Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2702653, 12 pages
https://doi.org/10.1155/2020/2702653

Research Article

Generation of Dynamical S-Boxes for Block Ciphers via Extended Logistic Map

B. B. Cassal-Quiroga and E. Campos-Cantón

División de Matemáticas Aplicadas, Instituto Potosino de Investigación Científica y Tecnológica A. C., Camino a la Presa San José 2055, Col. Lomas 4 Sección, C.P. 78216, San Luis Potosí, S.L.P., Mexico

Correspondence should be addressed to E. Campos-Cantón; eric.campos@ipicyt.edu.mx

Received 14 November 2019; Revised 5 February 2020; Accepted 24 February 2020; Published 19 March 2020

Guest Editor: Lazaros Moysis

In this work, we present a simple algorithm to design $n \times n$-bits substitution boxes (S-boxes) based on chaotic time series of the logistic map for different carrying capacities. The use of different carrying capacities in the chaotic map leads to low computational complexity, which is desirable to get high-speed communication systems. We generate a main sequence by means of two auxiliary sequences with uniform distribution via the logistic map for different carrying capacities. The elements of the main sequence are useful for generating the elements of an S-box. The auxiliary sequences are generated by considering lag time chaotic series; this helps to hide the chaotic map used. The U-shape distribution of logistic chaotic map is also avoided, in contrast with common chaos-based schemes without considering lag time chaotic series, and uncorrelated S-box elements are obtained. The proposed algorithm guarantees the generation of strong S-boxes that fulfill the following criteria: bijection, nonlinearity, strict avalanche criterion, output bits independence criterion, criterion of equiprobable input/output XOR distribution, and maximum expected linear probability. Finally, an application premised on polyalphabetic ciphers principle is developed to obtain a uniform distribution of the plaintext via dynamical S-boxes.

1. Introduction

Nowadays, we are in the era of informatics, and due to a large number of attacks, it is important to adequately protect the information to be transmitted and avoid the possible misuse of it. The aforementioned comment motivates the generation of different approaches to have secure cryptographic systems. In general, cryptosystems can be divided into two classes: stream cipher and block cipher. A stream cipher takes one bit and transforms it into one output bit. Meanwhile, a block cipher takes m input bits and transforms them into m output bits. The core of this transformation in block ciphers is static S-boxes. The S-boxes give cryptosystems the confusion property described by Shannon [1] and are used in conventional block ciphers such as the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). The security in these cryptographic systems depends mainly on the properties of S-boxes used. A strong S-box fulfills the following criteria: bijection, nonlinearity, strict avalanche criterion (SAC), and the output bit independence criterion (BIC) [2]. If an S-box fulfills the above criteria, then it is called "good S-box." Other desirable characteristics are being resistant to linear and differential cryptanalysis attacks. The construction of cryptographically secure dynamic S-boxes is a field of interest in the area of cryptography.

In recent years, many papers have been reported and focus on studying cryptosystems based on chaos, see ref. [3–16], this is, because of the relationships that exists between the chaotic system properties and the cryptosystem properties. In ref. [17] the relationship between these properties are given; for instance, confusion is related to ergodicity, the diffusion property with sensitivity to initial conditions and the deterministic dynamic with the deterministic pseudorandomness. Taking advantage of the chaotic system's properties, a strong and dynamic S-Box is proposed that fulfills the criteria of a good S-box.

Regarding the generation of S-box based on chaos, some algorithms have been developed using discrete dynamical systems. For example, in ref. [3, 5, 8–10, 16], the generation of substitution boxes was introduced through using only a time series of a map or by combining two time series of different maps. In the same way, there are algorithms that use continuous chaotic dynamical systems [4, 6, 11, 12]. Nevertheless, these algorithms do not guarantee that the series used have a uniform distribution, in contrast with the approach used in this work. There are also algorithms that consider the mixing of time series of continuous and discrete dynamical systems [7, 13], and in ref. [18] the algorithm is built via time-delay series. Other types of encryption algorithms consider so-called hidden attractors [15]. The advantage of using discrete chaotic dynamical systems is that from one iteration to another, the elements of the time series are uncorrelated. However, this does not happen if a continuous chaotic dynamical system is used, and the elements of the time series are strongly correlated. Therefore, many iterations are needed, and the calculation of the mutual information between elements of the time series is necessary to be able to say when they are uncorrelated, which implies a higher computational cost.

In chaos-based encryption schemes, pseudorandom sequences via chaotic maps are generally used as one-time pad for encrypting messages. Since encryption schemes, based on low dimensional chaotic map, have low computational complexity, they can be analyzed with low computational cost using iteration and correlation functions [19]. Time-delay chaotic series have complex behavior and erase the trace of the mapping that generates them. In ref. [20], the generation of pseudorandom series with good statistical properties was proposed using lag time series from the logistic map. Using this kind of lag time series, it is possible to hide the map used to build them. Usually, the chaotic maps used for cryptography have been normalized to map the interval [0, 1] to itself, *i.e.*, f: [0, 1] $\longrightarrow$ [0, 1]. Now, in this work, we explore a different carrying capacity to map the interval [0, 2^8] to itself, *i.e.*, f: [0, 256] $\longrightarrow$ [0, 256], for this case, the logistic map is known as the extended logistic map. This simple maneuver of using the extended logistic map allows the generation algorithm to calculate dynamic S-boxes faster than its standardized version.

In this paper, we present an algorithm to design $n \times n$-bits S-boxes based on a main sequence generated by mixing two auxiliary sequences using lag time chaotic series. These chaotic time series are computed with the logistic map considering different carrying capacities. Because of the logistic map considers a carrying capacity parameter, it is easy to adjust this parameter to map the interval [0, 2^8] to itself, so each iteration generates a byte instead of a bit. The generation of a byte with each iteration helps with the computational time to produce S-boxes because fewer iterations are needed. Notice that the generation of bytes can be realized in a different way, for example, by partitioning the interval [0, 1] into 256 subintervals. So each subinterval corresponds to a byte. The difference between both approaches of byte generation is the computing program; then we implement the approach based on the extended logistic map. Two lag time chaotic series are mixed to favor a uniform distribution in the main sequence that generates S-boxes. In addition, the logistic map is hidden if the first return map is plotted with the main sequence. Several statistical tests are carried out to evaluate the performance of these proposed S-boxes. The results show that all the criteria are met for a good S-box and with high immunity to resist differential cryptanalysis and linear cryptanalysis.

The rest of the work is organized as follows: in Section 2, an algorithm for generating substitution boxes based on binary sequences of the logistic map is presented. In Section 3, a dynamic analysis of the extended logistic map and the proposed scheme to generate a dynamic S-box based on the map are presented. In Section 4, the criteria for a "good" $n \times n$-bits S-box are described. In Section 5, the performance analysis of the proposed algorithms are provided and compared with other S-boxes reported in the literature. In Section 6, an application to obtain a uniform distribution of the plaintext via dynamical S-boxes is presented. Finally, conclusions are drawn in Section 7.

2. Dynamical S-Boxes Based on CSPRNG

First, we consider an algorithm to generate dynamic S-boxes based on a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG), which was proposed by García-Martínez and Campos-Cantón [20] and tested with the suite of NIST. The CSPRNG is based on two lag time series generated with the logistic map, f_L: $I \longrightarrow I$, which is defined as

$$f_L(\alpha, x_i) = \alpha x_i (1 - x_i), \tag{1}$$

where x is the state variable of the logistic map, and α is the parameter of the system.

2.1. The First Algorithm for S-Box Design via CSPRNG. The steps of the algorithm are simple as shown below:

Step 1. Select initial conditions x_{01} and x_{02} for CSPRNG in order to generate the stream of bits $s_0, s_1, s_2, \ldots$

Step 2. Generate the block sequence of n-bits each, $C_0 = (s_0, s_1, \ldots, s_{n-1})$, $C_1 = (s_n, s_{n+1}, \ldots, s_{2n-1})$, $C_2 = (s_{2n}, s_{2n+1}, \ldots, s_{3n-1})$, $\ldots$

Step 3. Convert the blocks $C_0, C_1, C_2, \ldots$ of n-bits to integer numbers $D_0, D_1, D_2, \ldots$.

Step 4. Discard the repeated elements D's to select 2^n different values. The rule to discard an element is as follows: if $D_i = D_j$ with $i < j$ then discard D_j.

Step 5. Create the S-Box with the 2^n different elements of D's.

Once the procedure is over, the proposed algorithm returns a $n \times n$-bits S-box with distinct 2^n values. Note that D_0 is the first element of the S-box, but the second element could be not D_1 if $D_0 = D_1$. However, enough 2^n elements have been generated to build the S-box. Each block C's is comprised by n-bits, $s_j, s_{j+1}, \ldots, s_{j+n-1}$, which are related with the functions f_i, with $i = 1, \ldots, n$.

To exemplify the generation of dynamic S-boxes, consider $n = 8$, this allows us to build a 8×8 bits S-Box. The other parameters are set as follows: $\alpha_1 = 4$, $\alpha_2 = -2$ and two arbitrary initial conditions x_{01} and x_{02}, then the S-Box obtained is shown in Table 1 for $x_{01} = 0.8147$, and $x_{02} = 0.9058$. This proposed substitution box has the properties of confusion and diffusion, which are of vital importance for the block ciphers. In this case, the time needed to calculate an S-box was 1.096 seconds.

3. Analysis of the Extended Logistic Map

In the context of mathematics, it is possible to consider the logistic map at different intervals, $[0, \kappa]$, with $0 < \kappa \in \mathbb{R}$, and the parameter α positive and restricted to an interval such that the orbit of an initial condition $x_0 \in [0, \kappa]$ does not escape to infinity. In this context, the logistic map for $1 < \kappa$ is called extended logistic map. For our interest, we consider the extended logistic map as $f_{LE}(\alpha, x)$: $[0, 2^n] \longrightarrow [0, 2^n]$, so that it is defined as follows:

$$f_{LE}(\alpha, x) = \alpha x_i \left(2^n - x_i\right), \quad n \in \mathbb{Z}^+, \tag{2}$$

for the bifurcation parameter $\alpha \in [0, (4/2^n)]$ and $x_0 \in [0, 2^n]$.

Nevertheless, as it is explained in [21], mathematically, it is possible to consider negative values. As mentioned above, the extended logistic map is now studied with α in the interval $[-(2/2^n), 0)$. The dynamical system (2) has one or two fixed points located at $x_1^* = 0$ and at $x_2^* = 2^n\alpha - 1/\alpha$, for $\alpha \neq 0$.

For cryptographic purposes, the value of $n = 8$ is chosen because a pixel, specifically, is represented by 8 bits (2^8 colors), and 1 byte = 8 bits. For $\alpha \in [-(2/2^8), (4/2^8)]$, the orbits do not escape to infinity for any initial conditions in an appropriate interval determined by the value and sign of the α parameter. Figure 1 shows the shape of the extended logistic map for $\alpha = -2/2^8$ in blue triangles and for $\alpha = 4/2^8$ in black crosses. It can be seen that the system has one or two fixed points depending on the value of α which are located at $x_1^* = 0$ and $x_2^* = (2^8\alpha - 1)/\alpha$ for $\alpha \neq 0$.

The stability of the fixed points is displayed in Figure 2 where a circle denotes a stable or attractive fixed point, while a cross denotes that the fixed point is unstable or repulsive. The stability of these fixed points change according to the parameter α, *i.e.*, when $|f'_{LE}(x_1^*)| < 1$ and $|f'_{LE}(x_2^*)| < 1$; then the fixed points x_1^* and x_2^* are stable, respectively, and they are unstable when $|f'_{LE}(x_1^*)| > 1$ and $|f'_{LE}(x_2^*)| > 1$. The case of interest is the last, because the system presents complex behavior; this is, both fixed points are repulsive, $|f'_{LE}(x_1^*)| = |2^8\alpha| > 1$ and $|f'_{LE}(x_2^*)| = |-2^8\alpha - 2| > 1$. The x_1^* fixed point is repulsive for $-2/2^8 \leq \alpha < -1/2^8$ or $1/2^8 < \alpha \leq 4/2^8$. On the other hand, the x_2^* fixed point is repulsive for $-2/2^8 \leq \alpha < 1/2^8$ but $\alpha \neq 0$, or $3/2^8 < \alpha \leq 4/2^8$. So the interesting values are $\alpha \in [-2/2^8, -1/2^8] \cup [3/2^8, 4/2^8]$; this is the condition to have both repulsive fixed points.

The dynamical system (2) bifurcates when $|f'_{LE}(x_1^*)| = 1$ and $|f'_{LE}(x_2^*)| = 1$, this happens for x_1^* when $\alpha = \pm 1/2^8$, and for x_2^* the bifurcation values are given by $\alpha = 1/2^8$ and $3/2^8$. It is possible to analyze the behavior of the system by means of a bifurcation diagram, which is shown in Figure 3. This diagram shows orbit's values as a function of α parameter and the route to chaos are period-doubling bifurcations at $\alpha = 3/2^8$ and period-halving bifurcations at $\alpha = -1/2^8$. There are intervals for the parameter α near to $-2/2^8$ and $4/2^8$ where the extended logistic map $f_{LE}(\alpha, x)$ behaves chaotically.

There are several approaches to demonstrate that a system is chaotic; one of them is to prove that the dynamical system fulfills the definition given by Devaney [22] and another approach is based on the Lyapunov exponent [23, 24]. We use this last concept to prove chaos and the Lyapunov exponent of Equation (2) is shown in Figure 4. The graph of Lyapunov exponents is symmetric with respect to $\alpha = 1/2^8 \approx 0.0039$, the chaotic behavior of the extended logistic map appears for values of the parameter α near $-2/2^8 \approx -0.0078$ and $4/2^8 \approx 0.0156$. The local stability of the fixed points are in accordance with the Lyapunov exponent values, for example, when $\alpha \in (-1/2^8 \approx -0.0039, 3/2^8 \approx 0.0117)$, the orbits of the system converge at a fixed point, and when the bifurcations occur, the orbits converge at periodic orbits up to chaos appears.

The aim is to use the extended logistic map to generate a time series with uniform distribution and without evidencing the mapping used. To achieve this, there is an approach based on two chaotic time series of the extended logistic map given in [20]. Following Lyapunov exponent analysis, the values of α are arbitrarily selected within the chaos region, so it is considered $\alpha = -2/2^8$ and $4/2^8$.

The extended logistic map for these parameter values is invariant in different intervals as follows:

$$\begin{aligned} f_{-2/2^8}\colon\ [-128, 384] &\longrightarrow [-128, 384], \\ f_{4/2^8}\colon\ [0, 256] &\longrightarrow [0, 256]. \end{aligned} \tag{3}$$

It is worth mentioning that the time series generated with both parameter values have a U-shape distribution.

3.1. S-Box Construction via Integer Chaotic Lag Time Series. The main idea of the proposed algorithm for the generation of dynamic S-boxes is to mix two lag time series based on the extended logistic map $f_{LE}\colon I \longrightarrow I$. The interval I is determined by the parameter α. Let $M1$ and $M2$ be two time series generated with the extended logistic map by means of the following considerations: (i) given two arbitrary initial conditions x_{01}, x_{02}, such that, $x_{01} \neq x_{02}$; (ii) two different bifurcation parameter values α_1 and α_2; and (iii) l-units of memory for each time series $x_{(i-k_{l-1})1}, \ldots, x_{(i-k_2)1}, x_{(i-k_1)1}, x_{i1}$ and $x_{(i-k'_{l-1})2}, \ldots, x_{(i-k'_2)2}, x_{(i-k'_1)2}, x_{i2}$. So the orbits have uniform distribution independent of the U-shape distribution of the extended logistic map. In order to illustrate the algorithm, we have chosen the bifurcation parameter values at $\alpha_1 = -2/2^8$ and $\alpha_2 = 4/2^8$ for the time series $M1$ and $M2$, respectively. These parameter values ensure that system (2) has chaotic behavior in both cases; see Figure 4.

To guarantee that the generator presents good statistical properties, it is necessary to generate time series with uniform distribution and also it is desirable to eliminate the extended logistic map shape in these new time series. This is achieved by means of the number of lags involved.

Table 1: The S-box generated by proposed algorithm.

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

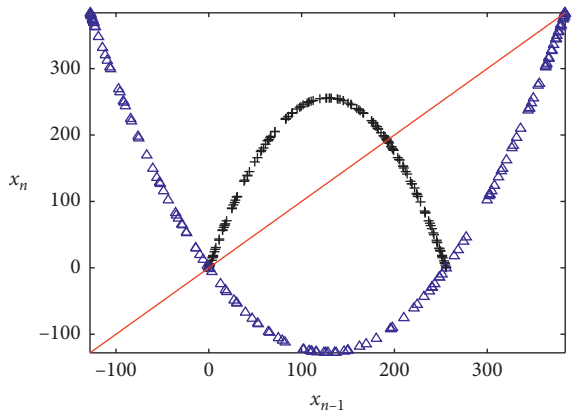

Figure 1: The extended logistic map for different α values: $4/2^8$ (black crosses) and $-2/2^8$ (blue triangles).

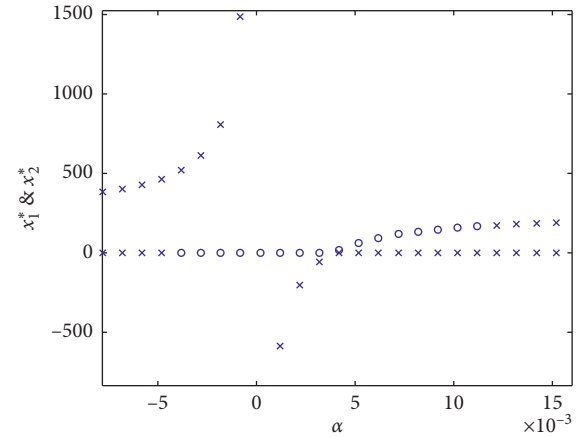

Figure 2: Fixed points stability where an asterisk and a circle denote repulsive and attracting fixed points, respectively. The extended logistic map have both repulsive fixed points at $\alpha \in [-2/2^8 \approx -0.0078, -1/2^8 \approx -0.0039] \cup [3/2^8 \approx 0.0117, 4/2^8 \approx 0.0156]$.

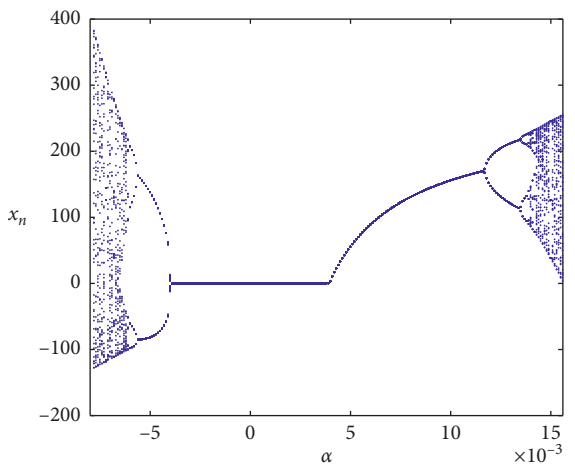

Figure 3: Bifurcation diagram for the extended logistic map given by (2). The bifurcations occur for x_1^* at $\alpha = \pm 1/2^8 \approx \pm 0.0039$, and for x_2^*, the bifurcations occur at $\alpha = 1/2^8 \approx 0.0039$ and at $3/2^8 \approx 0.0117$.

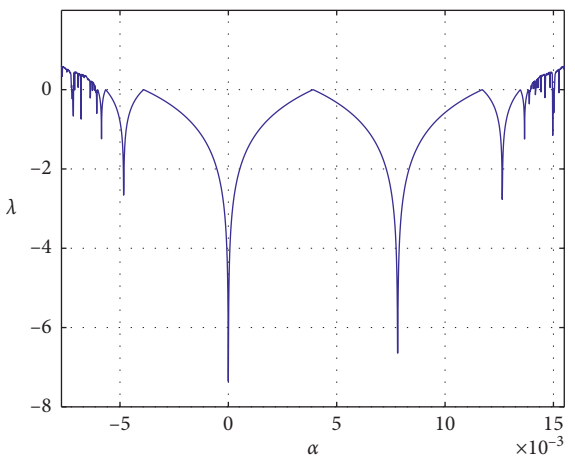

Figure 4: Lyapunov exponent as a function of parameter α.

For example, for the time series $M2 = m_{02}, m_{12}, m_{22}, \ldots$, if we consider two memory units and $\alpha_2 = 4/2^8$, the elements m_{i2} are given as follows:

$$m_{i_2} = M2\left(x_{(i-k1)2}, x_{i_2}\right) = x_{(i-k1)2} + x_{i2}, \bmod 2^8, \tag{4}$$

where $k_1 = 5$. Figure 5 shows the plot of $m_{(n-1)2}$ against m_{n2}, now many points are outside the curve of the extended logistic map. But it is still possible to distinguish the shape of the extended logistic map. The number of iterations considered in the delay does not matter to observe the shape of the extended logistic map and the points never spread enough on the plane $(m_{(n-1)2}, m_{n2})$, so it is necessary to consider more memory units.

Therefore, we consider three memory units, the elements m_{i2} of the time series $M2$ are given as follows:

$$\begin{aligned} m_{i_2} &= M2\left(x_{(i-k2)2}, x_{(i-k1)2}, x_{i2}\right) \\ &= x_{(i-k2)2} + x_{(i-k1)2} + x_{i2}, \bmod 2^8, \end{aligned} \tag{5}$$

where $k1 = 10$ and $k2 = 5$. Now, there are too many points outside the curve of the extended logistic map that look like a cloud of points on the plane $(m_{(n-1)2}, m_{n2})$; see Figure 6. Also the shape of the extended logistic map almost disappears, so three memory units are enough. The problem of considering more memory units has a computational price of information storage. For this reason, two delays, k_1, k_2, and the present state of the time series of the extended logistic map are used. Also the lags must not be contiguous in order to avoid regular patterns which directly affect the data distribution.

As a summary, for two given orbits $x_{01}, x_{11}, x_{21}, \ldots$, and $x_{02}, x_{12}, x_{22}, \ldots$, we consider different delays, $k_2' = k_2 = 10$, $k_1' = 5$ and $k_1 = 6$, to generate both time series $M1$ and $M2$. Then, the series $M1$ is conformed by the sum of two delay states $x_{(i-10)1}$, $x_{(i-5)1}$, and the actual state x_{i1}. In the same way for $M2$, $x_{(i-10)2}$, $x_{(i-6)2}$ and x_{i2}. The values of the time series are limited by the operation mod 2^8, this guarantees that $M1, M2 \in [0, 2^8) \subset \mathbb{R}$. Explicitly $M1\,(x_{(i-101)}, x_{(i-51)}, x_{i1})$ and $M2(x_{(i-10)2}, x_{(i-6)2}, x_{i2})$ are expressed in the following way:

$$\begin{aligned} m_{i_1} &= M1\left(x_{(i-10)1}, x_{(i-5)1}, x_{i1}\right) \\ &= x_{(i-10)1} + x_{(i-5)1} + x_{i1}, \bmod 2^8, \end{aligned} \tag{6}$$

$$\begin{aligned} m_{i_2} &= M2\left(x_{(i-10)2}, x_{(i-6)2}, x_{i2}\right) \\ &= x_{(i-10)2} + x_{(i-6)2} + x_{i2}, \bmod 2^8. \end{aligned} \tag{7}$$

Finally, these time series $M1 = m_{01}, m_{11}, m_{21}, \ldots$ and $M2 = m_{02}, m_{12}, m_{22}, \ldots$ given by (6) and (7), respectively, are mixed and the operation mod 2^8 is applied again, this process generates a new time series Z_i given as follows:

$$Z_i = m_{i_1} + m_{i_2}, \bmod 2^8. \tag{8}$$

Note that $Z_i \in [0, 2^8) \subset \mathbb{R}$. Equations (2), (6)–(8) define a delayed map that hides the structure of the chaotic map used.

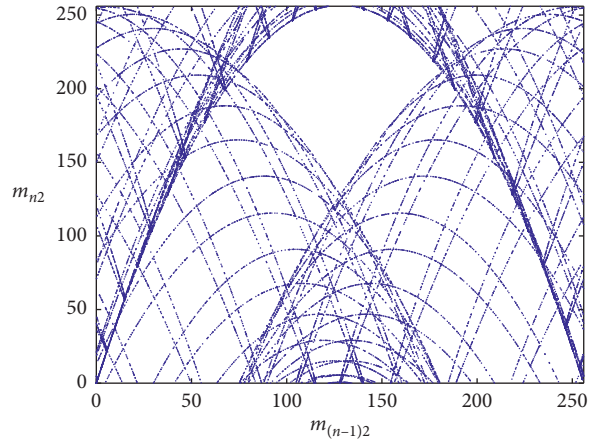

Figure 5: $(m_{(n-1)2}, m_{n2})$ from the time series $M2(x_{(i-k1)2}, x_{i2})$ considering two memory units.

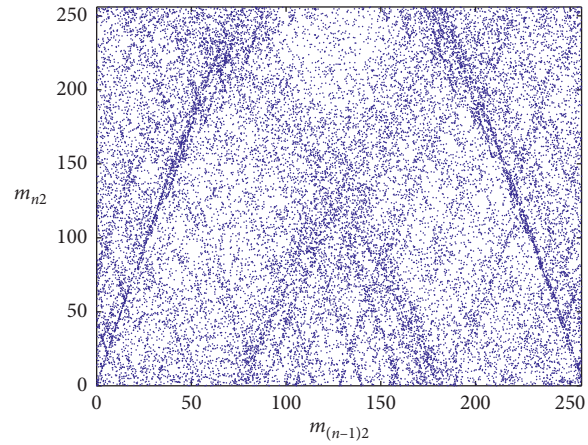

Figure 6: $(m_{(n-1)2}, m_{n2})$ from the time series $M2(x_{(i-k2)2}, x_{(i-k1)2}, x_{i2})$ considering three memory units.

For instance, the plot of the time series x_n onto the plane (x_{n-1}, x_n) reveals the map used; see Figure 7(a), *i.e.*, the extended logistic map is shown. In contrast, the time series z_n does not reveal the extended logistic map onto the plane (z_{n-1}, z_n); see Figure 7(c). In addition, the delayed map has a uniform probability distribution instead of a "U-shaped" probability distribution [25] that the extended logistic map has. Figures 7(b) and 7(d) show the probability distribution of the time series x_n and z_n, respectively. This is an important characteristic that makes easier the construction of S-box since all values have the same probability of occurrence in contrast to the use of a single time series.

Because the elements of the time series Z_i are real numbers, they are discretized to obtain a time series that is useful for cryptosystems. The symbolic dynamics of Z_i time series is given by using the floor function to obtain the series s_i. So the elements of s are integer numbers, $s_i(Z_i) \in \{0, 1, 2, \ldots, 2^8\}$; thus, the process for getting the integer number series is as follows:

$$s_i = \lfloor Z_i \rfloor. \tag{9}$$

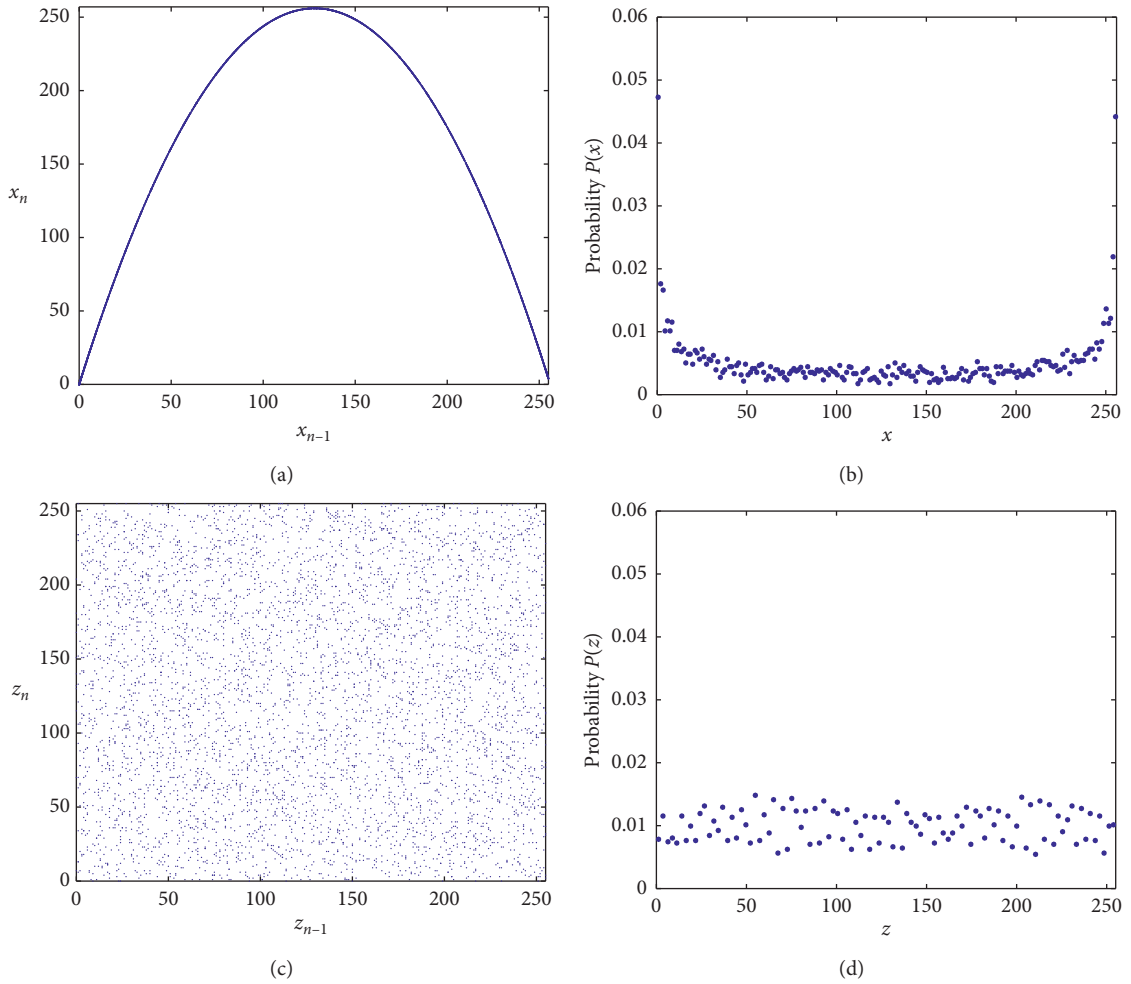

FIGURE 7: (a) The extended logistic map given by x_n against x_{n-1}; (b) "U-shaped" probability distribution of the extended logistic map; (c) Delayed map given by z_n against z_{n-1}; (d) Uniform probability distribution of the delayed map.

A CSPRNG based on a discrete dynamical system is given from equations (2) to (9).

3.2. The Second Algorithm for S-Box Design via Extended Logistic Map. In this subsection, we introduce an algorithm to generate $n \times n$ S-boxes based on the extended logistic map. The algorithm steps are simple as shown below.

Step 1. Select initial conditions x_{01}, x_{02}, and use equations (2) to (9) to generate the stream of byte sequence $s_0, s_1, s_2, \ldots$

Step 2. Discard the repeated elements D's to select 2^n different values. The rule to discard an element is as follows: if $s_i = s_j$ with $i < j$ then discard s_j.

Step 3. Create the S-Box with the 2^n different elements of s_i's.

Once the procedure is finished, the proposed algorithm returns a $n \times n$ S-box with 2^n different values. Note that s_0 is the first element of the S-box, but the second element could be not s_1 if $s_0 = s_1$. However, enough 2^n elements have been generated to build the S-box.

For example, if $n = 8$, $x_{01} = 191$, $x_{02} = 209$, $\alpha_1 = 4/2^8$, and $\alpha_2 = -2/2^8$, then the 8×8 S-Box is obtained and shown in Table 2. This proposed substitution box has the confusion and diffusion properties, which are vital for the block ciphers. In this case, the time needed to calculate an S-box is 0.014 seconds.

4. Criteria for a Good $n \times n$ S-Box

A compilation of six important and well-known criteria reported in the literature to generate cryptographically good S-boxes is presented. These criteria are bijection; nonlinearity, strict avalanche criterion, independence criterion of output bits, XOR distribution of equiprobable input/output, and maximum expected linear probability.

4.1. Bijective Criterion. Let $S(x)$ be an S-box, which is bijective if and only if their Boolean functions f_i satisfy the following condition:

$$wt\left(a_1 \cdot f_1 \oplus a_2 \cdot f_2 \oplus \cdots \oplus a_n \cdot f_n\right) = 2^{n-1}, \tag{10}$$

Table 2: The S-box generated by proposed algorithm.

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

where $a_i \in \mathbb{F}$, $(a_1, a_2, \ldots, a_n) \neq (0, 0, \ldots, 0)$ and $wt(\cdot)$ is the Hamming weight [2, 26], the corresponding S-box is guaranteed to be bijective.

4.2. Nonlinearity Criterion

Definition 1 (see [27]). The nonlinearity of a Boolean function $f: \mathbb{F}^n \longrightarrow \mathbb{F}$ is denoted by

$$N_f = \min_{l \in A_{w,c}(x)} d_H(f, l), \tag{11}$$

where $A_{w,c}(x)$ is an affine function set and $d_H(f, l)$ is the Hamming distance between f and l.

The minimum distance between two Boolean functions can be described by means of the Walsh spectrum [28]:

$$\min_{l \in A_{w,c}(x)} d_H(f, l) = 2^{n-1}\left(1 - 2^{-n} \max_{\omega \in \mathbb{F}^n} \left|\widehat{S}_{(f)}(\omega)\right|\right), \tag{12}$$

where the Walsh spectrum of $f(x)$ is defined as follows:

$$\widehat{S}_{(f)}(\omega) = \left|\widehat{F}_f(\omega)\right| = \sum_{x \in \mathbb{F}^n} (-1)^{f(x) \oplus x \bullet \omega}, \tag{13}$$

with $\omega \in \mathbb{F}^n$, and $x \bullet \omega$ is the dot product between x and ω as

$$x \bullet \omega = x_1 \cdot \omega_1 \oplus \cdots \oplus x_n \cdot \omega_n. \tag{14}$$

4.3. Strict Avalanche Criterion (SAC). This criterion was first introduced by Webster and Tavares [29]. A Boolean function f satisfies SAC if complementing any single input bit changes the output bit with the probability of half. So, a Boolean function f satisfies SAC, if and only if

$$\sum_{x \in \mathbb{F}^n} f(x) \oplus f(x \oplus e_i) = 2^{n-1}, \quad \forall i\colon 1 \leq i \leq n, \tag{15}$$

where $e_i \in \mathbb{F}^n$ such that $wt(e_i) = 1$.

4.4. Output Bits Independence Criterion (BIC). Output Bit Independence Criterion was also introduced by Webster and Tavares [29]. It means that all the avalanche variables should be pairwise independent for a given set of avalanche vectors generated by complementing a single plaintext bit.

Adam and Tavares introduced another method to measure the BIC, for Boolean functions, f_i and f_j $(i \neq j)$ of two output bits in a S-box, if $f_i \oplus f_j$ is highly nonlinear and comes as close as possible to satisfy SAC [2]. Additionally, $f_i \oplus f_j$ can be tested with a Dynamic Distance (DD). The DD of a function f can be defined as

$$\mathrm{DD}(f) = \max_{\substack{d \in \mathbb{F}^n \\ wt(d)=1}} \frac{1}{2}\left|2^{n-1} - \sum_{x=0}^{2^n-1} f(x) \oplus f(x \oplus d)\right|. \tag{16}$$

If the value of DD is a small integer and close to zero, the function f satisfies the SAC.

4.5. Criterion of Equiprobable Input/Output XOR Distribution. Biham and Shamir [30] introduced differential cryptanalysis which attacks S-boxes faster than brute-force attack. It is desirable for an S-box to have differential uniformity. This can be measured by the maximum expected differential probability (MEDP). Differential probability for a given map S can be calculated by measuring differential resistance and is defined as follows:

$$\mathrm{DP}_f = \max_{\Delta x \neq 0, \Delta y}\left(\frac{\#\{x \in \mathbb{F}^n | S(x) \oplus S(x \oplus \Delta x) = \Delta y\}}{2^m}\right), \tag{17}$$

where 2^n is the cardinality of all the possible input values (x), Δx and Δy are called input and output differences, respectively, for the S. Thus, the smaller value of DP_f gives better cryptographic property, *i.e.*, better resistance to differential cryptanalysis.

4.6. Maximum Expected Linear Probability. The Maximum Expected Linear Probability (MELP) is the maximum value of the unbalance of an event. Two randomly selected masks a and b are given, where a is used to calculate the mask of all possible values of an input x, and b is used to calculate the mask of the output values of the corresponding S-box. The parity of the input bits mask a is equal to the parity of the

output bits the mask b. MELP of a given S-box can be computed by the following equation:

$$\mathrm{LP}_f = \max_{a,b\in\mathbb{F}^n\setminus\{0\}} \left(2^{-n} \sum_{x\in\mathbb{F}^n} (-1)^{a\cdot x+b\cdot f(x)} \right)^2. \quad (18)$$

The closer the MELP is to zero, the greater the resistance against linear cryptanalysis.

5. Performance Analysis of the Generated S-Box

In this section, we analyze the proposed S-boxes based on the second algorithm with the six important cryptographic criteria. Furthermore, our results are compared with results reported in the literature using other approaches.

5.1. Bijective Criterion. The computed value of proposed S-box is the desired value of $2^{n-1} = 128$, with $n = 8$, according to formula (10), the bijective criterion is satisfied. So, the proposed S-box is one-to-one, surjective, and balanced.

5.2. Nonlinearity Criterion. Nonlinearity criterion ensures that an S-box is not a linear function between input vectors and output vectors. The nonlinearity gives the degree of dissimilarity between the Boolean function f and n-bit linear function l. If the function has high minimum Hamming distance is said to have high nonlinearity, *i.e.*, by reducing the Walsh spectrum in (12). An S-box contains n Boolean functions and the nonlinearity of each Boolean function must be calculated. The nonlinearities of the proposed S-box are 96, 104, 106, 102, 104, 102, 108, and 96. High nonlinearity ensures the strongest ability to resist powerful modern attacks such as linear cryptanalysis.

5.3. Strict Avalanche Criterion (SAC). The avalanche effect indicates the randomness of an S-box when an input has a change. The generated S-box is given in Table 3. We obtain a maximum SAC equal to 0.6094, the minimum is 0.4219, and its average value 0.5059 is close to the desired value 0.5. With these results, we conclude that the S-box generated by the proposed method fulfills the property of SAC.

5.4. Output Bits Independence Criterion (BIC). The BIC criterion guarantees that there is no statistic pattern or dependency between output vectors. The obtained BIC results are shown in Tables 4, 5, and 6.

The mean value of BIC-nonlinearity is 103.50, the mean value of BIC-SAC is 0.5050, and maximum value of DD is 12 which indicates that S-box satisfies the BIC criterion.

5.5. Criterion of Equiprobable Input/Output XOR Distribution. The equiprobable input/output XOR Distribution analyzes the effect in particular differences of input pairs of the resultant output pairs to discover the key bits. The idea is to find the high probability difference pairs for an S-Box under attack. In Table 7, the maximal value of the

Table 3: SAC criterion result of the generated S-box.

0.4688	0.4688	0.4531	0.5313	0.4844	0.5156	0.4531	0.5469
0.4844	0.4688	0.5781	0.5938	0.5469	0.4688	0.5313	0.5469
0.5156	0.5156	0.5156	0.5000	0.5938	0.5469	0.5469	0.4688
0.4844	0.5625	0.5000	0.5000	0.4375	0.5000	0.5625	0.4688
0.5625	0.4844	0.5313	0.5313	0.4844	0.5156	0.4844	0.4375
0.5000	0.4844	0.5781	0.5313	0.4688	0.4688	0.5000	0.4531
0.5156	0.4688	0.5000	0.4531	0.5781	0.4688	0.5156	0.4219
0.4844	0.4531	0.6094	0.5625	0.5313	0.4531	0.4375	0.5469

Table 4: BIC-nonlinearity criterion result of the generated S-box.

0	106	106	108	106	104	102	102
106	0	102	104	100	106	106	106
106	102	0	100	106	102	104	102
108	104	100	0	104	104	104	100
106	100	106	104	0	104	104	96
104	106	102	104	104	0	104	102
102	106	104	104	104	104	0	104
102	106	102	100	96	102	104	0

Table 5: BIC-SAC criterion result of the generated S-box.

0	0.4785	0.5176	0.5098	0.5039	0.5195	0.5195	0.4707
0.4785	0	0.5000	0.4922	0.5195	0.5137	0.4941	0.5000
0.5176	0.5000	0	0.5469	0.5137	0.5020	0.4863	0.4980
0.5098	0.4922	0.5469	0	0.5098	0.5176	0.5137	0.4922
0.5039	0.5195	0.5137	0.5098	0	0.5176	0.4863	0.5156
0.5195	0.5137	0.5020	0.5176	0.5176	0	0.4785	0.4863
0.5195	0.4941	0.4863	0.5137	0.4863	0.4785	0	0.5371
0.4707	0.5000	0.4980	0.4922	0.5156	0.4863	0.5371	0

Table 6: The DD of the generated S-box (BIC-SAC criterion).

0	0	4	4	8	12	2	2
0	0	4	4	4	2	6	2
4	4	0	6	2	4	2	0
4	4	6	0	4	10	4	8
8	4	2	4	0	4	4	12
12	2	4	10	4	0	8	6
2	6	2	4	4	8	0	2
2	2	0	8	12	6	2	0

Table 7: Equiprobable input/output XOR distribution approach table for the generated S-box.

6	8	6	8	6	6	6	10	8	10	6	6	6	8	6	6
8	6	6	6	6	8	8	8	8	6	6	4	12	6	8	6
8	4	8	10	8	6	8	6	6	8	6	10	6	6	8	8
6	6	6	6	8	6	6	4	8	6	6	6	6	8	8	6
6	6	6	6	6	6	8	6	6	8	8	6	6	6	8	6
6	6	6	6	6	6	6	6	8	8	6	6	6	4	6	6
6	8	8	6	6	6	8	6	8	8	6	6	6	6	8	6
8	6	6	6	6	8	8	6	6	6	6	8	8	6	6	6
6	6	6	6	8	8	6	8	8	6	6	6	6	8	6	6
8	8	6	8	6	8	6	6	6	6	6	8	6	6	8	6
6	6	6	6	8	6	6	6	6	10	6	6	6	6	6	6
6	6	6	6	8	6	8	6	6	8	6	6	6	6	6	6
8	4	8	8	6	6	6	6	8	6	8	6	6	6	8	6
8	6	6	6	8	8	6	8	8	6	8	6	8	6	6	6
6	8	6	6	6	8	8	8	8	6	6	6	6	6	6	6
6	8	8	6	6	6	6	8	6	6	8	6	8	6	6	—

Table 8: Comparison of our S-boxes and others S-boxes used in typical block ciphers.

	Bijection	Nonlinearity			SAC			BIC			I/O XOR	MELP
		Min	Max	Avg	Min	Max	Avg	SAC	Nonlinearity	DD		
Gray [31]	128	112	112	112	0.4375	0.5625	0.4998	0.5026	112	112	0.0156	0.0156
AES [32]	128	112	112	112	0.4531	0.5625	0.5049	0.5046	112	112	0.0156	0.0156
Skipjack [33]	123	100	108	105.12	0.3906	0.5938	0.5027	0.5003	104.03	109	0.0469	0.0549
APA [34]	128	112	112	112	0.4375	0.5625	0.5007	0.4997	112	112	0.0156	0.0156
Ref. [3]	128	102	108	105.25	0.4375	0.5781	0.5056	0.5019	103.78	108	0.0391	0.0977
Ref. [4]	128	104	110	106.25	0.4219	0.5938	0.5039	0.5059	103.35	108	0.0391	0.0791
Ref. [6]	128	90	108	103	0.3438	0.6094	0.4851	0.5018	103.78	108	0.0469	0.0665
Ref. [8]	128	98	107	103.25	0.3828	0.5938	0.5059	0.5033	104.21	108	0.0469	0.0665
Ref. [9]	128	96	106	102.50	0.3906	0.6719	0.5178	0.4790	102.64	106	0.2109	0.1077
Ref. [10]	128	106	108	106.75	0.4219	0.6250	0.5034	0.5015	103.78	108	0.0391	0.0706
Ref. [11]	128	104	108	105.75	0.4219	0.5938	0.4976	0.5013	104.50	108	0.0391	0.0625
Ref. [14]	128	106	108	107.25	0.4219	0.6094	0.5034	0.4980	105.28	108	0.0469	0.0706
Ref. [15]	128	106	108	106.75	0.3594	0.5781	0.4917	0.4998	104.14	108	0.0391	0.0706
Ref. [35]	128	112	112	112	0.4219	0.5469	0.5115	0.4982	108.71	112	0.0313	0.0479
Ref. [36]	128	102	108	106	0.4219	0.5938	0.5002	0.5016	104.42	108	0.0391	0.0881
Ref. [37]	128	106	108	106.75	0.4063	0.5938	0.4971	0.5008	102.92	106	0.0391	0.0791
Ref. [38]	129	103	109	104.87	0.3984	0.5703	0.4966	0.5044	102.96	109	0.0391	0.0706
Ref. [39]	128	96	106	103	0.3906	0.6250	0.5039	0.5010	100.35	106	0.5000	0.0881
Ref. [40]	128	110	112	110.50	0.4375	0.5625	0.4937	0.5033	103.85	106	0.0391	0.0625
Ref. [41]	128	106	108	106.75	0.4219	0.6250	0.5034	0.5015	103.78	108	0.0391	0.0706
Ref. [42]	128	106	108	106.75	0.4219	0.5781	0.5010	0.5005	104.07	108	0.0391	0.0706
Proposal 1	128	96	104	101.75	0.3906	0.5781	0.5012	0.5066	103.42	108	0.0391	0.0706
Proposal 2	128	96	108	102.25	0.4219	0.6094	0.5059	0.5050	103.50	108	0.0469	0.0625

generated S-box is 12, which indicates that the S-box is resistant to differential cryptanalysis.

5.6. MELP Criterion. This criterion is computed according to equation (18) and the average value for the proposed S-Box is 0.0625, which indicates resistance against linear cryptanalysis.

5.7. Performance Comparation. A performance comparison of our S-box and others' good S-boxes that were reported in the literature is presented in Table 8. The proposed S-box fulfills the most important condition and bijection and accomplishes good results to the rest of the test values expected [3, 4, 6, 8–11, 14, 31–42].

Comparing these S-boxes, we find that

(i) The proposal fulfills the expected value 128.

(ii) The nonlinearity is 102.25, and our proposal has similar value or above that approaches reported in references [6, 9, 39].

(iii) The mean value of SAC of our proposed algorithm is 0.5059. This value is close to the ideal value, 0.5.

(iv) The BIC-nonlinearity average is 103.42, and the BIC-SAC average is 0.5066. Table 8 shows that all the S-boxes have a good BIC property.

(v) The value of the XOR distribution of equiprobable input/output is 0.0469 which indicates resistance against to differential attack.

(vi) The maximum MELP is 0.0625 which is a good value in comparison with that the S-boxes reported in the literature.

The computational time depends on the algorithm and the computer used to simulate the algorithm. For example, algorithms based on chaos need more steps for generating one iteration if the chaotic systems are based on ordinary differential equations (ODEs) instead of mappings. References [4, 6, 15, 36, 39] are based on ODEs which imply more computational operations to solve the systems than references [2, 3, 9, 10, 35, 38, 40] based on iterated maps. It would be unfair to compare the computational time between these algorithms that all of them generate good S-boxes.

Our two proposals are based on iterated logistic maps, the first proposal maps the interval [0, 1] and second proposal maps $[0, 2^8]$. One of the characteristics of the second proposal (extended logistic map) is the use of fewer iterations to generate S-boxes than the first proposal (logistic map). Fewer iterations are possible because the extended logistic map employs the interval $[0, 2^8]$ instead of the logistic map [0, 1]. The use of the interval $[0, 2^8]$ allows to generate S-boxes with less iteration that the original logistic map. To scale the interval allows us to glimpse that the algorithms reported in the literature can be scaled to operate in the interval $[0, 2^8]$ to generate less iterations to build good S-boxes.

6. Dynamical Generation of S-Boxes and its Application

The Alberti cipher was one of the first polyalphabetic ciphers where the principle is substitution, using multiple substitution alphabets such that the output has a uniform distribution. Nowadays, the Alberti cipher is considering a codification instead of a cipher. Taking this idea of

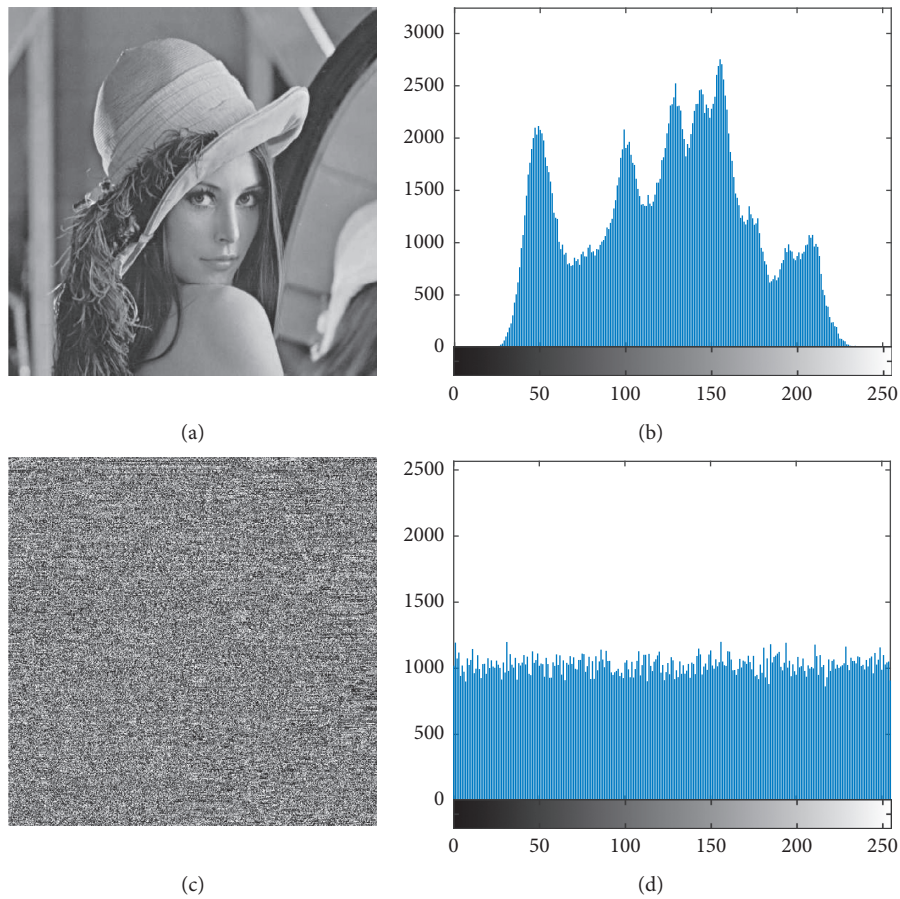

Figure 8: (a) The plain image of Lenna; (b) the grayscale histogram of plain Lenna image; (c) the encoded image of Lenna; (d) the grayscale histogram of encoded image.

"polyalphabetic ciphers," we present an application of dynamical S-boxes, where a particular intensity of a pixel is substituted by different intensities in the same round. Usually, an S-box is used to substitute all the pixels of an image of size $p \times q$ in the same way. The idea of "polyalphabetic ciphers" is to use a dynamical S-box to achieve uniform distribution in the encoded image.

Our encoded approach, to get a uniform distribution, is given by applying dynamical S-box which changes in each pixel row. The codification input is the grayscale Lenna image (Figure 8(a)), and the process is to substitute pixels of a row according to an S-box, but different rows use different S-boxes. The codification output is shown in Figure 8(c), and Figures 8(b) and 8(d) show the image histograms of plain image of Lenna and encoded Lenna image, respectively.

Furthermore, in cryptography, a uniform distribution is always desired; since this property was achieved by simple substitution with the S-boxes, a good result is expected for a full cryptographic algorithm based on these S-boxes.

It is well known that many ciphers have been successfully analyzed with the help of statistical analysis and several statistical attacks have been devised on them. Therefore, an effective cipher should be robust against any statistical attack, for instance, the information entropy, the correlation of two adjacent pixels, Peak Signal to Noise Ratio (PSNR), Unified Average Changing Intensity (UACI), and others, which it is not the purpose of this article.

It is important to point out that this substitution is a simple and useful approach intended to catch a glimpse of possible applications of dynamical S-boxes presented in this assignment.

7. Concluding Remarks

In this work, simple algorithms to design $n \times n$-bits substitution boxes are presented. The algorithms are based on two lag time chaotic series of the logistic map and the extended logistic map. In both approaches, two lag time series are generated by considering different carrying capacity parameter values. The mixing of these lag time series favors two things: a uniform distribution, and the concealment of the chaotic map used.

Two proposals were presented, the former generates bits and the latter generates bytes. The generation of bytes instead of bits helps to generate S-boxes with less iterations. Although, for this work, we use the extended logistic map, it is possible to employ the logistic map or different chaotic maps to generate bytes.

To evaluate the performance of the proposed S-boxes, several statistical tests were carried out. The numerical

analysis results for both proposed algorithms show that all good S-box criteria were fulfilled with high immunity to resist differential cryptanalysis and linear cryptanalysis. The number of operations is considerably reduced when we generate bytes instead of bits. The Lyapunov exponent, the bifurcations, and the local stability of logistic map are preserved in the extended logistic map. We contrast the performance of our S-boxes with other S-boxes reported in the literature and our results are in the average of good S-boxes. Finally, a simple and useful S-box application approach for coding that provides a uniform distribution was presented.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

B.B.C.Q. received a CONACYT scholarship under registration 262247 through the project supported by the Sectoral Fund for Research for Education, CONACYT-SEP, under grant number A1-S-30433. E.C.C. acknowledges CONACYT for the financial support through Project No. A1-S-30433.

References

[1] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[2] C. Adams and S. Tavares, "The structured design of cryptographically good S-boxes," *Journal of Cryptology*, vol. 3, no. 1, pp. 27–41, 1990.

[3] A. Belazi, M. Khan, A. A. A. El-Latif, and S. Belghith, "Efficient cryptosystem approaches: S-boxes and permutation-substitution-based encryption," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 337–361, 2017.

[4] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, "A novel approach for strong S-box generation algorithm design based on chaotic scaled Zhongtang system," *Nonlinear Dynamics*, vol. 87, no. 2, pp. 1081–1094, 2017.

[5] G. Chen, "A novel heuristic method for obtaining S-boxes," *Chaos, Solitons & Fractals*, vol. 36, no. 4, pp. 1028–1036, 2008.

[6] A. K. Farhan, R. S. Ali, H. Natiq, and N. M. G. Al-Saidi, "A new S-box generation algorithm based on multistability behavior of a plasma perturbation model," *IEEE Access*, vol. 7, pp. 124914–124924, 2019.

[7] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, "A novel design of chaos based S-boxes using genetic algorithm techniques," in *Proceedings of the 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pp. 678–684, Doha, Qatar, November 2014.

[8] G. Jakimoski and L. Kocarev, "Chaos and cryptography: block encryption ciphers based on chaotic maps," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 2, pp. 163–169, 2001.

[9] M. Khan and Z. Asghar, "A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and s8 permutation," *Neural Computing and Applications*, vol. 29, no. 4, pp. 993–999, 2018.

[10] D. Lambić, "A novel method of S-box design based on chaotic map and composition method," *Chaos, Solitons & Fractals*, vol. 58, pp. 16–21, 2014.

[11] G. Liu, W. Yang, W. Liu, and Y. Dai, "Designing S-boxes based on 3-d four-wing autonomous chaotic system," *Nonlinear Dynamics*, vol. 82, no. 4, pp. 1867–1877, 2015.

[12] F. Özkaynak and A. B. Özer, "A method for designing strong S-boxes based on chaotic Lorenz system," *Physics Letters A*, vol. 374, no. 36, pp. 3733–3738, 2010.

[13] Y. Tian and Z. Lu, "S-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm," *Journal of Systems Engineering and Electronics*, vol. 27, no. 1, pp. 232–241, 2016.

[14] A. Ullah, S. S. Jamal, and T. Shah, "A novel scheme for image encryption using substitution box and chaotic system," *Nonlinear Dynamics*, vol. 91, no. 1, pp. 359–370, 2018.

[15] X. Wang, Ü. Çngsref, S. Kacar et al., "S-box based image encryption application using a chaotic system without equilibrium," *Applied Sciences*, vol. 9, no. 4, p. 781, 2019.

[16] Y. Wang, K.-W. Wong, X. Liao, and T. Xiang, "A block cipher with dynamic S-boxes based on tent map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 3089–3099, 2009.

[17] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[18] F. Özkaynak and S. Yavuz, "Designing chaotic S-boxes based on time-delay chaotic system," *Nonlinear Dynamics*, vol. 74, no. 3, pp. 551–557, 2013.

[19] Y. Zhou, L. Bao, and C. L. P. Chen, "Image encryption using a new parametric switching chaotic system," *Signal Processing*, vol. 93, no. 11, pp. 3039–3052, 2013.

[20] M. García-Martínez and E. Campos-Cantón, "Pseudo-random bit generator based on lag time series," *International Journal of Modern Physics C*, vol. 25, no. 4, Article ID 1350105, 2014.

[21] D. S. Dendrinos and M. Sonis, "Socio-spatial stocks and antistocks; the logistic map in real space," *The Annals of Regional Science*, vol. 27, no. 4, pp. 297–313, 1993.

[22] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, Westview Press, Boulder, CO, USA, 2003.

[23] C. Li and G. Chen, "Estimating the lyapunov exponents of discrete systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 14, no. 2, pp. 343–346, 2004.

[24] C. Yang, C. Q. Wu, and P. Zhang, "Estimation of lyapunov exponents from a time series for n-dimensional state space using nonlinear mapping," *Nonlinear Dynamics*, vol. 69, no. 4, pp. 1493–1507, 2012.

[25] J. Urías, E. Campos, and N. F. Rulkov, *Random Finite Approximations of Chaotic Maps*, Springer, New York, NY, USA, 2006.

[26] C. Adams and S. Tavares, "Good S-boxes are easy to find," in *Advances in Cryptology—CRYPTO' 89 Proceedings*, G. Brassard, Ed., Springer, New York, NY, USA, 1990.

[27] Y. Tian and Z. Lu, "Chaotic S-box: intertwining logistic map and bacterial foraging optimization," *Mathematical Problems in Engineering*, vol. 2017, Article ID 6969312, 11 pages, 2017.

[28] W. Millan, "How to improve the nonlinearity of bijective S-boxes," in *Information Security and Privacy*, C. Boyd and E. Dawson, Eds., Springer, Berlin, Heidelberg, Germany, 1998.

[29] A. F. Webster and S. E. Tavares, "On the design of S-boxes," in *Advances in Cryptology-CRYPTO'85 Proceedings*, H. C. Williams, Ed., Springer, Berlin, Heidelberg, Germany, 1986.
[30] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.
[31] M. T. Tran, D. K. Bui, and A. D. Duong, "Gray S-box for advanced encryption standard," in *Proceedings of the 2008 International Conference on Computational Intelligence and Security*, vol. 1, IEEE, Suzhou, China, pp. 253–258, December 2008.
[32] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard, Springer Science and Business Media*, Berlin, Germany, 2002.
[33] I. Hussain, T. Shah, M. A. Gondal, and Y. Wang, "Analyses of SKIPJACK S-box," *World Applied Sciences Journal*, vol. 13, no. 11, pp. 2385–2388, 2011.
[34] L. Cui and Y. Cao, "A new S-box structure named affine-power-affine," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 3, pp. 751–759, 2007.
[35] A. Belazi, A. A. Abd El-Latif, A.-V. Diaconu, R. Rhouma, and S. Belghith, "Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms," *Optics and Lasers in Engineering*, vol. 88, pp. 37–50, 2017.
[36] F. Islam and G. Liu, "Designing s-box based on 4D-4wing hyperchaotic system," *3D Research*, vol. 8, no. 1, pp. 1–9, 2017.
[37] F. Özkaynak, "Construction of robust substitution boxes based on chaotic systems," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3317–3326, 2019.
[38] G. Tang and X. Liao, "A method for designing dynamical S-boxes based on discretized chaotic map," *Chaos, Solitons & Fractals*, vol. 23, no. 5, pp. 1901–1909, 2005.
[39] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, and I. Hussain, "A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems," *Nonlinear Dynamics*, vol. 70, no. 3, pp. 2303–2311, 2012.
[40] Y. Wang, P. Lei, and K.-W. Wong, "A method for constructing bijective S-box with high nonlinearity based on chaos and optimization," *International Journal of Bifurcation and Chaos*, vol. 25, no. 10, Article ID 1550127, 2015.
[41] D. Lambić, "A novel method of S-box design based on discrete chaotic map," *Nonlinear Dynamics*, vol. 87, no. 4, pp. 2407–2413, 2017.
[42] D. Lambić, "A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design," *Nonlinear Dynamics*, pp. 1–13, 2020.