EAI Endorsed Transactions
on Smart Cities

Research Article **EAI.EU**

An Efficient Substitution Box design with a chaotic logistic map and Linear Congruential Generator for secure communication in Smart cities

Muhammad Asim Hashmi[1,2,*], Noshina Tariq[3]

[1]Department of Electronics, Quaid-i-Azam University, Islamabad, Pakistan (E-mail: mahashmi@ele.qau.edu.pk)
[2]Department of Electrical and Computer Engineering, Air University, Islamabad, Pakistan
[3]Department of Avionics Engineering, Air University, Islamabad, Pakistan (E-mail: noshina.tariq@mail.au.edu.pk)

Abstract

The study provides a unique method for creating an efficient substitution box (S-box) for advanced encryption standards using a Chaotic Logistic Map (CLM) and a Linear Congruential Generator (LCG) (AES) for secure communications in a smart city. The Pseudo-Random Number Generator (PRNG), which is further examined, is constructed using an extensive search of reasonable possibilities for the initial seed and set parameters. Using statistical testing, the performance analysis of the new S-box is assessed. Additionally, the resilience of differential, as well as linear cryptanalysis, is shown. It is derived using other features, including nonlinearity, the Bit Independence Criterion (BIC), and the Strict Avalanche Criterion (SAC). The suggested S-box has good potential and is usable for symmetric key cryptography, according to the features of the new S-cryptographic box.

Keywords: Security, smart city, Cryptography, Encryption, AES, s-box

Received on 07 November 2022, accepted on 19 January 2023, published on 23 March 2023

doi: 10.4108/eetsc.v7i1.2845

[*]Corresponding author. Email: mahashmi@ele.qau.edu.pk

1. Introduction

Irrevocable protection of both a transmitter and a receiver is essential for a secure communication network in smart cities. For all modern techno-driven smart automation, the security of the communication between the application and the controlling network is one of the major concerns. For this considerable problem both encryption and decryption are practical answers to this significant issue. Encryption makes communication unintelligible for any unauthorized user or intruder. To make communication networks in smart cities, modern encryption schemes provide a solution for secure information flow. Some markedly essential applications for which security and privacy are topmost concerns include (but are not limited to) Raspberry Pi-based automation systems for smart homes and SMEs that primarily need secure communications [1]. Communications in smart cities meal preparation [2], and waste management [3] needs end-to-end security. All protocols and architecture presented in [4] require secure communications.

The modern standards for data encryption, which are known as Advanced Encryption Standards (AES), were first released by NIST in the year 2000 and comprised four primary operations, which are Substitution Byte, Shift Rows, Add Round Key, and Mix Columns [5] [6]. This algorithm's replacement step, essential to encryption, is carried out via a 256-element array known as the Substitution box (S-box). The design of this S-box in AES is crucial since it makes the algorithm difficult to break and causes confusion and dissemination [7]. Plain and encrypted text in a cryptogram that uses a simple substitution approach is identical. Substitution and permutation are the two fundamental conditions for a nonlinear encryption method that must be met for a system to be impermeable to frequency analysis [8].

For this reason, the S-box is tested for additional cryptographic features and intended to resist cryptanalysis. Bit Independence Criterion (BIC), nonlinearity, Strict Avalanche Criterion (SAC), Bijectivity, Linear and Differential Approximated Probabilities, and SAC-BIC analysis all contribute to the S-box robustness. Most researchers employed chaotic maps to create new S-boxes last year to solve this research problem; some of the most prominent ones are covered in the literature review.

This paper presents a simple and quick approach to creating a cryptographically efficient S-box for AES. It offers a circular shift approach and develops an S-box with many iterations for a single output at the level of random number generation. Additionally, the approach permutes the intended vector to enhance statistical tests used in cryptography. With the help of this technique, the suggested S-Box is further statistically assessed for the cryptographic application by the needed cryptographic features, demonstrating the method's broad applicability. The following is a list of the suggested model's key contributions:

1. A novel and fast method is proposed to design a cryptographically efficient S-box for AES.
2. More than one iteration is used to generate a single output at a random number generation level while designing the proposed S-box.
3. A robust S-box has been presented by introducing the circular shift technique.
4. A permutation matrix of 256 entries-based maps is presented based on simple programming.

The next sections of the paper are as follows: Section 2 provides the literature review. Section 3 gives some insight into the background. Section 4 presents the proposed model to design an S-box for AES. The statistical testing and performance of the proposed S-Box are analyzed, and discussions are presented in Section 5, while Section 6 is the conclusion section.

2. Literature Review

From classical data security requirements to modern applications like Energy efficient routing protocols [30], unmanned aerial vehicles [31], block chain technology [32], efficient operations in data storage [33], mobile communication networks [34], the cryptographic algorithm plays a vital role in data security. By examining the effects of the Chaos base approach on block ciphers, Jakimoski et al. [9] produced a Chaos-based S-box. This S-box is significantly nonlinear and appropriate for Cryptography applications where encryption is needed, and substitution is part of the encryption technique. Grouping Tang et al. [10] developed a technique for designing S-boxes that yields dynamically powerful cryptographic substitution box. It used a two-dimensional (2D) discretized chaotic Baker map cryptographically superior to Jakimoskie's S-box. Gondal et al. [11] introduced a novel approach for S-box design that was significantly nonlinear. The approach relied on a chaotic Bakers map and a scaled-down version of an 8-bit block cipher. The behavior in a chaotic logistic map renders the algorithm incomprehensible, adding to the unpredictability. Iqtidar et al. [12] applied a chaotic logistic map's output to a linear functional transformation. They presented a novel method for creating a considerably nonlinear S-box with all the cryptographic features. Zhongyun et al. [13] suggested a unique strategy equivalent to previous relevant S-boxes using the entire Latin square method. Qing et al. [14] developed a more extensive chaotic range and many chaotic features utilizing the Logistic-Sine System. Akram et al. [15] suggested a novel approach for designing an S-box based on a chaotic sine map. The approach utilized to create this S-box is straightforward to apply. This method secures the permutations and maps generated values with a permutation matrix of 256 entries. The map used (in this method) is based on simple programming and does not have solid mathematical roots. Using credibility complex fuzzy sets (CCFS), Yahya et al. [28] proposed a novel scheme for designing an S-box for the encryption of images and discussed the results for the suitability of the proposed S-box for image encryption.

3. Background

The methods, which are chaos-based Pseudo-Random Number Generators (PRNGs), played a vital role in designing robust cryptographic algorithms in the previous two decades. Some markedly on the top are included in section 1.1. We took two different PRNGs to design a novel S-box. The structure of the AES algorithm for a single round of encryption is shown in Figure 1 [29].

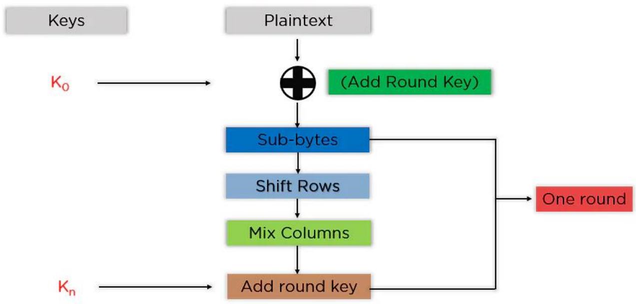

Figure 1. AES encryption process

Except for these four steps, all other steps, such as adding a round key, mixing columns, and shifting rows, are linear operations [12] [15]. That is why the S-box is the only factor that introduces nonlinearity in an algorithm. The substitution operation is described in Figure 2.

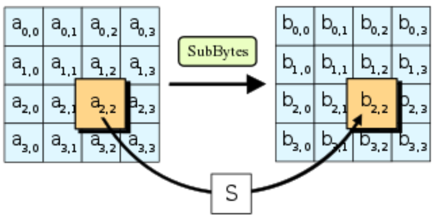

Figure 2. Substitution in AES

In the substitution process, the entry or plain text gets changed according to the value of the location in the S-box. For example, if the value of the plain text is "3F," then the value at the 63rd location of the S-box will be substituted accordingly. The number of rows and columns is equal (16x16) for AES, whichfulfills all the required substitution possibilities in the American Standard Code for Information Interchange (ASCII) [5]. This substitution is possible only if a randomly permuted unique string of [0-255] elements exists. An unpredictable random number generator is required to generate this string unintelligibly and robustly [13]. A chaotic logistic map is a well-known chaos-based random number generator for its sensitive output upon a slight change in initial conditions.

A good PRNG means highly unpredictable output for miner input values change. For all PRNGs, there are some fixed parameters and seed values. In our experiment for both PRNGs, which are chaotic logistic maps and linear congruential generators, we have some fixed parameters and a seed value, as described in Tables 1 and 3 of Section 3.

4. Proposed Architecture

The design scheme is presented by division into two subsections. Subsection 4.1 elaborates on the scheme of random permutations with the help of CLM, while subsection 4.2 describes the mapping vector. Finally, a novel S-box is generated using both vectors, as shown in Fig. 3.

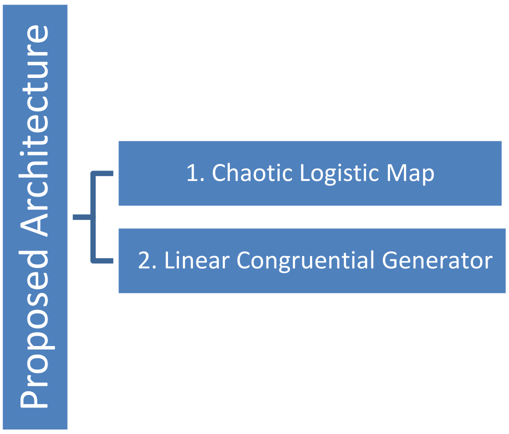

Figure 3. Proposed Architecture

4.1. Chaotic Logistic Map

A logistic map is highly sensitive to initial conditions and is an efficient chaotic map [12]. Mathematically CLM is defined in Equation 1.

$$X_{n+1} = Xn(1 - Xn) \qquad (1)$$

In Equation 1, X_{n+1} refers to the output of the seed and initial conditions for the n^{th} iteration. The Xn defines the seed value for the first iteration or the output of previous iterations. In this study, values ranging from 0-255 are extracted from **GF** (2^8), and a proportional gain F is applied to make the output suitable for usage with **GF** (2^8). Following a significant amount of trial and error, the initial values that were established for seed, modulus, and constant variable are shown in Table 1.

Table 1. Initial values for Chaotic Logistic map

Variable	Value
F	19731
X_n	0.167

Create a string using a hundred thousand iterations, then reformat it into a 100x1000 matrix. Following that, a circular shift of ten columns and ten rows, respectively, is applied to the matrix. Now, choose just the tenth or its multiplier part of this string to generate an S-box, which will provide us with an array containing 10000 items. After removing any instances of duplication from this array, the resultant matrix (G) will consist of the 256 items presented in the following order. Figure 4 presents the matrix in its entirety, designed using Matlab.

Algorithm 1 CLM Random Permutations

Required: Permuted unique random values in **GF(2^8)**
1: Parameters: PRNG equation, scaling factor, seed
2:**Int** seed Xn (Xn = seed value)
3: **Int** scaling factor S (S = 19731)
4: **Int** modulus M (M = 256)
5: **For** iteration 1:1000000
6: $X_{(i+1)} = X_{(i+1)} * (1-X_{(i+1)}) \bmod M$
7: Output = Circular shift (output,10,10)
8: **For**I = 1:100000
9: CLM (i) = Output (i*10)
10: end

131	174	167	229	127	29	178	209	92	143	241	130	103	246	89	58
65	22	126	80	64	15	63	247	55	31	119	62	25	35	215	129
188	164	105	195	163	138	74	11	236	109	220	23	139	144	1	96
3	243	114	203	4	69	81	252	10	95	175	158	124	204	191	150
61	7	228	193	45	102	125	115	59	47	170	83	48	176	132	2
172	94	182	210	140	110	122	146	136	206	222	180	99	91	211	181
41	67	14	121	66	219	254	194	43	218	135	245	44	33	54	19
231	202	251	78	86	169	221	230	73	154	196	100	145	184	185	57
168	20	161	162	84	223	108	166	26	142	147	87	104	117	21	156
165	116	256	56	71	97	76	216	70	79	111	238	9	42	250	187
186	217	106	107	112	197	128	208	6	5	30	253	68	157	60	40
224	36	123	207	199	198	232	52	98	173	239	85	249	213	235	28
153	227	190	38	27	17	212	155	34	32	159	113	151	189	51	134
255	49	24	179	237	192	88	248	75	242	201	177	16	214	50	226
234	8	90	101	160	225	244	82	39	141	118	205	148	171	12	200
93	233	137	37	240	149	72	77	133	152	53	183	120	46	18	13

Figure 4. The Figure presents the Initial 16X16 matrix designed by applying initial conditions to the chaotic logistic map. The simulations are made using Matlab.

4.2. Linear Congruential Generator

The quickest random number generator is a linear congruential pseudo-random number generator (LCG) [17]. Equation 2 mathematically defines the LCG.

$$X_{(n+1)} = (aX_n + c) \; mod \; M \qquad (2)$$

The values of both multiplicative factor a and additive factor c lie between 0 and the value of modulus M. $X_{(n+1)}$, which refers to the output value of the nth iteration. *In contrast*, $X_{(n)}$ refers to the seed value for the nth iteration. In our experiment, we employ LCG as a mapping vector. Table 3 shows the beginning values for LCG in this experiment.

Table 2. Initial Conditions for Linear Congruential generator

Variable	Value
Multiplicative factor (a)	11
Addition factor (c)	7
Modulus (M)	19731

Since the values of both the multiplicative and additive factors are less than 19731 and greater than 0, it fulfills the primary requirement of LCG. The output vector is confined to modulus N using these starting values, as shown in Equation 3.

$$M(i) = X(i) \; mod \; N \qquad (3)$$

Using this pseudo-random number generator, this work creates an initial string after 10000 iterations. It builds a vector from them by defining them in mod 257 and generates a vector of (1-256). In this regard, Table 4 shows the permutation matrix (P).

Algorithm 2 LCG Permutations

Required: Permuted unique random values from (1-256)
1: Parameters: PRNG equation, Additive factor, seed
2:**Int** seed Xn (Xn = seed value)
4: **Int** multiplicative factor “A” (A = 7)
5: Int multiplicative factor “c” (c = 11)
6: **Int** modulus M (M = 256)
7: Int modulus M (N = 257)

8: **For** iteration 1:10000
6: $X_{(n+1)} = (aX_n + c) \mod M$
8: **For**I = 1:1000
9: LCG (i) = Output (i*10) Mod 257 ; **Unique**
10: G(i) = LCG (i);
11: end

$$S(P(i)) = G(i) \qquad (4)$$

This vector specifies the permutation positions for the Matrix G. Figure 4 depicts the planned S-box. This mapping vector leads to the final design of the substitution box. The proposed s-box is presented in Figure 6. Row 1 and column 1 in Figure 5 determine the locations of entries.

Algorithm 3 Mapping

1: Int P, G
2: For i=1:256
3: S(P(i)) = G (i)

4: End

18	205	206	64	40	256	62	175	50	76	80	182	69	227	108	59
92	198	162	81	185	102	159	73	155	4	142	201	195	245	24	221
125	72	61	255	51	54	84	218	42	187	215	199	148	10	133	243
110	197	93	60	70	213	211	23	177	45	132	91	237	189	146	67
147	129	207	120	17	86	240	143	38	27	116	172	249	234	44	105
226	97	112	20	202	231	168	214	106	236	33	121	83	183	138	248
223	228	163	233	41	77	152	94	82	229	246	118	169	160	1	167
36	251	115	161	95	128	141	16	47	68	75	203	242	156	14	78
150	7	113	222	252	126	43	194	244	144	165	56	119	122	180	158
12	98	13	173	3	63	48	90	96	135	8	6	30	151	184	109
31	149	103	145	209	34	216	99	217	2	66	219	87	153	65	241
46	49	32	71	193	140	188	196	5	15	230	58	253	166	22	117
9	224	29	19	191	52	127	208	239	157	192	101	85	21	39	238
130	114	250	124	89	74	134	104	123	57	200	37	139	79	210	178
174	254	55	220	171	170	28	176	235	179	186	88	131	232	212	204
26	35	137	154	100	190	136	164	11	247	25	53	107	181	225	111

Figure 5. Figure shows the mapping functions for initial CLM permutations. The matrix is designed using Linear Congruential Generator with initial conditions

54	45	222	36	60	80	94	161	7	107	78	10	199	112	18	40
5	131	197	92	30	43	116	49	255	191	126	226	166	50	172	145
71	38	233	163	150	99	208	251	148	61	14	141	25	81	246	96
31	210	231	17	154	74	240	170	175	139	77	21	52	12	67	26
98	189	177	183	103	151	113	28	1	217	147	46	243	69	218	35
238	235	59	138	4	102	15	245	88	168	213	130	22	146	239	135
20	252	51	90	207	68	106	104	79	42	19	91	23	220	110	152
111	215	127	87	164	205	248	193	72	254	122	250	201	128	171	181
153	53	66	242	247	237	2	180	137	157	44	64	62	6	121	48
144	93	236	85	192	216	136	117	211	37	225	118	140	11	176	24
133	47	108	58	196	158	120	75	16	55	160	109	100	253	39	202
165	244	132	203	143	65	89	179	198	184	83	232	200	101	206	84
256	190	95	105	219	114	187	188	221	32	56	204	174	129	124	249
185	9	224	178	57	194	3	223	119	156	234	149	167	162	134	8
173	186	73	97	214	86	13	212	115	63	209	27	228	41	34	195
230	123	76	229	241	70	182	169	142	33	82	159	155	227	125	29

Figure 6. The resultant substitution box after mapping locations of CLM with the function of LCG.

EAI Endorsed Transactions
on Smart Cities
01 2023 - 03 2023 | Volume 7 | Issue 1 | e6

5. Performance Analysis

The suggested S-cryptographic box's features are subjected to a statistical analysis in which the probability of nonlinearity, BIC, bijectivity, SABIC, SAC, differential approximation, and linear are considered.

5.1 Bijectivity

The S-box is bijective [14] if and only if every input has a unique mapping on the output and correspondingly unique values in **GF** (2^8).

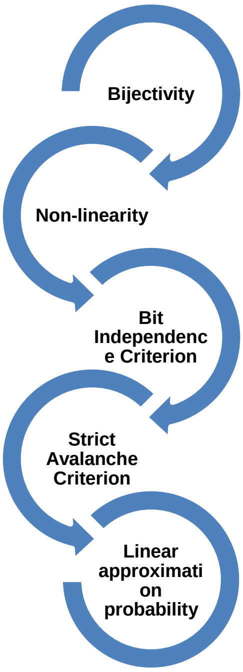

Figure 7. Proposed steps in Performance analysis

5.2 Nonlinearity

High nonlinearity is the most crucial statistical feature of an S-box. This feature reveals a shift in the bits between two successive encrypted sentences [19]. A nonlinear Boolean function g(x) may be represented by its Walsh spectrum [20]. Figure 8 depicts the suggested S-box nonlinearity from a function perspective.

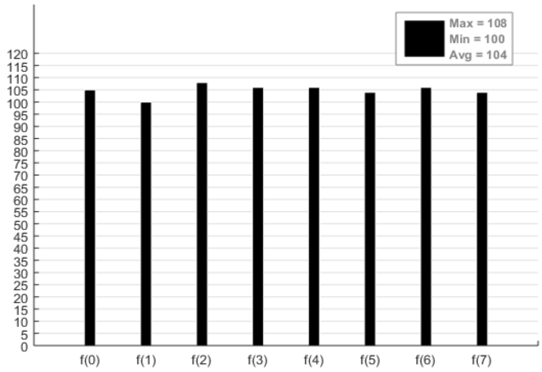

Figure 8. Proposed s-box Nonlinearity

5.3 Bit Independence Criterion

For this reason, Webster and Tavares [22] developed the Bit independence criteria. Analyzing the S-box's strength using this technique is standard practice. It indicates that any shift in the bits sent out does not affect any other pairs. That is to say, during nonlinearity in sequence or the avalanche effect, if a single bit in the input is altered, its behaviour at the output is unrelated to any preceding bits. The BIC-SAC and BIC nonlinearity are calculated, shown in Figure 9 and Figure 10, and a comparison is given in Table 4.

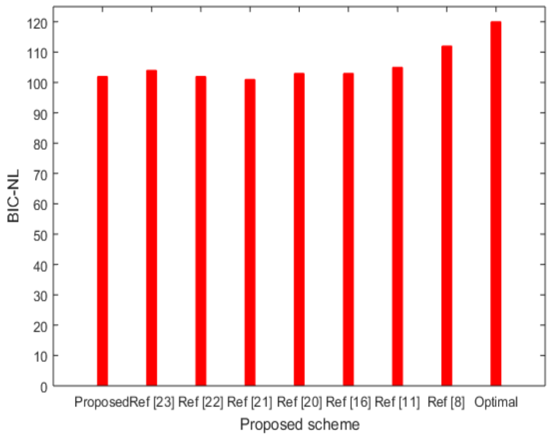

Figure 9. BIC-NL comparison

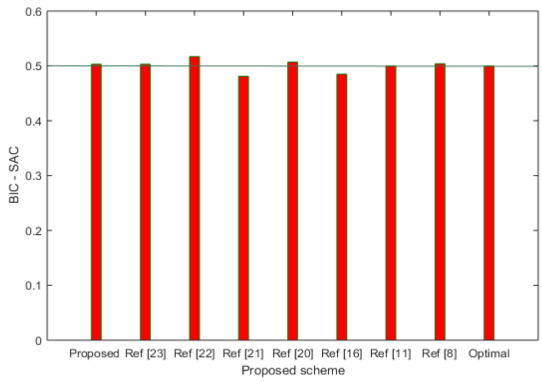

Figure 10. BIC-SAC comparison

5.4 Strict Avalanche Criterion

The Strict Avalanche Criterion (SAC) was developed by Webster and Tavares [22]. By this definition, if a single bit of input is complemented, then all bits of the output will change with probability half. Thus the function satisfies the SAC. Half of the encryption bits will be reversed if one bit of plain text is inverted. Table 6 presents the results of the SAC analysis; Figure 11 provides a visual comparison. Whereas Table 4 compares the value to that of other well-known coded S-boxes.

Table 3. Strict Avalanche Criterion results

SAC Maximum	0.59
SAC Minimum	0.41
Average Value	0.498
Variance	0.042

.

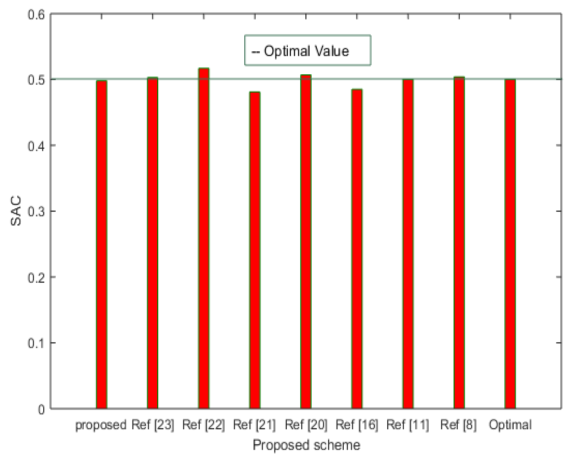

Figure 11. SAC Comparison

Table 4. Comparison table of Results

Scheme	NL	BIC	SAC	SAC-BIC	LAP	DAP
Proposed	104	102	0.498	0.503	0.132	0.0390
Anees et al.[24]	102	103	0.507	0.502	0.141	0.0468
Khan et al.[25]	100	101	0.481	0.496	0.171	0.0625
Khan et al.[26]	102	102	0.517	0.479	0.164	0.210
Wang et al.[20]	104	103	0.485	0.0.476	0.141	0.0390
Balezi et al. [15]	105	105	0.500	0.500	0.125	0.0468
Kim et al.[27]	104	104	0.503	0.503	0.109	0.0468
Hussain et al.[12]	112	112	0.504	0.504	0.062	0.0156
Optimal	120	120	0.500	0.500	0.062	0.0156

5.5 Linear approximation probability

Linear Approximation Probability (LAP) is the most significant value of an event's imbalance. In order to provide an equal number of output and input bits, the mask selects the parity of the bits [23]. The proposed S-box show the LAP values better than Ref [20, 24, 25, 26] and comparable to Ref. [15, 27]. The graphical comparison of LAPs is in Figure 12, and the comparison is given in Table 4.

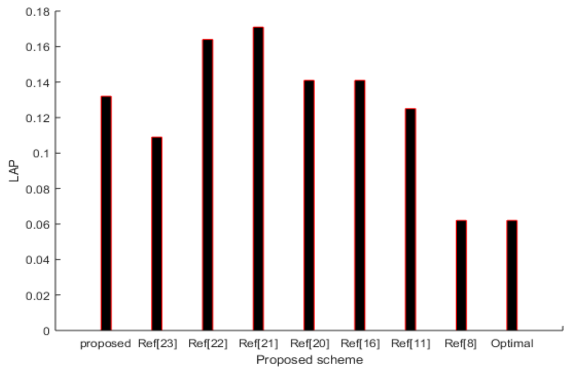

Figure 12. LAP Comparison

5.6 Differential approximation probability

An S-box's Differential Approximation Probability (DAP) measures differential uniformity [23]. The DAP method ensures that each differential at the input is uniquely mapped at the output. It is ideal for making this approximation probability as low as possible. The optimal value of this probability is 0.062. The comparative analyses of LAP and DAP are provided in Table 4. It shows that the proposed method has DAP values better than Ref. [15, 20, 24, 25, 26, 27]. The graphical comparison of DAPs is shown in Figure 13, and the comparison is given in Analysis Table.

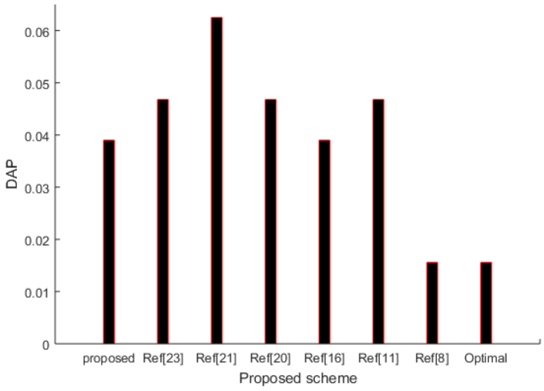

Figure 13. DAP Comparison

The comparison table shows that the nonlinearity of the proposed s-box is 3.921% better than Khan et al. [25] and 1.942 % better than Anees et al. [24] and Khan et al. [26]. The BIC values are 0.985% better than Khan et al. [25]. The difference in SAC from the optimal value is 4.27% better than the lowest value [25] in Table 4. LAP values are lower than Anees et al. [24], Khan et al. [25], Khan et al. [26], and Wang et al. [20]. Similarly, the DAP values are better than Anees et al. [24], Khan et al. [25], Khan et al. [26] and Balezi et al. [15], and Kim et al. [27]. The

Results show that the proposed method for designing the s-box is prominently applicable to cryptographic applications.

6. Conclusions

This article presents a basic but effective way of creating S-boxes. A chaotic logistic map and a linear congruential pseudo-random number generator create a reliable S-box architecture. The created S-box is compared to the codified S-box to assess its resistance to cryptanalysis assaults. The effectiveness of the created S-box demonstrates the tremendous potential of this AES S-box for cryptographic applications. For all applications in smart cities where encryption is required, this is a vital part of the algorithm on application level uses. Future applications of this technique include the encryption of still images and moving video by breaking a movie down into individual frames and encrypting each one in turn. The cryptographic properties of this work show that the method fulfills all required properties for secure communication between a transmitter and a receiver.

References

[1] Tirumala, S. S., Nepal, N., & Ray, S. K. (2022). Raspberry pi-based intelligent cyber defense systems for SMEs and smart-homes: An exploratory study. *EAI Endorsed Transactions on Smart Cities*, *6*(18), e4-e4.

[2] Namasivayam, B. (2022). AI for Healthy Meal Preparation in Smart Cities. *EAI Endorsed Transactions on Smart Cities*, *6*(4), e1-e1.

[3] McCurdy, A., Peoples, C., Moore, A., & Zoualfaghari, M. (2021). Waste Management in Smart Cities: A Survey on Public Perception and the Implications for Service Level Agreements. *EAI Endorsed Transactions on Smart Cities*, *5*(16).

[4] Sajid, A., Shah, S. W., & Magsi, T. (2022). Comprehensive Survey on Smart Cities Architectures and Protocols. *EAI Endorsed Transactions on Smart Cities*, *6*(18).

[5] Daemen J, Rijmen V. The Design of RIJNDAEL: AES The Advanced Encryption Standard.SpringerVerlag: Berlin, 2002.

[6] Khan, M., Azam, N. A. (2015). Right-translated AES gray S-boxes. Security and Communication Networks, 8(9), 1627-1635.

[7] Ferguson N, Schroeppel R, Whiting D. A simple algebraic representation of Rijndael. In Selected Areas in Cryptography SAC01, LNCS2259, 2001; 103?11.

[8] Shannon, C.E., 1949. Communication theory of secrecy systems. The Bell system technical journal, 28(4), pp.656-715.

[9] Jakimoski, G., Kocarev, L.: Chaos and cryptography: Block encryption ciphers based on chaotic maps. IEEE Trans. Circuits Syst. 48(2), 163 (2001)

[10] G. Tang, X. Liao, Y. Chen, A novel method for designing S-boxes based on chaotic maps, Chaos Solitons Fractals 23 (2005) 41319

[11] Muhammad Asif Gondal, Abdul Raheem, Iqtadar Hussain, A scheme for obtaining secure S-Boxes based on chaotic baker map, 3D Res. 5 (August)(2014) 17

[12] Hussain, I., Shah, T., Gondal, MA and Mahmood, H., 2013. An efficient approach for the construction of LFT S-boxes using chaotic logistic map. Nonlinear Dynamics, 71(1), pp.133-140.

[13] Hua, Z., Li, J., Chen, Y., and Yi, S., 2021. Design and application of an S-box using a complete Latin square. Nonlinear Dynamics, 104(1), pp.807- 825.

[14] Lu, Q., Zhu, C. and Deng, X., 2020. An efficient image encryption scheme based on the LSS chaotic map and single S-box. IEEE Access, 8, pp.25664-25678.

[15] Belazi, A. and Abd El-Latif, A.A., 2017. A simple yet efficient S-box method based on chaotic sine map. Optik, 130, pp.1438-1444.

[16] Radwan, A.G., 2013. On some generalized discrete logistic maps. Journal of advanced research, 4(2), pp.163-171.

[17] Marsaglia, G., 1972. The structure of linear congruential sequences. In Applications of number theory to numerical analysis (pp. 249-285). Academic Press.

[18] Zamli, K. Z., Kader, A., Din, F., Alhadawi, H. S. (2021). Selective chaotic maps Tiki-Taka algorithm for the S-box generation and optimization. Neural Computing and Applications, 1-18

[19] Javeed, A., Shah, T. (2020). Design of an S-box using RabinovichFabrikant system of differential equations perceiving third order nonlinearity. Multimedia Tools and Applications, 79(9), 6649-6660

[20] Wang, Y., Xie, Q., Wu, Y., Du, B. (2009, June). A software for Xbox performance analysis and test. In 2009 International Conference on Electronic Commerce and Business Intelligence (pp. 125-128). IEEE

[21] Pedro Miguel Sosa. Calculating nonlinearity of boolean functions with Walsh-Hadamard transform. 2016

[22] A. Webster, S. Tavares, On the design of S-boxes Advances in Cryptology: Proc. of Crypto?5, Santa Barbara, USA. Lecture

[23] M. Matsui, Linear cryptanalysis method of DES cipher Advances in Cryptology, Proc. Eurocrypt?3. LNCS, vol. 765, Springer, Berlin, 1994, pp. 386

[24] Anees, A. and Ahmed, Z., 2015. A technique for designing substitution box based on van der pol oscillator. Wireless Personal Communications, 82(3), pp.1497-1503.

[25] Khan, M., Shah, T. and Batool, S.I., 2016. Construction of S-box based on chaotic Boolean functions and its application in image encryption. Neural Computing and Applications, 27(3), pp.677-685.

[26] Khan, M. and Asghar, Z., 2018. A novel construction of substitution box for image encryption applications with Gingerbreadman chaotic map and S8 permutation. Neural computing and applications, 29(4), pp.993-999

[27] Kim, J., Phan, R. C. W. (2009, June). A cryptanalytic view of the NSA's Skipjack block cipher design. In International Conference on Information Security and Assurance (pp. 368-381). Springer, Berlin, Heidelberg

[28] Yahya, M., Abdullah, S., Almagrabi, A. O., & Botmart, T. (2022). Analysis of S-Box Based on Image Encryption Application Using Complex Fuzzy Credibility Frank Aggregation Operators. IEEE Access, 10, 88858-88871.

[29] Heron, S. (2009). Advanced encryption standard (AES). Network Security, 2009(12), 8-12.

[30] Hassan, M. Abul, et al. "Energy efficient hierarchical based fish eye state routing protocol for flying ad-hoc networks." Indonesian Journal of Electrical Engineering and Computer Science 21.1 (2021): 465-471.

[31] Hassan, Muhammad Abul, et al. "Unmanned Aerial Vehicles Routing Formation using fisheye state routing for flying ad-hoc networks." the 4th international conference on future networks and distributed systems (ICFNDS). 2020.

[32] Javed, Abdul Rehman, et al. "Integration of blockchain technology and federated learning in vehicular (IoT) networks: A comprehensive survey." Sensors 22.12 (2022): 4394.

[33] Sajid, Faiqa, et al. "Secure and Efficient Data Storage Operations by Using Intelligent Classification Technique and RSA Algorithm in IoT-Based Cloud Computing." Scientific Programming 2022 (2022).

[34] Ali, Sher, et al. "New Trends and Advancement in Next Generation Mobile Wireless Communication (6G): A Survey." Wireless Communications and Mobile Computing 2021 (2021).