AIMS Mathematics

AIMS Mathematics, 9(5): 10977–10996.
DOI: 10.3934/math.2024537
Received: 09 December 2023
Revised: 17 February 2024
Accepted: 26 February 2024
Published: 20 March 2024

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

Research article

8×8 S-boxes over Klein four-group and Galois field $GF(2^4)$: AES redesign

Mohammad Mazyad Hazzazi[1,*], Amer Aljaedi[2], Zaid Bassfar[3], Misbah Rani[4] and Tariq Shah[4]

[1] Department of Mathematics, College of Science, King Khalid University, Abha 61413, Saudi Arabia
[2] College of Computing and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia
[3] Department of Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia
[4] Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan

* **Correspondence:** Email: mmhazzazi@kku.edu.sa.

Abstract: This research paper is supplemented with a unique formation to design state-of-the-art S-boxes. The invented approach is simple but has the capability of creating confusion in our newly proposed algorithm. Our core planned work refined the method of already designed S-boxes to accomplish more compact ones. Various structures were merged here, namely affine transformation, fractional linear transformation, structure of Klein four-group, and the algebraic structures of the Galois fields, $GF(2^4)$ and $GF(2^8)$. These structures were utilized to synthesize newly 1600 robust S-boxes. Besides, we discussed encryption steps of AES with these newly generated S-boxes. We highlighted some specific characteristics, performance of parameter's improvement, and their utilization. Nonlinear properties were mainly set to inspect the behavior of I/O bits and could apply image encryption. Then, the performance of proposed S-boxes and newly structured AES was tested in comparison with other prevailing S-boxes.

Keywords: AES; Klein four group V_4; S-boxes; Cartesian structure; Cartesian permutation; $V_4 \times V_4$
Mathematics Subject Classification: 68P25, 68U15

1. Introduction

In symmetric key cryptography, since 2000, the AES standard of 128 bits was adapted by the National Institute of Standard and Technology (NIST). For review, the original designers of AES, Joan

Daemen, and Vincent Rijmen (known as Rijndael) submitted this designed approach along various other algorithms. There was a requirement to develop a strong enough algorithm that can be further used for encryption application. The reason that urges the cryptographers to build the AES is to retain the US Government document in private for a minimum of 20 years. Due to the advancement in computing power, the former methods of securing data i.e., Data Encryption Standard (DES) (developed by IBM in 1975 and break in 1998) and triple-DES has become weak for security purposes. Thus, the government of USA offered an open proposal to increase the standard of data encryption. Finally, the Rijndael cryptosystem (AES) was adapted for encryption scheme and it was circulated by FIPS in 26 NOV 2001 [1]. AES is largely being used for secure transactions through the internet and to transfer of money through banks [2].

S-box plays a seed role in AES. By understanding the concepts of its functionality, it comes to know that S-box is a non-linear factor in AES. It is a heart of any block cipher cryptosystem. The Symmetric block key algorithm uses substitution boxes that yield confusion and permutation boxes that yield diffusion in data when it is going to be encrypted. The significance of S-box is that it creates hurdles for cryptanalysis. Thus, no unauthorized person can get unlawful access to original messages [3]. Several attempts have been consequently followed to design valuable, highly secure, and robust S-boxes that many ciphers utilize to encrypt data. The usefulness of the original AES S-box is to provide substitute data, which is based on keys along with permutation to develop a substitution-permutation network. S-boxes have resulted in many interesting properties that are appropriate for different ciphers. Due to the shuffling of input bits, output bits also change. These changing of bits are analyzed to determine the confusion creating capability and the strength of S-boxes. Various S-boxes are presented in literature such as APA [4], Gray [5], S_8 and residue prime [6–8], and AES [3], which have good cryptographic properties and algebraic complexity [9].

In literature, there is an insight upon the building of S-boxes under three irreducible polynomials of $GF(2^4)$. These three polynomials are used for the manufacture of small 4×4 S-boxes [10]. Affine transformation is utilized that carries the best choice of 4×4 transformation matrices and 4×1 constant matrices for all irreducible polynomials of $GF(2^4)$ [11]. It was very challenging and hard for code breakers to break the code because numerous irreducible polynomials are practiced instead of single ones like the Rijndael algorithm, which worked only for single S-box. Ten Small S-boxes are structured and the permutation of symmetric group S_4 (consists of 24 permutations) [12] is operated on each of the small 10 S-boxes, and 240 new ones are obtained by applying permutation, one after the other. Formerly, 10 random S-boxes are selected. The number of choices for arbitrary selection is 240C10. Those innovative S-boxes play a very vital role for hiding data so that nobody can crack the code easily in a limited time.

In our projected algorithm, a new scheme is developed, which has very efficient algebraic complexity for safeguarding data. It safeguards the data in both text and image form. Our optimized research work will utilize the 10 S-boxes of $GF(2^4)$ (Result of symmetric group permutation S_4) [10] and Cartesian permutations of Klein four-group V_4 [12] with itself. Our proposed algorithm seeks a new methodology that is sufficient for security purposes to develop 8×8 S-boxes using subfield $GF(2^4)$ of $GF(2^8)$. With the aid of this new stylist approach, 1600 independent S-boxes are obtained here and then random 10 S-boxes are picked. The key point is that we have billions of choices i.e., 1600C10≈(2.945764438E+37) to arbitrary pick any 10 S-boxes for utilization in AES. It is quite a large number that makes the process very impressive and safe, which nobody knows, instead of a receiver that is the choice of cryptographers to choose 10 S-boxes for encryption. For security purposes, the performance of our modified S-boxes is comparatively more accurate than other ones when we compare it with other S-boxes through different tests.

The structure of the paper layout is as follows. In Section 2, we briefly elaborate necessary algebraic expressions for AES S-boxes that use the Rijndael algorithm [1]. Section 3 displays the technique for already designed small substitution boxes [10]. In Section 4, we elaborate on the new scheme for the construction of modified S-boxes, permutation of Klein four group, and their use in proposed S-boxes. Section 5 depicts the random selection of S-boxes and pictorial representation of whole modified schemes. In Sections 6 and 7, message encryption and decryption by modified S-boxes, analysis of S-boxes, and image encryption by proposed schemes with their results is presented. In the last two sections, comparisons and conclusions are presented.

2. Algebraic structure of the AES S-box

2.1. Klein four group

The Klein four-group is the smallest noncyclic Abelian group in which every element has order 2. It is $\cong$ to the direct sum of two abelian groups $\mathbb{Z}_2 \times \mathbb{Z}_2$ where Addition is defined component wise under mod (2) [12]. In group notation, the Klein four-group is defined by,

$$V_4 =< i, j \mid i^2 = j^2 = (ij)^2 = e >. \tag{1}$$

Permutation Representation: The permutation illustration of this group entails four points [12].

$$V_4 =< e, (12)(34), (13)(24), (14)(23) >. \tag{2}$$

As V_4 is applicable only to four-bit data so for the application of this particular permutation on eight-bit data to increase the capability of diffusion of that cipher is to utilize the Cartesian structure of V_4. It comprises 16 permutations that are signified by μ_t; $1 \leq t \leq 16$ and the permutation chart according to their data type is given in Table 1.

Table 1. Use of permutation in S-boxes.

Permutation		**S-boxes**
$\mu_t(\boldsymbol{S_1^{1\neg}}); \mathbf{1 \leq t \leq 16}$	$\rightarrow$	$S_j^{1^*}; 1 \leq \mathrm{j} \leq 16$
$\mu_t(\boldsymbol{S_2^{1\neg}}); \mathbf{1 \leq t \leq 16}$	$\rightarrow$	$S_j^{2^*}; 1 \leq \mathrm{j} \leq 16$
$\mu_t(\boldsymbol{S_3^{1\neg}}); \mathbf{1 \leq t \leq 16}$	$\rightarrow$	$S_j^{3^*}; 1 \leq \mathrm{j} \leq 16$
……	……	……
……	……	……
$\mu_t(\boldsymbol{S_{10}^{10\neg}}); \mathbf{1 \leq t \leq 16}$	$\rightarrow$	$S_j^{10^*}; 1 \leq \mathrm{j} \leq 16$

2.2. Galois field

Let $F = F_q$ be a field and $F[x]$ is the Euclidean domain. For the extension of field $F[x]^m$, the quotient rings

$$F[x]/< f(x) >\cong GF(q^m),$$

where the maximal ideal $< f(x) >$ is generated by $f(x)$ an irreducible polynomial of degree m in $F[x]$. If we write α to denote the coset $x + (f(x))$, then $f(\alpha) = 0$ and

$$F[x]^m = \{a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{m-1}\alpha^{m-1} \colon \forall\, a_i \in F, i = 0,1,2,\ldots,m-1\}.$$

The field $F[x]^m$ is a Galois field (m-degree extension field of the field F).

2.3. AES S-box functions

Two sub steps are discussed here for S-box function of input bytes that is utilized in AES for safeguarding data [3,13].

1) ***Multiplicative inversion:*** Let a be a nonzero input byte. Taking its inverse in $GF(2^8)$ and acquire output byte $f(a)$.

$$f(a) = t = \begin{Bmatrix} a^{-1} & a \neq 0 \\ 0 & a = 0 \end{Bmatrix}. \tag{3}$$

2) ***Affine transformation:*** Next sub step is to use the affine transformation i.e.,

$$S - box = c = M\big(f(a)\big) \oplus\ b. \tag{4}$$

It is a required S-box function, where b is a stable byte and M is constant bit matrix. Affine transformation is given below [14,15].

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{pmatrix} = \begin{pmatrix} 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \\ 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

3. Designing 4×4 S-boxes

The small S-boxes comprises 16 elements that are defined over finite Galois field $GF(2^4)$ [10] and they are designed under three distinct irreducible polynomials [13] through the best choice of 4×4 transformation matrices and 4×1 constant matrices (Table 2). B represents each member of $GF(2^4)$ in transformation T, which is written in the form of 4×1 matrix.

The transformation matrices X and constant matrices C are fixed according to distinct irreducible polynomials P_1 (t), P_2 (t), and P_3 (t). The chart presented below represents the whole information about transformation and constant matrices according to respective polynomials.

After achieving 10 S-boxes, the symmetric group S_4 acts on them to permute the bytes $S_4 \times S_i$; $1 \leq i \leq 10$ (Section 3). A total of 240 S-boxes are obtained under permutation of S_4 [12]. Thus, 10 S-boxes have been arbitrarily picked for utilization in a cryptographic area.

Table 2. Technique for designing small S-boxes.

Transformation: $T = XB \oplus C$ [16]			
Polynomials	**Best Choice Matrix X**	**Inverse of Matrix X**	**Suitable constant Matric C (order 4×1)**
$P_1(t) = t^4 + t + 1$	$a_1 = \begin{pmatrix} 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}$	$(a_1)^{-1} = \begin{pmatrix} 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}$	$0 \times a \; and \; 0 \times f$
$P_2(t) = t^4 + t^3 + 1$	$a_2 = \begin{pmatrix} 1\,0\,1\,1 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \\ 0\,1\,1\,1 \end{pmatrix}$	$(a_2)^{-1} = \begin{pmatrix} 1\,1\,1\,0 \\ 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \end{pmatrix}$	$0 \times 3, 0 \times 9, 0 \times c \; and \; 0 \times d$
$P_3(t) = t^4 + t^3 + t^2 + t + 1$	$a_3 = \begin{pmatrix} 1\,1\,0\,0 \\ 0\,1\,0\,1 \\ 0\,1\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}$	$(a_3)^{-1} = \begin{pmatrix} 1\,1\,0\,1 \\ 0\,1\,0\,1 \\ 0\,1\,1\,1 \\ 0\,0\,0\,1 \end{pmatrix}$	$0 \times 4, 0 \times 5, 0 \times d \; and \; 0 \times f$

4. Proposed S-boxes

For making practical and effective use, we demonstrate the configuration of proposed 8×8 S-boxes in this section. This novel technique depends on four steps by utilizing 10 small 4×4 S-boxes of $GF(2^4)$ (output of Section 3). Also, the Cartesian structure of Klein four-group is practiced here and the performance of proposed algorithm is much better for security purpose.

Step 1. The initial step for designing 8×8 S-boxes is to follow the combination scheme of nibbles. As $S_i, S_j : GF(2^4) \to GF(2^4)$. So the mapping $\xi : S_i S_j \to S_j^i$ for combination is known as joint mapping that joins the nibbles of S_i and S_j for the formation of bytes of S_j^i. Where $S_i S_j : GF(2^4) \to GF(2^8)$ is defined as $\xi\left(S_i(u) S_j(v)\right) = S_j^i(uv) = S_j^i(t); 1 \le i, j \le 10$. Here u, v are nibbles and t represents byte.

The process of combination is that first nibble of S_i; $1 \le i \le 10$ is joint one by one by all nibbles of S_j; $1 \le j \le 10$ and makes one row of S_j^i. By following this scheme 100 new structured S-boxes $S_j^i; 1 \le i, j \le 10$ of $GF(2^8)$ are developed. The process of combination is presented in Table 3.

Table 3. Application of joint mapping.

S-box combination		Number of S-boxes	Representation
$\boldsymbol{\xi(S_1 S_i),\ 1 \le i \le 10}$	➡	10 S-boxes	$S_j^1; 1 \le j \le 10$
$\boldsymbol{\xi(S_2 S_i),\ 1 \le i \le 10}$	➡	10 S-boxes	$S_j^2; 1 \le j \le 10$
$\boldsymbol{\xi(S_3 S_i),\ 1 \le i \le 10}$	➡	10 S-boxes	$S_j^3; 1 \le j \le 10$
……….	………..	……	……….
$\boldsymbol{\xi(S_{10} S_i),\ 1 \le i \le 10}$	➡	10 S-boxes	$S_j^{10}; 1 \le j \le 10$

Step 2. In this step we utilize the change of 8×8 basis matrix [14] for computing the S-box function of given byte. The algebraic expression for transformation $\beta: GF(2^8) \to GF(2^8)$ is defined by $\beta(t) = Xt.$ It is a multiplication of 8 bit matrix of input block with this basis matrix X for enhancing complexity. Where t and $\beta(t)$ both are $8.$ bit input matrix. The approach of using this transformation is to convert every input byte into output ones. The change of constant bit basis matrix

X [2] is represented as,

$$X = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{5}$$

This transformation is applied to each byte of $S_j^i; 1 \leq i,j \leq 10$ (Step 1). So as a consequence of this transformation 100 modified S-boxes $S_j^{i\prime}; 1 \leq i,j \leq 10$ are achieved.

Step 3. Under this step, the affine linear transformation $eqn\ (1-2)$ is applied in quite consistent format to each of the distinct S-box $S_j^i; 1 \leq i,j \leq 10$ (Step 1) and resulting S-boxes are $S_j^{i\prime\prime}; 1 \leq i,j \leq 10$. Then the SubBytes transformation is practiced and transform different 8 bit to another different 8 bit data [13]. The key point in this substitution process is that unique 100 substitution boxes are utilized for subByte of modified 100 S-boxes $S_j^i; 1 \leq i,j \leq 10$. Thus $S_j^{i\gamma}$ is obtained by subByte of $S_j^{i\prime\prime}$ (substitution box) to $S_j^{i\prime}$ (Step 2) for each $1 \leq i,j \leq 10$. The subByte pattern that is depicted in Figure 1.

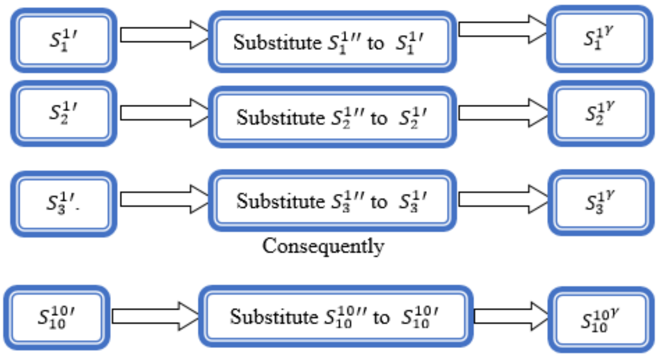

Figure 1. Substitution process.

Step 4. Now for enhancing more the algebraic power of the outcome of Step 3, it is to break each and every byte of input blocks $S_j^{i\gamma}$; $1 \leq i,j \leq 10$ into prefix and postfix nibbles for the evaluation of bijective transformation $f(z)$ defined on $GF(2^4)$. Where z represents an input byte. This specific bijective function is evaluated by the fixed values of $a', b', c'\, and\ d'$ chosen from $GF(2^8)$ against the range of z defined as $[0, 255]$. The respective transformation is given as,

$$f(z) = f(x)f(y),$$

where

$$\begin{cases} f(x) = (a'x + b')mod\ 16;\ x = prefix\ Nibble, \\ f(y) = (c'x + d')mod\ 16;\ x = postfix\ Nibble. \end{cases} \tag{6}$$

In above equation $f(z)$ represents a byte and later on the group action of projective general linear group $\eta : PGL(2, GF(2^8)) \times GF(2^8) \rightarrow GF(2^8)$ [6] is defined on $GF(2^8)$ referred as Mobius transformation. The expression for computational evaluation is $\eta\big(f(z)\big) = \frac{af(z)+b}{cf(z)+d}$ for fixed constants $a, b, c\ and\ d$ chosen from $GF(2^8)$. Large number of S-boxes have been synthesized by following this procedure but to make it easy for the reader the example is elaborated on this technique.

Example. This example will elaborate the whole technique of step 4 by fixed values $a' = 1B,\ b' = 39,\ c' = 25$ and $d' = 6B$ for evaluation of $f(z)$ and then particular Mobius transformation $\eta\big(f(z)\big) = \frac{35f(z)+15}{9f(z)+5}$, where $1B, 39, 25, 6B, 35, 15, 9, 5 \in GF(2^8)$ [17]. The resultant all entries after this transformation are from $GF(2^8)$ and form 100 S-boxes $S_j^{i\neg}; 1 \le i, j \le 10$. One of them is given in Table 4.

Table 4. $S_2^{1\neg}$.

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

4.1. Use of permutation in S-boxes

In this section, we will discuss how to get permuted S-boxes by diffusion process. The use of Cartesian permutation $V_4 \times V_4$ in $S_j^{i\neg}; 1 \le i, j \le 10$ (Step 4) is quite important step in our research paper. The technique of permutation is

$$for\ x \in GF(2^8)\ of\ any\ of\ S_j^{i\neg},$$

$$G: (V_4 \times V_4) \times GF(2^8) \to GF(2^8),$$

$$G(V_4 \times V_4, \boldsymbol{x}) = (V_4 \times V_4)(x), \tag{7}$$

$(V_4 \times V_4) \times S_j^{i\neg}\ 1 \le i, j \le 10\ (100\ S-boxes) = S_j^{i^*} 1 \le i \le 10, 1 \le j \le 16\ (1600\ S-boxes)$.

The whole description is tabulated as.

Here $\mu_t; 1 \le t \le 16$ are the mixture of two permutations σ and π. σ is applicable to prefix (N_1) and π is on postfix nibble (N_2) for each byte $x \in GF(2^8)$ of $S_1^{1\neg}$. The application of 8^{th}

permutation μ_8 on x is obtained as $\mu_8(x) = \sigma_8(N_1(\mathrm{x}))\pi_8(N_2(x))$. As a consequence, 1600 S-boxes $S_j^{i^*}$; $1 \leq \mathrm{i} \leq 10, 1 \leq \mathrm{j} \leq 10$ are achieved by the action of 16 permutations to each of $S_j^{i^\neg}$; $1 \leq i,j \leq 10$ (Step 4).

5. Random selection of S-boxes

The main step that is the heart of our algorithm is to select arbitrarily 10 S-boxes from 1600. The total possible choice for this selection is ${}^{1600}C_{10} \approx (2.945764438E + 37)$. The reason behind selection in this paper is to utilize in AES encryption that makes the system much more complicated for cryptanalysts. In our case selected 10 S-boxes are $\boldsymbol{S_2^{1}}^*$, $\boldsymbol{S_5^{4}}^*$, $\boldsymbol{S_7^{7}}^*$, $\boldsymbol{S_{10}^{5}}^*$, $\boldsymbol{S_1^{2}}^*$, $\boldsymbol{S_{10}^{10}}^*$, $\boldsymbol{S_{10}^{8}}^*$, $\boldsymbol{S_7^{4}}^*$, $\boldsymbol{S_9^{6}}^*$, $\boldsymbol{S_9^{4}}^*$, presented in Table 5.

Table 5. Final 10 S-boxes.

$\boldsymbol{S_2^{1^*}}$

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

$\boldsymbol{S_1^{2^*}}$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	180	253	126	44	234	188	41	139	116	133	228	19	12	146	16	64
1	166	49	171	135	246	87	111	81	183	231	100	99	222	46	164	7
2	140	144	109	218	232	102	203	98	214	160	244	223	56	104	201	63
3	137	10	181	110	250	219	17	103	121	161	75	28	23	35	77	176
4	122	127	136	117	209	221	195	147	68	205	130	236	157	115	3	94
5	238	237	217	177	131	113	105	52	48	9	167	235	186	185	229	193
6	73	79	184	50	173	165	251	108	2	212	172	220	13	119	199	226
7	149	57	34	155	197	224	27	120	47	90	1	32	88	96	170	43
8	36	200	15	163	192	37	247	182	112	69	76	168	153	80	123	202
9	53	25	190	42	150	59	58	141	189	118	6	230	106	124	78	215

Continued on next page

$S_1^{2^*}$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A	148	240	158	8	62	60	51	248	249	239	86	21	92	174	38	74
B	145	45	134	29	175	194	71	178	33	187	191	129	152	198	82	55
C	93	208	154	97	255	72	245	5	11	107	67	26	156	84	54	211
D	125	20	14	85	252	65	227	18	213	242	40	4	22	142	132	61
E	233	66	210	138	196	179	143	216	207	114	151	243	70	159	95	39
F	241	206	91	196	31	30	162	225	24	89	169	254	0	83	128	204

$S_5^{4^*}$

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

$S_7^{4^*}$

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

Continued on next page

								$S_7^{4^*}$								
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F	61	223	248	90	95	213	246	138	69	254	160	78	202	236	199	137

								$S_9^{4^*}$								
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	109	237	127	208	221	166	196	114	21	188	244	17	137	246	85	229
1	140	190	189	116	7	37	29	146	211	20	143	41	226	207	78	130
2	233	222	230	12	231	202	24	11	142	155	74	48	6	87	170	105
3	3	70	210	165	31	40	197	186	161	76	52	184	86	152	89	8
4	14	103	25	46	38	153	16	169	151	22	191	111	212	9	249	225
5	236	66	93	90	173	63	238	131	35	195	65	224	96	242	10	182
6	108	26	0	69	30	84	18	49	192	122	58	79	91	124	95	201
7	94	83	227	59	54	47	247	75	100	193	112	39	156	34	118	60
8	171	180	206	203	33	32	215	113	145	183	5	44	43	214	168	117
9	240	129	71	72	235	135	250	64	19	149	218	120	119	107	68	36
A	132	45	115	123	56	28	150	61	216	126	50	213	80	27	178	174
B	92	42	167	157	51	176	138	158	104	223	181	164	82	219	252	255
C	106	148	97	177	196	102	81	53	200	99	251	228	243	57	194	62
D	172	128	88	98	147	204	125	220	185	198	245	110	248	2	159	187
E	239	134	73	121	4	217	205	1	234	253	209	160	13	139	77	241
F	136	162	144	179	163	67	154	15	23	55	133	141	232	175	254	199

								$S_9^{6^*}$								
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	81	123	151	209	6	13	92	80	3	27	251	116	68	201	217	111
1	52	146	213	46	83	157	154	51	165	19	141	43	132	9	254	55
2	48	11	104	203	187	172	24	181	108	139	224	71	18	66	135	84
3	133	25	0	230	195	164	26	220	109	226	247	114	125	222	252	255
4	240	248	60	59	218	168	177	54	131	188	10	76	16	182	233	44
5	5	162	211	176	160	21	202	138	126	171	244	129	74	246	37	77
6	30	9	178	50	70	72	128	155	121	7	161	231	190	17	249	113
7	184	34	140	228	95	227	87	238	20	205	31	73	96	63	208	29
8	122	212	185	40	1	41	137	253	127	2	186	232	75	107	153	110
9	88	180	98	12	14	245	196	38	183	166	42	192	103	167	85	22
A	100	156	79	124	142	115	130	158	174	191	67	117	163	189	175	57
B	56	243	102	91	143	97	53	89	106	134	145	105	148	119	169	4
C	64	206	23	118	8	39	61	152	62	216	58	229	52	65	45	241
D	28	120	35	170	193	144	194	199	179	15	33	112	82	242	235	234
E	207	204	69	239	200	221	150	236	173	47	86	214	198	93	210	94
F	136	32	223	36	147	78	237	215	90	225	250	197	149	99	159	219

								$S_{10}^{5^*}$								
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	209	15	168	151	7	253	88	17	167	222	166	110	188	235	123	115
1	221	141	135	63	176	58	242	96	3	0	73	52	84	90	112	42

Continued on next page

$S_{10}^{5^*}$

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

$S_{7}^{7^*}$

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

$S_{10}^{8^*}$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	188	194	208	108	112	29	242	78	73	90	83	40	181	233	176	26
1	191	18	196	161	22	159	154	52	32	198	116	132	228	10	126	152
2	214	167	42	27	95	59	200	246	113	226	49	96	178	60	174	53
3	250	204	150	189	142	120	82	0	21	144	41	62	37	207	138	229
4	163	64	218	66	141	193	19	247	55	50	24	130	252	254	46	166
5	146	30	136	128	231	169	239	11	43	179	16	124	213	15	232	51
6	245	164	114	91	23	44	7	143	85	162	74	121	77	211	34	133

Continued on next page

$S_{10}^{8^*}$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7	20	227	145	25	104	107	39	70	129	84	54	101	160	68	209	220
8	79	192	153	241	157	81	147	238	249	33	221	47	65	118	182	243
9	63	251	45	244	222	175	71	131	137	171	111	195	203	212	58	69
A	234	158	235	253	135	61	134	17	105	155	88	80	219	13	89	180
B	67	184	72	168	6	206	103	9	139	123	187	151	248	48	86	94
C	110	36	75	119	148	117	14	165	127	202	109	156	186	35	93	199
D	2	28	216	38	255	5	201	102	217	99	172	205	224	1	12	177
E	87	8	215	140	223	97	100	125	225	183	122	106	57	4	185	197
F	236	115	173	3	92	230	56	170	210	52	237	190	240	31	76	98

$S_{10}^{10^*}$

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

6. Redesigning the AES algorithm

The Modified AES algorithm is the most important section in our research paper for utilization of newly generated robust S-boxes. In our modified algorithm, all means of AES-128 [1] are changed to improve the complexity and make it unpredictable. Our case is key dependent, just like the original AES because if we build a key dependent framework [13], then cryptanalysis turns out to be more troublesome. Also, the Mixed column matrix is not fixed, so a better approach of shifting is adapted and S-box is not static. The elaboration is displayed below.

1) Add round key

The initial step add round key is the same as in the original AES, so that in each round, a new key is Xored with state. The State matrix and key [13] are both equal in order. These keys are originally built by AES key expansion.

2) Substitute bytes

A new strategy for byte substitution is accomplished here. As a consequence of (Step 5), ten S-boxes are utilized one by one in each round for substitution. The way of substitution is the same as in AES [1].

3) Shift bytes

In this step, bytes are moved by utilizing a new scheme known as shift box. Diverse shifts are attempted in various rounds. Shift boxes are not fixed for each round [15,18]. Their selection is based on a substitution box. If $\boldsymbol{S_j^{i^*}}$ is used for substitution in Round 1 then S_i will play the role of shift box for that specific one (presented in Table 6).

Table 6. Shift S-box.

State Matrix				Shift Box				Result			
$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$	$S_{0,0}$	$S_{3,3}$	$S_{2,0}$	$S_{3,1}$	$S_{0,0}$	$S_{1,1}$	$S_{2,2}$	$S_{3,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$	$S_{0,1}$	$S_{2,2}$	$S_{2,3}$	$S_{1,0}$	$S_{2,1}$	$S_{3,2}$	$S_{2,0}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$S_{1,3}$	$S_{1,1}$	$S_{0,2}$	$S_{2,1}$	$S_{0,2}$	$S_{2,3}$	$S_{1,2}$	$S_{1,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	$S_{3,0}$	$S_{3,2}$	$S_{1,2}$	$S_{0,3}$	$S_{3,0}$	$S_{0,3}$	$S_{3,1}$	$S_{0,1}$

The way of shifting is that if the first component of shift box is 14, then the shifting is that the primary component of state moves towards the fourteenth position of state. Likewise, all elements are shifted. The technique is shown in the Figure 2.

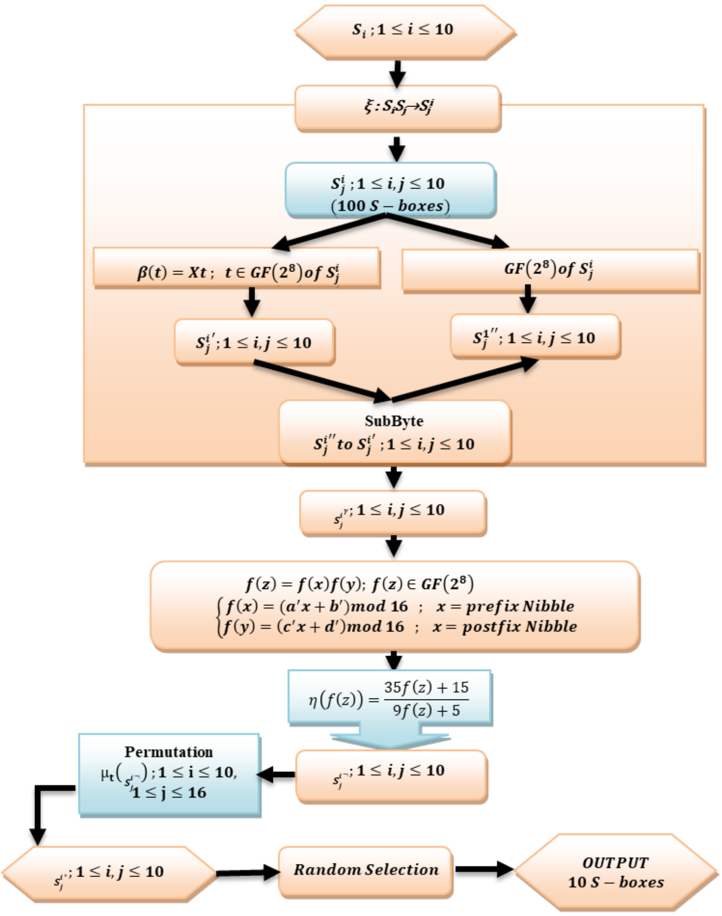

Figure 2. Flow chart of proposed algorithm.

4) Mix column

The mix column is an essential element of the encryption scheme. Like shift boxes [18], different mix columns are utilized in different rounds. Their selection is also based on the substitution box. If

$S_j^{i^*}$ is used for substitution in Round 1, then S_j will play the role of shift box for that specific one.

5) Round constant in key expansion

For each round, unique round constants (same as AES) [18], are used that are presented in Table 7.

Table 7. Round constants.

Rounds	1	2	3	4	5	6	7	8	9	10
Round constant	01	02	04	08	10	20	40	80	1b	36

The results of the message encryption and decryption by Modified AES structure are given in Table 8.

Table 8. Message encryption and decryption.

	Encryption	**Decryption**
	Plaintext TWO ONE NINE TWO	**Ciphertext**
Round 0	7F097A3667C0B786E59E7BED299B3853	70BAE70AEB6E28E5B61B87308F9A87F3
Round 1	0C2A223FBE983E29CF03F40EA1478E80	D5DAC6864CF89ED51A74F80D058117C4
Round 2	411F171705861FDEDA2B3BE898E986C9	97D05A6454C195589FA0C446C5CD62B4
Round 3	46159F31736643F2DBEC157BCCB659C3	3D3668E9EF4768A47F20EED2687AAE95
Round 4	E8145C9CBBAE049F029D20E31D8FA2C0	7AD27EB4EA69740BB33B44EC7B44A153
Round 5	18D695C3DDF6BBF2B58330EF533AE156	745ACBE0832FFFA49A0210BC69B94B59
Round 6	74613DDFBA037336CCD063823E2C2C32	0088533489FB1EEA73C544E6088D289C
Round 7	984E47AE1A5540DEF5D9E1C6A7EB2CE0	B24455F9A9AF562F98E16B0BA8ECB79A
Round 8	FDEE0A3DCBEDED7B8A75B3711F8997F6	732C69322B1EC0CB43EF5045BFBE16E0
Round 9	0F4643FC0948638C4A1172638FA068E8	382F51A9A2339F19785F40B61FD47504
Round 10	B8AE1A606BCA28D666DA0D6DD57A7282	54776F204F6E65204E696E652054776F
	Ciphertext	**Plaintext** TWO ONE NINE TWO

Moreover, the flow chart of modified encryption scheme are depicted in Figure 3.

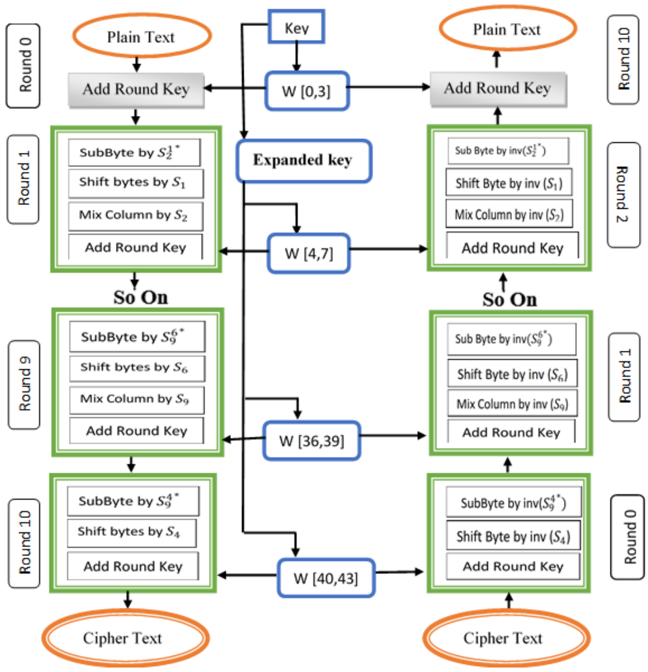

Figure 3. Modified encryption process.

7. Analysis of S-boxes

Many standard analyses have been depicted in the literature to evaluate the strength of S-boxes with fitting confusion. For testing security of S-boxes, many analyses like Bijectivity [13], SAC [16], Non-linearity [19], and differential and linear probability [7] are used in the checking process [20].

7.1. Bijectivity

S-boxes are bijective if all elements of $GF(2^8)$ are given as an input to a Boolean function and corresponding to each value, and a unique output of $GF(2^8)$ is obtained [21]. It is visible from the above 10 tables that those corresponding to a single input there is a single output. Thus, our all newly constructed 1600 S-boxes that are designed by means of four steps are 100% bijective.

7.2. Nonlinearity

Checking for Nonlinearity is essential to identify how much our S-box creates hurdle for linear attack. A function $\varphi(x)$ is defined from nth Boolean function $GF(2^n)$ to $GF(2)$. So the nonlinearity for function $\varphi(x)$ is defined in the form of

$$\delta_\varphi = min_{t \epsilon L_n} d_\delta(\varphi, t).$$

Here, members of L_n are all linear and affine functions as well, and $d_\delta(\varphi, t)$ represents the Hamming distance amongst φ and t.

The Nonlinearity through Walsh transform is predicted for function φ is

$$N.Lin(\varphi) = 2^{-n}(1 - max_{\sigma \epsilon GF(2^n)} \left| S_{<\varphi>}(\sigma) \right|).$$

For function φ the notation $S_{<\varphi>}(\sigma)$ is termed as cyclic spectrum and it is evaluated by [19],

$$S_{<\varphi>}(\sigma) = 2^{-n} \sum_{x \in GF(2^n)} (-1)^{\varphi(x) + x.\sigma},$$

where, $x.\sigma$ represents dot product for each $x, \sigma \in GF(2^n)$. The radius bound of Nonlinearity is $2^{n-1} - 2^{\frac{n}{2}-1}$. So, to counterattacks against linear cryptanalysis, the Nonlinearity of S-box should be close to its certain upper bound. It is represented in Table 9.

Table 9. Analysis of proposed S-boxes.

		S-box 1	S-box 2	S-box 3	S-box 4	S-box 5	S-box 6	S-box 7	S-box 8	S-box 9	S-box 10
Non-linearity	**Max**	106	106	107	106	106	108	107	107	108	108
	Min	100	100	98	100	100	96	97	100	100	96
	Average	103.25	103.375	103.875	102.875	103.625	102.375	103.625	104.125	104.375	103.875
Strict Avalanche Criteria (SAC)	**Max**	0.625	0.648438	0.601563	0.601563	0.59375	0.609375	0.625	0.585938	0.609375	0.578125
	Min	0.40625	0.0429688	0.429688	0.40625	0.382813	0.40625	0.375	0.40625	0.421875	0.390625
	Average	0.501221	0.503906	0.506592	0.501709	0.505615	0.502686	0.505859	0.499756	0.501221	0.499023
	S.D	0.0217624	0.0206006	0.0212301	0.0211541	0.0221775	0.0220563	0.0240203	0.0195796	0.0241116	0.0191802
Bit Independent Criteria (BIC)	**Min**	98	93	97	97	96	97	102	97	97	99
	Average	103.5	103.179	103.464	103.393	103.393	102.964	104.75	103.464	103.536	103.75
	S.D	2.5425	3.20773	2.89682	2.78182	3.2551	2.51399	1.7243	2.74489	2.93358	2.30876
BIC-SAC	**Min**	0.464844	0.482422	0.466797	0.470703	0.476563	0.462891	0.470703	0.46875	0.462891	0.458984
	Average	0.498117	0.503278	0.501744	0.503418	0.506836	0.504534	0.501395	0.503976	0.504534	0.50007
	S.D	0.0144139	0.011881	0.0193514	0.015125	0.0118402	0.0186685	0.0149098	0.0132311	0.0186685	0.0155593
Differential Probability	**D.P**	0.046875	0.015625	0.0390625	0.046875	0.0390625	0.046875	0.0390625	0.046875	0.046875	0.046875
Linear Probability	**Max**	162	161	165	163	161	162	163	159	162	160
	L.P	0.140625	0.144531	0.136719	0.144531	0.132813	0.140625	0.136719	0.144531	0.13281	0.125

7.3. Strict avalanche criterion (SAC)

SAC [16] interprets the evidence about altering bits in the output and that alteration is approximately half of the output bits that are altered. These bits are changed by changing one single bit of eight-bit input i.e., 0 to 1 or 1 to 0 [7]. This analysis is vital for examining the confusion aptitude of S-boxes.

7.4. Bit independent criterion (BIC)

If Exclusive OR of a Boolean function φ_j and two bits output of S-box φ_k is extremely Non-linear and fulfills the properties of SAC, then for each and every pair of output bit, the correlation coefficient is near to 0 by inverting one input bit. Thus, for BIC, use $\varphi_j\ xor\ \varphi_k\ for\ (j \neq k)$ [19] to determine whether it meets the criteria of nonlinearity and SAC.

7.5. Differential approximation probability

To conceptualize uniformity, apply the differential attack to input and notice the alteration in behavior and properties of output at the intermediate stage. Then, time linear and nonlinear responses due to differential attacks is observed. Here, corresponding input and output differentials are expressed by ∂x_i and ∂y_i. Mathematical expression is [7]

$$Dif.prob(\partial \mathrm{x} \to \partial \mathrm{y}) = \frac{\#\{x \in X | S(x) \oplus S(x \oplus \partial x) = \partial y\}}{2^m}.$$

The subsequent numerical values of $Dif.prob$ for the proposed S-box are inside a satisfactory range.

7.6. Linear approximation probability

At the output bit, the estimation for unbalancing events is monitored through drawing changes at input bits. ωx and ωy are applied to each parity of individual bits and analyze each individual response and its effect at the output stage. Mathematically [19],

$$Lin.prob = \max_{\omega x, \omega y \neq 0} \left| \frac{\#\{x | x.\omega x = S(x).\omega y\}}{2^n} - \frac{1}{2} \right|,$$

where 2^n represents cardinality and x represents input. The results are tabulated in Table 9. As smaller the $Lin.prob$, the more grounded the capacity of S-box for fighting against direct cryptanalysis attack. Thus, the consequence of $Lin.prob$ is better. The analyses of the proposed S-boxes are displayed in Table 9.

8. Comparison with prevailing S-boxes

The consequences of our proposed S-boxes and their average by means of comparison with others are depicted in Table 10. The proposed S-boxes' performance is concurrent with the aftereffects of the investigation for different S-boxes that are tried for comparison. The estimation of the proposed S-box is near to the ideal value that demonstrates its protection from attackers. The consequence of non-linearity [19] is near to Xyi and Skipjack S-box. An examination of SAC [16] for various S-boxes utilized as benchmarks is shown in this work. These outcomes run from a greatest value of 0.625 to at least 0.406, with an average estimation of 0.502. The average outcomes from SAC investigations yield esteems near the ideal estimation of 0.50. The subsequent numerical values of $Dif.prob$ [7] for the proposed S-box are inside a satisfactory range and bring it into comparison with various other S-boxes that are utilized in this paper for analysis. The smaller the $Lin.prob$, the more grounded the capacity of S-box for fighting against direct cryptanalysis attack. Thus, the consequence of $Lin.prob$ [19] is better.

Table 10. Comparison of performance indexes of proposed S-boxes 1–10 and different S-boxes.

S-boxes	Nonlinearity	SAC	BIC-SAC	BIC	DP	LP
AES S-box [3]	112	0.5058	0.504	112.0	0.0156	0.062
APA S-box [4]	112	0.4987	0.499	112.0	0.0156	0.062
Gray S-box [5]	112	0.5058	0.502	112.0	0.0156	0.062
Skipjack S-box [21]	105.7	0.4980	0.499	104.1	0.0468	0.109
Xyi S-box [19]	105	0.5048	0.503	103.7	0.0468	0.156
Residue Prime [8]	99.5	0.5012	0.502	101.7	0.2810	0.132
Proposed S-box 1	103.25	0.501221	0.498117	103.5	0.046875	0.140625
Proposed S-box 2	103.375	0.503906	0.503278	103.179	0.015625	0.144531
Proposed S-box 3	103.875	0.506592	0.501744	103.464	0.0390625	0.136719
Proposed S-box 4	102.875	0.501709	0.503418	103.393	0.046875	0.144531
Proposed S-box 5	103.625	0.505615	0.506836	103.393	0.0390625	0.132813
Proposed S-box 6	102.375	0.502686	0.504534	102.964	0.046875	0.140625
Proposed S-box 7	103.625	0.505859	0.501395	104.75	0.0390625	0.136719
Proposed S-box 8	104.125	0.499756	0.503976	103.464	0.046875	0.144531
Proposed S-box 9	104.375	0.501221	0.504534	103.536	0.046875	0.13281
Proposed S-box 10	103.875	0.499023	0.50007	103.75	0.046875	0.125
Average proposed results	103.538	0.502759	0.50279	103.539	0.041406	0.13789

9. Conclusions

It is concluded that this work is identified with the development of S-boxes. The whole process comprises a few stages i.e. Matrix multiplication, Affine Transformation, Substitution, Action of projective general linear group, and permutation with specific components of group $V_4 \times V_4$. The proposed strategy has numerous advantages. The principal advantage is that $GF(2^4)$ works behind the development of 8×8 S-boxes, and when cryptanalysts apply different inverse methods for code break, they will use $GF(2^8)$. Thus, they are not expected to break the code. Since the Galois field is cyclic, it implies one can develop every single other component of S-box with just a single generator. The second benefit is that it is known very well that diffusion is induced by permutation in a secure communication field. Thus, the shuffling by the permutations of cartesian product $V_4 \times V_4$ enhances the diffusion capacity of the cipher. In Section 3, with the proposed technique, we developed 1600 S-boxes subsequent to applying $V_4 \times V_4$ permutation. At that point, we chose 10 S-boxes arbitrarily, utilized it in AES calculations, and inspected their analyses in eminent cryptographic criteria. We inferred that our S-boxes have great cryptographic properties i.e., nonlinearity, differential probability, linear probability, SAC, and BIC. Since AES has used just a single S-box byte sub step and utilizes the same S-box in all, it relies on its key length. Hence, another advantage is that here we have 1600 boxes with great properties, chose 10 S-boxes, and use every one in distinctive rounds of AES. We have ${}^{1600}C_{10} \approx (2.945764438E + 37)$ possible outcomes to choose 10 S-boxes. Additionally, different steps of AES were changed. Therefore, we have new calculations and strong S-boxes that have crucial application in encryption algorithms of block cipher. We have given AES encryption and proposed encryption on which we have done work. In this way, it demonstrates that it is a profoundly secure framework, and in the event that somebody wants to break the code of this algorithm, they need to move through the opposite of all steps. That is the reason why it is difficult to break down the code of this proposed algorithm. Cryptanalysis requires numerous years to decode the message. Thus, they need to perform thousands or even millions of counts to register what 10 S-boxes have been chosen to conceal the information? This strategy is most noteworthy for anchoring information when two gatherings build up a secret communication with each other.

Use of AI tools declaration

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

Acknowledgments

The authors extend their gratitude to the deanship of scientific research of King Khalid University, for funding this work through a research project under grant R.G.P.2/238/44.

Conflict of interest

The authors declare no conflicts of interest.

References

1. J. Daemen, V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*, Berlin: Springer Science & Business Media, 2002. https://doi.org/10.1007/978-3-662-60769-5
2. D. Canright, A very compact S-box for AES, In: *Cryptographic hardware and embedded systems*, Berlin: Springer, 2005, 441–455. https://doi.org/10.1007/11545262_32
3. J. Rosenthal, A polynomial description of the Rijndael advanced encryption standard, *J. Algebra Appl.*, **2** (2003), 223–236. https://doi.org/10.1142/S0219498803000532
4. L. Cui, Y. Cao, A new S-box structure named affine-power-affine, *Int. J. Innov. Comput. I.*, **3** (2007), 751–759.
5. M. Tran, D. Bui, A. Duong, Gray S-box for advanced encryption standard, *Proceedings of International Conference on Computational Intelligence and Security*, 2008, 253–258. https://doi.org/10.1109/CIS.2008.205
6. E. Abuelyman, A. Alsehibani, An optimized implementation of the S-Box using residue of prime numbers, *Int. J. Comput. Sci. Net.*, **8** (2008), 304–309.
7. I. Hussain, T. Shah, H. Mahmood, M. Gondal, Construction of S_8 Liu J S-boxes and their applications, *Comput. Math. Appl.*, **64** (2012), 2450–2458. https://doi.org/10.1016/j.camwa.2012.05.017
8. I. Hussain, T. Shah, H. Mahmood, M. Gondal, U. Bhatti, Some analysis of S-box based on residue of prime number, *Proceedings of the Pakistan Academy of Sciences*, **48** (2011), 111–115.
9. P. Kumar, S. Rana, Development of modified AES algorithm for data security, *Optik*, **127** (2016), 2341–2345. https://doi.org/10.1016/j.ijleo.2015.11.188
10. T. Shah, A. Qureshi, A novel approach for generating small 8×8-bit S_4 S-boxes, *U.P.B. Sci. Bull., Series C*, **79** (2017), 153–162.

11. T. Shah, I. Hussain, M. Gondal, H. Mahmood, Statistical analysis of S-box in image encryption applications based on majority logic criterion, *Int. J. Phys. Sci.*, **6** (2011), 4110–4127. https://doi.org/10.5897/IJPS11.531
12. K. Erdmann, Blocks whose defect groups are Klein four groups, *J. Algebra*, **59** (1979), 452–465.
13. I. Hussain, T. Shah, H. Mahmood, A new algorithm to construct secure keys for AES, *Int. J. Contemp. Math. Sci.*, **5** (2010), 1263–1270.
14. D. Canright, A very compact S-box for AES, In: *Cryptographic hardware and embedded systems*, Berlin: Springer, 2005, 441–455. https://doi.org/10.1007/11545262_32
15. O. Sahoo, D. Kole, H. Rahaman, An optimized S-box for advanced encryption standard (AES) design, *Proceedings of International Conference on Advances in Computing and Communications*, 2012, 154–157. https://doi.org/1109/ICACC.2012.35
16. P. Mar, K. Latt, New analysis methods on strict avalanche criterion of S-boxes, *World Academy of Science, Engineering and Technology*, **48** (2008), 150–154.
17. I. Hussain, T. Shah, M. Gondal, W. Khan, H. Mahmood, A group theoretic approach to construct cryptographically strong substitution boxes, *Neural Comput. Appl.*, **23** (2013), 97–104. https://doi.org/10.1007/s00521-012-0914-5
18. T. Chandrasekharappa, Enhancement of confidentiality and integrity using cryptographic techniques, Ph.D, Manipal University, 2012.
19. A. Altaleb, M. 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
20. J. Cui, L. Huang, H. Zhong, C. Chang, W. Yang, An improved AES S-box and its performance analysis, *Int. J. Innov. Comput. I.*, **7** (2011), 2291–2302.
21. I. Hussain, T. Shah, M. Gondal, Y. Wang, Analyses of SKIPJACK S-box, *World Appl. Sci. J.*, **13** (2011), 2385–2388.