Multimedia Tools and Applications (2023) 82:16237–16256
https://doi.org/10.1007/s11042-022-14064-8

Modified advanced encryption standard (MAES) based on non-associative inverse property loop

Sadam Hussain[1] · Tariq Shah[1] · Adnan Javeed[1,2]

Received: 26 February 2021 / Revised: 23 September 2022 / Accepted: 9 October 2022 /
Published online: 2 November 2022

Abstract
In this paper, a cryptographic encryption standard is proposed whose model is same as presented in Rijndael Algorithm by Joan Daemen and Vincent Rijmen. The modification lies in the design of the Cipher, we have used inverse property (IP) loop instead of Extended Binary Galois Field (GF). The proposed mathematical structure is superior to GF in terms of complexity and has the ability to create arbitrary randomness due to a larger key space. Moreover, IP loop are non-isomorphic and have more than one Cayley table representation as compared to GF. This in result confirms the resistance against cryptanalytic attacks specifically on mathematical structures. The complete description of S-box, encryption and decryption of this cryptographic scheme is measured and evaluated critically to substantiate its multimedia applications.

Keywords Advanced encryption standard · Non-associative structure · Image privacy · Inverse property loop · Encryption analyses

1 Introduction

Communication over the globe is becoming basic need of populaces. The inventions of soft computing devices for communication drives are increasing exponentially as per the demand of the consumers. Speedy internet along with these devices increased the communicating behavior of the public. Variety of messages in terms of pictures, notifications of civil governments, undisclosed confidential military movement informations and sensitive medical reports are transported by individuals as well as organizations via electronic media. The ultimate loss and theft to the valuable data during transmission is causing serious concern among populaces.

✉ Adnan Javeed
ajaveed@math.qau.edu.pk

[1] Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan

[2] Government Associate College Daultala, Rawalpindi, Pakistan

Secure communication has attracted many research centers including national institute of standards NIST and military cyber units etc. to intervene in this grim matter. The prime objective is to alter the original information in non-readable form before transmitting it in an unsecure channel [34]. The receiver recovers the original message by using accurate set of keys along with de-ciphering mechanism. This journey is not very old as it started in late sixty's. Many recent developments in this field includes various renowned encryption standards. The encryption standards like blowfish [35], triple DES [11] and advanced encryption standard AES [14] are the most prominent enciphering standards of contemporary past. Among them, AES [14] is the most secure until now.

AES is a block cipher. The only nonlinear component of block cipher also known as substitution box (S-box) in AES is based on extended binary Galois Field. This fragment produces confusion in the cryptosystem. Which is one of the desired attribute acknowledged by the theory of Shannon [38], whereas the other one is diffusion which is achieved via column mixing, repetition of rounds and permutation. These two are used to gauge the strength of a cipher. Keeping this analogy, many recent developments have been published for the design of S-box. A brief literature review in terms of mathematical system is given hereafter.

The invention of S-box in our opinion is generally categorized into three major divisions of mathematical structures i.e. algebraic, chaotic and their merger. The foremost construction is based on algebra, specifically, it includes Galois field [9, 13–15], ring theory [18, 36, 37], and loop theory [17, 33]. The second mathematical system used for this purpose includes chaos [5, 25]. Its further division relies on continuous and discrete chaos. From discrete chaotic system literature reveals 1D chaotic system in [8, 28, 30, 39, 40], 2D chaotic system in [1, 29] and 3D or higher dimensional chaotic systems in [3, 10, 23, 32]. While continuous chaotic systems were used by the authors in [20–22]. Since current computing devices works on streams of numbers which only involves zero and one. For similarity such kind of sequences are mapped with the sequences generated by Deoxyribonucleic acid (DNA). Many articles include the construction based on DNA [12, 27]. Moreover, literature found some articles by the amalgamation of any two systems as explained above like [2, 6, 7, 19, 24, 26, 31].

All this motivated us to use such a mathematical structure that has superior properties required for the design of a cryptosystem. So Galois Field the crux of the matter in AES is replaced here in this article by non-associative structure from loop theory. Ultimate desirable traits in a cryptosystem are ergodicity, randomness, larger key space and complexity. All of these are attained with good strength by using IP loop theory. IP-Loop have many representations in terms of Cayley table as compared to one Cayley representation in GF. The inclusion of non-associative structures in the design of S-boxes will surely open a new gateway in cryptography.

The rest of article is organized in the following manner. Ongoing section explains the introduction. Section 2 explains the mathematical structure involved in the design of cryptosystem. Section 3 proposes the construction algorithm for S-box as well as the analyses of S-box. In Section 4, we have presented modified form of AES using IP-loop whereas Section 5 explains the reverse steps for the decryption process. Section 6 explains the key schedule for both encryption and decryption. In Section 7 detailed critical analyses are accomplished for the encryption scheme to establish its efficacy. Section 8 elaborates some examples for application purpose whereas whole piece of article is concluded in Section 9.

2 Preliminaries for IP-loop

This manuscript focuses on a new direction of mathematical structure i.e. inverse property loop (IP-Loop) [4]. A groupoid satisfying the left and right cancellation laws is known as a quasigroup. A quasi group L is a loop if and only if it has a right and left identity. Loop theory is a very comprehensive field. Almost every work on loops is considered as special cases. Steiner loops was one of the initial classes of loops, which postulates for a, $b \in L$:

$$(ab)b = a \tag{1}$$

$$a(ab) = b \tag{2}$$

In loops every element a has an inverse b (left inverse) such that $ba = e$ and an inverse c (right inverse) such that $ac = e$. In the case of Steiner loops both b and c are just a.

Lemma An IP-loop is a loop that has the inverse property [4], that is, a loop having the inverse such that for all elements a and b:

$$a^{-1}(ab) = b = (ba)a^{-1} \tag{3}$$

A homomorphism on a loop L is a single valued operation preserving from a loop to a same or different loop. The set of all those elements of loop L which are mapped with zero are called the Kernel of Homomorphism and it is normal sub loop of L.

Theorem A sub loop S of loop L is called normal sub loop If and only if for all $a \in L$, $aS = Sa$. Isomorphism theorems are also applicable on loop theory. The reason to choose non associative structures is that it has more variety of algebraic structures as compared to associative structures due to lesser constraints.

3 Design for $n \times n$ (S-boxes)

The substitution boxes are the basic building block in private key cryptosystems. All the symmetric key cryptosystems use the process of substitution boxes to create confusion in the algorithm. As a result, a large number of methods give the idea to design S-Boxes. For this purpose, cryptographers use different algebraic structure in order to increase the security of S-Boxes. So, Binary Galois Field Extension $GF(2^8)$, Local associative Algebras, Pseudo Random Number Generators (PRNG) and Elliptic curves have been used to construct S-box. Here an S-box is constructed over Inverse property Loop (IP-Loop). Following transformations have been used in the design of S-Box.

1. Inversion map: Inverts the elements of IP-Loop by using the mapping, $\sigma : L \to L$ (as)

$$\sigma(x) = x^{-1}, \forall\ x \in L \tag{4}$$

2. Right Translation map: Operate right translation map with a fixed element of loop to inverse generating in 1st step by mapping, $\varphi_u : L \rightarrow L$ (as)

$$\varphi_u(x) = (u * x) \oplus v, \; x \in L \tag{5}$$

where u, v are fixed elements of L.

3. Compose both of these mappings $\varphi_{\mathring{u}}\sigma : L \rightarrow L$ (as)

$$\varphi_u(\sigma(x)) = \left(u * x^{-1} \right) \oplus v \tag{6}$$

The 1st step inverts the elements of L and 2nd step perform the left translation with XOR of fixed element of L. The composition of these two steps gives us the elements of the required S-box. We can produce a variety of S-boxes by changing the values of elements $u, v \in L$. Table 1 gives the IP-Loop of order 16. This is a non-associative Loop in which the inverse of zero element is itself. While Tables 2 and 3 gives construction of S-box over loop of order 16, and Table 4 represents S-box based on loop of order 256.

The mapping to construct elements of S-box is given by the following equation.

$$\varphi_u(\sigma(x)) = 7*(x)^{-1} \oplus 11 \tag{7}$$

The generated S-box over L_{16} is given by the following table.

In the same way the S-box used in the AES algorithm can be generated from the Non-associative IP-Loop of order 256. Here we construct the S-box by fixing the elements 231, 181 $\in L$. The composition map is given by the equation:

$$\varphi_{231}(\sigma(x)) = \left(231 * x^{-1} \right) \oplus 181 \tag{8}$$

The process is given as follows:

Table 1 Inverse property loop of order 16

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

Table 2 Construction of proposed S-box over L_{16}

L_{16}	$\varphi_u(\sigma(x))=7*(x)^{-1}\oplus 11$	Entries of Proposed S-Box
0	$\varphi_u(\sigma(0))=7*(0)^{-1}\oplus 11$	12
1	$\varphi_u(\sigma(1))=7*(1)^{-1}\oplus 11$	15
.	.	.
.	.	.
.	.	.
14	$\varphi_u(\sigma(14))=7*(14)^{-1}\oplus 11$	13
15	$\varphi_u(\sigma(15))=7*(15)^{-1}\oplus 11$	9

Table 3 Proposed S-box in the form of 4 × 4 (matrix)

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

4 Description of encryption algorithm

In this cipher, the encryption scheme consists of 10 iterative rounds. The input of 128-bits is taken as state array. Before the start of 1st round, the encipher key is operated which was initially selected to encrypt data and then round function is performed. In this cipher, round function consists of 10 rounds. Final round is slightly different from others. After doing all of these rounds the result is output. Each round is performed with different keys which are generated in a key schedule by using the initial encipher key. The encryption scheme uses the following transformation for encryption. Substitution Bytes (Sub Bytes), Shifting of Rows (Shift Rows), Mixing of Columns (Mix Columns), Round Key Binding (Round Key Binding). In 10th round, the Mix Column () operation is not performed.

4.1 Sub bytes transformation

In this transformation, the bytes of state are substituted with bytes of S-Box (The process of S-Box designing is given above). It is a non-linear step, i.e. for the bytes k_i and k_j of the state $Sub\ Byte(k_i)\ *\ Sub\ Byte(k_j)\ \neq\ Sub\ Byte(k_i\ *\ k_j)$. In this transformation an invertible Substitution box is used in this transformation. The Sub Bytes () transformation is bijective. Each element of loop L is mapped onto some element of loop L. So, each element of loop L can be inverted in the decryption process. For example, the byte of a state i.e. $k_i = (65)_{16}$ can be substituted by using S-box in Table 5 as:

Table 4 Construction of proposed S-box over L_{256}

L_{256}	$\varphi_{231}(\sigma(x))=(\ 231*x^{-1})\oplus 181$	*Proposed S−box*
0	$\varphi_{231}(\sigma(0))=(\ 231*0^{-1})\oplus 181$	147
1	$\varphi_{231}(\sigma(1))=(\ 231*1^{-1})\oplus 181$	71
2	$\varphi_{231}(\sigma(2))=(\ 231*2^{-1})\oplus 181$	1
.	.	.
.	.	.
.	.	.
254	$\varphi_{231}(\sigma(254))=(\ 231*254^{-1})\oplus 181$	117
255	$\varphi_{231}(\sigma(255))=(\ 231*254^{-1})\oplus 181$	121

Table 5 Proposed S-box in the form of 16 × 16 (matrix)

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

Similarly, one can construct bulk of S-boxes by using different values of $u, v \in L$.

$$S\left((65)_{16}\right) = (54)_{16} \tag{9}$$

S-Box has no fixed points i.e. there is no byte in L such that $S(k_i) = k_i$. Even the identity element of loop L is substituted to some element of L, other than identity element.

$$S\left((0)_{16}\right) = (147)_{16} \tag{10}$$

4.2 Shift rows transformation

In the process of the Shift Rows, there is a byte shift which is cyclic across the rows of state. 1st row will be unchanged. The Shift Rows transformation is given as follows:

$$j^*_{r,c} = j_{r,(c+shift(r,Nb))modNb}, \;\; 0 \leq r < 4, 0 \leq c < Nb \tag{11}$$

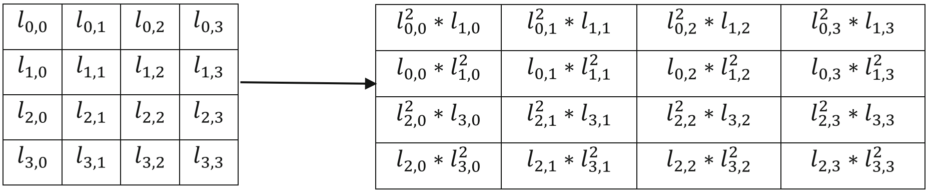
Here $Nb = 4$ and r shows the row number which decides $shift(r, Nb)$ (shift value). Shift value is given by

$$shift(1,4) = 1; \;\; shift(2,4) = 2; \;\; shift(3,4) = 3; \tag{12}$$

Shift Rows operation rotates the bytes of rows towards the right according to the rule given above. In this operation, Row n is moved n rounds right. So, every new column which is generated after this operation is created with bytes from all columns of state. This transformation keeps the columns away from linear independence. The Shift Rows transformation weakens the division of cipher into four independent block ciphers (Figs. 1, 2 and 3).

4.3 Mix columns transformation

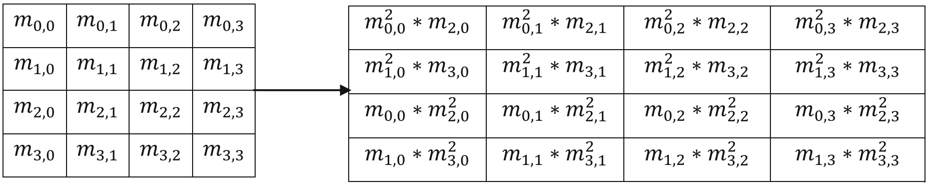
It is a column-wise transformation. This transformation is performed with the help of associative property of the IP-Loop. We have divided this transformation into further three sub-



$l_{0,0}$	$l_{0,1}$	$l_{0,2}$	$l_{0,3}$
$l_{1,0}$	$l_{1,1}$	$l_{1,2}$	$l_{1,3}$
$l_{2,0}$	$l_{2,1}$	$l_{2,2}$	$l_{2,3}$
$l_{3,0}$	$l_{3,1}$	$l_{3,2}$	$l_{3,3}$

$l_{0,0}^2 * l_{1,0}$	$l_{0,1}^2 * l_{1,1}$	$l_{0,2}^2 * l_{1,2}$	$l_{0,3}^2 * l_{1,3}$
$l_{0,0} * l_{1,0}^2$	$l_{0,1} * l_{1,1}^2$	$l_{0,2} * l_{1,2}^2$	$l_{0,3} * l_{1,3}^2$
$l_{2,0}^2 * l_{3,0}$	$l_{2,1}^2 * l_{3,1}$	$l_{2,2}^2 * l_{3,2}$	$l_{2,3}^2 * l_{3,3}$
$l_{2,0} * l_{3,0}^2$	$l_{2,1} * l_{3,1}^2$	$l_{2,2} * l_{3,2}^2$	$l_{2,3} * l_{3,3}^2$

Fig. 1 Mix Columns () 1st Sub Operation



$m_{0,0}$	$m_{0,1}$	$m_{0,2}$	$m_{0,3}$
$m_{1,0}$	$m_{1,1}$	$m_{1,2}$	$m_{1,3}$
$m_{2,0}$	$m_{2,1}$	$m_{2,2}$	$m_{2,3}$
$m_{3,0}$	$m_{3,1}$	$m_{3,2}$	$m_{3,3}$

$m_{0,0}^2 * m_{2,0}$	$m_{0,1}^2 * m_{2,1}$	$m_{0,2}^2 * m_{2,2}$	$m_{0,3}^2 * m_{2,3}$
$m_{1,0}^2 * m_{3,0}$	$m_{1,1}^2 * m_{3,1}$	$m_{1,2}^2 * m_{3,2}$	$m_{1,3}^2 * m_{3,3}$
$m_{0,0} * m_{2,0}^2$	$m_{0,1} * m_{2,1}^2$	$m_{0,2} * m_{2,2}^2$	$m_{0,3} * m_{2,3}^2$
$m_{1,0} * m_{3,0}^2$	$m_{1,1} * m_{3,1}^2$	$m_{1,2} * m_{3,2}^2$	$m_{1,3} * m_{3,3}^2$

Fig. 2 Mix Columns () 2nd Sub Operation

transformations. In the sub-transformation, two bytes of a column are mixed at a time. Mathematically the transformations are given as follows:

$$l_{r,c}^{2} * l_{r+1,c} = l_{r,c}^{*} \tag{13}$$

$$l_{r,c} * l_{r+1,c}^{2} = l_{r+1,c}^{*} \tag{14}$$

By using the above transformations, the 1st Sub-transformation is given in following figure.

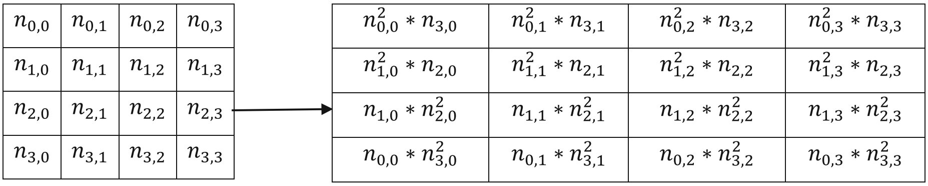
The 2nd Sub-transformation is given in following figure. In this transformation we let

$$l_{i,j}^{2} * l_{h,k} = m_{i,k} \tag{15}$$

The 3rd Sub-transformation is given in the following figure. In this transformation we let

$$m_{i,j}^{2} * m_{h,k} = n_{i,k} \tag{16}$$

This transformation operates on four bytes as input and the resulting output is also four bytes and uses an invertible linear transformation. In this transformation, each input byte modifies the four bytes of output. Shift Rows () and Mix Columns () transformations combined provides diffusion in the cipher.



$n_{0,0}$	$n_{0,1}$	$n_{0,2}$	$n_{0,3}$
$n_{1,0}$	$n_{1,1}$	$n_{1,2}$	$n_{1,3}$
$n_{2,0}$	$n_{2,1}$	$n_{2,2}$	$n_{2,3}$
$n_{3,0}$	$n_{3,1}$	$n_{3,2}$	$n_{3,3}$

$n_{0,0}^2 * n_{3,0}$	$n_{0,1}^2 * n_{3,1}$	$n_{0,2}^2 * n_{3,2}$	$n_{0,3}^2 * n_{3,3}$
$n_{1,0}^2 * n_{2,0}$	$n_{1,1}^2 * n_{2,1}$	$n_{1,2}^2 * n_{2,2}$	$n_{1,3}^2 * n_{2,3}$
$n_{1,0} * n_{2,0}^2$	$n_{1,1} * n_{2,1}^2$	$n_{1,2} * n_{2,2}^2$	$n_{1,3} * n_{2,3}^2$
$n_{0,0} * n_{3,0}^2$	$n_{0,1} * n_{3,1}^2$	$n_{0,2} * n_{3,2}^2$	$n_{0,3} * n_{3,3}^2$

Fig. 3 Mix Columns () 3rd Sub Operation

4.4 Round key binding transformation

Round Key Binding () transformation is the loop operation in which the bytes of state matrix are combined with bytes of key. If k_i, $0 \le i < 16$ are key bytes and s_i, $0 \le i < 16$ are the state byte. Round Key Binding () transformation is as follows:

$$R_{k_i}(s_i) = k_i{*}s_i \tag{17}$$

Each Round Key consists of the 4 words. In each round, different key is used which is constructed by the process of Key Schedule. Four words of round key and the columns of state combined as follows:

$$\left[l^*_{0,c}, l^*_{1,c}, l^*_{2,c}, l^*_{3,c}\right] = \left[w_{round*Nb+c}\right] \; * \; \left[l_{0,c}, l_{1,c}, l_{2,c}, l_{3,c}\right] \; 0{\le}c < 4$$

Here, keywords are denoted by $[w_i]$ and number of round is in the range $0 \le round \le Nr$. Before the start of round function, initial key is added where $round = 0$. In all rounds, the round keys are added where $1 \le round \le Nr$ and $l = round.\ Nb$.

5 Inverse cipher

All the transformations used above are invertible and one can easily find the plaintext from the cipher text applying the inverse process. The inverse cipher or Decryption cipher of the encryption algorithm consists of the following transformations Inv. Shift Rows (), Inv. Sub Bytes (), Inv. Mix Columns () and Inv. Round key Binding (). These transformations are explained in the following paragraphs.

5.1 Inverse shift rows transformation

The inverse process of Shift Rows is the Inv. Shift Rows transformation. In this transformation, the bytes of the state are rotated left cyclically according to the rule except the 1st row. 1st row $r = 0$ will be unchanged. The rotation of bytes of following rows are given by $Nb - shift\ (\ r,\ Nb\ )$ and $shift\ (\ r,\ Nb\ )$ depends upon the rows number as follows:

$$shift(1,4) = 1;\;\; shift(2,4) = 2;\;\; shift(3,4) = 3; \tag{18}$$

5.2 Inverse sub bytes () transformation

Inv. Sub Bytes operation is the inverse process of the Sub Bytes transformation, in which the bytes of the state are updated from the bytes of inverse Substitution box given in Table 6. The process of constructing the inverse S-box is same as constructing S-box by using inverse map. First applying inverse of linear map and then apply inversion map for the construction of inverse S-box.

$$(\varphi_u(\sigma(x)))^{-1} = \sigma^{-1}\left({\varphi_u}^{-1}(x)\right) = \left(u^{-1}{*}(x \oplus v)\right)^{-1} \tag{19}$$

Table 6 Proposed inverse S-box in the form of 16 × 16 (matrix)

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

Where $u, v \in L$ are the fixed elements of IP-Loop that are used for the construction of S-box. As $u, v \in L$ are the fixed elements of IP-Loop. So, which structure of IP-loop is to be used at the decryption end is only known to an authorized person. Thus receiver can easily find the inverse of u to find the inverse S-box for decryption.

In the S-box, we have constructed in Sub bytes, $u = 231$ and $v = 181$ used as fixed elements of IP-Loop. The inverse of 231 is 141 in IP-Loop that have been used. So, the inverse mapping for the construction of inverse S-box in Inv. Sub Bytes () step is given by:

$$\sigma^{-1}\left(\varphi_u{}^{-1}(x)\right) = \left(231^{-1}*(x \oplus 181)\right)^{-1} = (141*(x \oplus 181))^{-1} \tag{20}$$

The inverse S-box is given in the following table.

5.3 Inverse mix columns transformation

Inv. Mix Columns is inverse process of Mix Columns operation. This transformation applies on the state in column-wise manner. The Mix Column transformations are given by:

$$l^2_{r,c} * l_{r+1,c} = l^*_{r,c} \tag{21}$$

$$l_{r,c} * l^2_{r+1,c} = l^*_{r+1,c} \tag{22}$$

Here, $l^*_{r,c}, l^2_{r+1,c}$ are the output values. By using the power-associativity and di-associativity of the IP-Loop, the inverse of Mix Column transformation can be easily done, which is described as follows:

From *Eq.* (22), we can get

$$l_{r,c} = l^*_{r+1,c} * \left(l^{-1}_{r+1,c}\right)^2 \tag{23}$$

By using this value in Eq. (23), we get

$$\begin{gathered}
\left(l^{*}_{r+1,c}*\left(l^{-1}_{r+1,c}\right)^{2}\right)^{2}*l_{r+1,c}=l^{*}_{r,c}\\
\left(l^{*}_{r+1,c}\right)^{2}*\left(l^{-1}_{r+1,c}\right)^{4}*l_{r+1,c}=l^{*}_{r,c}\\
\left(l_{r+1,c}\right)^{3}=\left(l^{*}_{r+1,c}\right)^{2}*\left(l^{*}_{r,c}\right)^{-1}\\
l_{r+1,c}=\left(\left(l^{*}_{r+1,c}\right)^{2}*\left(l^{*}_{r,c}\right)^{-1}\right)^{1/3}\\
l_{r,c}*\left(\left(\left(l^{*}_{r+1,c}\right)^{2}*\left(l^{*}_{r,c}\right)^{-1}\right)^{1/3}\right)^{2}=l^{*}_{r+1,c}\\
l_{r,c}=l^{*}_{r+1,c}*\left(\left(l^{*}_{r,c}*\left(\left(l^{*}_{r+1,c}\right)^{2}\right)^{-1}\right)^{1/3}\right)^{2}
\end{gathered}\tag{24}$$

5.4 Inverse round key binding transformation

The inverse process of Round Key Binding transformation is called Inv. Round Key Binding transformation. This transformation is also a loop operation in which the state matrix bytes are combined with inverses key bytes. For $0 \le i < 16$, k_i and s_i are key and state bytes respectively. Inv. Round Key Binding () transformation is given as follows:

$$R_{k_i^{-1}}(s_i)=k_i^{-1}*s_i \tag{25}$$

For inverse round key words are combined with state columns as follows:

$$\left[l^{*}_{0,c},l^{*}_{1,c},l^{*}_{2,c},l^{*}_{3,c}\right]=\left[w_{round*Nb+c}\right]^{-1}*\left[l_{0,c},l_{1,c},l_{2,c},l_{3,c}\right]\ 0\le c<4$$

Here, keywords are denoted by $[w_i]$ and the number of rounds are in the range $0 \le round \le Nr$.

6 Key schedule

In the Encryption algorithm, 128-bit key is used, which is combined with the state in each round as there are 10 rounds in the cipher and key ties with state 10 times in each encryption process. It is detriment to tie same key in each round. Some transformations are applied on the key to make non-linearity in key to use it in different rounds. This process of key transformation is known as Key Schedule. Here, we discuss the expansion of cipher key K of length 128-bits and generates 11 subkeys (one initial key and 10 new subkeys), one key(initial key) is for key whitening used before the start of round function and remaining 10 sub keys for 10 rounds.

Algorithm takes the cipher key K as input and break it in four blocks or rows of 16 bytes, called words w []. Then apply the transformations known as Word Rotation (), Sub Word () and Round Constant Binding [i] and generates 44 words denoted by w[0], w[1], ………, w[42], w[43]. The bytes of the initial key are k_0, k_1,k_2, ………k_{15}. Where K_0 the original is key selected for the encryption. The bytes of this key generated the first four elements of the key array w. The other elements of array can be computed as follows:

It is clear that the first word of the sub key $w[4i]$, i = 1, 2, 3, …10 is computed as follows:

$$w[4i] = w[4(i{-}1)]^*\ g(\ w[4i{-}1]\ )$$

Here g() is function which is linear. It takes four bytes as input and output is also four bytes. A recursive process is used to construct the other three words of the subkey.

$$w[4i+j] = w[4\ (i{-}1)\ ]^*\ w[4i{-}1+j],\ \ i = 1, 2, \ldots\ldots, 10,\ \ j = 1, 2, 3$$

The function g() consists of the 3 operations Sub Word (), Word Rotation () and Round Constant Binding [i].

Sub word () is an operation in which input word consists of the four bytes and manipulates each of these bytes from the S-box and produce four-byte output word.

Word Rotation () operation takes four byte input word i.e. $[a_0,\ a_1,\ a_2,\ a_3]$ and apply a cyclic permutation on the bytes of the word and produce a four byte output word i.e. $[a_1,\ a_2,\ a_3,\ a_0]$.

Round constant Binding [i] operation consists of a round constant word array $[\{ii\},\ \{00\},\ \{00\},\ \{00\}]$ where $\{ii\}$ is an element of L and apply as Right translation to the 1st byte of the word $w[4i]$. The values of round coefficient $\{ii\}$ for the sub keys of the different rounds are given by:

$$\begin{aligned} RoundConstantTie[1] &= (11)_{16}\\ RoundConstantTie[2] &= (22)_{16}\\ RoundConstantTie[3] &= (33)_{16}\\ &\vdots\\ RoundConstantTie[10] &= (AA)_{16} \end{aligned}$$

This function g() is used for two purposes:

1- To add the nonlinearity in the Key Schedule.
2- To remove the symmetry in the AES.

6.1 Inverse key schedule

Inverse Key Schedule is the inverse process of Key Schedule. All the transformations in Key Schedule are invertible and can easily be inverted by using inverse mappings. The inverse process of Word Rotation () is the same as inversion of the Sub Bytes () step as explained in the previous section. The inverse of the Sub Word () step is also the same as the inverse of Sub Bytes () step as explained in the previous section.

The operation of Binding Round Constant () is invertible. Let $y = \{ii\}$ is the round constant and operated to the 1st byte x of the word $w[4i]$ and the new byte generated is z given by:

$$z = y\ *\ x \tag{26}$$

Since y is the element of L and L is IP-Loop. So, its inverse is also an element of IP-Loop L. i.e. $y^{-1}\ \in\ L$. The inverse mapping is given by:

$$\begin{gathered} y^{-1} *\ z = y^{-1} *\ (\ y\ *\ x\ ) \\ x = y^{-1} *\ z \end{gathered} \tag{27}$$

Here, y is round constant and z is the output byte, both are known to the person at decryption end. So, he can easily find the original byte by the process mentioned above. The pseudo code of the algorithm is as follow.

7 Security analysis of proposed S-box

7.1 Non-linearity (NL)

Non-linearity is defined in n-variable as the minimum of the hamming distance among the set of all non-regular linear combos of issue feature and the set of all affine features on GF(2^n). Whereas, the whole rely of positions at which the corresponding output is extraordinary is the Hamming distance. The proposed scheme presented in this section outcomes in excessive non-linearity cost.

The generated S-box by means of the proposed scheme has the avg. Non- linearity is 111.5 (Table 7). Now compares the nonlinearities of other designed S-boxes generated with the aid of the chaos-primarily based method. It's proven from the table the common values of other methods are smaller than our proposed scheme value.

7.2 Differential approximation probability (DP)

The nonlinear transformation of S-box should ideally have differential uniformity. An input differential Δx_i should uniquely map to an output differentialΔy_i, thereby ensuring a uniform mapping probability for each i.

7.3 Linear approximation probability (LP)

The LP is the maximum imbalance value of the case variance. The parity of the input bits selected for the mask is equal the parity of the output bits selected for the mask.

Table 7 Nonlinearity of proposed S-box and comparison with other well-known S-boxes

S-box	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	Average
Proposed	**112**	**112**	**110**	**112**	**112**	**112**	**112**	**110**	**111.5**
AES	112	112	112	112	112	112	112	112	112
S_8 AES	112	112	112	112	112	112	112	112	112
Ref [17]	108	105	110	104	106	106	106	110	106.87
Ref [26]	106	106	106	106	106	106	108	108	106.5
Ref [22]	112	110	112	112	112	110	112	112	111.5
Ref [19]	106	108	110	110	108	104	100	108	106.75

7.4 Bit Independence criterion (BIC) and strict avalanche criterion (SAC)

In the case of a given set of avalanche vectors, all the avalanche variables should be paired skillfully independently for all other statistical properties due to the performance of the BIC for an S-field. To examine the output of an S-box, while keeping some of its input bits unchanged, we are implementing the BIC definition for an S-container. SAC is the general statement of the effect and completeness of avalanche. If an unmarried input bit is inserted, all output bits will deviate with a 1/2 probability (Tables 8 and 9).

7.5 Majority logic criterion test

Majority logic criterion (MLC) explain the comparison between plain image and encrypted image and give the accurate evaluation of encryption scheme. Majority logic criterion is explained in [16], mostly homogeneity, entropy, energy, contrast and correlation are measured in MLC analysis. The formulations and explanation of these analysis are given in [16]. The results of these analysis show the performance and strength of encryption scheme and hence used S-boxes. For these analyses we used here 256 × 256-pixel Lena image. Table 10 represent the MLC analysis Lena grey image encrypted by 16 × 16 S-box. In this table comparison with other well-known S-boxes are also given which show that our proposed technique has better results and is good for encryption. Encrypted image and histogram of Lena with newly designed S-box and comparison with other well-known S-boxes are given in Fig. 4.

7.6 Security analysis of proposed encryption algorithm

All the cryptographic primitives are used for the sack of information security. The modern advancement in cryptanalysis techniques and computation speeds, the security of many cryptosystems is compromised. So, the cryptographers are working to construct new secure cryptographic primitives and improving the structures of the existing cryptographic primitives to meet the security needs of this era. Therefore, new techniques are applied in this field such as the arrival of quantum cryptography. In quantum cryptography, quantum bits are used, whose values are not restricted at 0 and 1 but can be varied between 0 and 1. This is the most

Table 8 Pseudo code of the algorithm

```
Encryption (byte in [16], byte out [16], key w [44])
 Begin
Byte state [16]
                State = in
Round Key Addition (state, & w [0-3])
                For $i = 1$ , step 1 to 9
Sub Bytes (state)
Rows Shift (state)
Columns Mix (state)
Round Key Addition (state, & w [4-40])
                    End for
Sub Bytes (state)
Rows Shift (state)
Round Key Addition (state, & w [41-44])
                 State= Out
               End
```

Table 9 Algebraic analyses of proposed S-box and comparison with other well-known S-boxes

S-Box	BIC			SAC			LP		DP
	Average	Min.	Square Deviation	Min.	Average	Square Deviation	Max Value	Max LP	Max DP
Proposed	**111.6**	**110**	**0.765**	**0.437**	**0.487**	**0.016**	**146**	**0.070**	**0.023**
AES	112	112	0	0.390	.493	0.020	144	0.062	0.0156
S8 AES	112	112	0	0.462	.500	0.015	144	0.062	0.0156
Ref [17]	106.107	102	1.877	0.437	0.509	0.013	157	0.113	0.0312
Ref [26]	104.071	100	2.234	0.421	0.500	0.018	162	0.132	0.039
Ref [22]	111.3	110	0.934	0.437	0.505	0.016	146	0.070	0.015
Ref [19]	106.27	104	1.578	0.401	0.504	0.018	161	0.125	0.0267

advance form of cryptography and many cryptographers are working in this field. Hence designing new complex and ambiguous cryptosystem are need of epoch. Some new foundations are also introduced in modern cryptography.

In this paper, we have presented a new scheme for encryption in symmetric key cryptosystem. This new scheme worked on the lines of Rijndael Algorithm (AES) but based on a different algebraic structure. It also uses a key of length 128-bits and encrypts a 128-bit block of data at a time. The encryption scheme consists of 10 rounds as AES. Each round contains the four components Sub Bytes () Transformation, Shift Rows () Transformation, Mix Columns () Transformation and Round Key Binding () Transformation. 10 different sub keys are generated by a Key Schedule to use them in each round of round function. Therefore, as for as the internal structure of this scheme, it has the same security parameters as in AES. But in this cipher, we have used a different algebraic structure known as Non-associative Moufang Loop of order 256 instead of the Galois field $GF2^8$. That makes it different from AES and in some prospectus more secure. In the complete cipher scheme, we have used the binary operation, from which the Non-associative IP-Loop is formed, binary multiplication under modulo primitive irreducible polynomial (Table 11).

The main points of new algorithm are:

1- This algorithm also used key of 128 bits which is enough secure under the brute force attack due to large key space of 2^{128}.

2- It is simple and flexible cipher with good performance.

3- The cipher is designed in such a way that it can protect it against known attack and conservative design.

Table 10 Results of MLC analyses by 16 × 16 S-box

S-Boxes	Entropy	Contrast	Correlation	Energy	Homogeneity	MAD
Proposed	**7.9353**	**9.9764**	**.0487**	**.0161**	**.4131**	**38.3543**
Ref. [33]	7.9633	8.5969	.0019	.0174	.4070	38.5639
AES [14]	7.7301	7.3220	.0879	.0244	.4835	36.3630
S_8 AES	7.7094	8.1685	.2309	.0227	.4870	43.5662
Prime	7.6595	6.3683	.0996	.0260	.4984	36.3084

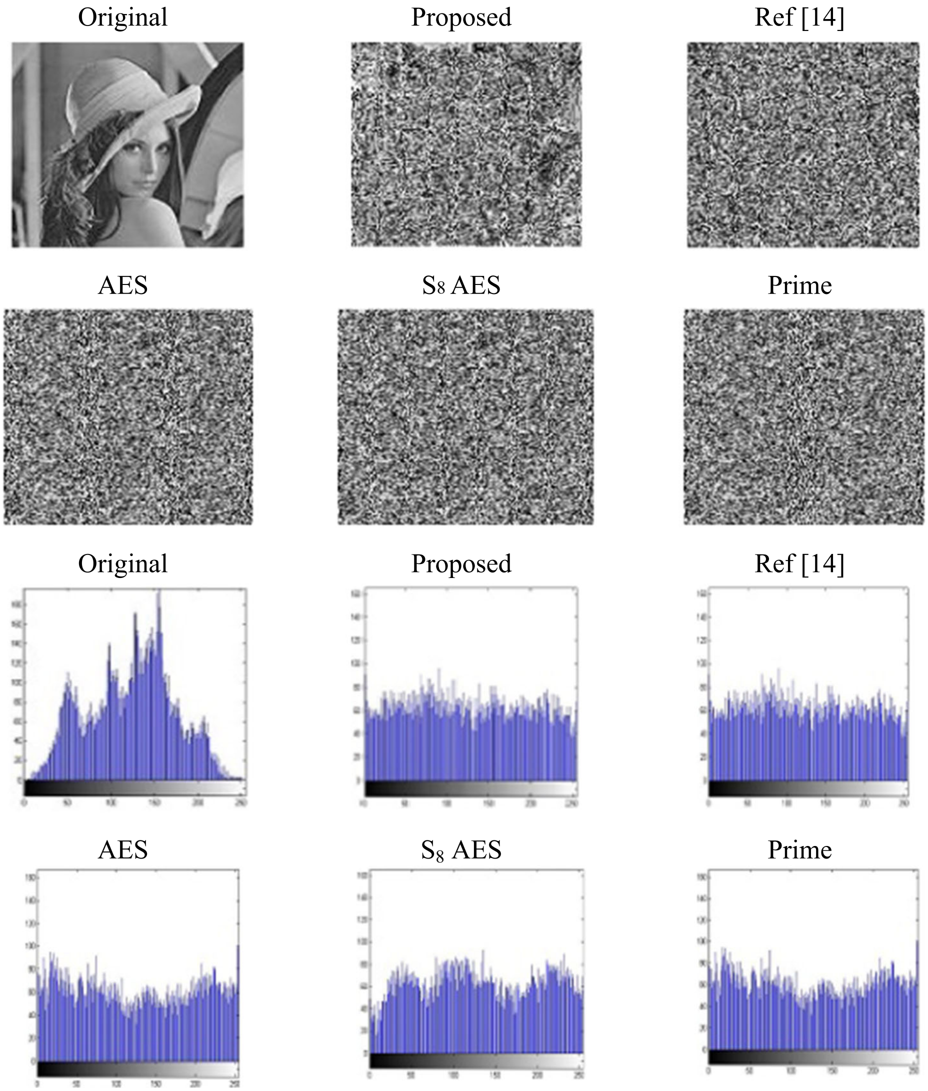


Fig. 4 Encrypted images and their respective histograms

4- The new cipher does not have only 128 bits key. But the loop of order 256 is also used as a key. Because without any knowledge of Loop used in Cipher, No one can decipher the text even if he has the knowledge of key.

Table 11 Comparison of No. of binary Galois fields and non-associative IP-Loop

N	8	16	32	64	128	256
$M(n)$	0	5	71	4262	?	?
$GF(n)$	1	1	3	3	9	8

5- In AES, we have only limited structures of Binary Galois Field of order 256. But in this cipher, we have used Moufang Loop of order 256, of which we have large number of Moufang loops of order 256.

6- Since in this cipher the binary operation depends upon the Loop, which is used in our encryption scheme. This loop is non-commutative, in which the same numbers operated in different way can give us different results. So, it is also difficult to get any information from the energy consumed in this operation.

7- This study will bring the cryptographers toward the algebraic structures other than Binary Galois Field and diversify the basis of the symmetric cryptography from the Binary Galois Field.

8- This study will also boost up the research in the Non associative Algebraic Structures and their uses in different scientific and technological areas.

7.7 Time analysis

The speed of encryption algorithm is very important criteria, especially on large data encryption operations. To demonstrate that the proposed encryption algorithm speed performance in this study, the comparisons are made with only the chaotic system of the encryption algorithm and AES algorithm. The encryption, decryption and the total time of encryption algorithms are given in Table 12. According to test results, it is seen that it performs encryption and decryption process approximately 20 times less than the AES algorithm and very close with the chaos encryption algorithm.

8 Cipher example

The following illustration shows the values in the State array as the Cipher progresses for a block length and a Cipher Key length of 16 bytes each (i.e., $Nb = 4$ and $Nk = 4$).

$$Input = Logical\ Thoughts$$
$$In\ Hexadecimal\ form : 76\ 111\ 67\ 69\ 63\ 61\ 108\ 20\ 54\ 68\ 111\ 75\ 67\ 68\ 74\ 73$$
$$Cipher\ Key = Pure\ Mathematics$$
$$In\ Hexadecimal\ form : 50\ 75\ 72\ 65\ 20\ 77\ 61\ 74\ 68\ 65\ 109\ 61\ 74\ 69\ 63\ 73$$

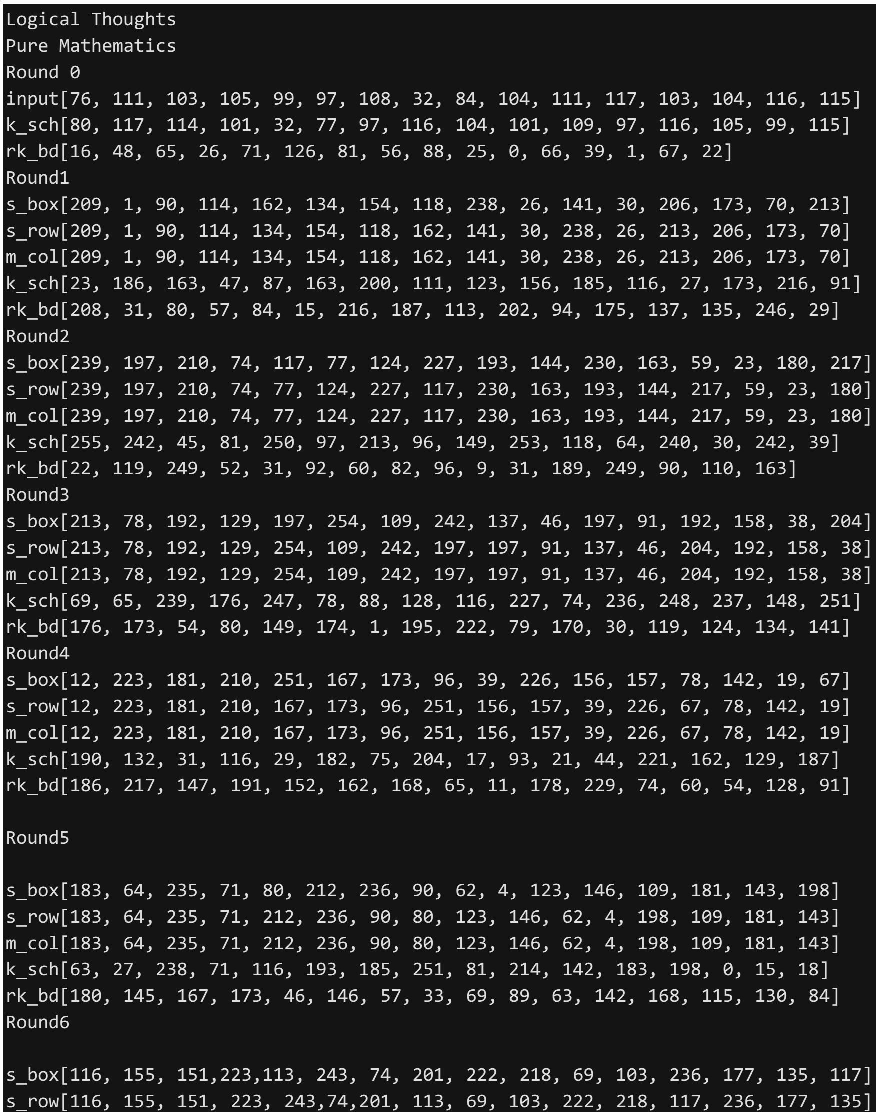
The values of round keys are taken from the Round Key Schedule Given in the previous section.

Table 12 Encryption/Decryption time analyses and comparison

	Chaos algorithm	AES algorithm	Proposed algorithm
Information entropy	795,454	795,912	795,667
Encryption quality	29,67,574	35,84,217	35,87,899
Encryption time (sec.)	121,468	27,36,952	167,342
Decryption time (sec.)	116,734	29,21,683	130,321
Total time (sec.)	238,202	56,58,635	297,663

Legend for Cipher (Encrypt).
Input: Cipher Input.
s_box: State after Sub Bytes ().
s_row: State after Shift Rows ().
m_col: State after Mix Columns ().
rk_bd: State after Round Key Binding ().
output: Cipher Output.

AES-128(Nk = 4, Nr = 10)

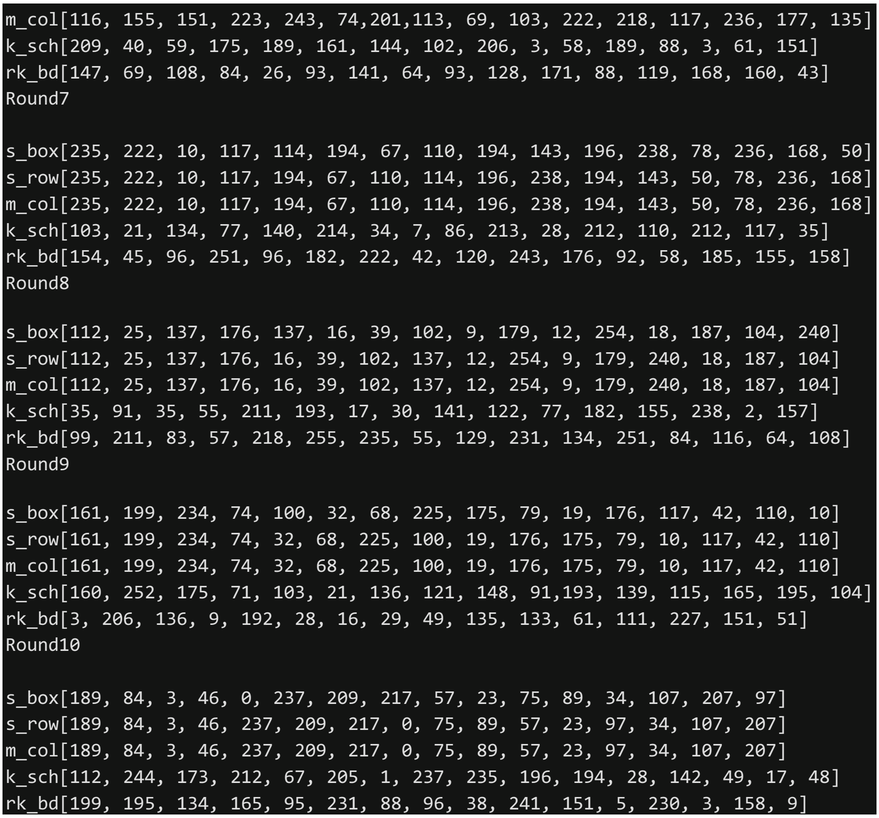


```
Logical Thoughts
Pure Mathematics
Round 0
input[76, 111, 103, 105, 99, 97, 108, 32, 84, 104, 111, 117, 103, 104, 116, 115]
k_sch[80, 117, 114, 101, 32, 77, 97, 116, 104, 101, 109, 97, 116, 105, 99, 115]
rk_bd[16, 48, 65, 26, 71, 126, 81, 56, 88, 25, 0, 66, 39, 1, 67, 22]
Round1
s_box[209, 1, 90, 114, 162, 134, 154, 118, 238, 26, 141, 30, 206, 173, 70, 213]
s_row[209, 1, 90, 114, 134, 154, 118, 162, 141, 30, 238, 26, 213, 206, 173, 70]
m_col[209, 1, 90, 114, 134, 154, 118, 162, 141, 30, 238, 26, 213, 206, 173, 70]
k_sch[23, 186, 163, 47, 87, 163, 200, 111, 123, 156, 185, 116, 27, 173, 216, 91]
rk_bd[208, 31, 80, 57, 84, 15, 216, 187, 113, 202, 94, 175, 137, 135, 246, 29]
Round2
s_box[239, 197, 210, 74, 117, 77, 124, 227, 193, 144, 230, 163, 59, 23, 180, 217]
s_row[239, 197, 210, 74, 77, 124, 227, 117, 230, 163, 193, 144, 217, 59, 23, 180]
m_col[239, 197, 210, 74, 77, 124, 227, 117, 230, 163, 193, 144, 217, 59, 23, 180]
k_sch[255, 242, 45, 81, 250, 97, 213, 96, 149, 253, 118, 64, 240, 30, 242, 39]
rk_bd[22, 119, 249, 52, 31, 92, 60, 82, 96, 9, 31, 189, 249, 90, 110, 163]
Round3
s_box[213, 78, 192, 129, 197, 254, 109, 242, 137, 46, 197, 91, 192, 158, 38, 204]
s_row[213, 78, 192, 129, 254, 109, 242, 197, 197, 91, 137, 46, 204, 192, 158, 38]
m_col[213, 78, 192, 129, 254, 109, 242, 197, 197, 91, 137, 46, 204, 192, 158, 38]
k_sch[69, 65, 239, 176, 247, 78, 88, 128, 116, 227, 74, 236, 248, 237, 148, 251]
rk_bd[176, 173, 54, 80, 149, 174, 1, 195, 222, 79, 170, 30, 119, 124, 134, 141]
Round4
s_box[12, 223, 181, 210, 251, 167, 173, 96, 39, 226, 156, 157, 78, 142, 19, 67]
s_row[12, 223, 181, 210, 167, 173, 96, 251, 156, 157, 39, 226, 67, 78, 142, 19]
m_col[12, 223, 181, 210, 167, 173, 96, 251, 156, 157, 39, 226, 67, 78, 142, 19]
k_sch[190, 132, 31, 116, 29, 182, 75, 204, 17, 93, 21, 44, 221, 162, 129, 187]
rk_bd[186, 217, 147, 191, 152, 162, 168, 65, 11, 178, 229, 74, 60, 54, 128, 91]

Round5

s_box[183, 64, 235, 71, 80, 212, 236, 90, 62, 4, 123, 146, 109, 181, 143, 198]
s_row[183, 64, 235, 71, 212, 236, 90, 80, 123, 146, 62, 4, 198, 109, 181, 143]
m_col[183, 64, 235, 71, 212, 236, 90, 80, 123, 146, 62, 4, 198, 109, 181, 143]
k_sch[63, 27, 238, 71, 116, 193, 185, 251, 81, 214, 142, 183, 198, 0, 15, 18]
rk_bd[180, 145, 167, 173, 46, 146, 57, 33, 69, 89, 63, 142, 168, 115, 130, 84]
Round6

s_box[116, 155, 151,223,113, 243, 74, 201, 222, 218, 69, 103, 236, 177, 135, 117]
s_row[116, 155, 151, 223, 243,74,201, 113, 69, 103, 222, 218, 117, 236, 177, 135]
```



```
m_col[116, 155, 151, 223, 243, 74,201,113, 69, 103, 222, 218, 117, 236, 177, 135]
k_sch[209, 40, 59, 175, 189, 161, 144, 102, 206, 3, 58, 189, 88, 3, 61, 151]
rk_bd[147, 69, 108, 84, 26, 93, 141, 64, 93, 128, 171, 88, 119, 168, 160, 43]
Round7

s_box[235, 222, 10, 117, 114, 194, 67, 110, 194, 143, 196, 238, 78, 236, 168, 50]
s_row[235, 222, 10, 117, 194, 67, 110, 114, 196, 238, 194, 143, 50, 78, 236, 168]
m_col[235, 222, 10, 117, 194, 67, 110, 114, 196, 238, 194, 143, 50, 78, 236, 168]
k_sch[103, 21, 134, 77, 140, 214, 34, 7, 86, 213, 28, 212, 110, 212, 117, 35]
rk_bd[154, 45, 96, 251, 96, 182, 222, 42, 120, 243, 176, 92, 58, 185, 155, 158]
Round8

s_box[112, 25, 137, 176, 137, 16, 39, 102, 9, 179, 12, 254, 18, 187, 104, 240]
s_row[112, 25, 137, 176, 16, 39, 102, 137, 12, 254, 9, 179, 240, 18, 187, 104]
m_col[112, 25, 137, 176, 16, 39, 102, 137, 12, 254, 9, 179, 240, 18, 187, 104]
k_sch[35, 91, 35, 55, 211, 193, 17, 30, 141, 122, 77, 182, 155, 238, 2, 157]
rk_bd[99, 211, 83, 57, 218, 255, 235, 55, 129, 231, 134, 251, 84, 116, 64, 108]
Round9

s_box[161, 199, 234, 74, 100, 32, 68, 225, 175, 79, 19, 176, 117, 42, 110, 10]
s_row[161, 199, 234, 74, 32, 68, 225, 100, 19, 176, 175, 79, 10, 117, 42, 110]
m_col[161, 199, 234, 74, 32, 68, 225, 100, 19, 176, 175, 79, 10, 117, 42, 110]
k_sch[160, 252, 175, 71, 103, 21, 136, 121, 148, 91,193, 139, 115, 165, 195, 104]
rk_bd[3, 206, 136, 9, 192, 28, 16, 29, 49, 135, 133, 61, 111, 227, 151, 51]
Round10

s_box[189, 84, 3, 46, 0, 237, 209, 217, 57, 23, 75, 89, 34, 107, 207, 97]
s_row[189, 84, 3, 46, 237, 209, 217, 0, 75, 89, 57, 23, 97, 34, 107, 207]
m_col[189, 84, 3, 46, 237, 209, 217, 0, 75, 89, 57, 23, 97, 34, 107, 207]
k_sch[112, 244, 173, 212, 67, 205, 1, 237, 235, 196, 194, 28, 142, 49, 17, 48]
rk_bd[199, 195, 134, 165, 95, 231, 88, 96, 38, 241, 151, 5, 230, 3, 158, 9]
```

Output in string Ã ├åÑ_þX`&±ùENQµETX×HT

9 Conclusion

This article presents a modified scheme of encryption i.e. modification of AES (MAES). The construction of S-box is different over here. It is developed using IP-Loop. The superiority of the structure over the extended binary Galois field is due to the larger key space i.e. larger number of possibilities are available here as compared to Galois field. IP-Loop have many representations in terms of Cayley table as compared to one Cayley representation in GF. It includes 128 bits key along with IP-Loop of order 256. If an attacker has the knowledge of key but don't have any information about loop, he can't succeed to break this. Moreover, the proposed mathematical system is non-commutative making it harder to break. Different analyses were used to investigate the proposed scheme to verify its strength. All the standard tests were showing fruitful results ensuring its practical applications in image encryption, internet of things and E-health care system as well as in e-commerce etc.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Abdullah A, Noor M, Majid K, Iqtadar H (2020) An encryption scheme based on discrete quantum map and continuous chaotic system. Int J Theor Phys 59(4):1227–1240
2. Adnan J, Tariq S, Attaullah (2020) A color image privacy scheme established on nonlinear system of coupled differential equations. Multimed Tools Appl 79:32487–32501
3. Ahmad M, Doja MN, Beg SMM (2018) "Security analysis and enhancements of an image cryptosystem based on hyper chaotic system," Journal of King Saud University, Comput Inf Sci
4. Ali A, Slaney J (2008) Counting loops with the inverse property. Quasigroups Related Struct 16:13–16
5. Alligood KT, Sauer TD, Yorke JA (1996) Chaos an introduction to dynamical systems. Springer-verlag, New York
6. Asif M, Shah T (2019) BCH codes with computational approach and its applications in image encryption. J Intell Fuzzy Syst 37(3):3925–3939
7. Attaullah SJ, Shah T (2018) "A novel scheme for image encryption using substitution box and chaotic system.," Nonlinear Dyn, vol. 91
8. Attaullah AJ, Shah T (2019) Cryptosystem techniques based on the improved Chebyshev map: an application in image encryption. Multimed Tools Appl 78:31467–31484
9. Attaullah S, Jamal S, Shah T (2017) A novel construction of substitution box using a combination of chaotic maps with improved chaotic range. Nonlinear Dyn 88(4):2757–2769
10. Attaullah S, Jamal S, Shah T (2018) A novel scheme for image encryption using substitution box and chaotic system. Nonlinear Dyn 91(1):359–370
11. De Canniere, C (2005) "Triple-DES," Encyclopedia of cryptography and security, triple-DES. https://doi.org/10.1007/0-387-23483-7_437
12. Chai X, Chen Y, Broyde L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88:197–213
13. Cui L, Cao Y (2007) A new S-box structure named affine power-affine. Int J Innov Comput Inf Control 3(3):45–53
14. Daemen J, Rijmen V (2002) The design of Rijndael-AES: the advanced encryption standard, Springer
15. Hussain I, Shah T, Muhammad Asif G, Khan WA (2011) Costruction of cryptographically strong 8*8 S-boxes. World Appl Sci J 13(11):2389–2395
16. Hussain I, Shah T, Gondal MA, Mahmood H (2012) Generalized majority logic criterion to analyze the statistical strength of S-boxes. Z Naturforsch A 67a:282–288
17. Hussain S, Jamal SS, Shah T, Hussain I (2020) A power associative loop structure for the construction of non-linear component of block cipher. IEEE Access, pp 123492–123506
18. Jahangir S, Shah T (2020) "Designing S-boxes triplet over a finite chain ring and its application in RGB image encryption.," Multimed Tools Appl, no. 79, p. 26885–26911
19. Jamal SS, Shah T, Attaullah (2017) "A group action method for construction of strong substituion box," 3D Research, vol. 8, no. 2
20. Javeed A, Shah T, Attaullah (2020) Lightweight secure image encryption scheme based on chaotic differential equation. Chin J Phys. https://doi.org/10.1016/j.cjph.2020.04.008
21. Javeed A, Shah T, Attaullah (2020) Design of an S-box using Rabinovich-Fabrikant system of differential equations perceiving third order nonlinearity. Multimed Tools Appl 79:6649–6660
22. Javeed A, Shah T, Attaullah (2020) "Construction of non-linear component of block cipher by means of chaotic dynamical system and symmetric group," Wireless Personal Commun, pp. 1–14
23. Kaur M, Singh D, Kumar V, Gupta BB, El-Latif AAA (2021) Secure and energy efficient-based E-health care framework for green internet of things. IEEE Trans Green Commun Netw 5(3):1223–1231
24. Khan M, Shah T (2014) A novel image encryption technique based on Henon chaotic map and S8 symmetric group. Neural Comput 25:1717–1722

25. Kocarev L (2001) Chaos-based cryptography: a brief overview. IEEE Circut Syst 1:6–21
26. Lambic D (2020) A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design. Nonlinear Dyn. https://doi.org/10.1007/s11071-020-05503-y
27. Li X, Wang L, Yan Y, Liu P (2016) An improvement color image encryption algorithm based on DNA operations and real and complex chaotic systems. Opt-Int J Light Electron 127(5):2558–2565
28. Liu H, Kadir A, Xu C (2020) Cryptanalysis and constructing S-Box based on chaotic map and backtracking, vol. 376
29. Majid K, Zeeshan A (2018) A novel construction of substitution box for image encryption applications with Gingerbreadman chaotic map and S8 permutation. Neural Comput & Applic 29:993–999
30. Meshram C, Obaidat M, Meshram S (2018) "Chebyshev chaotic map-based ID-based cryptographic model using subtree and fuzzy-entity data sharing for public key cryptography,," Secur Priv, p. 1:e12
31. Naseer Y, Shah T, Attaullah, Javeed A (2020) Advance image encryption technique utilizing compression, dynamical system and S-boxes. Math Comput Simul178:207–217
32. Naseer Y, Shah D, Shah T (2018) A Novel Approach to improve multimedia security utilizing 3D Mixed Chaotic map. Microprocess Microsyst 65. https://doi.org/10.1016/j.micpro.2018.12.003
33. Naseer Y, Shah T, Sadam H, Asif A (2019) Steps towards redesigning cryptosystems by a non-associative algebra of IP-loops. Wirel Pers Commun 108:1379–1392
34. Paar C, Pelzl J (2010) Understanding cryptography: a textbook for students and practitioners. Springer-Verlag, New York
35. Schneier B (1994) The blowfish encryption algorithm. Dr Dobb's J 19(4):38–40
36. Shah T, Qamar A, Hussain I (2013) Substitution box on a maximal cyclic subgroup of units of a Galois ring. Zeitschrift fur Natureforschung A 68:567–572
37. Shah T, Ali A, Khan M et al (2020) Galois Ring GR(23,8) Dependent 24×24 S-Box Design: An RGB Image Encryption Application. Wireless Pers Commun 113:1201–1224
38. Shannon CE (1948) A mathematical theory of communication. In the Bell Syst Tech J 27(3):379–423
39. Si Y, Liu H, Chen Y (2021) "Constructing keyed strong S-Box using an enhanced quadratic map.," Int J Bifurcation Chaos, vol. 31, no. 10
40. Tian Y, Lu Z (2017) "Chaotic S-box: intertwining logistic map and bacterial foraging optimization," Math Problems Eng 2017

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.