Mathematics

AIMS Mathematics, 8(9): 21644–21667.
DOI: 10.3934/math.20231104
Received: 24 February 2023
Revised: 26 May 2023
Accepted: 06 June 2023
Published: 07 July 2023

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

Research article

Construction of nonlinear component of block cipher using coset graph

Adil Waheed[1], Fazli Subhan[1,2], Mazliham Mohd Suud[2,*], Muhammad Yasir Hayat Malik[3], Alina Mirza[4] and Farkhanda Afzal[4]

[1] Faculty of Engineering and Computer Sciences, National University of Modern Languages, Islamabad, Pakistan
[2] Faculty of Computer and Information, Multimedia University, Cyberjaya, Malaysia
[3] Department of Mathematics, Division of Science and Technology, University of Education Lahore, Lahore, Pakistan
[4] MCS, National University of Sciences and Technology, Islamabad, Pakistan

* **Correspondence:** Email: mazliham@mmu.edu.my.

Abstract: In recent times, the research community has shown interest in information security due to the increasing usage of internet-based mobile and web applications. This research presents a novel approach to constructing the nonlinear component or Substitution Box (S-box) of block ciphers by employing coset graphs over the Galois field. Cryptographic techniques are employed to enhance data security and address current security concerns and obstacles with ease. Nonlinear component is a keystone of cryptography that hides the association between plaintext and cipher-text. Cryptographic strength of nonlinear component is directly proportional to the data security provided by the cipher. This research aims to develop a novel approach for construction of dynamic S-boxes or nonlinear components by employing special linear group $PSL(2,\mathbb{Z})$ over the Galois Field $GF(2^{10})$. The vertices of coset diagram belong to $GF(2^{10})$ and can be expressed as powers of α, where α represents the root of an irreducible polynomial $p(x) = x^{10} + x^3 + 1$. We constructed several nonlinear components by using $GF^*(2^{10})$. Furthermore, we have introduced an exceptionally effective algorithm for optimizing nonlinearity, which significantly enhances the cryptographic properties of the nonlinear component. This algorithm leverages advanced techniques to systematically search for and select optimal S-box designs that exhibit improved resistance against various cryptographic attacks.

Keywords: S-box; nonlinear component; coset diagram; nonlinearity; security; block cipher
Mathematics Subject Classification: 05C10, 08A72

1. Introduction

Cryptology has two basic areas, cryptography and cryptanalysis, in which one sees the sketch and cracking of cryptosystems. While re-arranging a cryptosystem, the inspection of its security plays a vital role. In cryptosystems, the key role features are confidentiality, authentication and integrity of data [1]. Earlier in [2], cryptosystems have been handled by armed forces. In this century, now-a-days everyone wants a fully controlled security by means of cryptographic skills.

The day-by-day advancements in growing industries, particularly the communication sector, have facilitated the flow of huge data over vast areas within a short period of time. It has been a hot topic to discuss how modern security systems can be improved to allow for reliable data communication. The cryptographic roots are adopted to understand secure contact between credible parties for giving hidden data in a protected way [3]. The block ciphering method has been widely used for protected communication and storage in the last era. Cryptographers constructed advanced block ciphers to go with the pace of the modern era [4]. Shanon in 1949 gave the idea to deal with confusion and diffusion occurring in block ciphers via substitution boxes [5]. Well known block ciphers are AES, DES, RC4, Blowfish, IDEA, RC5, RC6, and many more. These ciphers actuate the nonlinearity for secure information. S-boxes play a vital role in communicating the information/data more securely. Nonlinear components are an integral part of encryption algorithms and play a crucial role in achieving confusion and nonlinearity, which are necessary for thwarting attacks and protecting sensitive data. The motivation for studying nonlinear components also includes the field of image encryption [6] owing to the growing reliance on digital images for storage, transmission, and communication of sensitive information. Overall, the motivation behind nonlinear component is to improve cryptographic systems and address the particular difficulties associated with image encryption, ultimately advancing secure communication, data security, and privacy preservation.

Mathematically, an $n \times m$ S-box does not follow a linear mapping. This indicates that confusion component has a nonlinear mapping S from Galois field $GF(2^n) \to GF(2^m)$, where $n \geq m$. It works as a Boolean function, which is comprised of bits. The effectiveness of an 8×8 S-box in encrypting data is so high that it catches the attention of cryptographers as a strong encryption method [7,8]. Numerous design patterns are examined and they came with an 8×8 S-box with best cryptographic features. The methods involved in the construction process are: Mobius transformation, linear trigonometric transformation, complete latin square, Bent function and logistic chaotic system, affine transformation, and symmetric group composition. S-boxes are also playing vital role in image encryption algorithms. Liu et al. [9] introduced a color image encryption scheme based on chaos, emphasizing the utilization of a randomly sampled noise signal. In [10], the encryption scheme involves creating six pseudo-random arrays to cyclically shift the red, green- and blue components both horizontally and vertically, followed by using the exclusive OR (XOR) operation to diffuse three color components. Liu et al. [11] proposed an image encryption scheme based on GF and chaotic systems to transmit pathological images over the network.

Currently, there is a lot of focus on robust S-box construction methods, and significant research has been proposed in [12–20]. This paper [21] introduces a new system model with improved chaotic characteristics by suggesting a piece-wise quadratic polynomial chaotic map that operates in one dimension (1D). In [12], a projective general linear group is used as a method of construction for the S-box for block ciphers. Islam et al. [13] explores the construction of an S-box with four dimensions and four wings hyperchaotic system. Husain et al. [14] designed cryptographically a strong nonlinear component based on a particular class of linear fractional transformation. Ahmad et al. [15] presents a technique, which involves utilizing chaotic maps and artificial intelligence based methodology.

Attaullah et al. [16] employed algebraic techniques to create the S-box. Özkaynak et al. [17] derived the S-box from a fractional order chaotic Chen system. In [22] a new S-box for encryption that utilizes the Lorenz equation was presented. In [23], a combination of chaotic maps is used to develop the S-box by improving chaotic range. Zheng et. al [24] outlines a dynamic S-box dependent image encryption method, comprising of four stages: creating encryption keys, S-box construction, image permutation, and image diffusion. In [25], the authors introduced a technique for constructing an S-box that fulfills the strict avalanche criterion. This paper [26] puts forth a three-layer optimization technique for producing high-performance S-boxes using a novel chaotic map and artificial jellyfish optimization algorithm.

A new method for creating S-boxes using coset diagrams and a one-to-one mapping was developed by Razaq et al. [18] in their study. Si et. al [27] created a chaotic map with an exponential quadratic function in two dimensions that has the capability to function as a generator of pseudo-random numbers. The article [19] describes the design of a confusion component using tangent delay chaotic sequence and a special kind of permutation from a symmetric group. Liu [20] et al. presented a technique in which S-box elements are shuffled randomly using a permutation operation performed between independent chaotic sequences. Liu et al. [28] gave an image encryption algorithm for the new dynamic S-box. In [29], confusion component is constructed based on linear fractional transformation using Galois field $GF(2^8)$. Farah et al. [30] used a teaching-learning based optimization to the design S-box. Jamal et al. [31] S-box construction method is controlled by a linear group over the finite commutative ring. In [32], Lambić demonstrated an efficient technique of designing a confusion component by composition method. Azam et al. [33] introduced a nonlinear component that is both cryptographically robust and injective specifically for elliptic curves. The paper [34] introduces the Q-learning naked mole rat algorithm, a new variant of the metaheuristic algorithm based on the naked mole rat, for building and optimizing substitution boxes. As opposed to the majority of competing works, which frequently include five chaotic maps (Singer, Chebyshev, logistic, circle, and sinusoidal) as a part of the algorithm itself. The key innovation and distinctive features of this paper are outlined below:

1) Our proposal presents a method for constructing S-boxes in a way that is highly efficient, by utilizing coset graph for the action of $PSL(2,\mathbb{Z})$ over the Galois field $GF(2^{10})$.
2) The proposed nonlinear components undergo a comprehensive analysis and are compared to other commonly used nonlinear components, which are generated through various algebraic structures. The purpose of this comparison is to evaluate the performance and potency of the proposed S-boxes in terms of their capacity to add nonlinearity to cryptographic systems. The findings demonstrate that the proposed nonlinear components are more efficient and resistant to algebraic attacks.

The rest of this article is structured as follows: coset diagrams for modular group background knowledge is introduced in Section 2. Use of coset diagram in the construction of the proposed nonlinear component is presented in Section 3. In Section 4, we propose a novel nonlinear enhancement algorithm for increasing the nonlinearity of any confusion component of the block cipher. Section 5 presents statistical analysis and simulation results. The results of the proposed S-box and performance analysis criteria are evaluated in the same section. Moreover, the proposed S-box construction scheme is compared with well-known S-boxes according to good S-box criteria and is also examined in the same section. In conclusion, this article presents a summary of a coset based S-box generation scheme.

2. Preliminaries

The modular group $PSL(2, \mathbb{Z})$ is comprised of a set of linear fractional mappings, which include by $l: s \to -{}^{1}/_{s}$ and $m: s \to {}^{s-1}/_{s}$. The finite display of $PSL(2, \mathbb{Z})$ is $< l, m: l^2 = m^3 = 1 >$. The modular group is the most significant infinite discrete group due to its extensive utilization in number theory, advanced group theory, geometry, and topology. There is a rich history of studying the actions of the modular group, particularly on finite sets, dating back to the late 19th century. G. Higman for the very first time used coset diagrams for this group in 1978. The coset diagrams [35–37] are derived from the way $PSL(2, \mathbb{Z})$ operates on the projective line over the finite field $GF(p^n)$, which is represented as $PL(F_{p^n}) = GF(p^n) \cup \{\infty\}$. Here, p represents a prime number. We use triangles to represents cycles of m because of order three. The elements of $GF(p^n) \cup \{\infty\}$ that form the nodes of the triangles undergo an anti-clockwise permutation by m. We utilize an edge in the coset diagram to connect a pair of nodes belonging to the triangles due to the presence of order two. The term "order two" indicates that these elements/nodes have a specific property where, when combined with themselves, they return to their original state after two repetitions. This property may be relevant or significant in the context of the coset diagram, influencing the decision to connect the corresponding nodes with an edge. Fixed points are denoted by thick dots, if they exist. Consider the action of a modular group on $GF(23) \cup \{\infty\} = \{0, 1, 2, 3, \dots, 22, \infty\}$. The permutation representations of l and m can be calculated by $l : s \to -{}^{1}/_{s}$ and $m : s \to {}^{s-1}/_{s}$.

$$l : (0\ \infty)(1\ 22)(2\ 11)(3\ 15)(4\ 17)(5\ 9)(6\ 19)(7\ 13)(8\ 20)(10\ 16)(12\ 21)(14\ 18)$$

$$m : (0\ \infty\ 1)(2\ 12\ 22)(3\ 16\ 11)(4\ 18\ 15)(5\ 10\ 17)(6\ 20\ 9)(7\ 14\ 19)(8\ 21\ 13).$$

It is evident that the permutation of m results in 8 cycles, thus implying the existence of 8 triangles in the coset diagrams. The vertices 2, 12 and 22 of a triangle corresponds to a cycle $(2\ 12\ 22)$. So 8 triangles can be drawn. Subsequently, we connect these triangles by permuting l. For example, the cycle (1 22) in l mean the nodes 1 and 22 are connected by an edge. The following coset diagram appears as a result of permutations of l and m. Since the image of 0 under x does not exist in $GF(p^n)$, it is essential to consider alternative mappings or transformations to ensure a comprehensive coverage of the desired range. Thus, it is not feasible for $PSL(2, \mathbb{Z})$ to act in this scenario. For this, we add ∞ in $GF(p^n)$ for the action of $PSL(2, \mathbb{Z})$. The utilization of coset diagrams has led to the resolution of numerous problems in group theory. This paper explores the implementation of these coset diagrams in the field of cryptography. Here, we formulate the coset diagram that we utilized in the designing of the proposed S-boxes.

In Figure 1, coset diagram illustrates the action of the modular group on $GF(23) \cup \{\infty\}$. Aslo, Figure 1 gives a clear understanding of how modular group operates on $GF(23) \cup \{\infty\}$ and distinguish the resulting cosets.

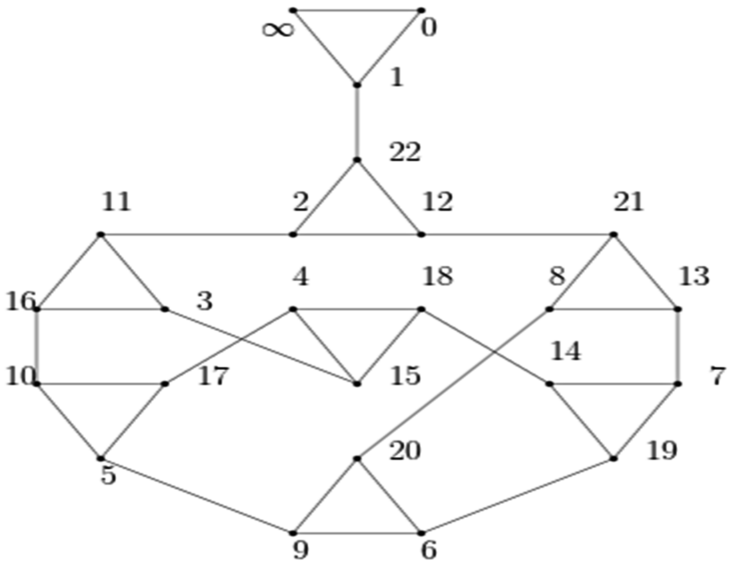

Figure 1. The coset graph for the modular group's action on GF(23) ∪ {∞}.

3. Design of proposed nonlinear component (S-box) by utilizing coset diagram

Let's examine a primitive polynomial $p(x) = x^{10} + x^3 + 1$ that cannot be factored into lower-degree polynomials over $\mathbb{Z}_2$, then $GF(2^{10}) = \frac{Z_2[x]}{<x^{10}+x^3+1>}$. In Table 1, the components of $GF(2^{10})$ are represented as certain exponents of α, where α corresponds to the root of $p(x)$. Let's examine how $PSL(2,\mathbb{Z})$ operates on $PL(F_{2^{10}}) = GF(2^{10}) \cup \infty$. The permutation representations of l and m can be computed using the operation of $(s)l = \frac{-1}{s}$ and $(s)m = \frac{s-1}{s}$. So,

$$l : (0\ \infty)(1)(\alpha^1\ \alpha^{1022})(\alpha^2\ \alpha^{1021})(\alpha^3\ \alpha^{1020})(\alpha^4\ \alpha^{1019})(\alpha^5\ \alpha^{1018})(\alpha^6\ \alpha^{1017})(\alpha^7\ \alpha^{1016})(\alpha^8\ \alpha^{1015})$$
$$(\alpha^9\ \alpha^{1014})(\alpha^{10}\ \alpha^{1013})(\alpha^{11}\ \alpha^{1012})(\alpha^{12}\ \alpha^{1011})(\alpha^{13}\ \alpha^{1010})(\alpha^{14}\ \alpha^{1009})(\alpha^{15}\ \alpha^{1008})(\alpha^{16}\ \alpha^{1007})\ \ldots$$
$$\ldots(\alpha^{497}\ \alpha^{526})(\alpha^{498}\ \alpha^{525})(\alpha^{499}\ \alpha^{524})(\alpha^{500}\ \alpha^{523})(\alpha^{501}\ \alpha^{522})(\alpha^{502}\ \alpha^{521})(\alpha^{503}\ \alpha^{520})(\alpha^{504}\ \alpha^{519})$$
$$(\alpha^{505}\ \alpha^{518})(\alpha^{506}\ \alpha^{517})(\alpha^{507}\ \alpha^{516})(\alpha^{508}\ \alpha^{515})(\alpha^{509}\ \alpha^{514})(\alpha^{510}\ \alpha^{513})(\alpha^{511}\ \alpha^{512}),$$

$$m : (\infty\ 1\ 0)(\alpha^{341})(\alpha^{682})(\alpha^1\ \alpha^{76}\ \alpha^{946})(\alpha^2\ \alpha^{152}\ \alpha^{869})(\alpha^3\ \alpha^7\ \alpha^{1013})(\alpha^4\ \alpha^{304}\ \alpha^{715})(\alpha^5\ \alpha^{508}\ \alpha^{510})$$
$$(\alpha^6\ \alpha^{14}\ \alpha^{1003})(\alpha^8\ \alpha^{608}\ \alpha^{407})(\alpha^9\ \alpha^{315}\ \alpha^{699})(\alpha^{10}\ \alpha^{1016}\ \alpha^{1020})(\alpha^{11}\ \alpha^{189}\ \alpha^{823})(\alpha^{12}\ \alpha^{28}\ \alpha^{983})\ldots$$
$$\ldots(\alpha^{599}\ \alpha^{668}\ \alpha^{779})(\alpha^{604}\ \alpha^{833}\ \alpha^{609})(\alpha^{614}\ \alpha^{807}\ \alpha^{625})(\alpha^{617}\ \alpha^{636}\ \alpha^{793})(\alpha^{620}\ \alpha^{680}\ \alpha^{746})$$
$$(\alpha^{633}\ \alpha^{760}\ \alpha^{653})(\alpha^{635}\ \alpha^{658}\ \alpha^{753})(\alpha^{641}\ \alpha^{677}\ \alpha^{728})(\alpha^{650}\ \alpha^{683}\ \alpha^{713})$$
$$(\alpha^{652}\ \alpha^{738}\ \alpha^{656})(\alpha^{654}\ \alpha^{697}\ \alpha^{695}).$$

The action's coset diagram comprises a sole instance of both π and Δ, alongside 170 iterations of γ, constituting a total of 172 orbits.

By refereeing to Figure 2, for the depiction of the orbit, labeled as γ, in the coset diagram. The operation of modular group on the set of $GF(2^{10}) \cup \{\infty\}$ is shown in this figure. By examining Figure 2, we can see interaction between the elements of modular group and the resulting orbit. From Figure 3, a deeper understanding can be obtained regarding the orbit labeled as γ_j in the coset diagram. By referencing the Figure 4, there is distinct replica of γ_j within which the vertex α^1 is located. A specific element or state within the mathematical structure is indicated by the presence of α^1 in the coset diagram.

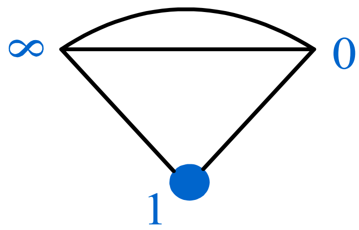

Figure 2. Representation of modular group on $GF(2^{10}) \cup \{\infty\}$ in the form of a coset diagram contains the orbit λ [18].

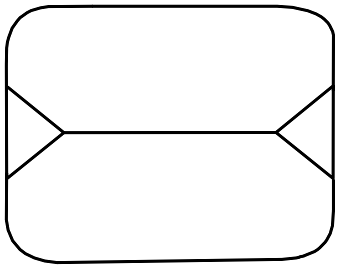

Figure 3. Representation of modular group on $GF(2^{10}) \cup \{\infty\}$ in the form of a coset diagram contains the orbit γ_j [18].

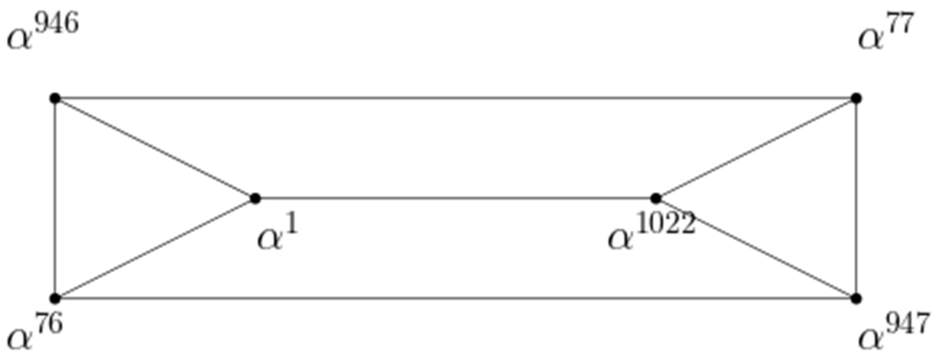

Figure 4. A copy of γ_j with the vertex α^1 in the coset diagram.

Table 1. Demonstration of the components belonging to $GF(2^{10})$.

	$GF(2^{10})$	Binary values	$GF(2^{10})$	Binary values	$GF(2^{10})$	Binary values	$GF(2^{10})$
0000000000	0	0000000001	1	0000000010	α^1	0000000100	α^2
0000001000	α^3	0000010000	α^4	0000100000	α^5	0001000000	α^6
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
0100100110	α^{1015}	1001001100	α^{1016}	0010010001	α^{1017}	0100100010	α^{1018}
1001000100	α^{1019}	0010000001	α^{1020}	0100000010	α^{1021}	1000000100	α^{1022}

Algebraic approaches have been employed in the study of Galois fields in various publications. According to this novel methodology, the nodes of the coset diagram are used to disrupt the initial sequence of the Galois field. $GF^*(2^{10})$ represents the subset of $GF(2^{10})$ consisting of elements that can be expressed as even powers of α. Our initial objective is to compose a 16×16 matrix of the coset diagram nodes. To accomplish this, we exclusively choose nodes that are a part of $GF^*(2^{10})$. The following is the procedure for constructing an S-box using a coset graph.

Table 1 provides an illustrative display showcasing the elements in binary form that are part of $GF(2^{10})$. This table enables a comprehensive examination of each element, making it possible to examine each of its elements thoroughly. Table 2 consists of 256 elements between 0 to 255 for $GF(2^8)$, each element is represented by an 8-bit binary sequence. Table 3 displays the results obtained after completing the second step of the procedure. This table present data into matrix of size 16×16, when each item in the matrix, as a result of the proposed calculations, has a remainder of zero. In Table 4, the proposed S-boxes (S-box-1) is presented in a manner where each of its elements displays a remainder of zero. Table 5 presented matrix of size 16×16, where element in the matrix has reminder of one. Table 6 shows proposed S-box-2 for remainder one after application of permutation. In Table 7, all elements of matrix have reminder two, which is obtained after completing the second step. Table 8 depicts proposed S-box-3 developed after applying process of permutation. In Table 9, elements are organized into 16×16 matrix, where all elements have remainder of three. Table 10 demonstrates proposed S-box-4, designed through the application of permutation.

Step 1. We generate a 16x16 matrix using elements from $\mathrm{GF}^*(2^{10})$ by implementing the subsequent approach, taking into account that our coset diagram encompasses 172 orbits.

We can begin by examining the orbit in the coset diagram that contains α^1. We can refer to this orbit as γ_1, and we can apply the transformation $lmlm^{-1}l$ 1 to α^1 which will result in α^{947}. While traversing through this path, we encounter $\alpha^{1022}, \alpha^{77}, \alpha^{946}$ and α^{76} before finally arriving at α^{947}. We can represent α^{76} as the final element of the first row of the 16×16 look-up table. After writing 1 node of any $\gamma_k \epsilon \{\gamma_j : j = 1, 2, 3, \ldots, 170\}$, in order to choose the next copy from γ_j, we find a node $v = \alpha^{i_1+1}$, where $\alpha^{i_1}, \alpha^{i_2}, \alpha^{i_3}, \alpha^{i_4}, \alpha^{i_5}$, and α^{i_6} are the nodes of γ_k, such that $i_1 < i_j$, where $j = 2, 3, 4, 5, 6$. If α^{i_1+1} is utilized in earlier selected copies of γ_j, then we go on to the copy of γ_j containing $v = \alpha^{i_1+2}$ and so on. For moving on to each node we apply $lmlm^{-1}l$ on v. Note that, in each γ_j, there are $0, 1, 2$ or 3 nodes belonging to $GF^*(2^{10})$ out of 6. Write these nodes in the 16×16 lookup table in a specific order. The function keeps iterating until all the 1022 nodes of γ_j are utilized. Next, select 0 from π and place it as the first element of the last row.

Step 2. Let $\mathrm{h} : \mathrm{GF}^*(2^{10}) \rightarrow \mathrm{GF}(2^8)$ be defined by the equation $\mathrm{f}(\alpha^{\mathrm{n}}) = \omega^{\frac{\mathrm{n}}{4}}$, where the elements of

F_{2^8} are represented using powers of ω as demonstrated in Table 2. The irreducible polynomial $p(x)=x^{10}+x^3+1$ over $\mathbb{Z}_2$ is used in this context. In this stage, we execute the function h on every matrix utilized in the initial step. By following this approach, we generate a 16x16 matrix consisting of elements from $GF(2^8)$. Subsequently, we transform each element of the matrix into binary form, and eventually, we convert them into decimal form. Similarly we have done the same task for remainders 1, 2 and 3. For the remainders 1, 2 and 3, we use mappings $f(\alpha^n)=\omega^{\frac{n-1}{4}}, \omega^{\frac{n-2}{4}}$ and $\omega^{\frac{n-3}{4}}$ respectively. In this way we have obtained 4 S-boxes (Tables 3, 5, 7, 9) having nonlinearity 101, 102, 103 and 104 respectively. These S-boxes are acceptable in encryption processes and are up to the mark. We increase their strength by utilizing permutations of the group S_{16} in the following manner:

Algorithm 1: Permutations of the group S_{16}.

a. In Table 4, we apply the following permutation $(1\ 7\ 4\ 3\ 16\ 13\ 8\ 14\ 11\ 10\ 12\ 9\ 15\ 6)(2\ 5)$ of the group S_{16} to generate our proposed S-box 1.
b. In Table 6, we apply the following permutation $(1)(2\ 4)(3\ 15\ 7\ 8\ 14\ 5\ 13\ 11\ 10\ 9\ 16)$ of the group S_{16} to generate our proposed S-box 2.
c. In Table 8, we apply the following permutation $(1\ 3\ 5\ 2\ 12\ 13\ 16\ 14\ 11\ 4)(6\ 10\ 8)(7\ 15)(9)$ of the group S_{16} to generate our proposed S-box 3.
d. In Table 10, we apply the following permutation $(1\ 13)(2\ 15\ 11\ 8\ 9)(3\ 5\ 7\ 14\ 12)(4)(6\ 10\ 16)$ of the group S_{16} to generate our proposed S-box 4.

Table 2. Demonstration of the components belonging to $GF(2^8)$.

Binary values	$\boldsymbol{GF(2^8)}$	**Binary values**	$\boldsymbol{GF(2^8)}$	**Binary values**	$\boldsymbol{GF(2^8)}$	**Binary values**	$\boldsymbol{GF(2^8)}$
11011000	ϖ^{251}	10101101	ϖ^{252}	01000111	ϖ^{253}	10001110	ϖ^{254}
10000011	ϖ^{247}	00011011	ϖ^{248}	00110110	ϖ^{249}	01101100	ϖ^{250}
01111101	ϖ^{243}	11111010	ϖ^{244}	11101001	ϖ^{245}	11001111	ϖ^{246}
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
10000000	ϖ^{7}	00011101	ϖ^{8}	00111010	ϖ^{9}	01110100	ϖ^{10}
00001000	ϖ^{3}	00010000	ϖ^{4}	00100000	ϖ^{5}	01000000	ϖ^{6}
00000000	0	00000001	1	00000010	ϖ^{1}	00000100	ϖ^{2}

Table 3. After the 2nd step, 16×16 square matrix having remainder zero.

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

Table 4. Proposed S-box 1 for remainder zero evolved after permutations.

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

Table 5. 16×16 matrix having remainder one evolved after 2nd step.

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

Table 6. Proposed S-box 2 for remainder one evolved after permutations.

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

Table 7. 16×16 matrix having remainder two evolved after 2nd step.

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

Table 8. Proposed S-box 3 for remainder two evolved after permutations.

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

Table 9. 16×16 matrix having remainder three evolved after 2nd step.

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

Table 10. Proposed S-box 4 for remainder three evolved after permutations.

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

4. Proposed novel nonlinearity booster algorithm

In order to increase the nonlinearity of bijective S-box (n×m), a strategy of dividing a larger S-box into smaller S-boxes makes sense. In this strategy, a larger S-box of 2048 bits is arranged into 16 smaller blocks. Each element of the S-box consists of one byte and each smaller S-box contains 128 bits. In the first round, start dividing the standard/required S-box into smaller S-boxes column wise and store them in an arrayList. In the second round, repeat the same process row wise. After that, in order to increase the nonlinearity [38], swap each smaller S-box with another S-box while keeping an eye on nonlinearity.

Algorithm 2 is introduced, which utilizes the divide and conquer approach, beginning with the development of an S-box using a coset graph.

Algorithm 2: Divide and conquer strategy based nonlinear booster algorithm.

Step1: S_1 ← the function F(n) generates bijective S-box $S_1(n\times m)$ using coset graph.
Step2: S_2 ←S_1 ∴ Here we are generating temporary copy of actual S-box,
While 1: n ∴Setting a loop that continue to execute loop body (Step 3 to 7) as long as condition holds true
Step 3: Received 16 blocks of size 4×4 ← divide the S-box (S_2) into blocks of size 128 bits.
Step4: Received updated S-box (S_2) ← Swap the one block size 4×4 with another one.
Step5: NewNL ← calculate the nonlinearity of updated S-box.
Step 6: Compare new NL with NL of actual S-box.
Stept 7: If the new nonlinearity (NL) is greater than the actual NL, make this change permanent. Otherwise, reverse the change.
end
Step 8: We will receive an S-box with improved nonlinearity.

Results after applying Algorithm 1

In Tables 11–14, we are presenting optimized S-boxes using Algorithm 2. The proposed S-boxes have a nonlinearity of 112.

Table 11. An optimized proposed S-box 1 with nonlinearity 112 using Algorithm 2.

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

Table 12. An optimized proposed S-box 2 with nonlinearity 112 using Algorithm 2.

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

Table 13. An optimized proposed S-box 3 with nonlinearity 112 using Algorithm 2.

082	123	151	201	210	152	006	176	228	158	029	118	117	213	126	081
171	248	032	121	080	120	025	012	155	194	203	156	150	084	250	061
148	038	063	089	231	141	197	095	056	107	030	221	208	161	115	053
255	185	237	193	218	241	192	196	005	235	190	128	022	187	039	251
083	175	183	130	142	062	002	238	104	099	067	143	229	047	106	108
059	239	249	073	233	180	090	091	163	174	004	114	222	068	064	100
207	140	055	027	125	102	164	216	093	076	243	060	111	145	077	230
058	186	041	028	014	070	031	189	166	212	159	088	247	000	078	253
045	160	103	219	036	157	169	065	105	177	016	245	240	001	003	127
232	136	138	252	037	225	162	168	096	137	023	110	149	008	098	195
170	009	033	215	226	011	085	050	153	021	206	191	013	094	246	242
182	042	181	113	79	179	122	224	133	204	109	217	147	040	154	057
167	205	173	178	044	043	198	112	086	046	017	024	026	072	200	132
188	139	172	071	124	209	054	010	034	075	244	116	097	101	066	007
254	214	087	052	236	165	144	131	019	146	184	051	015	223	119	220
211	049	069	020	129	234	135	092	199	227	134	202	048	018	074	035

Table 14. An optimized proposed S-box 4 with nonlinearity 112 using Algorithm 2.

150	219	211	254	098	194	033	023	198	148	225	195	012	199	224	025
131	107	015	084	160	109	203	240	034	141	166	218	209	068	003	132
065	061	044	100	137	040	181	094	035	041	011	173	079	083	247	237
016	031	054	202	167	136	077	006	182	248	170	037	221	104	253	067
081	060	215	189	080	128	097	164	039	106	184	046	186	200	208	112
056	102	146	214	229	062	233	238	116	122	156	169	021	127	070	174
178	239	180	045	117	147	246	192	222	206	129	172	118	099	171	119
213	090	130	149	216	242	126	087	022	210	110	075	052	093	236	232
140	008	051	004	145	227	074	176	228	066	095	013	059	231	076	155
036	255	201	055	157	153	159	175	103	017	113	071	001	252	187	154
005	020	250	096	125	057	092	196	197	124	072	142	163	101	204	115
028	042	029	114	089	053	193	223	027	191	135	220	226	014	120	165
134	111	241	230	139	082	235	207	058	143	019	002	177	162	190	158
188	030	151	078	185	064	088	026	108	183	018	243	212	000	050	152
105	032	179	121	091	038	069	217	048	234	024	063	144	009	123	138
161	007	073	245	010	249	043	049	168	086	133	251	085	047	244	205

5. Results from detailed statistical analysis and simulation

In order to keep security precautions in place, we have worked on cryptanalysis in this section. We ran a number of security measure tests to determine key characteristics of our proposed S-boxes. We can utilize the proposed S-boxes in various coding schemes and secure communication by examining its cryptographic properties. Our proposed S-boxes are evaluated by standard evaluation criteria including, nonlinearity, bit independence criterion (BIC), strict avalanche criterion (SAC), linear approximation probability (LP), differential approximation probability (DP), fixed point (Fp), and reverse fixed point (OFp). We examined the proposed S-box's outcomes and contrasted them with those of known S-boxes. Let's look at these tests in more detail for a better understanding.

In Table 15, average nonlinearity values of well-known S-boxes are shown, including the comparison of these values with the value of our proposed S-boxes. Table 16 presents a detailed Bit Independence Criterion analysis for the proposed S-box-4. This table provides an in-depth exploration of BIC. Table 17 depicts a comprehensive BIC comparison between our proposed S-boxes and other existing S-boxes. Table 18 shows a detailed breakdowns of the SAC for our S-boxes. Additionally, this table also includes the average SAC. Table 19 demonstrates the LP analysis for our S-boxes, along with comparison to other S-boxes. Table 20 contains comprehensive DP analysis for proposed S-box-4. Table 21 illustrates the analysis of fixed point, reverse fixed pint for our S-boxes, alongside a comparative assessment with other S-boxes.

5.1. Nonlinearity

In 1988, Pieprzyk and Finkelstein [39] introduced the term nonlinearity. The strength of the S-box is measured by this tool. It is very important in order to know the non-linear properties of the encrypted or coded material. A nonlinear component (S-box) with greater nonlinearity is generally considered more secure than one with lower nonlinearity. The mathematical formula is as follows:

$$N_k = 2^{l-1}\left(1 - 2^{-l} max\left|S_{(k)}(j)\right|\right),$$

where

$$S_{(k)}(j) = \sum_{j \in F_2^l} (-1)^\wedge (k(x) \otimes +. j.$$

The newly proposed S-boxes have an average nonlinearity value of 112. Table 15 presents a comparison between the proposed S-boxes and other robust S-boxes.

Table 15. Average nonlinearity values of robust well-known S-boxes.

S-box	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	Average
Proposed 1 (Table 11)	112	112	112	112	112	112	112	112	112
Proposed 2 (Table 12)	112	112	112	112	112	112	112	112	112
Proposed 3 (Table 13)	112	112	112	112	112	112	112	112	112
Proposed 4 (Table 14)	112	112	112	112	112	112	112	112	112
Zhu [40]	108	108	106	102	108	102	108	104	105.75
Zahid [41]	110	112	112	112	112	112	112	112	111.75
Hussain [42]	112	112	112	112	112	112	112	112	112
Gautam et al. [43]	108	106	104	98	102	102	98	74	99
Prime [44]	94	100	104	104	102	100	98	94	99.5
S_8 AES [45]	112	112	112	112	112	112	112	112	112
Xhi [46]	106	104	106	106	104	106	104	106	105
AES [47]	112	112	112	112	112	112	112	112	112
Skipjac and Kea [48]	104	108	108	108	108	104	104	106	106.75
Alkhaldi et al. [19]	108	104	106	106	102	98	104	108	104
Chen et al. [49]	100	102	103	104	106	106	106	108	104.3
Tang et al. [50]	100	103	104	104	105	105	106	109	104.5
Khan et al. [37]	102	108	106	102	106	106	106	98	104.25
Belazi et al. [51]	106	106	106	104	108	102	106	104	105.25
Hua [52]	106	106	108	106	102	102	108	104	105.25
Javeed [53]	108	106	106	110	106	108	108	108	107.50

5.2. Bit Independence Criterion (BIC)

The pairwise avalanche vectors' independent behavior and the variations in input bits are primarily evaluated using the bit independence criterion [29,30]. We have tested and sorted out the nonlinearity of the proposed S-box via BIC. In Table 17, the proposed S-box's minimum and average BIC values are compared with other well-known S-boxes' square deviation values.

Table 16. Detailed BIC analysis for proposed S-box-4.

	0	1	2	3	4	5	6	7
Detailed BIC Analysis for proposed S-box-4	----	102	104	100	106	106	106	106
	102	----	102	108	104	108	104	102
	104	102	----	104	108	108	102	104
	100	108	104	----	104	108	106	108
	106	104	108	104	----	96	104	98
	106	108	108	108	96	----	104	106
	106	104	102	106	104	104	----	104
	106	102	104	108	98	106	104	----
	Average BIC: 104.35							

Table 17. BIC comparison of the proposed S-boxes with various well-known S-boxes.

S-boxes	Minimum value	Average	Square deviation
Proposed 1	96	103.42	2.56
Proposed 2	98	102.86	2.38
Proposed 3	96	104.57	2.41
Proposed 4	96	104.35	2.81
Hussain [42]	112	112	0
Gautam [43]	92	103	3.5225
Prime [44]	94	101.71	3.53
S8 AES [54]	112	112	0
Xyi [46]	98	103.78	2.743
AES [47]	112	112	0
Skipjac [48]	102	104.14	1.767

5.3. Strict Avalanche Criterion (SAC)

Tavares and Webster [39] introduced Strict Avalanche Criterion. In this article, they gave the idea of avalanche and completeness effect. The purpose of the SAC component is achieved if there is a 0.5 probability that each output bit is changed by modifying a single input bit. In Table 18, the SAC of proposed S-box is displayed.

Table 18. Strict avalanche criterion of the proposed S-box (Table 11).

	0	1	2	3	4	5	6	7
SAC Results (S-box-1)	0.484375	0.531250	0.484375	0.546875	0.484375	0.515625	0.484375	0.468750
	0.546875	0.500000	0.515625	0.546875	0.453125	0.515625	0.515625	0.468750
	0.484375	0.500000	0.500000	0.484375	0.515625	0.500000	0.500000	0.515625
	0.453125	0.546875	0.578125	0.515625	0.500000	0.578125	0.531250	0.484375
	0.531250	0.562500	0.484375	0.515625	0.515625	0.515625	0.515625	0.484375
	0.531250	0.468750	0.500000	0.500000	0.484375	0.484375	0.484375	0.515625
	0.515625	0.515625	0.531250	0.468750	0.500000	0.562500	0.500000	0.531250
	0.453125	0.484375	0.546875	0.562500	0.500000	0.484375	0.500000	0.531250
Average SAC (S-box-1)	0.508301							
Average SAC (S-box-2)	0.504150							
Average SAC (S-box-3)	0.499756							
Average SAC (S-box-4)	0.506348							

5.4. Linear Approximation Probability (LP)

We investigate about the highest imbalance of an event in Linear Approximation Probability [55]. The uniformity of the input bits should be similar to that of output bits. Every input bit is examined separately and its results are scrutinizing by output bits. The masks denoted by ω_x and ω_y respectively are applied on the uniformity of both input and output bits. It is given as

$$LP = \max_{\omega_x,\omega_y \neq 0} \left| \frac{\#\{f \mid f.\omega_x = S(f) * \omega_y\}}{2^n} - \frac{1}{2} \right|,$$

where f is the collection of all possible inputs and 2^n represents the total number of elements. The results of our proposed S-box and different renowned S-boxes via LP are shown in Table 15. Our proposed S-box is built so well and strong enough to avoid the linear attacks while comparison.

Table 19. LP analysis of various S-boxes.

S-boxes	Max value	Max LP
Proposed 1 (Table 11)	160	0.125
Proposed 2 (Table 12)	162	0.133
Proposed 3 (Table 13)	164	0.141
Proposed 4 (Table 14)	158	0.117
AES [47]	144	0.062
Hussain [42]	144	0.062
Skipjack [48]	156	0.109
Prime [44]	162	0.132
Gautam [43]	164	0.2109
S_8 AES [54]	144	0.062
Xyi [46]	168	0.156

5.5. *Differential Approximation Probability (DP)*

A non-linear mapping is used to sort out the differential uniformity. It is a relation between input and output bits. The input differential has a distinct transformation with the output differential. It is given as

$$D_{p^s}(\Delta g \rightarrow \Delta h) = \frac{[\#\{g \in I | S(g) \oplus S(g \oplus \Delta g) = \Delta h\}]}{2^n}.$$

Here, Δg and Δh refer to the differentials of the input and output, respectively. We applied this test on our S-box and the results are as follows (Table 20).

Table 20. Detailed DP analysis for proposed S-box-4.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	8	6	6	8	6	6	6	6	6	8	6	8	6	6	6
6	6	8	6	6	6	6	6	6	8	6	6	6	8	6	6
6	6	8	8	6	8	8	6	6	6	6	6	8	6	8	6
6	8	6	8	6	6	6	6	6	10	8	6	6	6	6	6
8	8	6	8	8	6	8	4	6	8	8	6	6	6	6	6
6	8	6	6	6	8	6	4	6	6	8	6	8	6	4	8
6	10	12	6	6	6	8	6	6	6	6	6	6	6	6	6
6	8	6	6	6	8	6	6	6	8	6	6	6	6	8	6
6	8	6	6	6	6	8	6	6	6	6	10	8	6	6	8
8	6	6	6	6	6	6	8	6	8	6	6	4	8	6	6
6	6	6	8	6	6	6	6	8	8	6	8	6	6	8	6
8	6	6	6	6	8	8	6	6	6	6	6	6	6	8	6
6	6	8	8	6	8	8	6	6	6	6	8	6	6	6	6
8	6	8	8	6	6	6	6	8	8	8	6	6	6	8	6
8	6	8	8	8	6	8	8	6	6	6	6	6	8	8	6
6	8	8	8	8	8	6	8	6	6	6	6	8	8	6	6
Max Val: 12															

5.6. *Fixed point and reversed fixed point analysis*

A fixed point in an S-box refers to an input value that remains unchanged after undergoing the substitution process. In other words, if a specific input value maps to the same value as the output, that input value is said to be a fixed point. Fixed points potentially impact the resistance of the S-box against certain attacks. Therefore, an S-box designer ensures that there are no fixed points (FP) in the particular S-box and such an S-box is very useful in image encryption [6]. On the other hand, when an output value is used as an input, it yields the original input value, which is known as a reverse fixed point. Additionally, short cycles are specific patterns that appear in the output values of an S-box. Liu et al. [56] proposed an improved coupling quadratic map (ICQM) algorithm for S-box design, and in order to remove fixed point and reverse fixed point criteria. To prevent leakage in any statistical cryptanalysis, the number of F_p and OF_p should be kept as low as possible. In modern S-box designs, researchers have successfully eliminated the F_p and OF_p by employing a 2D enhanced quadratic map [57].

Table 21. Fixed point and revers fixed point analysis.

S-box	No. of fixed point	No. of reverse fixed point
Lambić [32]	2	3
Jamal [58]	18	None
Tian [59]	1	1
Çavuşoğlu [60]	0	2
Özkaynak [61]	4	1
Ullah [23]	4	None
Proposed S-box-1 (Table-11)	3	1
Proposed S-box-2 (Table-12)	1	1
Proposed S-box-3 (Table-13)	2	None
Proposed S-box-4 (Table-14)	3	1

6. Conclusions

The main accomplishment of this research is to create a way to use coset graph and optimization algorithms to develop secure S-boxes with enhanced nonlinearity. In this article, we propose a new, simple, and efficient S-box construction scheme based on a coset graph over the finite field $GF(2^{10})$. Various evaluations are conducted to determine the efficacy of the proposed S-box construction method, and the results are completely satisfactory. The experimental results demonstrated that the proposed S-boxes are strong enough to provide security against various algebraic attacks. Furthermore, the proposed nonlinearity enhancement algorithm improves the cryptographic properties of the constructed S-boxes. The application of the proposed algorithm is not restricted to the proposed S-boxes, but it can also improve the nonlinearity of any S-box.

Use of AI tools declaration

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

Conflict of interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. H. C. A. Tilborg, *Fundamentals of cryptology: a professional reference and interactive tutorial*, Boston: Kluwer Academic Publishers, 2000.
2. K. Larew, D. Kahn, *The codebreakers: the story of secret writing*, 1 Ed., New Yourk: McMillan, 1967.
3. A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone, *Handbook of applied cryptography*, 1 Ed., CRC Press, 1996.
4. D. R. Stinson, M. B. Paterson, *Cryptography: theory and practice*, 4 Eds., CRC Press, 1995.
5. C. E. Shannon, Communication theory of secrecy systems, *Bell Syst. Tech. J.*, **28** (1949), 656–715.

6. H. Liu, J. Liu, C. Ma, Constructing dynamic strong S-box using 3D chaotic map and application to image encryption, *Multimed. Tools Appl.*, **82** (2023), 23899–23914. https://doi.org/10.1007/s11042-022-12069-x
7. L. Cui, Y. Cao, A new S-box structure named affine-power-affine, *Int. J. Innov. Comput. Inf. Control*, **3** (2007), 751–759.
8. I. Hussain, T. Shah, Literature survey on nonlinear components and chaotic nonlinear components of block ciphers, *Nonlinear Dyn.*, **74** (2013), 869–904. https://doi.org/10.1007/s11071-013-1011-8
9. H. Liu, A. Kadir, X. Sun, Chaos-based fast colour image encryption scheme with true random number keys from environmental noise, *IET Image Process.*, **11** (2017), 324–332. https://doi.org/10.1049/iet-ipr.2016.0040
10. H. Liu, A. Kadir, Asymmetric color image encryption scheme using 2D discrete-time map, *Signal Process.*, **113** (2015), 104–112. https://doi.org/10.1016/j.sigpro.2015.01.016
11. H. Liu, A. Kadir, J. Liu, Color pathological image encryption algorithm using arithmetic over Galois field and coupled hyper chaotic system, *Opt. Lasers Eng.*, **122** (2019), 123–133. https://doi.org/10.1016/j.optlaseng.2019.05.027
12. I. Hussain, T. Shah, H. Mahmood, M. A. Gondal, 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
13. F. ul Islam, G. Liu, Designing S-box based on 4D-4wing hyperchaotic system, *3D Res.*, **8** (2017), 9. https://doi.org/10.1007/s13319-017-0119-x
14. I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, Construction of cryptographically strong 8x8 S-boxes, *World Appl. Sci. J.*, **13** (2011), 2389–2395.
15. M. Ahmad, M. N. Doja, M. M. S. Beg, ABC optimization based construction of strong substitution-boxes, *Wirel. Pers. Commun.*, **101** (2018), 1715–1729. https://doi.org/10.1007/s11277-018-5787-1
16. Attaullah, S. S. Jamal, T. Shah, A novel algebraic technique for the construction of strong substitution box, *Wireless Pers. Commun.*, **99** (2018), 213–226. https://doi.org/10.1007/s11277-017-5054-x
17. F. Özkaynak, V. Çelik, A. B. Özer, A new S-box construction method based on the fractional-order chaotic Chen system, *Signal, Image Video Process.*, **11** (2017), 659–664. https://doi.org/10.1007/s11760-016-1007-1
18. A. Razaq, A. Yousaf, U. Shuaib, N. Siddiqui, A. Ullah, A. Waheed, A novel construction of substitution box involving coset diagram and a bijective map, *Secur. Commun. Networks*, **2017** (2017), 5101934. https://doi.org/10.1155/2017/5101934
19. A. Hussain Alkhaldi, I. Hussain, M. A. Gondal, A novel design for the construction of safe S-boxes based on TD ERC sequence, *Alex. Eng. J.*, **54** (2015), 65–69. https://doi.org/10.1016/j.aej.2015.01.003
20. L. Liu, Y. Zhang, X. Wang, A novel method for constructing the S-box based on spatiotemporal chaotic dynamics, *Appl. Sci.*, **8** (2018), 2650. https://doi.org/10.3390/app8122650
21. S. Zhu, X. Deng, W. Zhang, C. Zhu, Secure image encryption scheme based on a new robust chaotic map and strong S-box, *Math. Comput. Simul.*, **207** (2023), 322–346. https://doi.org/10.1016/j.matcom.2022.12.025
22. F. A. Khan, J. Ahmed, J. S. Khan, J. Ahmad, M. A. Khan, A novel substitution box for encryption based on Lorenz equations, *2017 International Conference on Circuits, System and Simulation (ICCSS)*, 2017, 32–36. https://doi.org/10.1109/CIRSYSSIM.2017.8023176

23. A. Ullah, S. S. Jamal, T. Shah, A novel construction of substitution box using a combination of chaotic maps with improved chaotic range, *Nonlinear Dyn.*, **88** (2017), 2757–2769. https://doi.org/10.1007/s11071-017-3409-1
24. J. Zheng, Q. Zeng, An image encryption algorithm using a dynamic S-box and chaotic maps, *Appl. Intell.*, **52** (2022), 15703–15717. https://doi.org/10.1007/s10489-022-03174-3
25. L. Li, J. Liu, Y. Guo, B. Liu, A new S-box construction method meeting strict avalanche criterion, *J. Inf. Secur. Appl.*, **66** (2022), 103135. https://doi.org/10.1016/j.jisa.2022.103135
26. Y. Su, X. Tong, M. Zhang, Z. Wang, A new S-box three-layer optimization method and its application, *Nonlinear Dyn.*, **111** (2023), 2841–2867. https://doi.org/10.1007/s11071-022-07956-9
27. Y. Si, H. Liu, M. Zhao, Constructing keyed strong S-box with higher nonlinearity based on 2D hyper chaotic map and algebraic operation, *Integration*, **88** (2023), 269–277. https://doi.org/10.1016/j.vlsi.2022.10.011
28. Y. Liu, X. Tong, J. Ma, Image encryption algorithm based on hyper-chaotic system and dynamic S-box, *Multimed. Tools Appl.*, **75** (2016), 7739–7759. https://doi.org/10.1007/s11042-015-2691-5
29. I. Hussain, T. Shah, M. A. Gondal, M. Khan, W. A. Khan, Construction of new S-box using a linear fractional transformation, *World Appl. Sci. J.*, **14** (2011), 1779–1785.
30. 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
31. D. Shah, T. Shah, Y. Naseer, S. S. Jamal, S. Hussain, Cryptographically strong S-P boxes and their application in steganography, *J. Inf. Secur. Appl.*, **67** (2022), 103174. https://doi.org/10.1016/j.jisa.2022.103174
32. D. Lambić, A novel method of S-box design based on discrete chaotic map, *Nonlinear Dyn.*, **87** (2017), 2407–2413. https://doi.org/10.1007/s11071-016-3199-x
33. N. A. Azam, U. Hayat, I. Ullah, An injective S-box design scheme over an ordered isomorphic elliptic curve and its characterization, *Secur. Commun. Networks*, **2018** (2018), 3421725. https://doi.org/10.1155/2018/3421725
34. 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
35. P. J. Cameron, Cayley graphs and coset diagrams group actions, *Encycl. Des. Theory*, **1** (2006), 1–9.
36. P. M. Cohn, W. Magnus, A. Karrass, D. Solitar, *Combinatorial group theory.*, **74** (1967). https://doi.org/10.2307/2314941
37. M. Khan, T. Shah, M. A. Gondal, An efficient technique for the construction of substitution box with chaotic partial differential equation, *Nonlinear Dyn.*, **73** (2013), 1795–1801. https://doi.org/10.1007/s11071-013-0904-x
38. M. M. Dimitrov, On the design of chaos-based S-boxes, *IEEE Access*, **8** (2020), 117173–117181. https://doi.org/10.1109/ACCESS.2020.3004526
39. J. Pieprzyk, G. Finkelstein, Towards effective nonlinear cryptosystem design, *IEE Proc. E-Comput. Digital Tech.*, **135** (1988), 325–335.
40. D. Zhu, X. Tong, M. Zhang, Z. Wang, A new s-box generation method and advanced design based on combined chaotic system, *Symmetry*, **12** (2020), 1–17. https://doi.org/10.3390/sym12122087

41. A. H. Zahid, A. M. Iliyasu, M. Ahmad, M. M. U. Shaban, M. J. Arshad, H. S. Alhadawi, et al., A novel construction of dynamic S-box with high nonlinearity using heuristic evolution, *IEEE Access*, **9** (2021), 67797–67812. https://doi.org/10.1109/ACCESS.2021.3077194
42. I. Hussain, T. Shah, M. A. Gondal, H. Mahmood, Generalized majority logic criterion to analyze the statistical strength of S-boxes, *Z. Naturforsch. A*, **67** (2012), 282–288. https://doi.org/10.5560/ZNA.2012-0022
43. A. Gautam, G. S. Gaba, R. Miglani, R. Pasricha, Application of chaotic functions for construction of strong substitution boxes, *Indian J. Sci. Technol.*, **8** (2015), 1–5. https://doi.org/10.17485/ijst/2015/v8i28/71759
44. I. Hussain, T. Shah, H. Mahmood, M. A. Gondal, U. Y. Bhatti, Some analysis of S-box based on residue of prime number, *Proc. Pakistan Acad. Sci.*, **48** (2011), 111–115.
45. I. Hussain, A new algorithm to construct secure keys for AES, **5** (2010), 1263–1270.
46. X. Yi, S. X. Cheng, X. H. You, K. Y. Lam, Method for obtaining cryptographically strong 8×8 S-boxes, *GLOBECOM 97. IEEE Global Telecommunications Conference. Conference Record*, 1997, 689–693. https://doi.org/10.1109/glocom.1997.638418
47. J. Daemen, V. Rijmen, *The design of rijndael*, New York: Springer, 2002.
48. National Institute of Standards and Technology, SKIPJACK and KEA Algorithm Specifications, 1998. Available From: https://csrc.nist.gov/Presentations/1998/Skipjack-and-KEA-Algorithm-Specifications.
49. G. Chen, Y. Chen, X. Liao, An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps, *Chaos, Solitons Fract.*, **31** (2007), 571–579. https://doi.org/10.1016/j.chaos.2005.10.022
50. G. Tang, X. Liao, Y. Chen, A novel method for designing S-boxes based on chaotic maps, *Chaos, Solitons Fract.*, **23** (2005), 413–419. https://doi.org/10.1016/j.chaos.2004.04.023
51. A. Belazi, M. Khan, A. A. A. El-Latif, S. Belghith, Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption, *Nonlinear Dyn.*, **87** (2017), 337–361. https://doi.org/10.1007/s11071-016-3046-0
52. 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
53. A. Javeed, T. Shah, Attaullah, Design of an S-box using Rabinovich-Fabrikant system of differential equations perceiving third order nonlinearity, *Multimed. Tools Appl.*, **79** (2020), 6649–6660. https://doi.org/10.1007/s11042-019-08393-4
54. I. Hussain, A new algorithm to construct secure keys for AES, *Int. J. Contemp. Math. Sci.*, **5** (2010), 1263–1270.
55. E. Biham, A. Shamir, Differential cryptanalysis of DES-like cryptosystems, *J. Cryptol.*, **4** (1991), 3–72. https://doi.org/10.1007/BF00630563
56. H. Liu, A. Kadir, C. Xu, Cryptanalysis and constructing S-box based on chaotic map and backtracking, *Appl. Math. Comput.*, **376** (2020), 125153. https://doi.org/10.1016/j.amc.2020.125153
57. Y. Si, H. Liu, Y. Chen, Constructing keyed strong S-box using an enhanced quadratic map, *Int. J. Bifurcat. Chaos*, **31** (2021), 2150146. https://doi.org/10.1142/S0218127421501467
58. S. S. Jamal, M. U. Khan, T. Shah, A watermarking technique with chaotic fractional S-box transformation, *Wireless Pers. Commun.*, **90** (2016), 2033–2049. https://doi.org/10.1007/s11277-016-3436-0
59. Y. Tian, Q. Liu, D. Liu, Y. Kang, P. Deng, F. He, Updates to Grasselli's peak shear strength model, *Rock Mech. Rock Eng.*, **51** (2018), 2115–2133. https://doi.org/10.1007/s00603-018-1469-2

60. Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, S. Kaçar, A novel approach for strong S-box generation algorithm design based on chaotic scaled Zhongtang system, *Nonlinear Dyn.*, **87** (2017), 1081–1094. https://doi.rog/10.1007/s11071-016-3099-0
61. F. Özkaynak, Construction of robust substitution boxes based on chaotic systems, *Neural Comput. Appl.*, **31** (2019), 3317–3326. https://doi.org/10.1007/s00521-017-3287-y