electronics

MDPI

Article

Improvement of Radio Frequency Identification Security Using New Hybrid Advanced Encryption Standard Substitution Box by Chaotic Maps

Amira S. El Batouty [1], Hania H. Farag [2], Amr A. Mokhtar [2], El-Sayed A. El-Badawy [2] and Moustafa H. Aly [3,*,†]

[1] Department of Electrical and Communications Engineering, Alexandria Higher Institute of Engineering and Technology, Alexandria 21311, Egypt; amislh@yahoo.com
[2] Department of Electrical Engineering, Faculty of Engineering, Alexandria University, Alexandria 21521, Egypt; hania11@yahoo.com (H.H.F.); amromokhtar61@gmail.com (A.A.M.); sbadawe@ieee.org (E.-S.A.E.-B.)
[3] College of Engineering and Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria 21500, Egypt
* Correspondence: mosaly@aast.edu
† Member of the Optical Society of America (OSA).

Received: 23 June 2020; Accepted: 15 July 2020; Published: 18 July 2020

Abstract: Radio Frequency Identification (RFID) technology is widely utilized by businesses, organizations and wireless communication systems. RFID technology is secured using different ways of data encryption, e.g., Advanced Encryption Standard (AES). The Substitution Box (S-Box) is the core of AES. In this paper, a new algorithm is proposed to generate a modified S-Box with new keys, specifically a key and plaintext-dependent S-Box using an improved RC4 encryption algorithm with Logistic Chaotic Maps (LCM). The strength of the proposed S-Box is tested throughout the paper, and compared against the state-of-the-art S-Box implementations, namely, the static S-Box, dynamic S-box, KSA and PRGA S-Box, and RC4 S-Boxes with Henon chaotic maps. The comparison between the state-of-the-art S-Boxes and the proposed S-Box demonstrates that the use of the Logistic Chaotic Map increases the security of the S-Box and makes the differential and linear cryptography more sturdy. In particular, using the strict avalanche test, we demonstrate that the proposed S-Box improves the security by achieving a cipher text bit-flip ratio of 0.4765, which is closer to 0.5 (where half the bits are flipped), while maintaining a minimum elapsed time of 19 milliseconds for encryption and decryption.

Keywords: cryptography; RC4 algorithm; RFID security; encryption; logistic chaotic map

1. Introduction

Radio Frequency Identification (RFID) technology is an automatic identification technology that uses radio waves to identify the path, organize and wirelessly discover a variety of objects additionally as individuals, carriage, commodity, and property [1–4]. In some previous studies [4,5], RFID technology used the coding techniques as Data Encryption Standard (DES) and conjoined the Advanced Encryption Standard (AES) for information security. Nowadays, RFID systems have been enhanced by adopting the ways to secure the transferred data by safe encoding measures within the affordable RFID systems and to standardize the cryptography [6,7].

The two major classes of cryptosystems are asymmetric and symmetric keys. The stream cipher and block cipher are the two divisions of the symmetric key [8,9]. The Substitution Box (S-Box) is one of the foremost important parts and is the solely nonlinear component reassuring the confusion

property of the standard block ciphers like the AES. The strength of this algorithm depends on the planning of cryptographically sturdy S-Box [10,11].

In [12], the S-Box is generated by using the key scheduling algorithm and the dynamic S-Box is presented in [13,14]. The key and plaintext dependent S-Boxes RC4 algorithms are generated by Hossein Khani et al. [15]. The S-Box depends on the key and plaintext used for RC4 to induce the S-Box, whereas it is freelance from the key and plaintext used for the AES.

In this paper, we first generate some S-Boxes using three new different keys. We test their strength using the security tests; nonlinearity test, avalanche effect, percentage execution time performance efficiency (ETPE%), and the strict avalanche criteria. Then, we propose a novel method to get a secured S-Box; specifically, a key and plain dependent S-Box improved RC4 algorithm with the Logistic Chaotic Maps (LCM) algorithm. The new proposed S-Box is tested using the security tests and is compared with the previous S-Boxes, showing a better performance. Then, we use three different keys to generate three different S-Boxes with a comparison with other keys.

S. Zhu et al. [16] and Q. Lu et al. [17] proposed a new compound chaotic system called sine tent map which is two mixed chaotic maps (sine map and tent map) to generate new S-box used in image encryption. However, in our paper, we propose a new encryption algorithm (improved RC4 algorithm with Logistic Chaotic Map) to generate new S-box used in data encryption. Additionally, Lu et al. [17] used different ways to measure the Strict Avalanche Criteria (SAC) numerically. However, in our paper, three investigation techniques are utilized to measure the SAC: examination of the frequency of various Hamming weights, investigation of the frequency of various differential values, and investigation of Hamming weights as per the bit position.

The paper is organized as follows. Section 2 presents a brief survey for the RFID, followed by the clarification of the S-Box and its sorts. The modified algorithm is explained in Section 3 to get the modified S-Box. Section 4 explains the different tests to measure the strength of the modified S-Box. It also displays and discusses the obtained results. Section 5 is devoted to the main conclusions of this work.

2. Materials and Methods

2.1. RFID

The RFID system (Figure 1) uses RF electromagnetic fields to transfer data following tags that are very little and attached to things. Tags transmit single identifiers upon request by RFID readers that transmit powerful electromagnetic fields and skim the data unbroken within the tag. The tags are partitioned off into two categories. The primary one is a passive tag, which is cheaper and smaller because it needs no battery and is high-powered and skims at short ranges. The second category is an active tag that uses a neighbor power supply and emits radio waves. Thus, it must operate a few meters from the RFID reader [1–3].

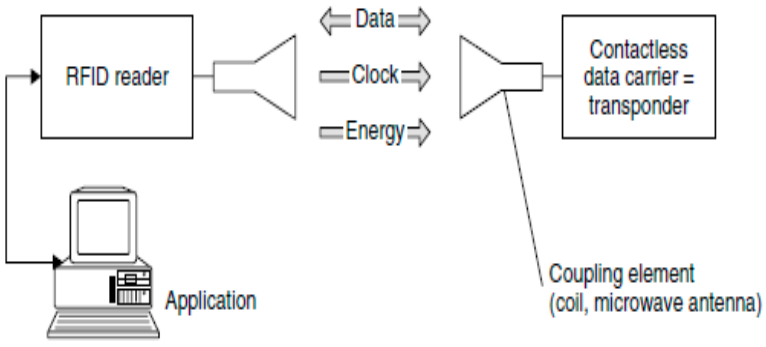

Figure 1. Radio Frequency Identification (RFID) system.

Information is electronically unbroken on the tag and can be scanned up to a few meters away. Since the application of RFID in the 1960s, and because of the obvious benefits of its low value and distinctive reorganization feature, the RFID systems are aware of human activity and site messages [8]. For security, both the reader and tag use identical cryptography algorithms to speak with each other.

2.2. S-BOXES

Confusion and diffusion concepts were identified by Shannon as the most important conditions for any cipher system security [9]. Diffusion covers the association between the plaintexts. In addition, the cipher text suggests that each symbol depends on some or all symbols among the plaintext. Confusion covers the association between the cipher text and the key which suggests if one bit among the keys is changed.

The S-Box is the only nonlinear component within the block cipher algorithm which supports the confusion function in reinforcing the encryption. The S-Box input may be an n-bit word, and the output may be an m-bit word, where m and n are not essentially equal. Two kinds of S-Boxes exist: the static S-Box which is fixed as explained in [3] and the dynamic S-Box, which has no relation with the key and is explained in [10].

2.2.1. Static S-Box

In [3], static S-Box is constructed as follows:

(1) The S-Box is initialized with byte values 00 to FF.
(2) Each byte is mapped to its inverse among the finite field.
(3) The affine transformation is applied to each byte among the S-Box [10,11].

2.2.2. Dynamic S-Box

The dynamic S-Box is a key-dependent S-Box [11,12] and depends on the Key Scheduling Algorithm (KSA) as shown in Table 1. The KSA is carried out by creating two 256-element arrays. The initial array is S"256" and is filled with the values 0 to 255. The second array is K"256" and is full of the shared secret key. The shared secret key is split into byte segments and is traced byte by byte into K. To finish key scheduling, the array should be irregular [14]. The algorithm for creating the KSA is shown in Figure 2, where S is a vector that contains a scheduled key and i, j are in the round iterations range. The output is 256 totally different values depending on the input key. Then, affine transformation is performed for the produced values [12,18], to avoid any fastened points and to form the dynamic S-Box shown in Table 1.

Table 1. Dynamic S-Box with the key "FEDCBA9876543210".

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	3B	64	3C	AD	58	39	F5	7E	EF	E0	97	92	07	DE	F9	D4
1	E1	0C	61	15	44	C1	FA	A8	7C	A2	DA	50	89	1F	96	6D
2	02	83	69	26	46	B4	38	CD	47	6A	75	5A	30	25	C3	D0
3	78	40	76	87	B1	32	21	51	EB	A7	8C	A5	48	2F	D9	DB
4	8B	A9	CF	53	9F	73	E5	B8	0F	00	FF	8D	AB	4D	63	70
5	0D	45	74	71	FB	7F	98	8A	FE	67	A3	A4	0B	B6	E8	95
6	9C	4B	84	BB	C6	12	65	7A	3F	54	4A	C4	2D	2A	49	0A
7	C5	D2	04	99	72	B0	31	6F	01	86	05	C9	3D	D5	EA	BF
8	90	06	1B	66	BE	77	35	CB	EE	F3	1C	03	88	28	6E	29
9	9A	A6	08	34	09	CA	D8	1A	52	56	E4	D6	7D	19	60	5C
A	37	82	4F	1E	AE	AA	11	1D	4E	FD	27	F1	80	E7	62	91
B	B2	5E	36	59	79	17	ED	55	FC	D1	5D	F4	8F	41	CC	9E
C	18	DF	81	23	93	EC	D3	9B	20	E6	A0	13	85	6B	D7	AF
D	CE	DC	7B	57	4C	33	C8	BC	0E	E2	B9	C0	DD	94	B5	B7
E	AC	BD	68	16	C7	6C	14	3E	A1	5F	F8	BA	E9	F6	8E	E3
F	24	2C	F0	3A	B3	C2	42	2E	2B	22	F7	F2	43	5B	9D	10

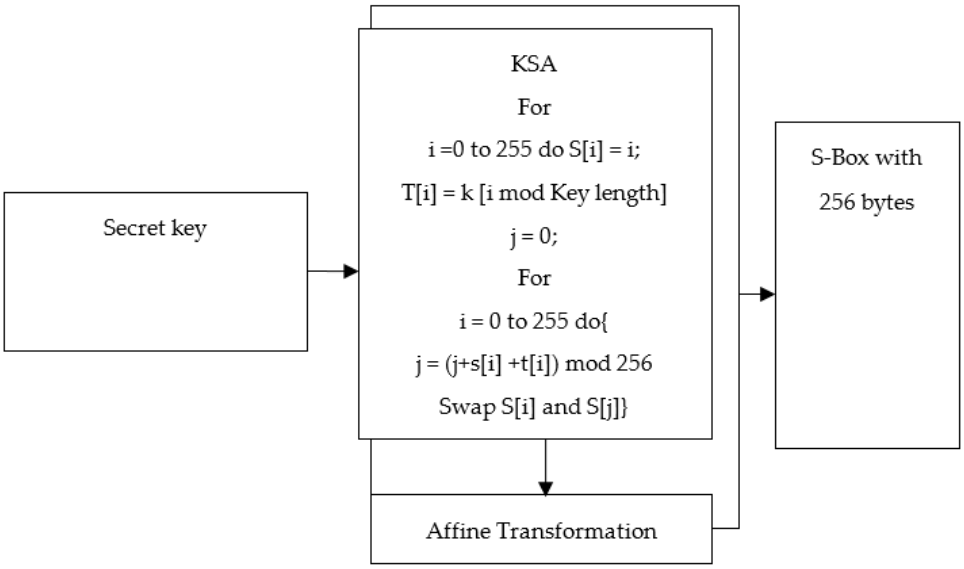

Figure 2. Block diagram of the key dependent S-Box RC4 algorithm.

2.3. RC4 Encryption Algorithm

The RC4 encryption algorithm is a stream cipher based on changes in the nonlinear data table changes, often used within real-time communications [9]. The RC4 algorithm consists of two partitions: the previously mentioned KSA and the Pseudo-Random Generation Algorithm (PRGA), which is used to get a single byte from KSA. Figure 3 shows the performance of the key and plaintext dependent S-Box with RC4 algorithm. In Figure 3, K is the key stream which is XORed with the plaintext that results in exiting the encrypted message. Every byte returned on every iteration forms a byte of encrypted messages. The variable i counts the number of rounds through the encryption process to synchronize the cipher stream.

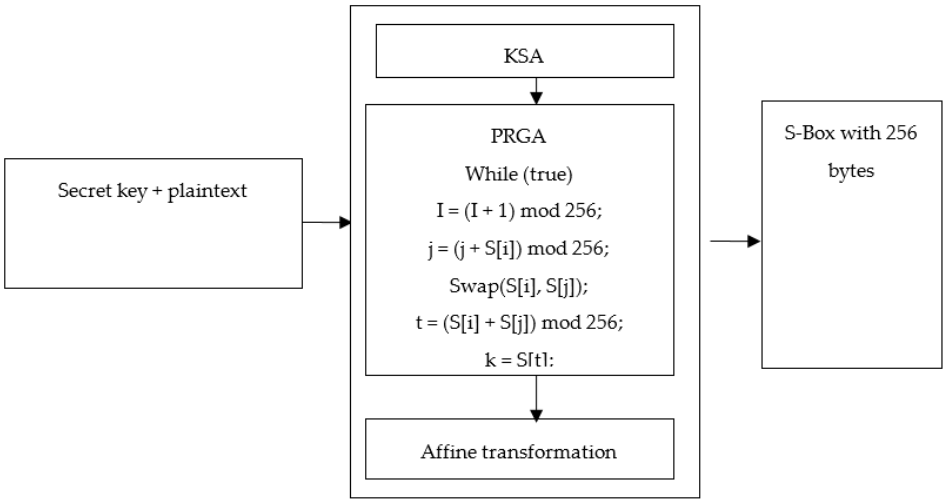

Figure 3. Block diagram of the key and plaintext dependent S-Box with RC4 algorithm.

2.3.1. Key and Plaintext Dependent S-Box with RC4 Algorithm

The S-Box uses two main elements of the RC4 encryption algorithm; particularly the KSA and PRGA. This leads to a robust S-Box and so performs the affine transformation of the produced value, to come up with a brand-new S-Box with 256 bytes. It is used with additional confusion and permutation that guarantee additional security against attacks [19,20]. Table 2 shows the key and

plaintext rely S-Box with RC4 algorithm. Clearly, there is no relation between two consecutive columns or two consecutive rows.

Table 2. Key and plaintext dependent S-Box enforced by RC4 (Key Scheduling Algorithm (KSA) and Pseudo-Random Generation Algorithm (PRGA)) with key "FEDCBA9876543210".

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0F	DB	F9	AD	E2	7F	A9	13	92	85	06	EC	5B	56	6E	49
1	A4	15	9C	37	67	BC	F7	87	F8	0A	EA	C4	39	FB	C3	69
2	31	3E	79	8E	D6	AB	6F	B6	FE	57	E3	32	D8	4A	21	6D
3	80	89	47	51	2A	59	03	D9	B9	BB	71	10	61	D5	84	52
4	EB	E5	9B	05	F5	0C	1C	35	48	D2	07	4F	25	0D	4D	7B
5	9A	65	B4	8B	E4	A0	83	ED	C2	1B	00	F2	62	CF	9E	C5
6	8F	FD	54	F0	F4	8A	8C	64	B7	60	34	B1	97	09	50	1A
7	72	CD	42	4B	96	12	16	0B	D4	93	0E	1E	73	CA	C0	CC
8	A7	45	F3	2B	5A	17	4C	02	88	04	66	40	DA	CE	91	3A
9	46	BA	18	76	86	9F	B0	2E	EE	7A	38	24	70	C8	27	2F
A	63	BF	FA	DF	C1	BD	78	C7	AF	5C	33	28	58	D7	68	22
B	B3	98	D0	E8	7E	82	A3	6A	C6	23	5F	A8	FC	01	AE	9D
C	EF	F1	2C	43	1F	30	29	D3	B5	8D	44	3B	A5	C9	BE	B8
D	E6	3F	19	B2	6B	E9	26	2D	7C	AA	99	DD	A1	7D	DC	1D
E	81	FF	55	11	36	94	AC	74	08	F6	E0	E7	5D	D1	A2	4E
F	3C	CB	20	90	14	41	E1	DE	5E	77	53	6C	A6	95	75	3D

2.3.2. Key and Plaintext Dependent S-Box with RC4 Enforced Henon Chaotic Map

The RC4 is an enforced Henon chaotic map, used to decrease time consumption and for generating a new S-Box with Henon chaotic mapping [19]. Inserting the two-dimensional Henon chaotic mapping into PRGA makes use of the complicated dynamic behavior of RC4 and the useful properties of the chaotic systems. Using a chaotic map ensures extra knowledge security because of confusion and diffusion, which reduces the time consumption of the algorithm without touching the protection [21]. Table 3 shows the S-Box with KSA and PRGA Henon chaotic map.

Table 3. Key and plaintext dependent S-Box with KSA and PRGA enforced by Henon Chaotic Map with key "FEDCBA9876543210".

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	BB	51	C2	BC	E5	54	F3	60	9A	AE	62	1A	F7	09	A8	9C
1	4B	CC	75	2F	50	BF	BA	4C	92	02	EB	78	6F	83	FB	12
2	56	87	61	23	82	6C	2C	79	BD	08	00	76	10	A9	D8	95
3	FE	71	9D	1D	3C	A7	64	D2	8C	1C	D5	FC	B0	D0	B7	A4
4	41	21	E9	59	16	52	5B	97	67	81	B6	66	58	B8	91	45
5	E8	B9	C5	98	27	F0	70	8F	34	4E	03	EF	25	14	29	D9
6	31	0C	57	DC	0D	1F	DE	15	42	26	7A	B4	FD	ED	3B	F4
7	DF	DD	AB	E3	B2	11	49	22	CE	A3	7F	BE	C3	01	36	7D
8	E6	FA	E0	D3	A6	EA	40	88	18	63	6D	CF	8B	94	EC	55
9	EE	7E	A1	D1	5E	39	96	CA	85	B5	7B	8A	06	D7	AF	DB
A	4A,	73	74	35	F9	E7	17	0F	0A	E2	43	F1	5A	C7	93	5C
B	6E	53	65	AA	AC	33	19	C9	13	A2	20	32	05	28	5D	9F
C	6B	48	3D	5F	F5	0B	C6	1E	89	F8	FF	3E	7C	84	4F	80
D	9E	68	CD	8D	6A	CB	3F	38	44	B3	2B	D4	C0	24	2A	69
E	C1	47	A0	8E	0E	1B	37	99	07	E4	A5	2D	90	9B	3A	2E
F	DA	77	B1	F2	AD	4D	F6	30	46	C8	E1	D6	C4	86	04	72

3. The Modified S-Box

Ahmad et al. proposed an encryption algorithm that depends on adding Logistic Chaotic Map (LCM) to the existing substitution process, which also depends on some interesting properties of the chaotic map and substitution boxes [22]. They divided the substituted image into non-overlapping blocks of $Z \times Z$ elements, for the purpose of other diffusion steps. A block of size $Z \times Z$ random values is generated through the chaotic map which is further XORed with the first block of the substituted image). However, in this paper, we propose a technique for generating a secured S-Box. It generates a dynamic S-Box that depends on RC4, enforced by the (LCM) within the PRGA part. In [23], the RC4 encryption is enforced by the Henon chaotic map [19] to decrease time consumption. Inserting a

chaotic signal that features a sensible randomicity into the RC4 algorithm offers higher confusion and diffusion impact on the encryption process. The LCM is an easy distinct algorithmic mathematical relation, which maps the output of one iteration of the operation onto the input of successive iteration.

The LCM mathematical equation is given by [24]:

$$x_{n+1} = r\, x_n(1 - x_n) \tag{1}$$

where r is a system parameter with a value between 0 and 4, x_n is the initial value at intervals between 0 and 1, and n is the number of iterations.

We use the logistic map with parameters r = 3.6, with an initial value = 0.6316 and number of iterations 256. This value is more suitable according to the work of Hamdi et al. [25]. Additionally, the key used in generating the S-box is different than that used in AES encryption algorithm, where the logistic map is used to improve the PRGA in RC4 algorithm to generate our S-box used in AES encryption algorithm

Inserting the one-dimensional LCM into the RC4 algorithm makes use of the extraordinarily complicated dynamic behavior of RC4, and also the sensible properties of the chaotic systems. The steps used are as follows. Initialize KSA after given input key and plaintext. Then, the PRGA is implemented with the LCM. The output is 256 bytes completely different in values; then the affine transformation is performed to urge the proposed S-Box with 256 bytes as shown in Figure 4.

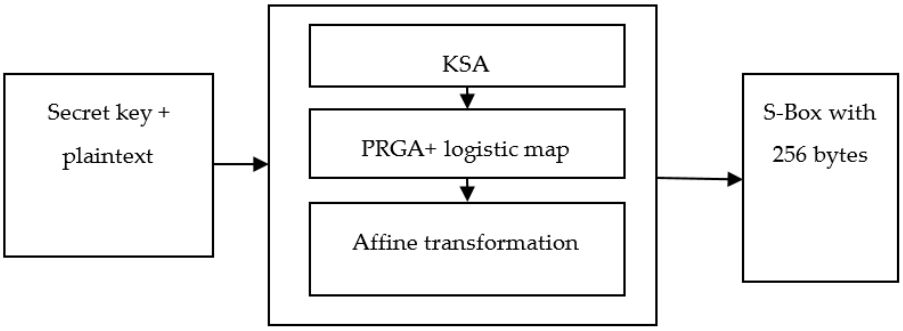

Figure 4. S-BOX using KSA and PRGA enforced by Logistic Chaotic Maps (LCM).

The difference between this S-Box and the dynamic S-Box is that the modified PRGA with the LCM is used in generating the S-Box. The proposed S-Box with different 256 bytes is presented in Table 4.

Table 4. Key and plaintext dependent S-BOX using KSA and PRGA enforced by Logistic Chaotic Maps (LCM) with key "FEDCBA9876543210".

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	1E	39	3A	46	B1	95	35	FA	F9	75	1D	21	D4	C7	E1	A8
1	C1	8D	0C	0D	06	A6	78	CF	E6	5A	02	54	B5	84	A4	22
2	DE	9E	64	A5	3B	C8	EC	C9	37	73	61	30	88	23	ED	89
3	2A	FC	B8	6D	EA	6A	1B	BB	C0	65	C4	AF	53	DF	D1	CD
4	18	15	AC	80	8B	97	43	F2	56	B4	82	71	11	C6	91	F3
5	7F	7B	86	87	52	0A	03	3F	E9	EB	17	0B	C3	9C	C5	2D
6	01	00	DD	F8	F4	83	66	BE	9D	F7	FB	77	1A	D6	74	6C
7	7C	4D	51	5E	F6	3E	E0	D8	90	D3	3D	8A	A9	92	42	7E
8	20	60	6E	DB	E3	FD	25	AD	D0	9A	24	E5	76	6B	34	2B
9	C2	E8	FE	4C	6F	31	A1	58	13	DC	EE	8C	B6	E7	32	5F
A	1F	8E	59	2F	12	79	B2	D7	1C	CE	08	04	85	AE	B3	7D
B	A0	AB	B0	26	4F	F5	67	45	49	2C	A7	9F	7A	E2	81	BC
C	63	33	BF	B9	0F	05	14	94	BD	5C	CC	69	36	2E	72	EF
D	F0	9B	16	3C	62	98	28	D2	CB	40	96	A2	D9	8F	4A	19
E	55	B7	BA	F1	5B	47	68	FF	44	5D	41	57	09	27	99	AA
F	DA	38	A3	4B	48	50	07	CA	E4	93	29	70	4E	0E	D5	10

4. Security Analysis and Results

To measure the degree of security of the S-Box, some cryptographic tests should be applied like, nonlinearity test, avalanche effect, ETPE%, and the strict avalanche criteria [15,23,26]. In this paper, these tests are used to measure the strength of the modified S-Boxes and to compare between the AES static S-Box, dynamic S-Box, S-Box with KSA and PRGA, S-Box with Henon chaotic maps and the modified S-Box with LCM. The simulations are achieved by a C++ program with three keys: "FEDCBA9876543210" "0123456789ABCDEF", "F9E8D7C6B5A4A0123" and the plaintext {00, 11, 22, 33, 44, 55, 66, 77, 88, 99, AA, BB, CC, DD, EE, FF}.

4.1. Nonlinearity

The S-Box's main goal is to provide a nonlinear change from the main text to the encrypted one. Nonlinearity means that the probability of the numbers of bits inverted from the input to the output is 0.5, which means half of all inputs and all changed bits [24,27]. In reality, it is unusual to have the probability exactly equal half. So, we aim to reach a probability of 0.5 or near to 0.5 as much as possible. The nonlinearity test is performed for the explained S-Boxes and the obtained results are shown in Table 5. According to nonlinearity test, the results show that the S-Box with Henon chaotic maps is more secure.

Table 5. Results of the security tests with key "FEDCBA9876543210".

	AES Static S-Box	**S-Box with KSA**	**S-Box with KSA and PRGA**	**S-Box with KSA and PRGA Enforced by Henon Chaotic Map**	**S-Box with KSA and PRGA Enforced by LCM**
Nonlinearity Old plaintext = 00112233445566778899AABBCCDDEEFF	0.546	0.484	0.49218	0.539	0.4843
Avalanche effect New plaintext = 01112233445566778899AABBCCDDEEFF	0.476	0.531	0.4531	0.4296	0.4765
Average time in seconds for generating the S-Box	0.1435	0.0913	0.09455	0.09285	0.0938
Average time in seconds for encryption	0.0297	0.0235	0.019	0.02125	0.019
Average time in seconds for decryption	0.0259	0.019	0.01825	0.0252	0.019
TEPE% (for generating the S-Box)	57.119%		51.77%	54.496%	52.932%
TEPE% for Encryption	26.38%		56.31%	39.76%	56.31%
TEPE% for Decryption	36.315%		41.91%	2.778%	36.31%

4.2. Avalanche Effect

The avalanche effect refers to a fascinating property of cryptanalytic algorithms and is defined by Equation (2). An S-Box is said to satisfy the avalanche effect if each output bits of the ciphertext changes when one input bit in the plaintext changes, provided that at least half the output bits will be flipped [28]. The new plaintext is "01, 11, 22, 33, 44, 55, 66, 77, 88, 99, AA, BB, CC, DD, EE, FF".

$$\text{Avalanche effect} = (\text{NC}/\text{NT}) \times 100 \tag{2}$$

where NC and NT are, respectively, the number of changed bits total number of bits in the cipher text.

The results of the avalanche effect test of the explained S-Boxes is shown in Table 5, the S-Box enforced by LCM seems more secure and has a stronger S-Box.

4.3. Execution Time Performance Efficiency (ETPE)

The execution time is a measure for the relative performance improvement when performing the algorithms. The percentage ETPE is defined as [28]

$$\text{ETPE\%} = \left(\frac{\text{T}_{\text{old}}}{\text{T}_{\text{new}}} - 1\right) \times 100 \tag{3}$$

where T_{old} is the average execution time of the S-Box without modification and T_{new} is the average execution time of the S-Box with modification calculated by C++ for 20 trials.

Table 5 presents the results of the nonlinearity test, avalanche effect test, average execution time (generating S-Box, encryption, and decryption), and ETPE% for the static S-Box, dynamic S-Box, the key and plaintext dependent S-Box with RC4—that depends on KSA and PRGA of RC4 encryption algorithm—and the key and plaintext dependent S-Box with RC4 enforced with Henon chaotic mapping, compared with the proposed key and plaintext dependent S-Box RC4 modified with LCM.

4.4. Strict Avalanche Criterion (SAC)

An S-Box is alleged to meet SAC if one small input bit of the S-Box changes every output bit with a probability of 0.5 [27]. The three investigation techniques for SAC used for testing and contrasting distinctive S-Boxes are: examination of the frequency of various Hamming weights, investigation of the frequency of various differential values ΔY, and investigation of Hamming weights as per the bit position.

4.4.1. Examination of the Frequency of Various Hamming Weights

Right now, the input is that the S-Box with 256 bytes, that is XORed with 2 whimsical numbers from the S-Box and afterwards the Hamming weight for the yield is calculable. The test is rehashed multiple times and afterwards, a number of reiterations is set. [29]

4.4.2. Investigation of the Frequency of Varied Differential Values ΔY

Here, the input is that the S-Box with 256 bytes, that is XORed with 2 of them at random numbers from the S-Box. The test is rehashed multiple times, and afterwards a number of reiterations within the outcome is set. [29]

4.4.3. Investigation of Hamming Weights Consistent with the Bit Position

In this methodology, the input is that the S-Box with 256 bytes, that is XORed with 2 irregular numbers from the S-Box and afterword Hamming weight is indicated by the bit position of redundancy of the result is set. [29]

Figure 5 displays the analysis of the frequency of varied Hamming weights and shows a comparison between the 5 S-Boxes. It is clear that the S-Box KSA and PRGA with LCM is sturdy against attacks because of the static AES S-Box.

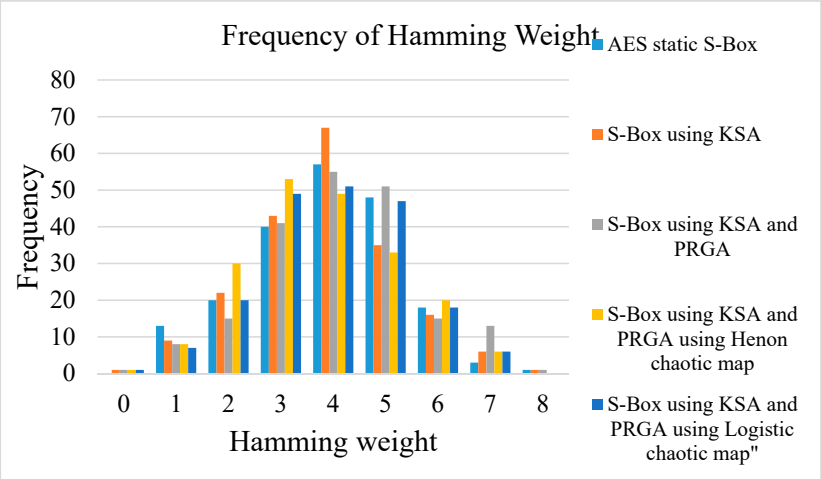

Figure 5. Hamming weights: A comparison between the five S-Boxes.

Figure 6 displays the analysis of the frequency of various differential values of ΔY for the five S-Boxes. Obviously, the S-Box KSA and PRGA with LCM is strong against attacks as the static AES S-Box.

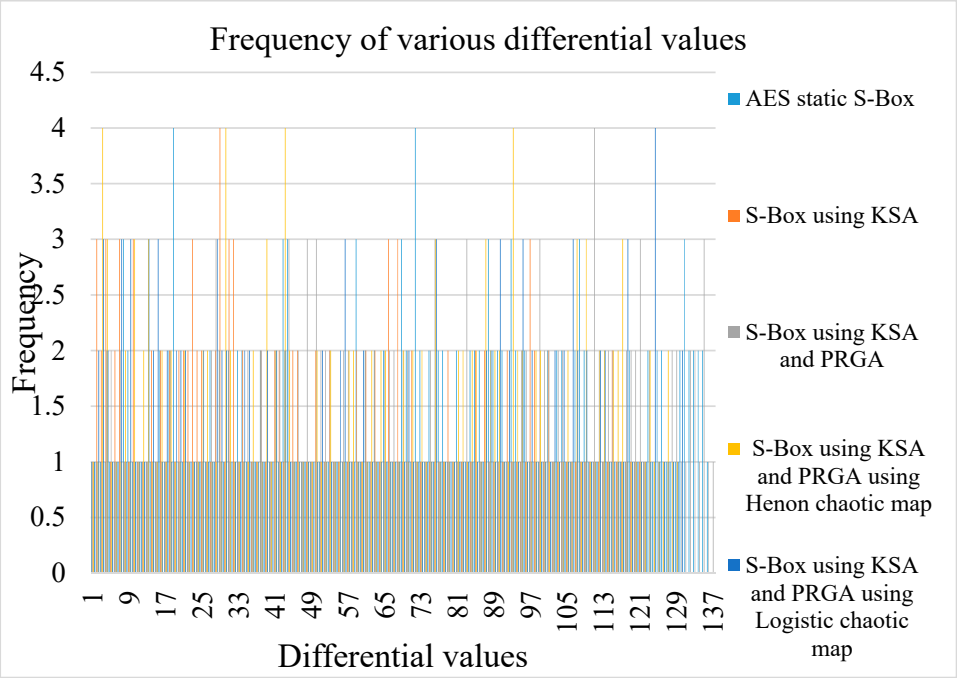

Figure 6. Analysis of the frequency of various values of ΔY for the 5 S-Boxes.

Figure 7 shows the analysis of Hamming weights according to the bit position and a comparison between the five S-Boxes. The S-Box KSA and PRGA enforced by LCM seems strong against attacks as the static AES S-Box.

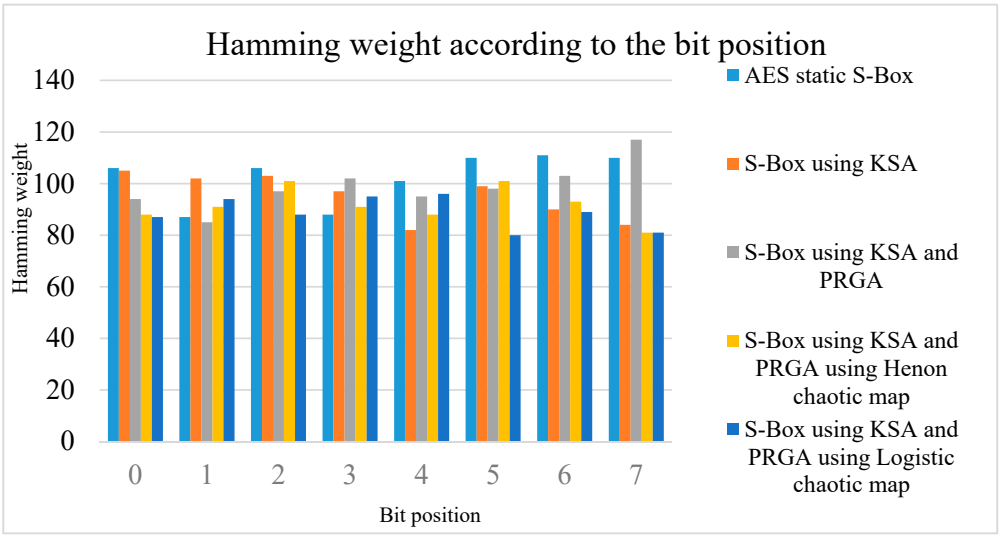

Figure 7. Analysis of Hamming weights according to the bit position for the five S-Boxes.

The procedure of generating the S-Boxes is repeated with another two different keys, "0123456789ABCDEF, and F9E8D7C6B5A40123. The new results are shown in Table 6 and Figures 8–10 for the key "0123456789ABCDEF and in Table 7 and Figures 11–13 for the key F9E8D7C6B5A40123.

Table 6. Results with the key "0123456789ABCDEF".

	AES Static S-Box	S-Box KSA	S-Box KSA and PRGA	S-Box KSA and PRGA Enforced by Henon Chaotic Map	S-Box KSA and PRGA Enforced by LCM
Nonlinearity Old plaintext = 00112233445566778899AABBCCDDEEFF	0.546	0.486	0.492	0.52	0.473
Avalanche effect New plaintext = 01112233445566778899AABBCCDDEEFF	0.476	0.4375	0.5781	0.4218	0.4921
Average time in seconds for generating the S-Box	0.14345	0.1059	0.1062	0.08895	0.1124
Average time in seconds for encryption	0.0297	0.019	0.02255	0.019	0.0213
Average time in seconds for decryption	0.0259	0.019	0.02105	0.01975	0.0196
TEPE% (for generating S-Box)	35.39%		35.075%	61.27%	52.932%
TEPE% for Encryption	56.315%		31.707%	56.315%	56.315%
TEPE% for Decryption	36.315%		23.04%	31.139%	36.315%

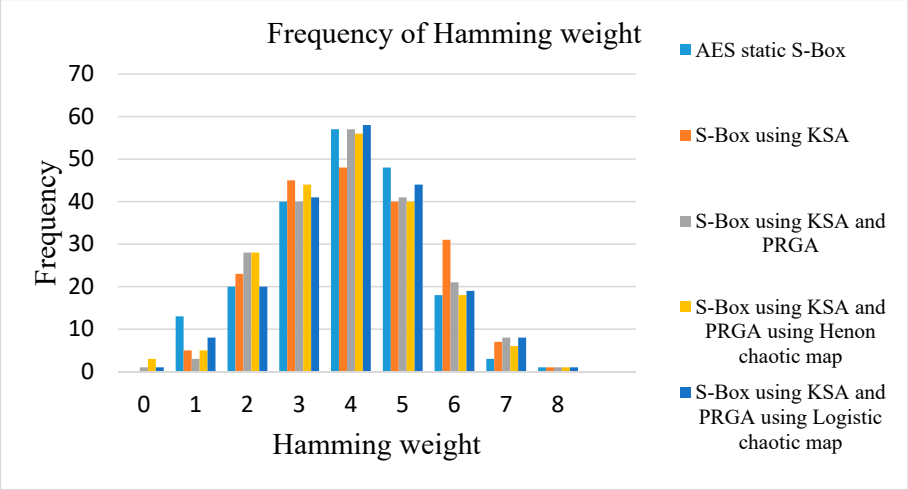

Figure 8. Hamming weights: a comparison between the five S-Boxes with the key "0123456789ABCDEF".

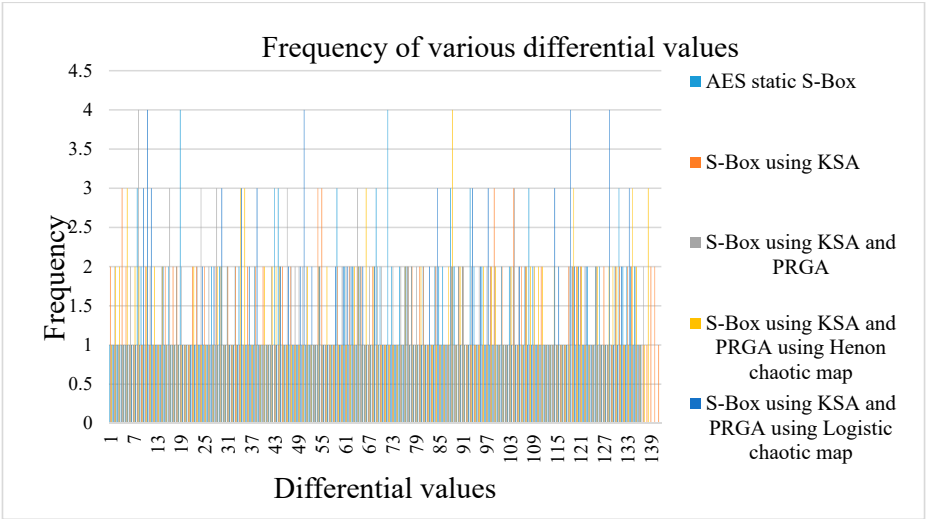

Figure 9. Analysis of the frequency versus differential values, ΔY, for the 5 S-Boxes with the key "0123456789ABCDEF.

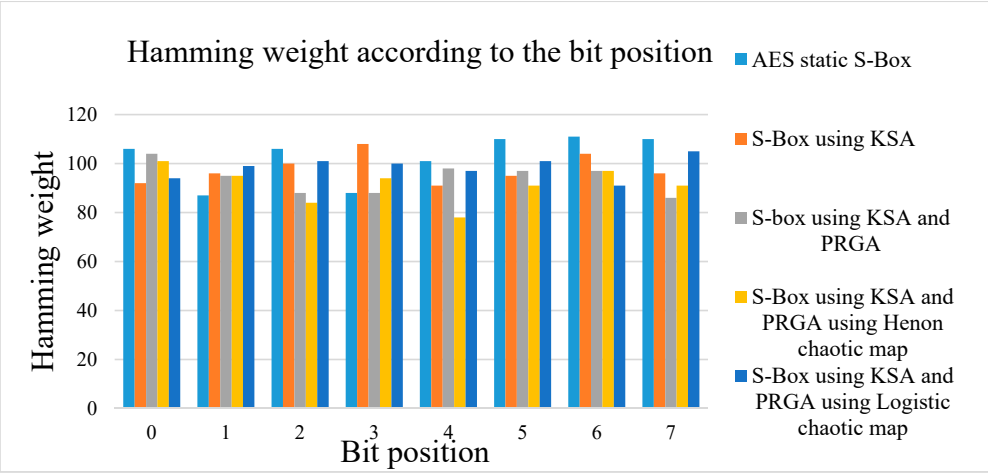

Figure 10. Analysis of Hamming weights according to the bit position for the five S-Boxes using the key "0123456789ABCDEF ".

Table 7. Results with the key "F9E8D7C6B5A40123".

	AES Static S-Box	S-Box KSA	S-Box KSA and PRGA	S-Box KSA and PRGA Enforced by Henon Chaotic Map	S-Box KSA and PRGA Enforced by LCM
Nonlinearity Old plain 00112233445566778899AABBCCDDEEFF	0.546	0.46875	0.484375	0.61718	0.5312
Avalanche effect New plaintext = 01112233445566778899AABBCCDDEEFF	0.476	0.437	0.5	0.5468	0.4765
Average time in seconds for generating the S-Box	0.1434	0.0984	0.0992	0.0946	0.10075
Average time in seconds for encryption	0.0297	0.0182	0.0190	0.0205	0.0175
Average time in seconds for decryption	0.0259	0.02135	0.0167	0.0228	0.01918
TEPE%(for generating the S-Box)	45.78%		44.606%	51.63%	42.382%
TEPE% for Encryption	62.74%		55.905%	44.878%	69.714%
TEPE% for Decryption	21.31%		55.089%	13.596%	35.036%

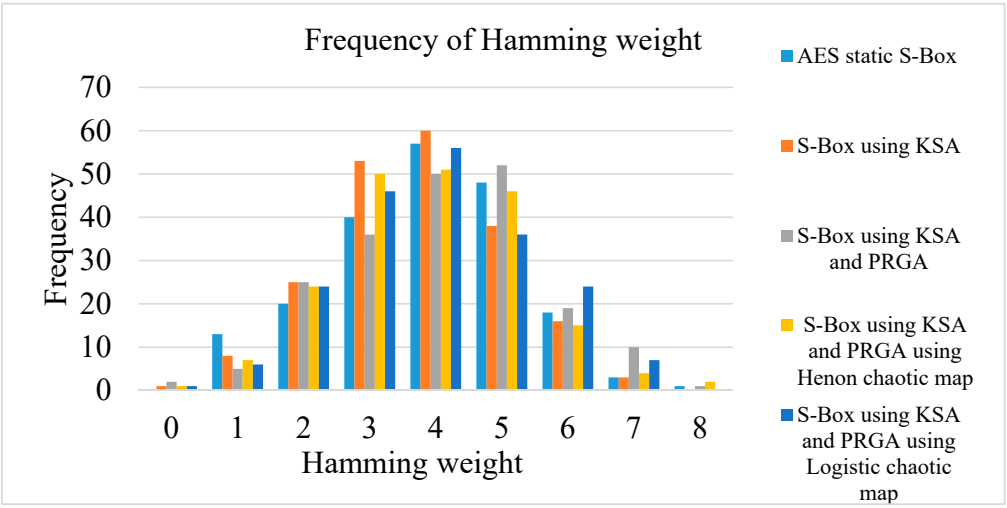

Figure 11. Hamming weights: a comparison between the five S-Boxes with the key "F9E8D7C6B5A40123".

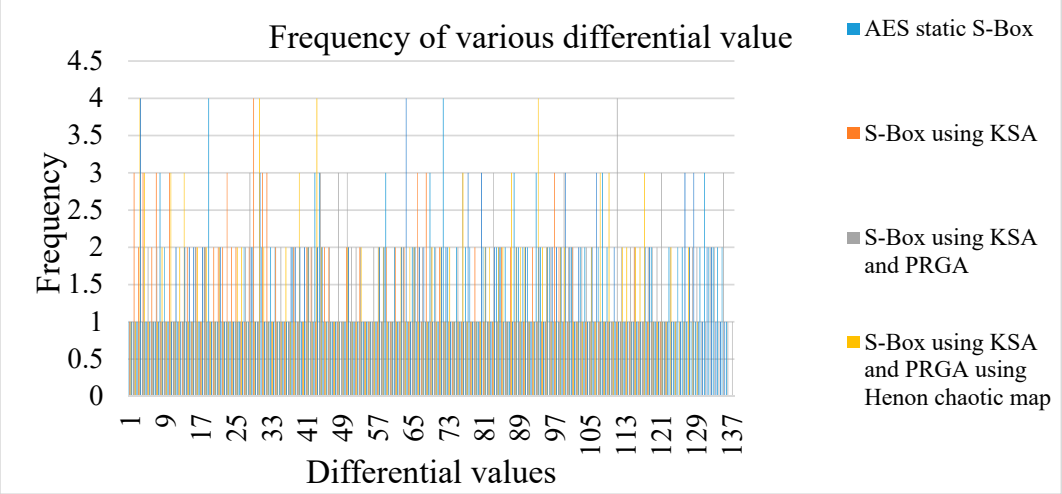

Figure 12. Analysis of the frequency of various differential values ΔY for the five S-Boxes with the key "F9E8D7C6B5A40123".

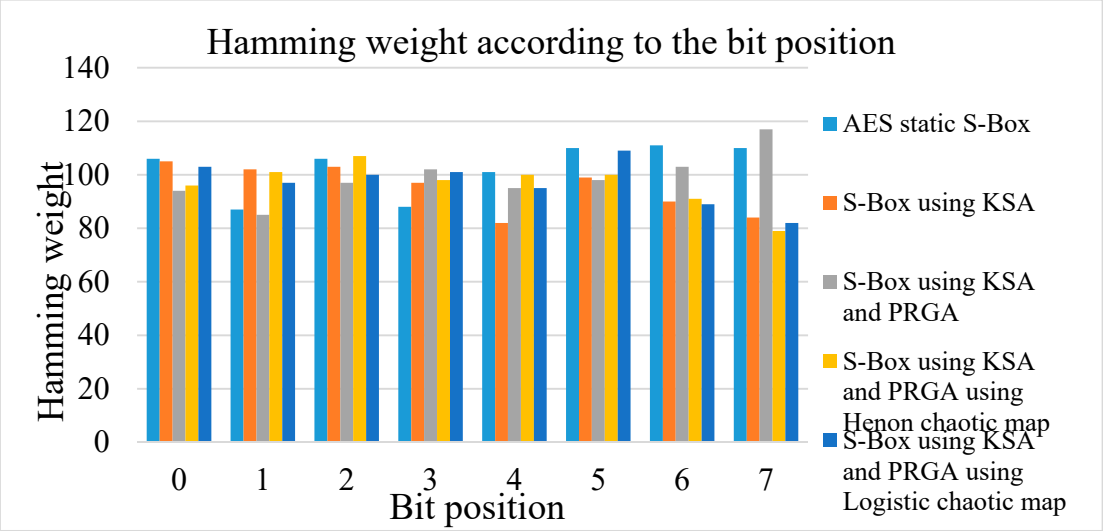

Figure 13. Analysis of Hamming weights according to the bit position for the five S-Boxes with the key "F9E8D7C6B5A40123".

For the same key, the nonlinearity test shows that the S-Box with KSA and PRGA is the nearest to 0.5, while for the avalanche effect test, the S-Box with KSA and PRGA enforced by LCM is the nearest to 0.5. The average time for generating the S-Box with KSA and PRGA enforced by LCM, for encryption and decryption, is the smallest time (0.019 s). This means this S-Box is strong and needs less time for execution. The use of different keys does not change the results significantly.

5. Conclusions

The S-Box is the keystone of symmetric cryptography systems. This paper proposes a new algorithm to generate a modified S-Box with new keys, specifically a key and plaintext-dependent S-Box using an improved RC4 encryption algorithm with Logistic Chaotic Maps (LCM). The paper studies the proposed S-Box in comparison to the state-of-the-art variants of the S-Boxes, namely, static S-Box, dynamic S-Box, KSA and PRGA S-Box, and RC4 S-Boxes with Henon chaotic maps. The experimental results demonstrate that the proposed S-Box leads to increased security. In particular, the strict avalanche test shows that proposed S-Box achieves a ciphertext bit-flip ratio of 0.4765 while maintaining a minimum time elapsed of 19 milliseconds for encryption and decryption.

In this paper, a new algorithm is proposed to generate a modified S-Box with new keys, specifically a key and plaintext-dependent S-Box using an improved RC4 encryption algorithm with LCM. This algorithm has a better time consumption in generating S-box using LCM, where it takes 19 ms in both AES encryption and decryption. Moreover, the execution time performance efficiency, which is a measure for the performance improvement, reached to 56.31% for AES encryption algorithm. The proposed S-Box depends on the RC4 algorithm which is improved with LCM which increases the permutation and randomness.

As future directions, we recommend the following suggestions: (i) The RFID system is used in many applications and needs data security. To increase the security, the S-box can be generated using the DNA genetic algorithms. (ii) Our proposed S-Box could be applied in an image encryption. (iii) More other tests for evaluating the obtained results could be considered.

Author Contributions: Conceptualization, H.H.F., A.A.M. and E.-S.A.E.-B.; methodology, A.S.E.B., H.H.F. and A.A.M.; software A.S.E.B. and H.H.F.; validation, A.S.E.B., M.H.A. and A.A.M.; formal analysis, A.S.E.B. and H.H.F.; investigation, A.A.M. and E.-S.A.E.-B.; resources, H.H.F., E.-S.A.E.-B., and A.A.M.; data curation, A.A.M. and M.H.A.; writing—original draft preparation, A.S.E.B.; writing—review and editing, A.A.M. and M.H.A.; visualization, A.A.M. and H.H.F.; supervision, A.A.M., E.-S.A.E.-B., and H.H.F.; project administration, A.S.E.B., M.H.A. and A.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kamdar, N.; Sharma, V.; Nayak, S. An Overview Paper on RFID Technology Its Applications and Classification of Security/Privacy Attacks and Solutions. *IRACST Int. J. Comput. Sci. Inf. Technol. Secur.* **2016**, *6*, 179–183.
2. Chari, S.; Jutla, C.; Rao, J.R.; Rohatgi, P. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards. In Proceedings of the Second Advanced Encryption Standard (AES) Candidate Conference, Rome, Italy, 22–23 March 1999; p. 15.
3. Finkenzeller, K. *RFID Handbook*, 2nd ed.; John Wiley & Sons Ltd.: Chichester, UK, 2003. [CrossRef]
4. Feldhofer, M.; Dominikus, S.; Wolkerstorfe, J. *Strong Authentication for RFID Systems*; Institute for Applied Information Processing and Communications, Graz University of Technology: Graz, Austria, 2004; pp. 350–370. [CrossRef]
5. *Victimization the AES Algorithm*; School of Transportation, Wuhan University of Technology: Wuhan, China, 2005. [CrossRef]
6. Weis, S.A.; Sarma, S.E.; Rivest, R.L.; Engels, D.W. *Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems*; Laboratory for Computer Science Auto-ID Center Massachusetts Institute of Technology: Cambridge, MA, USA, 2004; pp. 201–212. Available online: https://en.wikipedia.org/wiki/QD_code (accessed on 2 December 2018). [CrossRef]
7. Sarma, S.E.; Weis, S.A.; Engels, D.W. *RFID Systems and Security and Privacy Implications*; Auto-ID Center Massachusetts Institute of Technology: Cambridge, MA, USA, 2003. [CrossRef]
8. Mei, Y.; Jiang, Y. Secure RFID System Based on RC4 Chaotic Algorithm. *J. Comput. Inf. Syst.* **2013**, *9*, 2083–2091. Available online: www.researchgate.net (accessed on 25 January 2020).
9. Stallings, W. *Cryptography and Network Security, Principles and Practices*, 4nd ed.; Pearson Education, Prentice-Hall of India Pvt. Limited: New Delhi, India, 2006; Available online: https://dl.acm.org/doi/book/10.5555/280574 (accessed on 17 December 2019).
10. Shannon, C. Communication Theory of Secrecy Systems. *Bell Syst. Tech. J. MD Comput.* **1998**, *15*, 57–64. [CrossRef]
11. Ansari, S.; Ahmad, J.; Shah, S.S.; Bashir, A.K.; Boutaleb, T.; Boutaleb, S. Chaos-based privacy preserving vehicle safety protocol for 5G Connected Autonomous Vehicle networks. *Trans. Emerg. Telecommun. Technol.* **2020**, *5*, e3966.
12. The Mathworks: Galois Field Computations. Communications Toolbox. 2001. Available online: http://www.mathworks-.com/Access/help-desk/helptoolbox/comm./tutor3.shtml (accessed on 21 November 2019).

13. Wenceslao, V.F., Jr. Performance Efficiency of Modified AES Algorithm Using Multiple S-Boxes. *Int. J. New Comput. Archit. Appl.* **2015**, *5*, 1–9. [CrossRef]
14. Fahmy, F.; Salama, G. A Proposal for Key-Dependent AES. In Proceedings of the 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT2005), Susa, Tunisia, 27–31 March 2005; pp. 1–7. Available online: http://www.setit.rnu.tn/last_edition/setit2005/image-video/53.pdf (accessed on 14 February 2020).
15. Hosseinkhani, R.; Javadi, H.H.S. Using Cipher Key to Generate Dynamic S-Box in AES Cipher System. *Int. J. Comput. Sci. Secur.* **2012**, *6*, 19–28. Available online: https://www.cscjournals.org/manuscript/Journals/IJCSS/Volume6/Issue1/IJCSS-630.pdf (accessed on 15 January 2020).
16. Zhu, S.; Wang, G.; Zhu, C. A secure and fast image encryption scheme based on double chaotic s-boxes. *Entropy* **2019**, *21*, 790. [CrossRef]
17. Lu, Q.; Zhu, C.; Wang, G. A Novel S-Box Design Algorithm Based on a New Compound Chaotic System. *Entropy* **2019**, *21*, 1004. [CrossRef]
18. Webster, A.F.; Travares, S.E. *On The Design of S-Boxes*; Queen's University Kingston: Kingston, ON, Canada, 1998. [CrossRef]
19. Agrawal, B.; Agrawal, H. Implementation of AES and RSA Using Chaos System. *Int. J. Sci. Eng. Res.* **2013**, *4*, 1413–1417. Available online: https://www.semanticscholar.org/paper/Implementation-of-AES-and-RSA-Using-Chaos-System-Agrawal-Agrawal/0a200eca5f70d0f597ce0ac747ca40f30ce4b904 (accessed on 15 December 2019).
20. Sosa, P.M. *Calculating Nonlinearity of Boolean Functions with Walsh-Hadamard Transform*; UCSB: Santa Barbara, CA, USA, 2016; pp. 1–4. Available online: http://konukoii.com/blog/wp-content/uploads/2016/06/FinalPaper.pdf (accessed on 15 December 2019).
21. Eshmawi, A.A.; Mahmoud, E.E. Secure Communications Via Complex Phase Synchronization of Pair Complex Chaotic Structures with A Similar Structure of Linear Terms with Modifying in Nonlinear Terms. *Alex. Eng. J.* **2020**, *59*. in press. [CrossRef]
22. Ahmad, J.; Hwang, S.O. Chaos-based diffusion for highly autocorrelated data in encryption algorithms *Signal Image. Video Process.* **2015**, *13*, 1839–1850. [CrossRef]
23. El Batouty, A.S.; Farag, H.H.; Mokhtar, M.A.A.; El-Badawy, E.A. New Hybrid AES Static S-Box Algorithm Using Chaotic Maps. In Proceedings of the 2019 IEEE International Conference on Information Technologies (InfoTech-2019), Constantine and Elena, Bulgaria, 19–20 September 2019; pp. 1–5. [CrossRef]
24. Wen, H. *A Review of the Henon Map and i.ts Physical Interpretations*; School of Physics Georgia Institute of Technology: Atlanta, GA, USA, 2014; pp. 30332–30430. Available online: http://chaosbook.org/projects/Wen14.pdf (accessed on 22 December 2019).
25. Hamdi, M.; Rhouma, R.; Belghith, S. A Very Efficient Pseudo-Random Number Generator Based On Chaotic Maps and S-Box Tables. *Int. J. Electron. Commun. Eng.* **2015**, *9*, 481–485. [CrossRef]
26. Abd-ElGhafar, I.; Rohiem, A.; Diaa, A.; Mohammed, F. Generation of AES Key Dependent S-Boxes Using RC4 Algorithm. In Proceedings of the 13th International Conference on Aerospace Sciences & Aviation ASAT-13, Cairo, Egypt, 26–28 May 2009; pp. 1–9. [CrossRef]
27. Alkhaldi, A.H.; Hussain, I.; Gondal, M.A. A Novel Design for the Construction of Safe S-Boxes Based on TDERC Sequence. *Alex. Eng. J.* **2015**, *5*, 65–69. [CrossRef]
28. Kazlauskas, K.; Kazlauskas, J. Key-Dependent S-Box Generation in AES Block Cipher System, Informatica, Institute of Mathematics and Informatics. *Vilnius* **2009**, *20*, 23–34. [CrossRef]
29. Mar, P.P.; Latt, K.M. New Analysis Methods on Strict Avalanche Criterion of S-boxes. *J. Math. Comput. Phys. Electr. Comput. Eng.* **2008**, *2*, 150–154. Available online: https://www.researchgate.net/publication/242568097_New_Analysis_Methods_on_Strict_Avalanche_Criterion_of_S-_Boxes (accessed on 28 January 2020).