty\}$ that are fixed points of x and y are presented by small circles. A coset graph is composed of triangles, where each vertex of the triangle is linked to just one vertex of the triangle via an x-edge. For further elucidation on the coset graphs of M, we suggest consulting references [35–37].

In the subsequent example, we construct a coset graph of M over $GF(17) \cup \{\infty\}$.

***Example* 2.1** In order to draw the coset graph of M on $GF(17) \cup \{\infty\}$, we first apply $x: \gamma \longrightarrow \frac{-1}{\gamma}$ and $y: \gamma \longrightarrow \frac{\gamma-1}{\gamma}$ on each element of $GF(17) \cup \{\infty\}$. It is important to mention that, for every $\gamma \in GF(17) \cup \{\infty\}$, both $(\gamma)x$ and $(\gamma)y$ are fractions. Since 17 is zero in $GF(17)$, we continue to add 17 to the numerator until we reach an integral value. This method allows us to derive the permutation forms of x and y:

$$x = (0,\infty)(6,14)(2,8)(16,1)(3,11)(9,15)(5,10)(12,7),$$

$$y = (\infty,1,0)(13,14,7)(8,3,12)(2,9,16)(15,10,6)(4,5,11).$$

The coset graph in Figure 1 is the result of the aforementioned permutation forms of x and y.

2.2. Galois fields involved in the proposed method

An 8-bit S-box has 256 distinct entries and $2^8 = 256$, therefore Galois field $GF(2^8)$ plays a vital role in the 8-bit S-box construction scheme. In the literature, many S-box design proposals involving $GF(2^8)$ have been suggested. The S-box used in AES [5] is generated by the irreducible polynomial $1 + \Upsilon + \Upsilon^3 + \Upsilon^4 + \Upsilon^8$. In [38,39], the authors proposed the method of S-box design based on $1 + \Upsilon + \Upsilon^2 + \Upsilon^3 + \Upsilon^4 + \Upsilon^8$. Farwa et al. [40] generated an S-box by using $1 + \Upsilon^4 + \Upsilon^5 + \Upsilon^6 + \Upsilon^8$. In this work, instead of using the Galois field $GF(2^8)$ of 256 elements, we first involve $GF_1(2^7)$ and $GF_2(2^7)$ both having 128 elements, write their elements in a 16 × 16 matrix with the help of the vertices of their coset graphs, and then define a function $f: GF_1(2^7) \cup GF_2(2^7) \longrightarrow GF(2^8)$ to generate our initial S-box of reasonable strength. Furthermore, we increase complexity by reshuffling the initial S-box columns. The irreducible polynomials $f(\Upsilon) = 1 + \Upsilon^4 + \Upsilon^7$, $g(T) = 1 + T + T^2 + T^3 + T^5 + T^6 + T^7$, and $h(\delta) = 1 + \delta^4 + \delta^5 + \delta^6 + \delta^8$ are used to generate the elements of $GF_1(2^7)$, $GF_2(2^7)$, and $GF(2^8)$, respectively (see Tables 1–3).

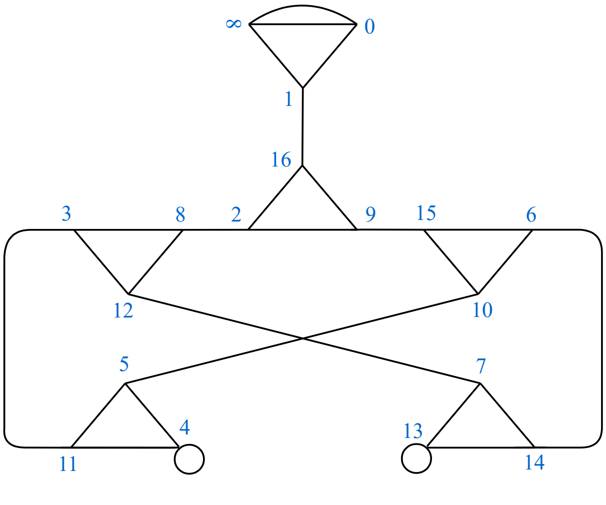


Figure 1: The coset graph of M on $GF(17) \cup \{\infty\}$.

Table 1. Structure description of $GF_1(2^7)$.

Binary Forms	$GF_1(2^7)$	Decimal	Binary Forms	$GF_1(2^7)$	Decimal	Binary Forms	$GF_1(2^7)$	Decimal	Binary Forms	$GF_1(2^7)$	Decimal
0000000	0	0	0000001	Υ^{127}	1	0000010	Υ^1	2	0000100	Υ^2	4
0001000	Υ^3	8	0010000	Υ^4	16	0100000	Υ^5	32	1000000	Υ^6	64
0010001	Υ^7	17	0100010	Υ^8	34	1000100	Υ^9	68	0011001	Υ^{10}	25
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…
0001101	Υ^{115}	13	0011010	Υ^{116}	26	0110100	Υ^{117}	52	1101000	Υ^{118}	104
1000001	Υ^{119}	65	0010011	Υ^{120}	19	0100110	Υ^{121}	38	1001100	Υ^{122}	76
0001001	Υ^{123}	9	0010010	Υ^{124}	18	0100100	Υ^{125}	36	1001000	Υ^{126}	72

Table 2. Structure description of $GF_2(2^7)$.

Binary Forms	$GF_1(2^7)$	Decimal	Binary Forms	$GF_1(2^7)$	Decimal	Binary Forms	$GF_1(2^7)$	Decimal	Binary Forms	$GF_1(2^7)$	Decimal
0000000	0	0	0000001	1	1	0000010	T^1	2	0000100	T^2	4
0001000	T^3	8	0010000	T^4	16	0100000	T^5	32	1000000	T^6	64
1101111	T^7	111	0110001	T^8	49	1100010	T^9	98	0101011	T^{10}	43
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…
1001111	T^{115}	79	1110001	T^{116}	113	0001101	T^{117}	13	0011011	T^{118}	26
0110100	T^{119}	52	1101000	T^{120}	104	0111111	T^{121}	63	1111110	T^{122}	126
0010011	T^{123}	19	0100110	T^{124}	38	1001100	T^{125}	76	1110111	T^{126}	119

Table 3. Structure description of $GF(2^8)$.

Binary Forms	$GF(2^8)$	Binary Forms	$GF(2^8)$	Binary Forms	$GF(2^8)$	Binary Forms	$GF(2^8)$
00000000	0	00000001	1	00000010	δ^1	00000100	δ^2
00001000	δ^3	00010000	δ^4	00000100	δ^5	01000000	δ^6
10000000	δ^7	01110001	δ^8	11100010	δ^9	10110101	δ^{10}
00011011	δ^{11}	00110110	δ^{12}	01101100	δ^{13}	11011000	δ^{14}
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
01011010	δ^{243}	10110100	δ^{244}	00011001	δ^{245}	00110010	δ^{246}
01100100	δ^{247}	11001000	δ^{248}	11100001	δ^{249}	10110011	δ^{250}
00010111	δ^{251}	00101110	δ^{252}	01011100	δ^{253}	10111000	δ^{254}

3. Proposed S-box construction method

The construction process of the generated S-box is based on coset graphs for $GF_1(2^7) \cup \{\infty\}$ and $GF_2(2^7) \cup \{\infty\}$ along with a certain column reshuffling pattern. This section is devoted to narrating the process used to complete the task.

3.1. Coset graphs used in the method

We designed the proposed S-box using two coset graphs D_1 and D_2 evolved through the action of M on $GF_1(2^7) \cup \{\infty\}$ and $GF_2(2^7) \cup \{\infty\}$, respectively.

In this section, we propose our S-box construction method based on the concepts described in the previous section.

3.1.1. Coset graphs of M for $GF_1(2^7) \cup \{\infty\}$

In order to form coset graphs of M for $GF_1(2^7) \cup \{\infty\}$, we first apply the generators $x\colon \gamma \longrightarrow \frac{-1}{\gamma}$ and $y\colon \gamma \longrightarrow \frac{\gamma-1}{\gamma}$ of M on each element of $GF_1(2^7) \cup \{\infty\}$ and obtain permutation representations of x and y, respectively. For instance,

$(\Upsilon^1)x = \frac{-1}{\Upsilon^1} = \frac{1}{\Upsilon^1} = \frac{\Upsilon^{127}}{\Upsilon^1} = \Upsilon^{126}$ and $(\Upsilon^{126})x = \frac{-1}{\Upsilon^{126}} = \frac{1}{\Upsilon^{126}} = \frac{\Upsilon^{127}}{\Upsilon^{126}} = \Upsilon^1$, that is, $(\Upsilon^1, \Upsilon^{126})$. Also, $(\Upsilon^1)y = \frac{\Upsilon^1-1}{\Upsilon^1} = \frac{\Upsilon^{97}}{\Upsilon^1} = \Upsilon^{96}$, $(\Upsilon^{96})y = \frac{\Upsilon^{96}-1}{\Upsilon^{96}} = \frac{\Upsilon^{126}}{\Upsilon^{96}} = \Upsilon^{30}$, and $(\Upsilon^{30})y = \frac{\Upsilon^{30}-1}{\Upsilon^{30}} = \frac{\Upsilon^{31}}{\Upsilon^{30}} = \Upsilon^1$, that is, $(\Upsilon^1, \Upsilon^{96}, \Upsilon^{30})$.

In a similar way, we find all remaining cycles of the permutations x and y which are given as:

$x = (\Upsilon^1,\Upsilon^{126})(\Upsilon^2,\Upsilon^{125})(\Upsilon^3,\Upsilon^{124})(\Upsilon^4,\Upsilon^{123})(\Upsilon^5,\Upsilon^{122})(\Upsilon^6,\Upsilon^{121})(\Upsilon^7,\Upsilon^{120})(\Upsilon^8,\Upsilon^{119})(\Upsilon^9,\Upsilon^{118})$
$(\Upsilon^{10},\Upsilon^{117})(\Upsilon^{11},\Upsilon^{116})(\Upsilon^{12},\Upsilon^{115})(\Upsilon^{13},\Upsilon^{114})(\Upsilon^{14},\Upsilon^{113})(\Upsilon^{15},\Upsilon^{112})(\Upsilon^{16},\Upsilon^{111})(\Upsilon^{17},\Upsilon^{110})(\Upsilon^{18},\Upsilon^{109})$
$(\Upsilon^{19},\Upsilon^{108})(\Upsilon^{20},\Upsilon^{107})(\Upsilon^{21},\Upsilon^{106})(\Upsilon^{22},\Upsilon^{105})(\Upsilon^{23},\Upsilon^{104})(\Upsilon^{24},\Upsilon^{103})(\Upsilon^{25},\Upsilon^{102})(\Upsilon^{26},\Upsilon^{101})(\Upsilon^{27},\Upsilon^{100})$
$(\Upsilon^{28},\Upsilon^{99})(\Upsilon^{29},\Upsilon^{98})(\Upsilon^{30},\Upsilon^{97})(\Upsilon^{31},\Upsilon^{96})(\Upsilon^{32},\Upsilon^{95})(\Upsilon^{33},\Upsilon^{94})(\Upsilon^{34},\Upsilon^{93})(\Upsilon^{35},\Upsilon^{92})(\Upsilon^{36},\Upsilon^{91})(\Upsilon^{37},\Upsilon^{90})$
$(\Upsilon^{38},\Upsilon^{89})(\Upsilon^{39},\Upsilon^{88})(\Upsilon^{40},\Upsilon^{87})(\Upsilon^{41},\Upsilon^{86})(\Upsilon^{42},\Upsilon^{85})(\Upsilon^{43},\Upsilon^{84})(\Upsilon^{44},\Upsilon^{83})(\Upsilon^{45},\Upsilon^{82})(\Upsilon^{46},\Upsilon^{81})(\Upsilon^{47},\Upsilon^{80})$
$(\Upsilon^{48},\Upsilon^{79})(\Upsilon^{49},\Upsilon^{78})(\Upsilon^{50},\Upsilon^{77})(\Upsilon^{51},\Upsilon^{76})(\Upsilon^{52},\Upsilon^{75})(\Upsilon^{53},\Upsilon^{74})(\Upsilon^{54},\Upsilon^{73})(\Upsilon^{55},\Upsilon^{72})(\Upsilon^{56},\Upsilon^{71})(\Upsilon^{57},\Upsilon^{70})$
$(\Upsilon^{58},\Upsilon^{69})(\Upsilon^{59},\Upsilon^{68})(\Upsilon^{60},\Upsilon^{67})(\Upsilon^{61},\Upsilon^{66})(\Upsilon^{62},\Upsilon^{65})(\Upsilon^{63},\Upsilon^{64})$.

$y = (0, \infty, 1)(\Upsilon^1,\Upsilon^{96},\Upsilon^{30})(\Upsilon^2,\Upsilon^{65},\Upsilon^{60})(\Upsilon^3,\Upsilon^{120},\Upsilon^4)(\Upsilon^5,\Upsilon^{45},\Upsilon^{77})(\Upsilon^6,\Upsilon^{113},\Upsilon^8)(\Upsilon^7,\Upsilon^{124},\Upsilon^{123})$
$(\Upsilon^9,\Upsilon^{83},\Upsilon^{35})(\Upsilon^{10},\Upsilon^{90},\Upsilon^{27})(\Upsilon^{11},\Upsilon^{93},\Upsilon^{23})(\Upsilon^{12},\Upsilon^{99},\Upsilon^{16})(\Upsilon^{13},\Upsilon^{36},\Upsilon^{78})(\Upsilon^{14},\Upsilon^{121},\Upsilon^{119})$
$(\Upsilon^{15},\Upsilon^{64},\Upsilon^{48})(\Upsilon^{17},\Upsilon^{58},\Upsilon^{52})(\Upsilon^{18},\Upsilon^{39},\Upsilon^{70})(\Upsilon^{19},\Upsilon^{21},\Upsilon^{87})(\Upsilon^{20},\Upsilon^{53},\Upsilon^{54})(\Upsilon^{22},\Upsilon^{59},\Upsilon^{46})$
$(\Upsilon^{24},\Upsilon^{71},\Upsilon^{32})(\Upsilon^{25},\Upsilon^{41},\Upsilon^{61})(\Upsilon^{26},\Upsilon^{72},\Upsilon^{29})(\Upsilon^{28},\Upsilon^{115},\Upsilon^{111})(\Upsilon^{31},\Upsilon^{126},\Upsilon^{97})(\Upsilon^{33},\Upsilon^{43},\Upsilon^{51})$
$(\Upsilon^{34},\Upsilon^{116},\Upsilon^{104})(\Upsilon^{37},\Upsilon^{117},\Upsilon^{111})(\Upsilon^{38},\Upsilon^{42},\Upsilon^{47})(\Upsilon^{40},\Upsilon^{106},\Upsilon^{108})(\Upsilon^{44},\Upsilon^{118},\Upsilon^{92})(\Upsilon^{48},\Upsilon^{15},\Upsilon^{64})$
$(\Upsilon^{49},\Upsilon^{91},\Upsilon^{114})(\Upsilon^{50},\Upsilon^{82},\Upsilon^{122})(\Upsilon^{55},\Upsilon^{101},\Upsilon^{98})(\Upsilon^{56},\Upsilon^{103},\Upsilon^{95})(\Upsilon^{57},\Upsilon^{88},\Upsilon^{109})(\Upsilon^{62},\Upsilon^{125},\Upsilon^{67})$
$(\Upsilon^{63},\Upsilon^{112},\Upsilon^{79})(\Upsilon^{66},\Upsilon^{86},\Upsilon^{102})(\Upsilon^{68},\Upsilon^{105},\Upsilon^{81})(\Upsilon^{69},\Upsilon^{110},\Upsilon^{75})(\Upsilon^{73},\Upsilon^{74},\Upsilon^{107})(\Upsilon^{76},\Upsilon^{84},\Upsilon^{94})$
$(\Upsilon^{80},\Upsilon^{85},\Upsilon^{89})$.

These permutations of x and y give rise to a disconnected coset diagram D_1, consisting of a total of 22 patches. It is worth mentioning that out of these 22 patches, 21 are of similar type, denoted by $D_1(\Gamma_i)$, where $i = 1,2,3,\ldots,21$, and the 22nd patch, which is a straight line connecting 1 and 0 through x, is denoted by $D_1(\Pi)$. Figures 2 and 3 show $D_1(\Pi)$ and one of the copies of $D_1(\Gamma_i)$, respectively.

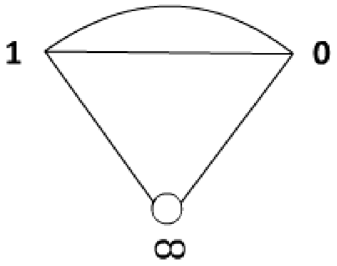


Figure 2. The patch $D_1(\Pi)$.

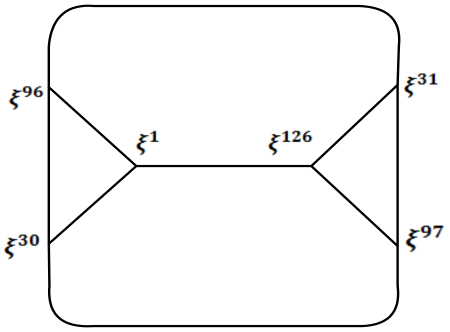


Figure 3. The patch $D_1(\Gamma_1)$.

3.1.2. Coset graphs of M for $GF_2(2^7) \cup \{\infty\}$

We denote this coset diagram by D_2, obtained as a result of the action of M on $GF_2(2^7) \cup \{\infty\}$. The permutation representations of x and y are given below:

$$x = (T^{1},T^{126})(T^{2},T^{125})(T^{3},T^{124})(T^{4},T^{123})(T^{5},T^{122})(T^{6},T^{121})(T^{7},T^{120})(T^{8},T^{119})(T^{9},T^{118})(T^{10},T^{117})$$
$$(T^{11},T^{116})(T^{12},T^{115})(T^{13},T^{114})(T^{14},T^{113})(T^{15},T^{112})(T^{16},T^{111})(T^{17},T^{110})(T^{18},T^{109})(T^{19},T^{108})$$
$$(T^{20},T^{107})(T^{21},T^{106})(T^{22},T^{105})(T^{23},T^{104})(T^{24},T^{103})(T^{25},T^{102})(T^{26},T^{101})(T^{27},T^{100})(T^{28},T^{99})$$
$$(T^{29},T^{98})(T^{30},T^{97})(T^{31},T^{96})(T^{32},T^{95})(T^{33},T^{94})(T^{34},T^{93})(T^{35},T^{92})(T^{36},T^{91})(T^{37},T^{90})(T^{38},T^{89})$$
$$(T^{39},T^{88})(T^{40},T^{87})(T^{41},T^{86})(T^{42},T^{85})(T^{43},T^{84})(T^{44},T^{83})(T^{45},T^{82})(T^{46},T^{81})(T^{47},T^{80})(T^{48},T^{79})$$
$$(T^{49},T^{78})(T^{50},T^{77})(T^{51},T^{76})(T^{52},T^{75})(T^{53},T^{74})(T^{54},T^{73})(T^{55},T^{72})(T^{56},T^{71})(T^{57},T^{70})(T^{58},T^{69})$$
$$(T^{59},T^{68})(T^{60},T^{67})(T^{61},T^{66})(T^{62},T^{65})(T^{63},T^{64}).$$

$$y = (0,\infty,1)(T^{1},T^{54},T^{72})(T^{2},T^{108},T^{17})(T^{3},T^{85},T^{39})(T^{4},T^{89},T^{34})(T^{5},T^{92},T^{30})(T^{6},T^{43},T^{78})$$
$$(T^{7},T^{13},T^{107})(T^{8},T^{51},T^{68})(T^{9},T^{16},T^{102})(T^{10},T^{57},T^{60})(T^{11},T^{37},T^{79})(T^{12},T^{86},T^{29})(T^{14},T^{26},T^{87})$$
$$(T^{15},T^{66},T^{46})(T^{18},T^{32},T^{77})(T^{19},T^{125},T^{110})(T^{20},T^{114},T^{120})(T^{21},T^{62},T^{44})(T^{22},T^{74},T^{31})$$
$$(T^{23},T^{71},T^{33})(T^{24},T^{45},T^{58})(T^{25},T^{111},T^{118})(T^{27},T^{36},T^{64})(T^{28},T^{52},T^{47})(T^{35},T^{122},T^{97})$$

$(T^{38},T^{123},T^{93})(T^{40},T^{101},T^{113})(T^{41},T^{115},T^{98})(T^{42},T^{124},T^{88})(T^{48},T^{90},T^{116})(T^{49},T^{84},T^{121})$

$(T^{50},T^{95},T^{109})(T^{53},T^{105},T^{96})(T^{55},T^{73},T^{126})(T^{56},T^{104},T^{94})$ $(T^{59},T^{76},T^{119})(T^{61},T^{112},T^{81})$

$(T^{63},T^{91},T^{100})(T^{65},T^{106},T^{83})(T^{67},T^{70},T^{117})(T^{69},T^{82},T^{103})(T^{75},T^{99},T^{80})$.

The coset diagrams D_1 and D_2 are similar except in the labeling of the vertices. In D_2, the 21 similar types of patches are denoted by $D_2(\Gamma_i)$, and the 22nd patch is represented by $D_2(\Pi)$. In Figures 4 and 5, $D_2(\Pi)$ and one of the copies of $D_2(\Gamma_i)$ are shown, respectively.

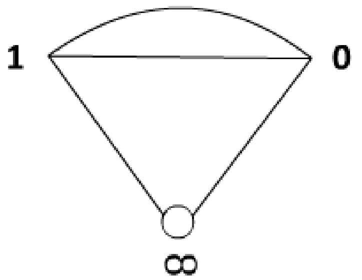


Figure 4. The patch $D_2(\Pi)$.

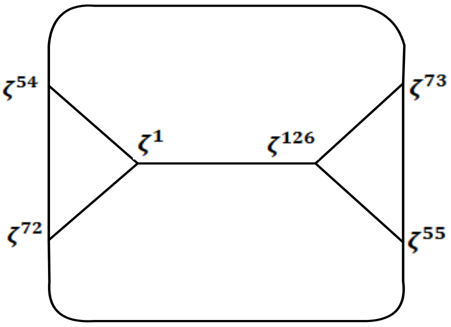


Figure 5. The patch $D_2(\Gamma_1)$.

3.2. Proposed method

The suggested S-box construction scheme involves three steps. The explanation of all three steps is provided below.

Step 1. In this step, we construct a square matrix of 256 elements by the above-mentioned two coset graphs, that is, coset graphs for $GF_1(2^7)$ and $GF_2(2^7)$. We pick one copy of fragments $D_1(\Gamma_i): i = 1,2,3,\dots,21$, which has a vertex with the least power of Υ, that is, Υ^1. Call it $D_1(\Gamma_1)$, and apply $xyxy^{-1}xy$ on $\Upsilon^1 \in D_1(\Gamma_1)$ such that we reach Υ^{126} by following the path:

$\Upsilon^1 \xrightarrow{x} \Upsilon^{126} \xrightarrow{y} \Upsilon^{97} \xrightarrow{x} \Upsilon^{30} \xrightarrow{y^{-1}} \Upsilon^{96} \xrightarrow{x} \Upsilon^{31}$ (see Figure 3). Insert $\Upsilon^1, \Upsilon^{126}, \Upsilon^{97}, \Upsilon^{30}, \Upsilon^{96}$, and Υ^{31} at the 1st, 2nd, 3rd, 4th, 5th and 6th places of the first row, respectively. Next, we choose a copy from the fragments $D_2(\Gamma_i): i = 1,2,3,\dots,21$ that contains a vertex with the least power of T, that is, T^1. Name it $D_2(\Gamma_1)$ (see Figure 5) and write all the vertices of $D_2(\Gamma_1)$ at the 7th, 8th, 9th, 10th, 11th, and 12th positions of the 1st row in a similar way as written in the case of $D_1(\Gamma_1)$. After that, select a copy from $\{D_1(\Gamma_i): i = 1,2,3,\dots,21\} - \{D_1(\Gamma_1)\}$ that has the least power of T (it is important to mention here that the vertex with the least power of T does not have to be T^2, because T^2 could be one of the vertices of $D_1(\Gamma_1)$. Name this copy $D_1(\Gamma_2)$, and write the six vertices of $D_1(\Gamma_2)$ as the next six elements (13th,14th, 15th and 16th elements of the 1st row and the 1st and 2nd elements of the 2nd row) of the matrix in a similar order mentioned in the case of $D_1(\Gamma_1)$. Then, we use a copy from $\{D_2(\Gamma_i): i =$

$1,2,3,\dots,21\} - \{D_2(\Gamma_1)\}$ to write 6 more elements, and this process continues until all the copies of $D_1(\Gamma_i)$ and $D_2(\Gamma_i)$ are exhausted. In this way, we have filled the matrix with the elements of $GF_1(2^7)$ and $GF_2(2^7)$ up to the 12th element of the 16th row. Lastly, place 1, 0 from fragment $D_1(\Pi)$ and 1, 0 from fragment $D_2(\Pi)$ at the 13th, 14th, 15th, and 16th positions of the last row. Thus, we were able to develop a square matrix (see Table 4) with 256 points from $GF_1(2^7)$ and $GF_2(2^7)$.

Table 4. Output of Step 1.

$\boldsymbol{\Upsilon^{1}}$	$\boldsymbol{\Upsilon^{126}}$	$\boldsymbol{\Upsilon^{97}}$	$\boldsymbol{\Upsilon^{30}}$	$\boldsymbol{\Upsilon^{96}}$	$\boldsymbol{\Upsilon^{31}}$	$\boldsymbol{T^{1}}$	$\boldsymbol{T^{126}}$	$\boldsymbol{T^{55}}$	$\boldsymbol{T^{72}}$	$\boldsymbol{T^{54}}$	$\boldsymbol{T^{73}}$	$\boldsymbol{\Upsilon^{2}}$	$\boldsymbol{\Upsilon^{125}}$	$\boldsymbol{\Upsilon^{67}}$	$\boldsymbol{\Upsilon^{60}}$
$\boldsymbol{\Upsilon^{65}}$	Υ^{62}	T^{2}	T^{125}	T^{110}	T^{17}	T^{108}	T^{19}	Υ^{3}	Υ^{124}	Υ^{123}	Υ^{4}	Υ^{120}	Υ^{7}	T^{3}	T^{124}
$\boldsymbol{T^{88}}$	T^{39}	T^{85}	T^{42}	Υ^{5}	Υ^{122}	Υ^{50}	Υ^{77}	Υ^{45}	Υ^{82}	T^{4}	T^{123}	T^{93}	T^{34}	T^{89}	T^{38}
$\boldsymbol{\Upsilon^{6}}$	Υ^{121}	Υ^{119}	Υ^{8}	Υ^{113}	Υ^{14}	T^{5}	T^{122}	T^{97}	T^{30}	T^{92}	T^{35}	Υ^{9}	Υ^{118}	Υ^{92}	Υ^{35}
$\boldsymbol{\Upsilon^{83}}$	Υ^{44}	T^{6}	T^{121}	T^{49}	T^{78}	T^{43}	T^{84}	Υ^{10}	Υ^{117}	Υ^{100}	Υ^{27}	Υ^{90}	Υ^{37}	T^{7}	T^{120}
$\boldsymbol{T^{20}}$	T^{107}	T^{13}	T^{114}	Υ^{11}	Υ^{116}	Υ^{104}	Υ^{23}	Υ^{93}	Υ^{34}	T^{8}	T^{119}	T^{59}	T^{68}	T^{51}	T^{76}
$\boldsymbol{\Upsilon^{12}}$	Υ^{115}	Υ^{111}	Υ^{16}	Υ^{99}	Υ^{28}	T^{9}	T^{118}	T^{25}	T^{102}	T^{16}	T^{111}	Υ^{13}	Υ^{114}	Υ^{49}	Υ^{78}
$\boldsymbol{\Upsilon^{36}}$	Υ^{91}	T^{10}	T^{117}	T^{67}	T^{60}	T^{57}	T^{70}	Υ^{15}	Υ^{112}	Υ^{79}	Υ^{48}	Υ^{64}	Υ^{63}	T^{11}	T^{116}
$\boldsymbol{T^{48}}$	T^{79}	T^{37}	T^{90}	Υ^{17}	Υ^{110}	Υ^{75}	Υ^{52}	Υ^{58}	Υ^{69}	T^{12}	T^{115}	T^{98}	T^{29}	T^{86}	T^{41}
$\boldsymbol{\Upsilon^{18}}$	Υ^{109}	Υ^{57}	Υ^{70}	Υ^{39}	Υ^{88}	T^{14}	T^{113}	T^{40}	T^{87}	T^{26}	T^{101}	Υ^{19}	Υ^{108}	Υ^{40}	Υ^{87}
$\boldsymbol{\Upsilon^{21}}$	Υ^{106}	T^{15}	T^{112}	T^{81}	T^{46}	T^{66}	T^{61}	Υ^{20}	Υ^{107}	Υ^{73}	Υ^{54}	Υ^{53}	Υ^{74}	T^{18}	T^{109}
$\boldsymbol{T^{50}}$	T^{77}	T^{32}	T^{95}	Υ^{22}	Υ^{105}	Υ^{81}	Υ^{46}	Υ^{59}	Υ^{68}	T^{21}	T^{106}	T^{83}	T^{44}	T^{62}	T^{65}
$\boldsymbol{\Upsilon^{24}}$	Υ^{103}	Υ^{95}	Υ^{32}	Υ^{71}	Υ^{56}	T^{22}	T^{105}	T^{96}	T^{31}	T^{74}	T^{53}	Υ^{25}	Υ^{102}	Υ^{66}	Υ^{61}
$\boldsymbol{\Upsilon^{41}}$	Υ^{86}	T^{23}	T^{104}	T^{94}	T^{33}	T^{71}	T^{56}	Υ^{26}	Υ^{101}	Υ^{98}	Υ^{29}	Υ^{72}	Υ^{55}	T^{24}	T^{103}
$\boldsymbol{T^{69}}$	T^{58}	T^{45}	T^{82}	Υ^{33}	Υ^{94}	Υ^{76}	Υ^{51}	Υ^{43}	Υ^{84}	T^{27}	T^{100}	T^{63}	T^{64}	T^{36}	T^{91}
$\boldsymbol{\Upsilon^{38}}$	Υ^{89}	Υ^{80}	Υ^{47}	Υ^{42}	Υ^{85}	T^{28}	T^{99}	T^{80}	T^{47}	T^{52}	T^{75}	Υ^{127}	0	T^{127}	0

Step 2. In Step 1, we created a matrix with 128 distinct entries at 256 positions, ensuring each entry occupies two positions. In this step, we construct our initial S-box, that is, a square matrix of order 16 with distinct entries (see Table 5) by defining a bijective map $f\colon GF_1(2^7) \cup GF_2(2^7) \longrightarrow GF(2^8)$ by

$$f(\Delta) = \begin{cases} \delta^{2n+1}, & if\ \Delta = \Upsilon^{2n+1} \\ \delta^{2n+128}, & if\ \Delta = \Upsilon^{2n} \\ \delta^{2n}, & if\ \Delta = T^{2n} \\ \delta^{2n+129}, & if\ \Delta = T^{2n+1} \\ 0 & if\ \Delta = 0 \in GF_1(2^7) \\ \delta^{128}, & if\ \Delta = 0 \in GF_1(2^7) \end{cases}.$$

The constructed initial S-box possesses satisfactory qualities to secure sensitive information. Its nonlinearity value is 104.50. In the next step, we further strengthen its security capabilities.

Table 5. Output of Step 2 (Initial S-box).

2	184	137	176	41	175	182	242	15	117	138	246	29	121	38	145
177	166	04	92	146	183	156	62	08	46	66	116	200	128	58	132
143	71	49	53	32	179	214	236	217	40	16	23	43	188	131	72
161	168	42	102	37	220	232	33	82	239	235	97	226	50	173	09
210	251	64	225	107	169	197	213	233	178	114	215	119	36	51	84
13	48	110	74	27	180	06	104	167	136	113	100	240	76	221	118
55	148	85	227	198	44	204	21	189	103	243	147	108	45	79	186
194	205	181	25	67	14	60	65	193	87	35	141	122	112	163	89
159	05	245	222	151	241	59	203	120	152	54	90	99	88	199	109
31	73	229	250	144	249	216	174	81	196	211	228	190	96	142	255
26	24	201	170	20	195	19	83	124	78	234	191	69	157	95	192
158	93	47	172	129	171	140	127	07	134	248	39	80	212	56	244
230	206	126	34	130	30	52	12	252	17	165	231	209	185	153	28
162	98	115	237	63	68	133	202	11	139	164	207	123	101	208	03
125	187	135	105	94	86	150	77	106	160	22	253	61	224	18	238
155	111	10	247	218	219	223	57	70	254	154	75	149	0	1	91

Step 3. In this step, we apply the action of a permutation group *G* with four generators on the initial S-box to generate our proposed S-box. The group G is generated by the elements *a,b,c,* and *d,* where the generators are given as:

$a =(1,148,162,24,38,93,152,90,78,41,13,54,35,213,155,89,98,127,192,211,5,200,186,117,96,99,206,37,208,229,250,109,203,181,25,107,170,246,14,33,207,150,77,232,97,240,112,231,95,122,182,252,74,57,66,129,6,30,102,251,23,165,9,151,120,134,234,55,12,184,256,239,218,188,244,104,216,233,131,177,224,164,214,220,195,15,42,227,48,198,50,31,118,83,156,88,147,124,179,47,80,222,19,111,169,32,221,76,139,67,140,245,110,238,132,103,56,115,63,125,161,201,86,194,167,20,39,199,128,79,126,84,176,226,58,119,28,49).$

$b = (2,149,65,193,73,52,8).$

$c = (3,105,254,43,174,175,166,137,7,17,173,133,145,144,51,138,75,249,253,64,146,197,159,235,191,121,94,172,114,236,142,160,92,189,85,255,153,27,196,141,91,100,187,185,204,183,180,59,26,68,53,219,36,82,71,178,60,10,157,11,44,4,248,45,202,106,61,72,243,168,87,22,225,136,101,163,209,21,190,116,247,158,242,18,70,210,143,212,113,205,237,62,130,29,228,123,40,215,171,108,154,46,16).$

$d =(34,81,241,230,69,217,135).$

With the help of GAP software, we determine that G has the finite presentation of the form:

$$\langle a,b,c,d\colon a^{138}=b^{7}=c^{103}=d^{7}=aba^{-1}b^{-1}=aca^{-1}c^{-1}=bcb^{-1}c^{-1}=ada^{-1}d^{-1}=bdb^{-1}d^{-1}=cdc^{-1}d^{-1}\rangle,$$

and the order of G is 696486. The actions of all these 696486 elements (permutations) on the initial S-box produce a new S-box. After exhaustive enumeration, we identify that the S-box resulting from the application of the element $a^{83}b^{5}c^{13}d^{4}$ achieves the highest nonlinearity score of 111.75. Thus, we select this S-box as our proposed S-box (see Table 6) for its enhanced security properties.

Table 6. Output of Step 3 (Proposed S-box).

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

4. Algebraic analyses

In this section, we undertake an assessment of the security characteristics of the newly created S-box. The evaluation of the properties of the proposed S-box is critical in determining its potential use in various encryption techniques and security contexts. To accomplish this objective, we apply five security performance tests. We then compare the results obtained from the proposed S-box to those of widely recognized S-boxes. The subsequent sections present a comprehensive explanation of the security tests employed on these S-boxes.

4.1. Bijection test

The bijection test assesses the distinctiveness of the output generated by an S-box. When an S-box meets the bijection criterion, the output values are distinct and are not repeated within the range of $[0,255]$. Additionally, a one-to-one correspondence exists between each input and output value. The suggested S-box has been found to meet the criteria for the bijection test. It generates distinct output values within the range of $[0, 255]$, establishing a one-to-one correspondence between each input and its corresponding output.

4.2. Nonlinearity

A Boolean mapping $\theta: Z_2^k \longrightarrow Z_2$ is nonlinear if it is at least as far away from the set of affine mappings as possible. This makes sure that the input vectors are not linearly mapped to the output vectors [41]. Its mathematical calculation is as follows:

$$\mathcal{N}_\theta = 2^{k-1} - \frac{1}{2}\left[\max_{v \in Z_2^k} \left(|\mathcal{S}_\theta(v)|\right)\right],$$

where $\mathcal{S}_\theta(v) = \sum_{u \in Z_2^k} (-1)^{\theta(u)} (-1)^{u.v}$ is the Walsh spectrum of $\theta(u)$ and $u.v$ represents the scalar product of u and v, respectively. Table 7 shows that the average nonlinearity of all eight Boolean mappings in the proposed S-box is 111.75. Table 12 presents a comparison of our S-box with other S-boxes in terms of nonlinearity analysis, demonstrating the proficiency of our S-box.

Table 7. Nonlinearity values of the proposed S-box

Boolean function	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	Mean
NL score	112	112	110	112	112	112	112	112	111.75

4.3. Strict avalanche criteria

The assessment of the quality of an S-box also encompasses the use of the strict avalanche criterion (SAC), which was introduced in 1985 [42]. This method assesses whether altering a single input bit results in a probability of half of the output bits changing. This analysis is characterized by the determination of the S-box's dependency matrix. A desirable S-box should have an average value for all dependency matrix elements that is closer to 0.50. Table 8 shows the dependency matrix of the developed S-box, with an SAC average value of 0.5007, which is nearly equal to the ideal value. Therefore, the developed S-box meets the SAC requirements.

Table 8. SAC values of the proposed S-box.

0.5	0.5469	0.4531	0.5156	0.4844	0.4531	0.4531	0.5625
0.4844	0.4844	0.5156	0.5312	0.4531	0.4844	0.5	0.5156
0.5156	0.4688	0.5	0.5312	0.5625	0.5	0.5312	0.4375
0.4375	0.4688	0.5312	0.4531	0.5156	0.4688	0.5	0.4688
0.5	0.5156	0.5	0.5312	0.5156	0.4844	0.5312	0.5156
0.5469	0.5312	0.4531	0.4688	0.4531	0.5	0.5156	0.4688
0.5312	0.5312	0.5156	0.5156	0.5	0.5312	0.5469	0.5156
0.4531	0.5156	0.5469	0.4688	0.4531	0.5625	0.4375	0.5625

4.4. Bits independence criteria

The bit independence criteria (BIC) is a set of strict rules for checking how well the output bits work and how changes affect the next encryption cycles. This scrutiny involves pairwise comparison of the variables to ascertain their level of independence. For intricate and dependable systems, a high degree of BIC-nonlinearity is essential. Table 9 illustrates the BIC-nonlinearity dependency matrix. Additionally, we apply the SAC to BIC. Table 10 showcases the dependency matrix for BIC-SAC. The findings indicate that the proposed S-box meets the BIC requirements.

Table 9. BIC nonlinearity scores for the proposed S-box.

-	112	112	112	112	112	112	112
112	-	110	112	110	110	110	112
112	110	-	112	112	112	112	110
112	112	112	-	110	110	112	112
112	110	112	110	-	112	112	112
112	110	112	110	112	-	112	112
112	110	112	112	112	112	-	112
112	112	110	112	112	112	112	-

Table 10. BIC-SAC Outcomes for the proposed S-box.

-	0.502	0.5312	0.5078	0.502	0.5195	0.4941	0.4961
0.502	-	0.4766	0.4941	0.5078	0.498	0.5059	0.4883
0.5312	0.4766	-	0.4941	0.4883	0.5059	0.4902	0.4961
0.5078	0.4941	0.4941	-	0.5156	0.5176	0.4941	0.498
0.502	0.5078	0.4883	0.5156	-	0.5039	0.4824	0.4961
0.5195	0.498	0.5059	0.5176	0.5039	-	0.502	0.4668
0.4941	0.5059	0.4902	0.4941	0.4824	0.502	-	0.5137
0.4961	0.4883	0.4961	0.498	0.4961	0.4668	0.5137	-

4.5. Linear probability

To ensure data privacy, modern block ciphers aim to increase the degree of indeterminacy and complexity in the encrypted data bits. This mechanism provides a shield against various techniques utilized by cryptanalysts to decipher the encrypted text. The primary way to achieve this is by deploying S-boxes. An S-box with a lower linear probability (LP) score is generally considered to be an effective countermeasure against linear cryptanalysis attacks. A mathematical formula [43] determines the LP of a substitution box, as shown below:

$$LP = \max_{f_u, g_u \neq 0} \left| \frac{\#\{u \in GF(2^k): u.f_u = S(u).g_u\}}{2^k} - \frac{1}{2} \right|,$$

where f_u and g_u represent input and output masks, respectively. The proposed S-box has an LP score of 0.0703.

4.6. Differential uniformity

A fundamental measure of an S-box's performance, differential uniformity (DU) [43], evaluates its resistance against differential attacks. We compute DU as the number of identical mappings from an input differential Δr to an output differential Δs. The S-box is considered efficient in countering differential attacks when it has a low DU value. The mathematical formula to calculate the DU value is as follows;

$$DU = \max_{\Delta r \neq 0, \Delta s} \#\{r \in Y: S(r) \oplus S(r + \Delta r) = \Delta s\}.$$

Table 11 shows the differential distribution for the proposed S-box. Our S-box has a maximum DU value of 6, which indicates that it is sufficiently strong to resist the effects of differential attacks.

Table 11. DU Values of the proposed S-box

4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	6	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	2	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	6	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	2	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	0

4.7. Comparison analysis discussion

The suggested S-box shows remarkable cryptographic properties, therefore stressing the strength of its developing technique in comparison to current S-boxes. By means of a thorough investigation of important performance measures in Table 12, the S-box exhibits robustness against several cryptographic assaults, surpassing many previously developed S-boxes using optimization, algebraic, and chaotic approaches. Significant findings on the strength of the suggested S-box architecture are presented below.

i. To withstand linear assaults, the S-box must have a large NL value. It is critical for defending against linear cryptanalysis by increasing the S-box's complexity and confusion. The proposed S-box achieves an average NL value of 111.75, which is significantly higher than S-boxes listed in Table 12. Therefore, the proposed S-box has a significant level of complexity and confusion, rendering it resistant to all available linear cryptanalysis techniques.

ii. A SAC value close to 0.5 shows that each output bit depends equally on each input bit. This way, an optimal diffusion effect can be achieved. The SAC value of the proposed S-box (0.5007) is approximately equal to 0.5 outperforming many S-boxes listed in Table 12. This shows the S-box's strong compliance with the SAC requirement, ensuring a robust spread of bit changes across output.

iii. In terms of BIC, both for SAC and NL, the proposed S-box exhibits superior performance. A high BIC score ensures that changes in individual bits are independent and propagate effectively, a critical feature for strong encryption. The proposed S-box achieves BIC-NL and BIC-SAC values that surpass most of the existing S-boxes, ensuring optimal bit independence.

iv. An efficacious S-box has a lower DU value. As shown in Table 12, the DU value of the proposed S-box is either less than or equal to the DU values of the S-boxes listed in Table 12.

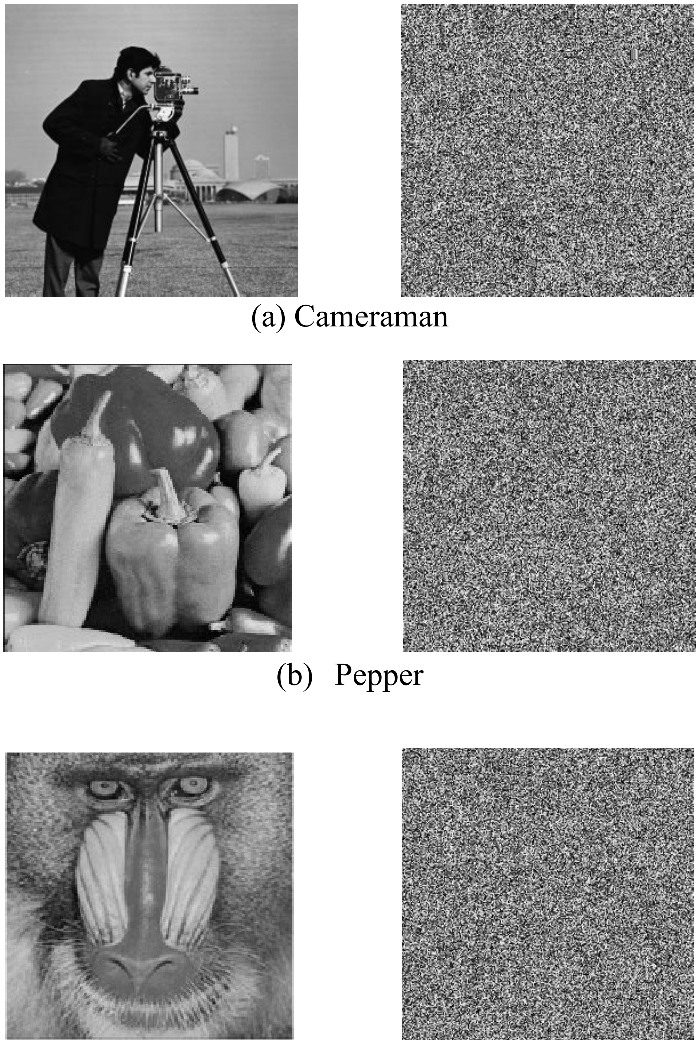
v. An S-box with a low LP score is less susceptible to linear cryptanalysis. Our S-box has an LP score of 0.0703, which is lower or equal to the LP values of the S-boxes listed in Table 12.

Table 12. Performance comparison of various analyses among different S-boxes.

S-box	Nonlinearity			SAC	BIC-SAC	BIC-NL	DU	LP
	min	max	mean					
Initial S-box	98	108	104.50	0.5037	0.5005	104.5	12	0.125
Suggested S-box	110	112	111.75	0.5007	0.4996	111.5	6	0.0703
Ref [9]	106	108	107.50	0.4944	0.4982	104.35	10	0.1250
Ref [44]	106	108	106.25	0.5112	0.4975	103.93	12	0.1484
Ref [45]	106	110	106.5	0.5010	0.4987	103.93	10	0.125
Ref [46]	106	108	107	0.4949	0.5019	102.29	12	0.141
Ref [47]	106	110	108.5	0.4995	0.5011	103.85	10	0.109
Ref [48]	108	110	109.75	0.5042	0.4987	110.6	6	0.0859
Ref [49]	102	110	106.5	0.4943	0.5019	103.35	12	0.1468
Ref [50]	104	108	105.5	0.5065	0.5031	103.57	10	0.1328
Ref [51]	104	110	107	0.4993	0,5050	103.29	10	0.1328
Ref [52]	102	112	108	0.5029	0.5020	104.43	14	0.1328
Ref [53]	102	108	105	0.5063	0.5002	104.07	10	0.1328
Ref [54]	110	112	111	0.5017	0.5018	111.43	6	0.0703
Ref [55]	108	110	109.75	0.4998	0.5041	104.14	10	0.1171
Ref [56]	108	110	109.50	0.4985	0.5012	104.07	10	0.1328
Ref [57]	104	110	106.50	0.4995	0.4983	104.57	10	0.1171
Ref [58]	108	110	108.5	0.491	0.5048	103.78	10	0.0791
Ref [59]	100	106	103.20	0.5048	0.5009	103.70	10	0.1289
Ref [60]	104	110	106.75	0.4995	0.5043	105.07	12	0.1289

5. Majority logic criterion for encryption analysis

The MLC [61] comprises a collection of evaluations, including contrast, correlation, energy, homogeneity, and entropy. The results of these tests assist in selecting the most suitable S-box for the encryption procedure. We assess the statistical competence of the S-box using the MLC for various encryption techniques. The available research describes a variety of statistical and analytical techniques for determining the S-box's potential to generate perplexity. Since the encryption process distorts the image, it is essential to understand the impact of statistical characteristics. A correlation test examines the relationship between plaintext and ciphertext. The entropy value depicts the level of randomness in the ciphertext image. Contrast analysis evaluates the brightness loss in the plaintext image during the encryption process. We can examine more features of the ciphertext by employing homogeneity and energy analyses. In light of the significance of these analyses' findings, we used our S-boxes to encrypt plaintext images and conduct MLC tests. For this purpose, we choose three 256×256 grayscale images of pepper, cameraman, and baboon. Figure 6 displays all images before and after encryption, while Table 13 lists the MLC outcomes. These results demonstrate the effectiveness of the developed substitution box for image encryption and its strong cryptographic attributes, making it suitable for use in safe data transmission algorithms.



(a) Cameraman

(b) Pepper

(c) Baboon

Figure 6. Original and encrypted images.

Table 13. MLC results.

Images	Entropy	Contrast	Correlation	Energy	Homogeneity
Cameraman Image					
Before	7.1025	0.4785	0.9292	0.1679	0.8964
After	7.9973	8.5093	- 0.0031	0.0161	0.3923
Pepper Image					
Before	7.5498	0.2668	0.9365	0.1477	0.9191
After	7.9958	8.4812	0.0003	0.0121	0.4032
Baboon Image					
Before	7.1273	0.7179	0.6782	0.1025	0.7669
After	7.9821	8.4728	- 0.0015	0.0160	0.4021

5.1. Differential analysis

The two major criteria, unified average changing intensity (UACI) and number of pixel change rate (NPCR), are employed for quantifying the impact of a single pixel alteration on the image encoded. The disparity in pixel counts between the two encoded images is quantified by NPCR, while the mean intensity variance is assessed using UACI. The pixel variation between two initial images is merely one, with their associated encoded images represented as $C_1(i,j)$ and $C_2(i,j)$. The NPCR and UACI scores are computed using the subsequent equations:

$$NCPR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\%,$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\%,$$

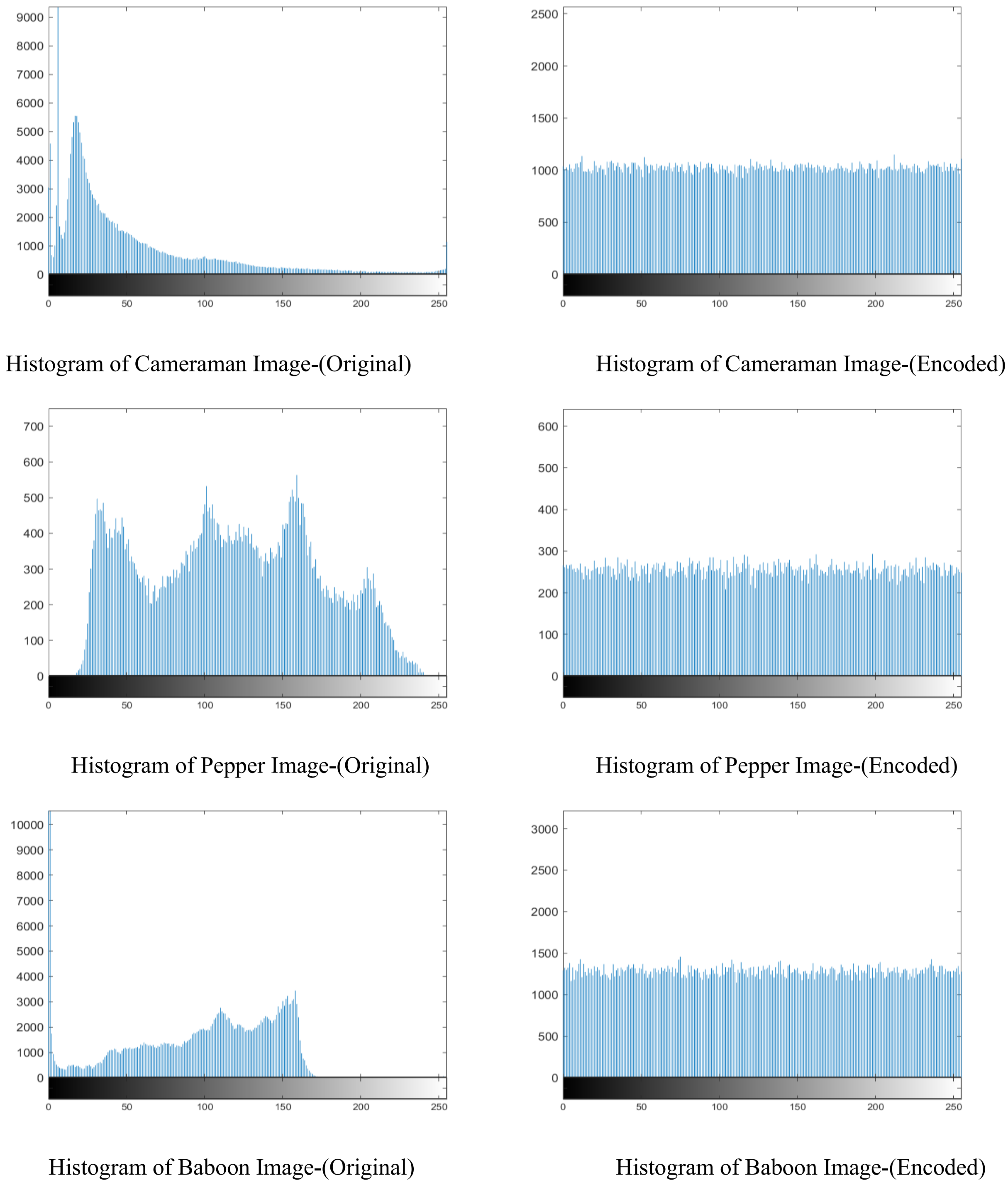
where $D(i,j) = 0$ if $C_1(i,j) = C_2(i,j)$ and otherwise, $D(i,j) = 1$. Also, M is the width and N is the height of the image. Table 14 provides a comprehensive overview of the UACI and NPCR measures.

Table 14. NPCR and UACI analysis of pixel sensitivity for different encoded images.

Image	*NCPR %*	*UACI %*
Cameraman	*99.4162*	*33.6588*
Pepper	*99.8957*	*33.5347*
Baboon	*99.1607*	*33.4927*

5.2. Histogram analysis

Histograms depict the distribution of pixel grey level intensities within an image. A cryptanalyst may employ the provided information to execute histogram attacks if the distribution is non-uniform. Nonetheless, the methodology has been developed to withstand histogram assaults, rendering data unidentifiable if the histogram is homogeneous and flattening. Analysing the histograms of the encoded and initial images reveals the disparities in colour intensity between the two. We performed analyses on the histograms of both the original and encrypted images and discovered that the histogram distribution of the encoded image, produced using the suggested S-box, markedly diverges from that of the original image. Figure 7 displays the histograms of both the original and encrypted versions of selected photographs for encryption. The histogram of the encrypted image has a notably uniform distribution, validating the efficacy of the suggested technique.



Histogram of Cameraman Image-(Original)

Histogram of Cameraman Image-(Encoded)

Histogram of Pepper Image-(Original)

Histogram of Pepper Image-(Encoded)

Histogram of Baboon Image-(Original)

Histogram of Baboon Image-(Encoded)

Figure 7. Histogram analysis.

6. Conclusions

This article introduces an innovative strategy for constructing dependable and resilient S-boxes with significant nonlinearity. The approach investigates the principles underlying coset graphs, which are derived from $GF_1(2^7) \cup \{\infty\}$ and $GF_2(2^7) \cup \{\infty\}$, in addition to a particular type of column rearrangement. Initially, an S-box is constructed by selecting vertices from two distinct graphs and placing them at predetermined positions within the matrix representing the S-box. Following the initial step, a permutation group of large order enhances the robustness of the initial S-box, ensuring its resistance against various cryptanalytic attacks. An execution evaluation quantifies the cryptographic quality of the generated S-box. When comparing the created S-box with other contemporary S-boxes, we see that it possesses superior cryptographic characteristics. Furthermore, the constructed S-box is applied to encrypt digital images, and the results obtained through the MLC indicate that the encrypted content exhibits favorable encryption quality. Therefore, the utilization of the suggested S-box in the domain of image encryption indicates its appropriateness for safeguarding data while it is being transmitted over an unsecured channel.

Author contributions

Abdul Razaq: Conceptualized the study, developed the methodology, and conducted the analysis. Muhammad Mahboob Ahsan: contributed to algorithm development, data validation, and manuscript review. Hanan Alolaiyan: Secured funding, contributed to mathematical framework design, and assisted in interpretation and revisions. Musheer Ahmad: Reviewed literature, linked to prior research, and refined the manuscript. Qin Xin: Supported computational implementation and manuscript preparation. All authors reviewed and approved the published version.

Use of Generative-AI tools declaration

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

Acknowledgments

This project is supported by the Researchers Supporting Project Number (RSP2024R317) King Saud University, Riyadh, Saudi Arabia.

Conflicts of interest

The authors declare no conflict of interest.

References

1. M. Zhang, Y. Zhang, Q. Cen, S. Wu, Deep learning-based resource allocation for secure transmission in a non-orthogonal multiple access network, *Int. J. Distr. Sensor Net.*, **18** (2022), 1975857866. https://doi.org/10.1177/15501329221104330
2. B. Bi, D. Huang, B. Mi, Z. Deng, H. Pan. Efficient LBS security-preserving based on NTRU oblivious transfer, *Wireless Pers. Commun.*, **108** (2019), 2663–2674. https://doi.org/10.1007/s11277-019-06544-2
3. R. Bhanot, R. Hans, A review and comparative analysis of various encryption algorithms, *Int. J. Secur. Its Appl.*, **9** (2015), 289–306. http://dx.doi.org/10.14257/ijsia.2015.9.4.27
4. G. Sun, Y. Li, D. Liao, V. Chang, Service function chain orchestration across multiple domains: A full mesh aggregation approach, *IEEE Tran. Network Service Manag.*, **15** (2018), 1175–1191. https://doi.org/10.1109/TNSM.2018.2861717
5. J. Daemen, V. Rijmen, *The design of Rijndael*, New York: Springer-verlag, 2002. https://doi.org/10.1007/978-3-662-04722-4
6. E. Biham, A. Shamir, Differential cryptanalysis of DES-like cryptosystems, *J. CRYPTOLOGY*, **4** (1991), 3–72. https://doi.org/10.1007/BF00630563

7. C. E. Shannon, Communication theory of secrecy systems, *Bell Syst. Technical J.*, **28** (1949), 656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x
8. N. Siddiqui, A. Naseer, M. Ehatisham-ul-Haq, A novel scheme of substitution-box design based on modified Pascal's triangle and elliptic curve, *Wirel. Personal Commun.*, **116** (2021), 3015–3030. https://doi.org/10.1007/s11277-020-07832-y
9. H. A. Ahmed, M. F. Zolkipli, M. Ahmad, A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map, *Neural Comput. Appl.*, **31** (2019), 7201–7210. https://doi.org/10.1007/s00521-018-3557-3
10. F. Masood, J. Masood, L. Zhang, S. S. Jamal, W. Boulila, S. U. Rehman, et al., A new color image encryption technique using DNA computing and Chaos-based substitution box, *Soft Comput.*, **26** (2022), 7461–7477. https://doi.org/10.1007/s00500-021-06459-w
11. A. Razaq, M. Ahmad, A. Yousaf, M. Alawida, A. Ullah, U. Shuaib, A group theoretic construction of large number of AES-like substitution-boxes, *Wirel. Personal Commun.*, **122** (2022), 2057–2080. https://doi.org/10.1007/s11277-021-08981-4
12. A. Razaq, S. Akhter, A. Yousaf, U. Shuaib, M. Ahmad, A group theoretic construction of highly nonlinear substitution box and its applications in image encryption, *Multi. Tools Appl.*, **81** (2022), 4163–4184. https://doi.org/10.1007/s11042-021-11635-z
13. F. Gonzalez, R. Soto, B. Crawford, Stochastic fractal search algorithm improved with opposition-based learning for solving the substitution box design problem, *Mathematics*, **10** (2022), 2172. https://doi.org/10.3390/math10132172
14. F. Artuğer, F. Özkaynak, SBOX-CGA: Substitution box generator based on chaos and genetic algorithm, *Neural. Comput. App.*, **34** (2022), 20203–20211. https://doi.org/10.1007/s00521-022-07589-4
15. M. S. Fadhil, A. K. Farhan, M. N. Fadhil, Designing substitution box based on the 1D logistic map chaotic system, *IOP Conf. Series: Mater. Sci. Eng.*, **1076** (2021), 012041. https://doi.org/10.1088/1757-899X/1076/1/012041
16. A. Razaq, Iqra, M. Ahmad, M. A. Yousaf, S. Masood, A novel finite rings based algebraic scheme of evolving secure S-boxes for images encryption, *Mult. Tools Appl.*, **80** (2021), 20191–20215. https://doi.org/10.1007/s11042-021-10587-8
17. I. Ullah, N. A. Azam, U. Hayat, Efficient and secure substitution box and random number generators over Mordell elliptic curves, *J. Infor. Security Appl.*, ***56*** (2021), 102619. https://doi.org/10.1016/j.jisa.2020.102619
18. Z. Hua, J. Li, Y. Chen, S. Yi, Design and application of an S-box using complete Latin square, *Nonlinear Dyn.,* **104** (2021), 807–825. https://doi.org/10.1007/s11071-021-06308-3
19. A. A. A. El-Latif, J. Ramadoss, B. Abd-El-Atty, H. S. Khalifa, F. Nazarimehr, A novel chaos-based cryptography algorithm and its performance analysis, *Mathematics*, **10** (2022), 2434. https://doi.org/10.3390/math10142434
20. A. Razaq, G. Alhamzi, S. Abbas, M. Ahmad, A. Razzaque, Secure communication through reliable S-box design: A proposed approach using coset graphs and matrix operations, *Heliyon*, **9** (2023). https://doi.org/10.1016/j.heliyon.2023.e15902
21. M. A. Khan, A. Ali, V. Jeoti, S. Manzoor, A chaos-based substitution box (S-Box) design with improved differential approximation probability (DP), *Iran. J. Sci. Technol. Trans. Electr. Eng.*, **42** (2018), 219–238. https://doi.org/10.1007/s40998-018-0061-9
22. F. Artuğer, F. Özkaynak, A novel method for performance improvement of chaos-based substitution boxes, *Symmetry*, **12** (2020), 571. https://doi.org/10.3390/sym12040571
23. A. Freyre-Echevarría, A. Alanezi, I. Martínez-Díaz, M. Ahmad, A. A. A. Abd El-Latif, H.

Kolivand, et al., An external parameter independent novel cost function for evolving bijective substitution-boxes, *Symmetry*, **12** (2020), 1896. https://doi.org/10.3390/sym12111896

24. L. Chu, Y. Su, X. Zan, W. Lin, X. Yao, P. Xu, et al, A deniable encryption method for modulation-based DNA storage, *Interdisciplinary Sciences: Comput. Life Sci.*, **16** (2024), 872–881. https://doi.org/10.1007/s12539-024-00648-5
25. X. Yao, R. Xie, X. Zan, Y. Su, P. Xu, W. Liu, A novel image encryption scheme for DNA storage systems based on DNA hybridization and gene mutation, *Inter. Sci. Comput. Life Sci.*, **15** (2023), 419–432. https://doi.org/10.1007/s12539-023-00565-z
26. S. Gao, R. Wu, X. Wang, J. Liu, Q. Li, X. Tang, EFR-CSTP: Encryption for face recognition based on the chaos and semi-tensor product theory, *Inf. Sci.*, **621** (2023), 766–781. https://doi.org/10.1016/j.ins.2022.11.121
27. S. Gao, H. H. C. Iu, J. Mou, U. Erkan, J. Liu, R. Wu, et al., Temporal action segmentation for video encryption, *Chaos Solitons Fract.*, **183** (2024), 114958. https://doi.org/10.1016/j.chaos.2024.114958
28. S. Gao, H. H. C. Iu, M. Wang, D. Jiang, A. A. Abd El-Latif, R. Wu, et al., Design, hardware implementation, and application in video encryption of the 2-D memristive cubic map, *IEEE Int. Things J.*, **11** (2024), 21807–21815. https://doi.org/10.1109/JIOT.2024.3376572
29. C. Fan, Q. Ding, A universal method for constructing non-degenerate hyperchaotic systems with any desired number of positive Lyapunov exponents, *Chaos Solitons Fract.*, **161** (2022), 112323. https://doi.org/10.1016/j.chaos.2022.112323
30. S. Gao, J. Liu, H. H. C. Iu, U. Erkan, S. Zhou, R. Wu, et al., Development of a video encryption algorithm for critical areas using 2D extended Schaffer function map and neural networks, *Appl. Math. Model.*, **134** (2024), 520–537. https://doi.org/10.1016/j.apm.2024.06.016
31. M. Wang, X. Fu, L. Teng, X. Yan, Z. Xia, P. Liu, A new 2D-HELS hyperchaotic map and its application on image encryption using RNA operation and dynamic confusion, *Chaos Solitons Fract.*, **183** (2024), 114959. https://doi.org/10.1016/j.chaos.2024.114959
32. J. A. Gallian, *Contemporary abstract algebra*, Chapman and Hall/CRC, 2021. https://doi.org/10.1201/9781003142331
33. Q. Mushtaq, A. Razaq, Homomorphic images of circuits in PSL(2,Z)-space, *Bull. Malaysian Math. Sci. Society,* **40** (2017), 1115–1133. https://doi.org/10.1007/s40840-016-0357-8
34. Q. Mushtaq, A. Razaq, A. Yousaf, On contraction of vertices of the circuits in coset diagrams for PSL(2, Z), *Proc. Math. Sci.*, **129** (2019), 1–26. https://doi.org/10.1007/s12044-018-0450-z
35. M. Conder, Three-relator quotients of the modular group, *Quart. J. Math.*, **38** (1987), 427–447. https://doi.org/10.1093/qmath/38.4.427
36. G. A. Jones, Maximal subgroups of the modular and other groups, *J. Group Theory*, **22** (2019), 277–296. https://doi.org/10.1515/jgth-2018-0144
37. A. Razaq, Q. Mushtaq, A. Yousaf, The number of circuits of length 4 in PSL(2, $\mathbb{Z}$)-space, *Commun. Algebra*, **46** (2018), 5136–5145. https://doi.org/10.1080/00927872.2018.1461880
38. I. Hussain, T. Shah, H. Mahmood, A projective general linear group based algorithm for the construction of substitution box for block ciphers, *Neural. Comput. Appl.*, **22** (2013), 1085–1093. https://doi.org/10.1007/s00521-012-0870-0
39. A. Altaleb, M. S. Saeed, I. Hussain, M. Aslam, An algorithm for the construction of substitution box for block ciphers based on projective general linear group, *AIP Adv.*, **7** (2017), 035116. https://doi.org/10.1063/1.4978264
40. S. Farwa, T. Shah, L. Idrees, A highly nonlinear S-box based on a fractional linear transformation, *Spr. Plus*, **5** (2016), 1658. https://doi.org/10.1186/s40064-016-3298-7

41. J. Pieprzyk, G. Finkelstein, Towards effective nonlinear cryptosystem design, *IEE Proc. E-Comput. Digital Tech.*, **135** (1988), 325–335. https://doi.org/10.1049/ip-e.1988.0044
42. A. F. Webster, S. E. Tavares, On the Design of S-boxes, *Adv. Crypt.-CRYPTO'85 Proc.*, **218** (1986). https://doi.org/10.1007/3-540-39799-X_41
43. M. Matsui, Linear Cryptanalysis Method for DES Cipher, *Adv. Crypt.-EUROCRYPT'93*, **765** (1994). https://doi.org/10.1007/3-540-48285-7_33
44. U. Hayat, N. A. Azam, H. R. Gallegos-Ruiz, S. Naz, L. Batool, A truly dynamic substitution box generator for block ciphers based on elliptic curves over finite rings, *Arabian J. Sci. Engineer.*, **46** (2021), 1–13. https://doi.org/10.1007/s13369-021-05666-9
45. S. Ibrahim, A. M. Abbas, Efficient key-dependent dynamic S-boxes based on permutated elliptic curves, *Inf. Sci.*, **558** (2021), 246–264. https://doi.org/10.1016/j.ins.2021.01.014
46. B. M. Alshammari, R. Guesmi, T. Guesmi, H. Alsaif, A. Alzamil, Implementing a symmetric lightweight cryptosystem in highly constrained IoT devices by using a chaotic S-box, *Symmetry*, **13** (2021), 129. https://doi.org/10.3390/sym13010129
47. H. S. Alhadawi, M. A. Majid, D. Lambić, M. A. Ahmad, A novel method of S-box design based on discrete chaotic maps and cuckoo search algorithm, *Multimed. Tools Appl.,* **80** (2021), 7333–7350. https://doi.org/10.1007/s11042-020-10048-8
48. M. Long, L. Wang, S-box design based on discrete chaotic map and improved artificial bee colony algorithm, *IEEE Access*, **9** (2021), 86144–86154. https://doi.org/10.1109/ACCESS.2021.3069965
49. R. Soto, B. Crawford, F. G. Molina, R. Olivares, Human Behaviour based optimization supported with self-organizing maps for solving the S-box design problem, *IEEE Access*, **9** (2021), 84605–84618. https://doi.org/10.1109/ACCESS.2021.3087139
50. W. Yan, Q. Ding, A novel S-box dynamic design based on nonlinear-transform of 1D chaotic maps, *Electronics*, **10** (2021), 1313. https://doi.org/10.3390/electronics10111313
51. P. Zhou, J. Du, K. Zhou, S. Wei, 2D mixed pseudo-random coupling PS map lattice and its application in S-box generation, *Nonlinear Dyn.*, **103** (2021), 1151–1166. https://doi.org/10.1007/s11071-020-06098-0
52. S. S. Jamal, M. M. Hazzazi, M. F. Khan, Z. Bassfar, A. Aljaedi, Z. ul Islam, Region of interest-based medical image encryption technique based on chaotic S-boxes, *Expert Syst. Appl.*, **238** (2024), 122030. https://doi.org/10.1016/j.eswa.2023.122030
53. M. Wang, H. Liu, M. Zhao, Construction of a non-degeneracy 3D chaotic map and application to image encryption with keyed S-box, *Mult. Tools Appl.*, **82** (2023), 34541–34563. https://doi.org/10.1007/s11042-023-14988-9
54. A. Razaq, L. A. Maghrabi, M. Ahmad, Q. H. Naith, Novel substitution-box generation using group theory for secure medical image encryption in E-healthcare, *AIMS Math.,* **9** (2024), 6207–6237. https://doi.org/10.3934/math.2024303
55. K. Z. Zamli, F. Din, H. S. Alhadawi, Exploring a Q-learning-based chaotic naked mole rat algorithm for S-box construction and optimization, *Neural Comput. Appl.*, **35** (2023), 10449–10471. https://doi.org/10.1007/s00521-023-08243-3
56. A. A. Alzaidi, M. Ahmad, H. S. Ahmed, E. A. Solami, Sine-cosine optimization-based bijective substitution-boxes construction using enhanced dynamics of chaotic map, *Complexity*, **2018** (2018), 1–16. https://doi.org/10.1155/2018/9389065
57. T. Farah, R. Rhouma, S. Belghith, A novel method for designing S-box based on chaotic map and teaching-learning-based optimization, *Nonlinear Dyn.*, **88** (2017), 1059–1074. https://doi.org/10.1007/s11071-016-3295-y

58. H.S. Alhadawi, D. Lambic, M.F. Zolkipli, M. Ahmad, Globalized firefly algorithm and chaos for designing substitution box, *J. Inf. Security Appl.*, **55** (2020), 1–13. https://doi.org/10.1016/j.jisa.2020.102671
59. F. Özkanak, A.B. Özer, A method for designing strong S-boxes based on chaotic Lorenz system, *Phys. Lett. A*, **374** (2010), 3733–3738. https://doi.org/10.1016/j.physleta.2010.07.019
60. Y. Aydin, A. M. Garipcan, F. Özkaynak, A novel secure S-box design methodology based on FPGA and SHA-256 hash algorithm for block cipher algorithms, *Arab. J. Sci. Eng.*, 2024, 1–14. https://doi.org/10.1007/s13369-024-09251-8
61. I. Hussain, T. Shah, M.A. Gondal, H. Mahmood, Generalized majority logic criterion to analyze the statistical strength of S-boxes, *Z Nat. A*, **67** (2012), 282–288. https://doi.org/10.5560/zna.2012-0022

Supplementary

Illustration and implementation details of step III in S-box construction

In Step III of the proposed method for S-box construction, we demonstrate that applying the element $a^{83}b^{5}c^{13}d^{4}$ to the initial S-box transforms it into the proposed S-box, achieving a nonlinearity score of 111.75. Here, we illustrate the process in detail. Since

a = *(1,148,162,24,38,93,152,90,78,41,13,54,35,213,155,89,98,127,192,211,5,200,186,117,96,99,206,37,208,229,250,109,203,181,25,107,170,246,14,33,207,150,77,232,97,240,112,231,95,122,182,252,74,57,66,129,6,30,102,251,23,165,9,151,120,134,234,55,12,184,256,239,218,188,244,104,216,233,131,177,224,164,214,220,195,15,42,227,48,198,50,31,118,83,156,88,147,124,179,47,80,222,19,111,169,32,221,76,139,67,140,245,110,238,132,103,56,115,63,125,161,201,86,194,167,20,39,199,128,79,126,84,176,226,58,119,28,49),*

b = *(2,149,65,193,73,52,8),*

c = *(3,105,254,43,174,175,166,137,7,17,173,133,145,144,51,138,75,249,253,64,146,197,159,235,191, 121,94,172,114,236,142,160,92,189,85,255,153,27,196,141,91,100,187,185,204,183,180,59,26,68,53, 219,36,82,71,178,60,10,157,11,44,4,248,45,202,106,61,72,243,168,87,22,225,136,101,163,209,21,190,116,247,158,242,18,70,210,143,212,113,205,237,62,130,29,228,123,40,215,171,108,154,46,16),*

d =*(34,81,241,230,69,217,135).*

Therefore,

$a^{83}b^{5}c^{13}d^{4}$=*(1,220,208,245,6,148,195,229,110,30,162,15,250,238,102,24,42,109,132,251,38,227,203, 103,23,93,48,181,56,165,152,198,25,115,9,90,50,107,63,151,78,31,170,125,120,41,118,246,161,134, 13,83,14,201,234,54,156,33,86,55,35,88,207,194,12,213,147,150,167,184,155,124,77,20,256,89,179, 232,39,239,98,47,97,199,218,127,80,240,128,188,192,222,112,79,244,211,19,231,126,104,5,111,95, 84,216,200,169,122,176,233,186,32,182,226,131,117,221,252,58,177,96,76,74,119,224,99,139,57,28,164,206,67,66,49,214,37,140,129)(2,52,193,149,8,73,65)(3,144,94,141,36,106,190,62,105,51,172, 91,82,61,116,130,254,138,114,100,71,72,247,29,43,75,236,187,178,243,158,228,174,249,142,185,60,168,242,123,175,253,160,204,10,87,18,40,166,64,92,183,157,22,70,215,137,146,189,180,11,225,210,171,7,197,85,59,44,136,143,108,17,159,255,26,4,101,212,154,173,235,153,68,248,163,113,46,133,191,27,53,45,209,205,16,145,121,196,219,202,21,237)(34,69,81,217,241,135,230).*

To illustrate the application of $a^{83}b^{5}c^{13}d^{4}$ on the initial S-box to generate the proposed S-box:

- Since 1 is mapped to 220, we shift the 1st element (which is 2) of the initial S-box to the 220th position (12th element in the 14th row, calculated as 13 × 16 + 12 = 220).

- Since 220 is mapped to 208, we shift the 220th element (207) of the initial S-box to the 208th position (16th element in the 13th row, 13 × 16 = 208).
- Continuing in this manner, 135 is mapped to 230, so we shift the 135th element (59) to the 230th position (6th element in the 15th row, 14 × 16 + 6 = 230).
- Finally, since 230 is mapped to 34, we shift the 230th element (86) to the 34th position (2nd element in the 3rd row, 2 × 16 + 2 = 34).

This iterative process continues for all mappings, thereby constructing the proposed S-box.