








Article

Chaos-Based S-Boxes as a Source of Confusion in Cryptographic Primitives

Élvio Carlos Dutra e Silva Junior ^{1,2,*}, Carlos Augusto de Moraes Cruz ², Isaías Abner Lima Saraiva ^{1,2}, Fávero Guilherme Santos ¹, Carlos Raimundo Pereira dos Santos Junior ^{1,2}, Leandro Soares Indrusiak ³, Weiler Alves Finamore ⁴ and Manfred Glesner ⁵

¹ SENAI Institute of Innovation in Microelectronics (ISI-ME), National Service for Industrial Apprenticeship (SENAI/AM), Av. Gal. Rodrigo Otávio Jordão Ramos, nr. 2394, Industrial District, Manaus 69075-005, Brazil; isaias.saraiva@am.senai.br (I.A.L.S.); favero.santos@am.senai.br (F.G.S.); carlos.pereira@am.senai.br (C.R.P.d.S.J.)

² Center of R&D in Electronic Technology and Information (CETELI), Federal University of Amazonas (UFAM), Av. Gal. Rodrigo Otávio Jordão Ramos, nr. 3000, Industrial District, Manaus 69077-000, Brazil; carlosamcruz@ufam.edu.br

³ Distributed Systems and Services Group, School of Computer Science, University of Leeds (UoL), Leeds LS2 9JT, UK; l.soaresindrusiak@leeds.ac.uk

⁴ Department of Electronic and Computer Engineering, Polytechnic School, Federal University of Rio de Janeiro (UFRJ), University City, Fundão Island, Rio de Janeiro 21941-914, Brazil; finamore@ieee.org

⁵ Microelectronic Systems (MES) Research Group, Department of Electrical Engineering and Information Technology (etit), Technische Universität Darmstadt, St. Merckstrasse, nr. 25, Room S3 | 06 343, 64283 Darmstadt, Germany; glesner@mes.tu-darmstadt.de

* Correspondence: elvio.dutra@am.senai.br or elvio.dutra@gmail.com

Abstract: In recent years, many chaos-based encryption algorithms have been proposed. Many of these are based on established designs and populate their S-boxes with values derived from chaotic maps, following conventional implementation strategies to enable comparison with their original non-chaotic counterparts. In contrast, this work proposes a novel approach: a Chaos-Based Substitution Box (CB-SBox) implementation, in which conventional ROM-based S-boxes are replaced by a digital circuit that directly executes a selected chaotic map. This method enables the construction of S-boxes with long word lengths through an FPGA-based programmable circuit that allows for variable S-box lengths, facilitating the analysis of S-boxes of varying sizes, and ultimately enhancing security, particularly for larger S-boxes, as demonstrated by increased resistance to linear and differential cryptanalysis. Furthermore, the proposed CB-SBox achieves reductions in both area and power consumption compared to size-comparable ROM-based S-boxes. A 19-bit chaos-based S-box consumes just 0.0238% of the area and 0.0241% of the power required by an equivalent ROM-implemented S-box while providing the same level of security. The inherent unpredictability of non-linear chaotic behavior causes the proposed chaos-based S-boxes to exhibit non-bijective characteristics, making them well suited for application in non-invertible cryptographic primitives, such as hash functions and Feistel networks. The proposed CB-SBox is implemented in a Feistel network as described in the literature, and the results are provided.

Keywords: substitution boxes (s-boxes); chaos; chaotic map; encryption; cryptographic primitives; Feistel networks; linear and differential cryptanalysis; field-programmable gate array (FPGA); FPGA-based programmable circuit; application-specific integrated circuit (ASIC)



Academic Editor: Alexander Barkalov

Received: 20 March 2025

Revised: 27 April 2025

Accepted: 30 April 2025

Published: 28 May 2025

Citation: Dutra e Silva Junior, É.C.; Cruz, C.A.d.M.; Saraiva, I.A.L.; Santos, F.G.; dos Santos Junior, C.R.P.; Indrusiak, L.S.; Finamore, W.A.; Glesner, M. Chaos-Based S-Boxes as a Source of Confusion in Cryptographic Primitives. *Electronics* **2025**, *14*, 2198. <https://doi.org/10.3390/electronics14112198>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since Lorenz's 1963 seminal article [1], more than forty non-linear differential equations presenting chaotic behavior have been reported [2–7]. They have been applied to many disciplines, such as mechanics, biology, ecology, astronomy, and cryptography [8–47].

Chaos and encryption are intimately tied: confusion and diffusion, two fundamental concepts of cryptography [48], are strongly related to the chaotic properties of ergodicity and sensitivity to initial conditions [49]. The sensitivity to initial conditions ensures that even minimal changes in input, such as slight variations in the secret key, produce significantly different outputs, enhancing key sensitivity and resistance to differential attacks. The property of ergodicity allows chaotic maps to eventually cover the entire state space, contributing to the statistical uniformity of encrypted outputs and increasing resilience against cryptanalysis. Although deterministic, chaotic maps exhibit highly complex dynamics, resulting in outputs that appear random and providing a desirable balance between reproducibility and unpredictability. This complexity reinforces confusion, while the topological mixing characteristic of chaos facilitates effective diffusion, ensuring that small changes in plaintext or key propagate throughout the ciphertext. These features make chaos an attractive and powerful foundation for the design of robust cryptographic systems.

This relation led many researchers to develop chaos-based encryption algorithms: Shujun Li et al. [8] mixed stream and block cipher techniques in a real-time scheme that applied a cascade of chaotic maps. Tenny et al. [9] presented a public key encryption scheme based on additive chaos. Bose [10] created a novel compression and encryption scheme based on coupled chaos. Zou [11] proposed a new method to strengthen the non-linear kinetic complexity and ergodicity of the Lorenz system, generating output sequences applicable to chaotic image encryption with good security and a high capacity to resist common attacks. Khan [12] integrated Gold sequences with chaotic maps in an SP-network, achieving high-dimensional cryptographic robustness. Feng et al. [13] proposed, in 2021, an image transmission scheme based on two chaotic maps, divided into two parts: secure image transmission between sensor nodes and sink nodes, and secure image transmission between sensor nodes and receivers. In 2023, they introduced a new fractional-order 3D Lorenz chaotic map and a hyperchaotic map [14], both applied to a multi-image encryption algorithm characterized by excellent practicability, high security, and efficiency. In 2024, they constructed a robust hyperchaotic map [15] and further developed an efficient image encryption algorithm based on this map and a pixel fusion strategy. Later that same year, they proposed a novel multi-channel image encryption algorithm [16] that leveraged pixel reorganization and two hyperchaotic maps to jointly generate chaotic sequences with a larger key space and improved randomness. Ma et al. [17] proposed a digital image encryption scheme based on the Tabu Search algorithm and Chen's hyperchaotic map, in which the encryption key and the initial values of the hyperchaotic map were derived from the plain image. Qian et al. [18] presented the design of a novel 2D polynomial hyperchaotic map, constructed by leveraging the cross-coupling of two titanium dioxide (TiO_2) memristors (non-linear circuit elements capable of retaining memory of past electrical states), and subsequently applied it to a multi-channel image encryption algorithm, demonstrating high security and notable efficiency. Yu et al. [19] proposed a new fractional-order memristive Hopfield neural network that exhibited rich dynamic behaviors, including transient chaos, which was implemented on a Field-Programmable Gate Array (FPGA), providing a theoretical basis for its application in encryption.

Many of these developed chaos-based encryption algorithms use chaos for constructing or populating tables and S-boxes: for Baptista's non-bijective algorithm [20], a table was built, where each plaintext symbol was associated with the number of iterations performed on a chaotic logistic map to move the cipher key-dependent chaotic attractor's

initial state to a final state corresponding to the plaintext symbol. Wong et al. [21] presented embedding compression in the Baptista-type scheme [20], which was further improved by Chen et al. [22] through dynamic look-up table updating. The ciphertext in the scheme by Alvarez et al. [23] is a triplet composed of x_0 , a given threshold U , and a parameter B used by the transmitter to locate the plaintext on a binary chain C constructed according to the threshold U . Wong [24] improved Baptista's cipher [20] by dynamically updating the look-up table that assigned plaintext to the corresponding attractor regions. Wong's cipher was further improved [25–28] by methods that used control parameters and hashing schemes. Zhang [29] proposed a dynamic S-box generation algorithm using a hyperchaotic map and quantum random walks, evaluating its security against cryptographic attacks. Jakimoski et al. [30] proposed a procedure for designing chaos-based block ciphers built on a Feistel network with 8-bit bijective S-boxes, which were precomputed prior to the encryption process using a four-step algorithm based on the N_{th} iteration of the logistic map. Peng [31] introduced a method for dynamically updating key-dependent S-boxes using a four-dimensional hyperchaotic Lorenz map to enhance the resistance of the encryption process. Guesmi [32] designed chaos-based S-boxes optimized with genetic algorithms, selecting the strongest ones through cryptographic criteria. Tang [33] proposed a dynamic S-box creation method based on discretized chaotic maps, improving resistance against cryptanalysis. Ibrahim [34] presented a 12-bit chaotic encryption scheme with a dynamic S-box optimized for grayscale medical images. Al-Maadeed [35] developed an image encryption algorithm integrating chaotic Lorenz sequences and algebraic S-boxes, ensuring robust diffusion and substitution. Ahmad [36] introduced an improved chaotic map for secure S-box generation, validated through cryptographic performance metrics. Nazir [37] exploited the 4D-hyperchaotic map to create three S-boxes (red, green, and blue) and used a logistic map to transform a plain image into DNA strands for enhanced color image encryption, demonstrating strong security properties. Manzoor [38] formulated a new chaotic S-box for block ciphers, leveraging a hybrid logistic–sine map with high Lyapunov exponents. Alabdullah [39] constructed an S-box based on Delannoy numbers and a five-dimensional hyperchaotic map, ensuring strong cryptographic characteristics. Goswami [40] implemented an FPGA-optimized SNOW 3G stream cipher with logic-gate-based S-boxes, improving efficiency. Lidong [41] developed a multi-image encryption scheme integrating chaotic maps, S-boxes, and image compression techniques. Jun [42] introduced a dynamic encryption step and extended the S-box size for improved image security. Ibrahim [43] proposed a generic medical image encryption framework using dynamic S-boxes and chaotic maps, validated through extensive security analysis. Zhang and Pasalic [44] proposed two methods for constructing highly non-linear and asymmetric S-boxes that could not be bijective. Piret et al. [45] proposed a new block cipher, PICARO, which was resistant to side-channel attacks and based on a Feistel network due to the use of a non-bijective S-box. Jassim and Farhan [46] proposed a new method for generating an 8-bit S-box using three integrated chaotic maps and the Flower Pollination Algorithm, in which bijectivity was guaranteed offline by randomly replacing repeated chaotic outputs. Zhu et al. [47] proposed an efficient and simple S-box generation method using a new compound chaotic map, the Sine–Tent one, and then introduced a new image encryption scheme based on double S-boxes.

In parallel, some studies investigated the conditions for applying chaos theory in encryption. The behavior of chaotic maps is highly sensitive to variations in initial conditions (IC), and in some cases, chaotic behavior may even vanish for specific IC values, causing the map to either converge to a fixed point or diverge away from the attractor. Dutra et al. [50] characterized the region of ICs responsible for driving the elementary chaotic map proposed by Linz and Sprott [7] into chaotic behavior and associated this

region with the key space calculation of chaotic ciphers. Building upon these concepts, a new methodology [51] was proposed for evaluating and ranking chaotic maps suitable for cryptographic applications. The chaotic behavior of natural chaotic maps is also highly sensitive to quantization errors introduced by fixed-point arithmetic. Shujun [52–55] proposed several measurable dynamic indicators to quantitatively assess the degradation of piecewise chaotic maps when implemented with finite precision (fixed-point arithmetic).

The main contribution of this work is to propose an S-box implementation in which traditional Look-Up Table (LUT) structures, such as those presented in [20–47], are replaced by an on-the-fly fixed-point arithmetic calculation of a non-linear differential equation. This structure is referred to as a Chaos-Based Substitution Box (CB-SBox) and offers notable advantages in terms of area, power consumption, and cryptographic strength. Its temporal evolution drives the initial state x_0 , also referred to as the initial condition (IC), toward a deterministically linked yet uncorrelated final state x_f . The final state is reached after executing the chaotic map for a fixed number of iterations. To the best of the authors' knowledge, this is the first attempt to design a chaotic S-box based on this on-the-fly approach.

The rest of this work is organized as follows: Section 2 discusses the fixed-point arithmetic implementation of six non-linear differential chaotic maps. The evaluation method proposed in [51] is applied to select a chaotic map suitable for a Chaos-Based Substitution Box. In Section 3, the chosen map is used to construct a Chaos-Based Substitution Box (CB-SBox) circuit. This construction allows the final state of the selected differential chaotic map (the circuit output) to be obtained by feeding an initial condition into the circuit's input. Section 4 compares the properties of the proposed CB-SBox with a typical memory-based LUT implementation. For a fair comparison, both implementations are carried out on an ASIC (Application-Specific Integrated Circuit) using VHDL (VHSIC Hardware Description Language). Parameters such as power, area, and time are analyzed in this section. The security improvements are evaluated by measuring resistance to differential and linear cryptanalysis [56]. Based on the proposed CB-SBox and the design decisions from the previous sections, Section 5 presents an application of the CB-SBox in a Feistel network block cipher from [30]. Finally, Section 6 concludes the article by summarizing the results and providing a glimpse of future work.

2. Chaotic Map Evaluation

The S-box proposed in this work is referred to as the Chaos-Based Substitution Box (CB-SBox). Its core is the fixed-point implementation of a non-linear differential system exhibiting chaotic behavior. Since Lorenz's seminal 1963 paper [1], more than forty such maps have been reported [2–7]. From this collection, six maps were selected and evaluated using the method proposed in [51], with the objective of identifying the most suitable candidate for the CB-SBox design. The evaluation set was limited to six maps due to the significant computational effort required to simulate the evolving states driven by different initial conditions (ICs) across the allowable state space. Notably, the evaluation included the Sprott N map instead of the Sprott A, which demonstrated poor Attractor Density (d_A) values [51], and the plate 39 configuration was retained for Chua's map.

The evaluation method described in [51] tracks four quantitative parameters (see Table 1): Attractor Density (d_A), Largest Subset Volume (V_{Sub}), Distance-to-Time Ratio (DTR), and Maximum Dispersion Time (MDT). The first parameter, Attractor Density (d_A), directly reflects the spatial distribution of a strange attractor \mathcal{A} [57]—the set of states in the state space of a chaotic map. Low values of d_A indicate that the chaotic map has poor dispersive capability over the state space, which is typically undesirable in applications such as chaotic encryption, where performance relies on the butterfly effect [58]. Let BP_A

be the smallest axis-aligned bounding parallelepiped that contains the attractor set \mathcal{A} , and suppose it is partitioned into smaller parallelepiped volumes sp . Let $Vol(\cdot)$ denote the volume function. If k of these sub-volumes contain at least one point of \mathcal{A} , then the Attractor Density is calculated as follows in Equation (1):

$$d_{\mathcal{A}} = \lim_{Vol(sp) \rightarrow 0} \left(\frac{\sum_{n=1}^k Vol(sp_n)}{Vol(BP_{\mathcal{A}})} \right) \quad (1)$$

The second parameter, obtained from [51], the Largest Subset Volume (V_{Sub}), is the subset volume that can be extracted from V_{IC} (volume of initial conditions—the three-dimensional volume containing all ICs capable of originating a strange attractor \mathcal{A} as shown in Figure 1). Let $|\cdot|$ be the cardinality function, let $[\cdot]^c$ denote the complement operation between sets, let \cup be the union operator, let SS be the set of ICs belonging to the state space, let \mathcal{A}_{all} be the set of all possible attractors of a chaotic map, and let \mathcal{A}_{cv} and \mathcal{A}_{dv} be, respectively, the subsets of degenerated attractors that converge to a pole and those that diverge to infinity, both belonging to \mathcal{A}_{all} . Then, the set of strange attractors \mathcal{A} is defined as $[\mathcal{A}_{cv} \cup \mathcal{A}_{dv}]^c$, and the V_{IC} is defined by Equation (2):

$$V_{IC} = \lim_{|SS| \rightarrow \infty} \left(\bigcup_{IC \rightarrow \mathcal{A}} [IC] \right) \quad (2)$$

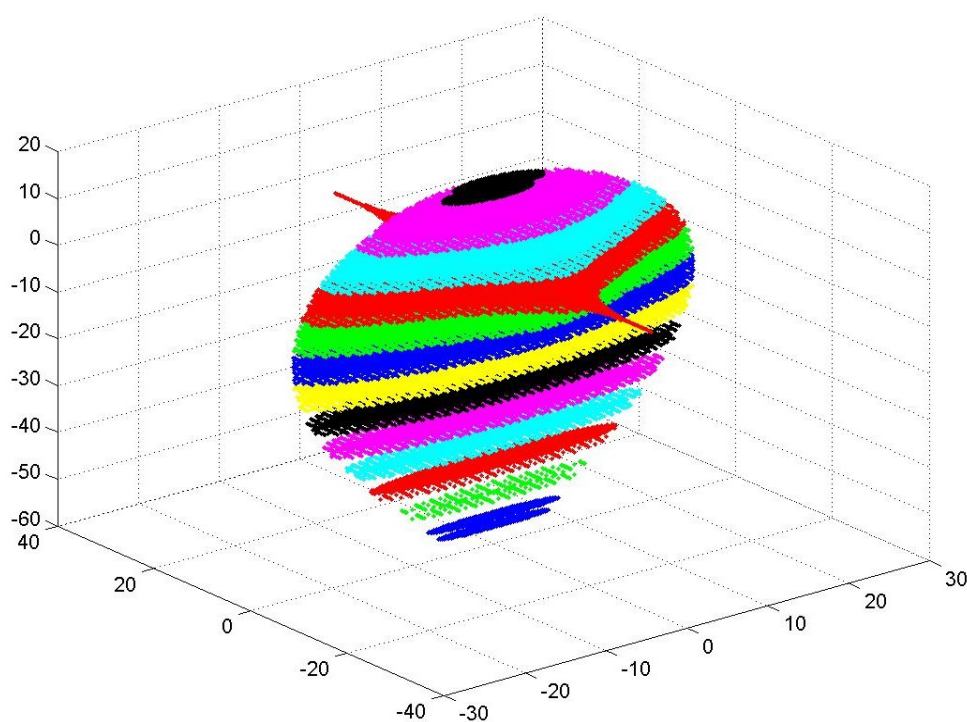


Figure 1. Three-dimensional volume of valid initial conditions (V_{IC}), computed for the Sprott B chaotic map [5], adapted from [50], and made available online in [59] to support three-dimensional visualization.

The V_{IC} of each chaotic system has an irregular shape, which is difficult to represent analytically, and is best visualized graphically (see Figure 1). To overcome this intractability, the Largest Subset Volume (V_{Sub}) is defined as the volume of the largest parallelepiped entirely contained within V_{IC} . Unlike V_{IC} , the V_{Sub} has tractable orthogonal boundaries,

making it suitable for microelectronic realizations. This subset can provide a sufficient number of ICs for a wide range of applications. The use of ICs not belonging to V_{IC} leads to degenerate attractors (\mathcal{A}_{cv} or \mathcal{A}_{dv}), which weakens chaotic maps realizations by making the map's future states predictable—an undesired effect, especially in encryption applications. Let $Var(\cdot)$ be the function that measures the maximum variation of a given variable, and let $h_{V_{IC}}$, $w_{V_{IC}}$, and $d_{V_{IC}}$ represent the three dimensions of the largest parallelepiped within V_{IC} . Then, the volume V_{Sub} is calculated by Equation (3):

$$V_{Sub} = Var(h_{V_{IC}}) \times Var(w_{V_{IC}}) \times Var(d_{V_{IC}}) \quad (3)$$

The third parameter, obtained from [51], Distance-to-Time Ratio (DTR), deals with the dispersion of the chaotic output states over \mathcal{A} . Let V_{Hist} be a normalized metric with $V_{Hist} \rightarrow 1$ representing a complete dispersion, let n be the number of dimensions of a chaotic map, let m be the number of output bins of a histogram function, let B_{mn} be the m -set histogram bins measured for dimension n , and let $min(\cdot)$ and $ave(\cdot)$ be the minimum and mean function, respectively. Then, V_{Hist} can be calculated by the product of Equation (4):

$$V_{Hist} = \prod_{i,j=1}^{n,m} \left(\frac{min(B_{ji})}{ave(B_{ji})} \right) \quad (4)$$

Because of the butterfly effect [58], given a set of ICs and considering the minimum distance ds between neighboring ICs, after some time, the corresponding final states disperse over \mathcal{A} beyond a given threshold measured by V_{Hist} . Smaller distances ds require more time to achieve this dispersion, leading to a negative ratio between ds and time. Let V_{Hist} be a given dispersion threshold, let $ave(\cdot)$ denote the mean function, let $d(\cdot)/dt$ represent the temporal derivative of a given function, and let $g(\cdot)$ be the function that relates the minimum distance ds between successive ICs to the time necessary for their outputs to spread beyond the V_{Hist} threshold. Then, the evaluation parameter DTR can be calculated by Equation (5). The reader should note that smaller absolute values of DTR imply that, for the same distance ds between successive ICs, less time is needed to disperse the state outputs over \mathcal{A} , which is a desirable property for chaotic systems:

$$DTR_{V_{Hist}} = ave \left(\frac{d(g_{V_{Hist}}(ds))}{dt} \right) \quad (5)$$

The fourth parameter, obtained from [51], Maximum Dispersion Time (MDT), also deals with the dispersion of the chaotic output states over \mathcal{A} . Let V_{Hist} be a given dispersion threshold, let $max(\cdot)$ be the function that measures a maximum peak value, and let $g(\cdot)$ be the function that relates the minimum distance ds between successive ICs to the time necessary to spread the outputs over \mathcal{A} , beyond the V_{Hist} dispersion threshold. Then, the evaluation parameter MDT can be calculated by Equation (6):

$$MDT_{V_{Hist}} = \lim_{ds \rightarrow 0} (max(g_{V_{Hist}}(ds))) \quad (6)$$

The MDT parameter represents an upper bound concerning the time necessary to disperse the output states of a given chaotic map, which does not depend on the distance ds between successive ICs. Observe that the distance ds is directly related to the Discretization Criterion adopted by the microelectronic implementation of a given chaotic map.

After calculating the four parameters $d_{\mathcal{A}}$, V_{Sub} , DTR, and MDT discussed in this section, we obtain the data presented in Table 1. Note that the best choice considering the Attractor Density ($d_{\mathcal{A}}$) is Sprott B, which exhibits the strongest ergodicity among the analyzed maps. Regarding V_{Sub} , the best option would be Sprott N, as it presents a larger parallelepiped

volume containing the valid initial conditions (V_{IC}). The Distance-to-Time Ratio (DTR) presents negative values as expected. Sprött B is the best choice for this parameter, as it has the smallest absolute values of DTR, implying that, for the same distance ds between successive ICs, less time is required to disperse the state outputs over \mathcal{A} . Finally, the Maximum Dispersion Time (MDT) shows that the best choice regarding this criterion is also Sprött B, which disperses its outputs over \mathcal{A} the fastest. For the final conclusion, the grades in Table 1 are normalized, with the best score set to 1. A final score is calculated based on a set of four weights (r, s, t, u) , defined as $(0.20, 0.34, 0.12, 0.34)$. The weighted average grade is shown in the last column of Table 1. The evaluation method ultimately allows for the selection of the highest-scoring map, Sprött B as highlighted in Table 1, making it the most suitable for the intended application.

Table 1. Performance comparison of six selected chaotic maps according to the four-parameter evaluation method proposed in [51].

| Equation | $d_{\mathcal{A}}$ | V_{Sub} | DTR | MDT | Grade |
|--------------|-------------------|---------------------|------|------|--------|
| Chua 39 [6] | 0.0574 | 3.200×10^1 | −486 | 8558 | 0.1353 |
| Rössler [2] | 0.0773 | 6.553×10^4 | −235 | 8251 | 0.1790 |
| Linz [7] | 0.0938 | 2.000×10^0 | −173 | 3610 | 0.2648 |
| Sprött N [5] | 0.0484 | 3.013×10^7 | −346 | 8243 | 0.4686 |
| Lorenz [1] | 0.0899 | 1.872×10^7 | −114 | 2700 | 0.5198 |
| Sprött B [5] | 0.1380 | 5.632×10^3 | −48 | 1015 | 0.6601 |

Discrepancies can be observed, as expected, when comparing the grades obtained in the present work (Table 1), which are derived from a fixed-point arithmetic implementation, with those reported in [51], which are based on floating-point arithmetic. Among the four evaluation parameters presented in Table 1, $d_{\mathcal{A}}$ is the least affected by the transition from floating-point to fixed-point arithmetic, as the volume of \mathcal{A} undergoes negligible variation in both implementations. On the other hand, the V_{Sub} parameter is significantly constrained by the fixed-point implementation since the size and shape of V_{IC} are altered due to the accumulation of quantization errors in both the ICs and the intermediate states of \mathcal{A} . Quantization errors also affect the trajectory of the intermediate states within \mathcal{A} , resulting in differences in the calculated DTR and MDT values.

Both floating-point and fixed-point arithmetic implementations are approximations of real chaotic maps found in nature. The digitization of natural chaotic systems inevitably introduces errors, which may cause the maps to deviate from chaotic behavior at some point during the simulation—a phenomenon known as dynamic degradation [52]. Such digital maps may eventually converge, diverge, or enter oscillatory modes, making their behavior predictable—an extremely undesirable characteristic in many applications of chaotic maps, particularly in cryptography. Although this behavior is inherent to the digital realization of chaotic maps—meaning it cannot be avoided with 100% certainty—the literature identifies three remedies [52] that can be applied to reduce the occurrence of dynamic degradation in digital chaotic systems: using higher finite precision, cascading multiple chaotic systems, and applying perturbation-based algorithms. In this work, we use higher finite precision both to combat dynamic degradation and to strengthen the encryption algorithm as evidenced in Section 4. The approach presented in this section also mitigates dynamic degradation by selecting a chaotic map that was extensively simulated prior to use in order to evaluate the four parameters $d_{\mathcal{A}}$, V_{Sub} , DTR, and MDT discussed above. The Maximum Dispersion Time (MDT) parameter was used to limit the digital simulation time, effectively helping to ensure that no dynamic degradation occurs during this early period of simulation ($0 < t < \text{MDT}$).

3. Chaos-Based Substitution Box (CB-SBox)

As stated in Section 2, the Sprott B chaotic map was selected for the construction of the Chaos-Based Substitution Box (CB-SBox) proposed herein. This map was introduced by Sprott in 1994 [5] as part of a mathematical effort to identify simple algebraic three-dimensional ordinary differential equations that exhibit chaotic behavior, resulting in the discovery of nineteen distinct systems. The Sprott B chaotic map is characterized by two critical points, $(1, 1, 0)$ and $(-1, -1, 0)$, and three Lyapunov exponents $\lambda = (0.210, 0, -1.210)$, where the largest exponent is positive as required for the emergence of chaos. Additionally, the system has a fractal dimension of 2.174, and its strange attractor \mathcal{A} is illustrated later in this paper (see Section 4.2). As shown in Equation (7), the Sprott B system features three integrators for \dot{x} , \dot{y} , and \dot{z} ; two multipliers for yz and xy ; and two subtractors for $x - y$ and $1 - xy$. In Section 5, the CB-SBox is applied in a practical example, serving as the source of confusion in a Feistel network based on the block cipher proposed by Jakimoski in 2001 [30]:

$$\begin{cases} \dot{x} = yz \\ \dot{y} = x - y \\ \dot{z} = 1 - xy \end{cases} \quad (7)$$

The S-boxes used in [30] are derived from Equation (8), where $g(\cdot)$ is obtained from a discrete chaotic map. The function f_j is defined in [30] through a procedure based on the N^{th} iteration of the logistic chaotic map. Essentially, the function f_j maps an 8-bit input to a pre-computed 8-bit output derived from the proposed procedure. The hardware implementation of f_j is straightforward: a ROM block with a depth of 256 positions and a word length of 8 bits. Jakimoski [30] employed eight such s-boxes in his design, resulting in a 64-bit block cipher:

$$f_j = g(x_1 \oplus x_2 \oplus \dots \oplus x_j \oplus z_j) \quad (8)$$

For the present CB-SBox design, the function $f_j : M \rightarrow M$ defined by Equation (8), where $M = \{0, 1, \dots, 255\}$, is replaced by a digital realization of the Sprott B chaotic map from Equation (7). The complete CB-SBox schematic circuit view (the model is available online in [59]) is shown in Figure 2a, while Figure 2b presents the details inside the Sprott_B subsystem. Observe in Figure 2b the three integrators responsible for solving the chaotic map from Equation (7). The CB-SBox shown in Figure 2a has a single input, which is sliced into three initial conditions $(x_0, y_0$ and $z_0)$, and correspondingly, a single output carrying the three final states $(x_f, y_f$ and $z_f)$. After properly setting the IC values inside V_{IC} (Figure 1), the circuit will run until an output uncorrelated from its input is obtained. The running time (called LongRun in this work) is calculated by using the parameter DTR in Table 1 according to the Discretization Criteria (DCs) discussed in [50]. The DTR parameter is dependent on the minimal distance between successive ICs from V_{IC} (Figure 1).

The parameters used to configure the circuit presented in Figure 2 are fully programmable and allow for the realization of the Sprott B-based CB-SBox in different lengths, i.e., the number of bits (NOB). Several precomputed parameters are stored in two configuration datasets, named PRM (Parallelepiped Relations Measurements) and Slice, which are available online in [59]. Each selected length (i.e., NOB) is associated with its own pair of datasets (PRM and Slice), built such that any CB-SBox input generates an initial condition (x_0, y_0, z_0) belonging to V_{IC} (see Figure 1). The values from the PRM and Slice datasets are used to configure the internal components of the programmable CB-SBox model (available in [59]), which are then implemented in the circuit shown in Figure 2.

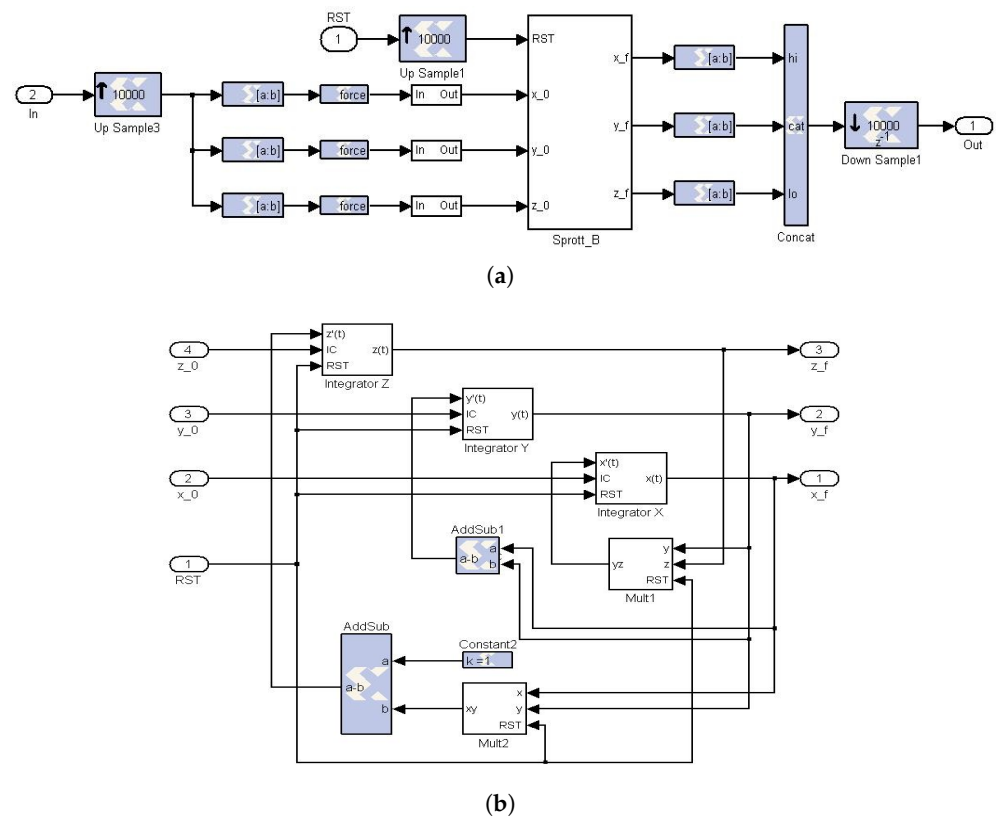


Figure 2. Schematic circuit view of the CB-SBox model and its Sprott B subsystem, available in [59]. (a) Chaos-Based Substitution Box (CB-SBox). (b) Sprott B [5] subsystem of Equation (7).

Each PRM dataset synthesizes, for a given NOB value, the dimensions of non-overlapping parallelepipeds within V_{IC} (see Figure 1) as exemplified in Table 2. Additionally, the PRM dataset lists, based on the selected dimensions and the applied Discretization Criterion (DC), the bit lengths before and after the binary point for the limiting vertices of the (x, y, z) parallelepipeds, including their respective axial distances (Δ).

Table 2. The dimensions of three non-overlapping parallelepipeds belonging to the volume of valid initial conditions (V_{IC}) shown in Figure 1.

| x_{min} | x_{max} | Δ | y_{min} | y_{max} | Δ | z_{min} | z_{max} | Δ |
|-----------|-----------|----------|-----------|-----------|----------|-----------|-----------|----------|
| −12.125 | −8.125 | 4 | −4 | 4 | 8 | −24 | 8 | 32 |
| −8 | 8 | 16 | −8 | 8 | 16 | −8 | 8 | 16 |
| 8.125 | 12.125 | 4 | −4 | 4 | 8 | −24 | 8 | 32 |

The Slice dataset is specified by the NOB parameter and the values stored in the corresponding PRM dataset. The parameters contained in the Slice dataset are used to configure the force and slice blocks of the model shown in Figure 2b. The circuit depicted in Figure 2 operates under the control of a reset signal, whose period is defined by the internal parameter LongRun. This reset signal initializes the CB-SBox to a new initial condition at the beginning of each LongRun cycle. It also controls the three integrators by triggering two dynamic registers within each one, both of which feature a configurable initial condition port. Additionally, the reset signal forces the two multipliers in Figure 2b to a null initial state at the start of each cycle. The integrators are further governed by an integration step size parameter (st), set to 0.01, and the multipliers operate with a latency parameter (Lat) of 5.

4. CB-SBox Performance Analysis

The main distinction between the block cipher presented in Section 5 and that introduced in [30] lies in the implementation of the S-boxes. In the design proposed here, they are realized using the Chaos-Based Substitution Box (CB-SBox) model described in Section 3 (see Figure 2). In contrast, the S-boxes in [30] are implemented as ROM blocks populated through an offline procedure that precomputes the outputs of a chaotic map.

For fair comparison purposes, both designs—the CB-SBox from Section 3 and a ROM-based version implemented by the authors based on the method described in Jakimoski [30]—were synthesized in ASIC using VHDL. The synthesis tool employed was DC Synopsys, version Z-2007.03-SP4, for Linux, using the UMC CMOS 0.13 μm technology library. This setup ensures a more balanced comparison than FPGA-based implementations with VHDL code automatically generated by the system generator since specific architectural features in most FPGAs—such as embedded multipliers and memory blocks—can disproportionately impact resource usage, particularly the number of slices consumed.

4.1. Area and Power Analysis

A comparison is presented between area (μm^2) and power consumption (μW), both plotted along the x -axis, and circuit size, plotted along the y -axis. These comparisons are illustrated in Figures 3 and 4, respectively, for the ROM-based and CB-SBox implementations. The size is defined by the number of bits (NOB) of the circuit input, which equals the output size due to design symmetry.

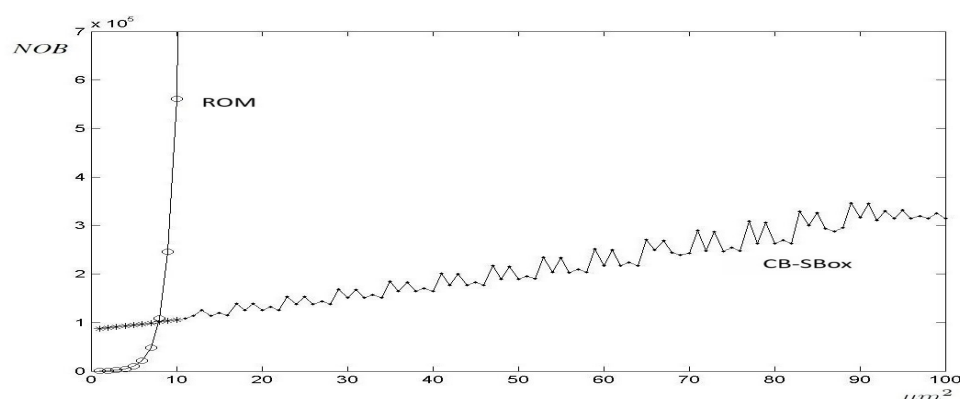


Figure 3. Area comparison between ROM-Based and CB-SBox implementations as a function of circuit input size (NOB).

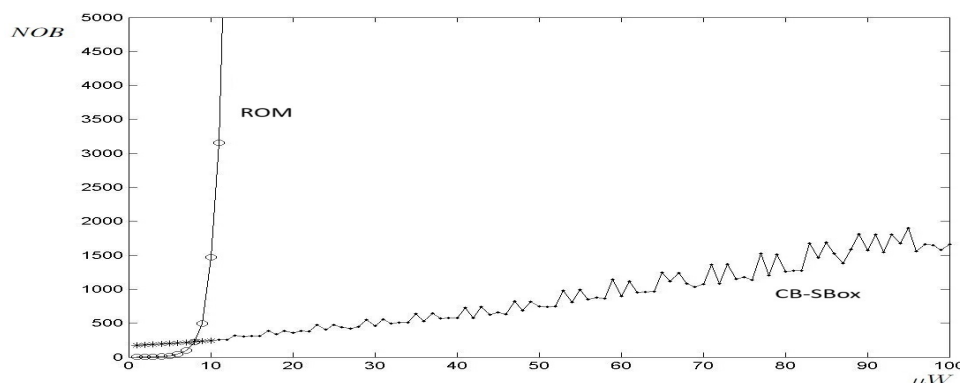


Figure 4. Power consumption comparison between ROM-Based and CB-SBox implementations as a function of circuit input size (NOB).

As shown in Figures 3 and 4, both the area and power consumption of the ROM-based S-boxes increase exponentially with NOB. For a ROM implementation with $\text{NOB} = 8$, meaning both input and output signals are 8 bits wide (corresponding to a memory depth of 256 positions), the area reaches $1.0861 \times 10^5 \mu\text{m}^2$ and the power consumption is $223.1284 \mu\text{W}$. When increasing the word length by one bit ($\text{NOB} = 9$), both area and power consumption nearly double, reaching $2.4612 \times 10^5 \mu\text{m}^2$ and $492.0213 \mu\text{W}$, respectively. Note that the ROM block was implemented up to $\text{NOB} = 13$; larger configurations could not be synthesized due to software and hardware limitations and had to be extrapolated.

In the case of the proposed CB-SBox, as shown in Figures 3 and 4, there is no strong exponential growth as observed in the ROM case. In fact, the CB-SBox consists of components (multipliers, adders, subtractors, multiplexers, and registers), whose area and power consumption exhibit a nearly linear dependence on the circuit size (NOB). The only CB-SBox components with a non-linear dependence on NOB are the multipliers, but they do not significantly affect the overall area and power consumption of the circuit. Consequently, the area and power consumption of the CB-SBox increase almost linearly with NOB.

The area and power consumption of both ROM and CB-SBox designs are plotted together in Figures 3 and 4, respectively. In these figures, the strong exponential behavior of the ROM implementation is evident, in contrast to the nearly linear behavior of the CB-SBox implementation. It can also be observed that the equilibrium point occurs near $\text{NOB} = 8$, leading to the conclusion that for circuits requiring a word length greater than 8 bits, the CB-SBox design outperforms the ROM-based implementation in terms of area and power consumption.

4.2. Timing Analysis

Section 4.1 demonstrates that the area and power consumption of an S-box based on the CB-SBox implementation are superior to those of the ROM-based design for S-boxes with $\text{NOB} > 8$. However, it is important to note that the latency of the CB-SBox implementation (defined by the LongRun parameter) is significantly higher compared to that of the ROM implementation. In chaos-based circuits, a substantially larger number of iterations is required to produce an output that is statistically uncorrelated with its input, whereas the ROM block delivers a precomputed value in a single clock cycle.

The LongRun parameter is determined by the selected Discretization Criterion (DC) [51]: the higher the DC, the larger the LongRun value required to decorrelate the CB-SBox input from its output. Based on the DC values stored in the PRM (Parallelepiped Relations Measurements) configuration dataset, Figure 5 shows the LongRun parameter (y-axis) as a function of the input size (NOB) of the CB-SBox circuit.

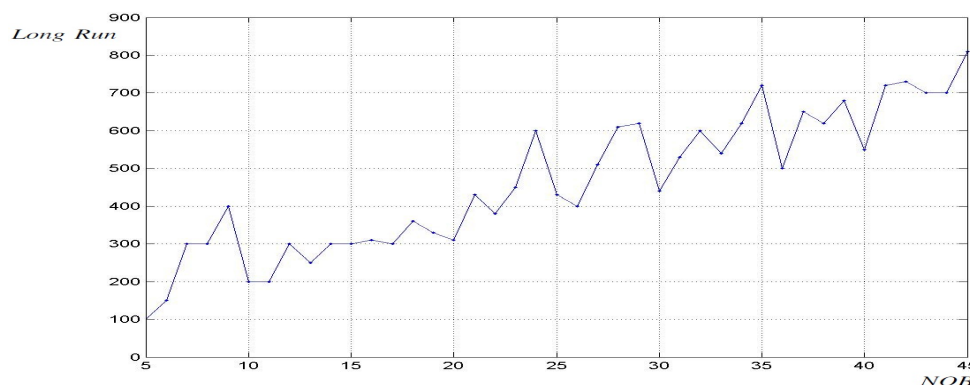
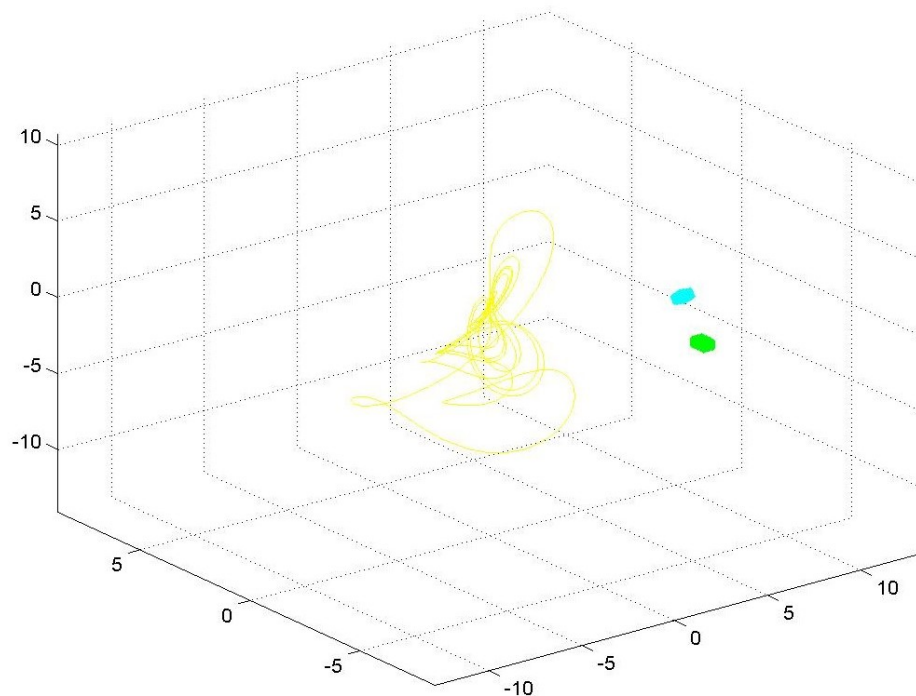


Figure 5. Time response of the CB-SBox: LongRun parameter versus circuit size (measured in NOB).

As shown in Figure 5, the relationship between the LongRun parameter and the NOB used in the CB-SBox exhibits an increasing trend. The data presented in Figure 5 were obtained through simulations of groups of 500 initial conditions (ICs). Each group of ICs was created using a specific Discretization Criterion ($4 \leq DC \leq 45$). The DC parameter is inversely proportional to the minimum distance between any pair of ICs within the group. The LongRun parameter represents the time required for each group of 500 ICs to fully disperse across the volume of the strange attractor \mathcal{A} [51].

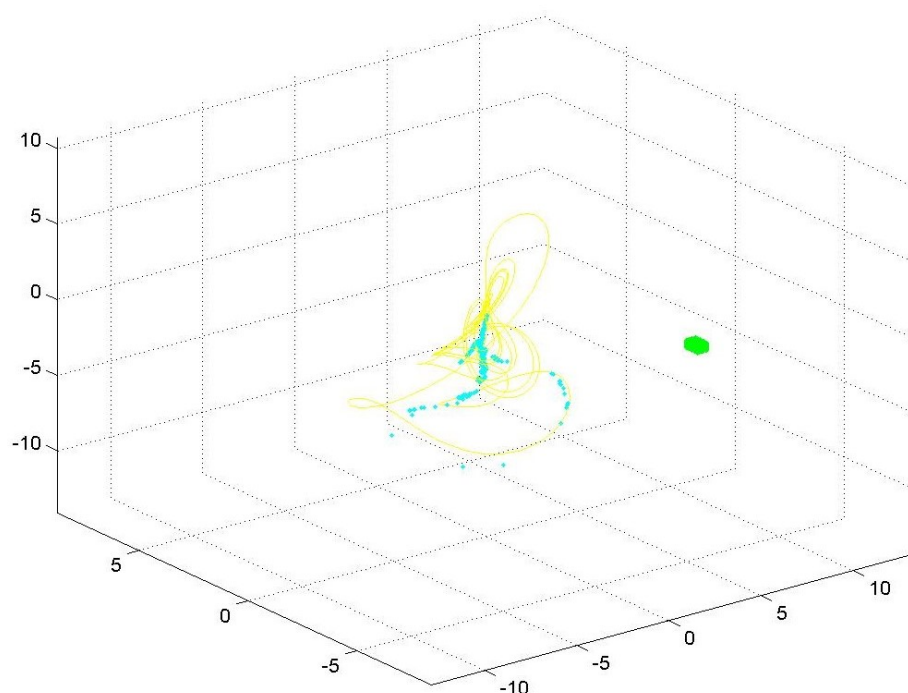
The location of each group of 500 ICs is a uniformly distributed random variable constrained within the volume of valid initial conditions (V_{IC}) [50]. Distinct groups, depending on their position within V_{IC} , exhibit different dispersion patterns. This behavior can be observed even in groups generated with the same DC, which results in the irregular behavior characterized by significant fluctuations as shown in Figure 5. This phenomenon is further analyzed below at the end of Section 4.2, in the penultimate paragraph.

For illustration purposes, a three-dimensional cube of 729 ICs is plotted in Figure 6 in green. Its location is randomly placed within $x \in [7.5085, 8.0085]$, $y \in [-3.3708, -2.8708]$, and $z \in [-2.1740, -1.6740]$. The Discretization Criterion (DC) is set to 4, so the distance between successive ICs is 0.0625, making it easy to identify the green cube in Figure 6. These 729 ICs were simulated using Equation (7), and their corresponding final states are plotted in blue. It can be observed that the final states tend toward the Sprott B Strange Attractor \mathcal{A} (shown in yellow) as expected, and their locations disperse on the attractor due to the butterfly effect. Figure 6 shows the three-dimensional dispersion pattern for 7 and 100 iteration steps. The project database, available online in [59], includes 27 additional three-dimensional dispersion patterns up to 1900 iteration steps, in both JPG and FIG formats.



(a) Final states (blue) dispersed in \mathcal{A} after 7 iteration steps

Figure 6. Cont.



(b) Final states (blue) dispersed in \mathcal{A} after 100 iteration steps

Figure 6. Visualization of a cube with 729 initial conditions (in green) and the three-dimensional dispersion of their corresponding final states (in blue) over the Sprott B attractor \mathcal{A} (in yellow). An additional 27 visualization steps are available online in the project dataset [59].

Despite extensive efforts (1300 groups of 500 ICs were simulated, each group constructed with $DC = 30$), no discernible pattern was identified linking the spatial locations of the IC groups within V_{IC} to the LongRun parameter. Figure 7 illustrates the variation of the LongRun parameter with respect to the spatial distribution of the IC groups within V_{IC} . This figure presents a representative subset of 79 IC groups. The center of each group is defined by its x- and y-axis coordinates at a fixed elevation of $z = -17.0625$. The LongRun parameter is encoded using the following seven-symbol scheme (not all symbols appear in the subset shown in Figure 7): a blue point • indicates that the required LongRun parameter lies within the interval $(420, 460]$; a green circle ○ corresponds to $\text{LongRun} \in (460, 500]$; a red square ■ to $\text{LongRun} \in (500, 540]$; a cyan multiplication sign × to $\text{LongRun} \in (540, 580]$; a magenta addition sign + to $\text{LongRun} \in (580, 620]$; a yellow lozenge ◇ to $\text{LongRun} \in (620, 660]$; and a black asterisk * to $\text{LongRun} \in (660, 700]$. The lack of a clear spatial correlation between group locations and the LongRun parameter suggests the influence of other contributing factors, potentially arising from the system's inherent chaotic dynamics, which warrant further investigation.

The timing analysis presented in this Section 4.2 is intrinsically linked to the circuit throughput. For a ROM-based S-box, its contents are precomputed prior to runtime, and during operation, the output becomes available with minimal latency. In contrast, the CB-SBox proposed in this work generates outputs dynamically, which requires additional computation time during operation. This latency is governed by the LongRun parameter, which increases as the DC value decreases as previously discussed. Consequently, this behavior directly impacts the circuit throughput as will be further analyzed in Section 5, where the proposed CB-SBox is applied to a practical implementation of a Feistel block cipher based on [30].

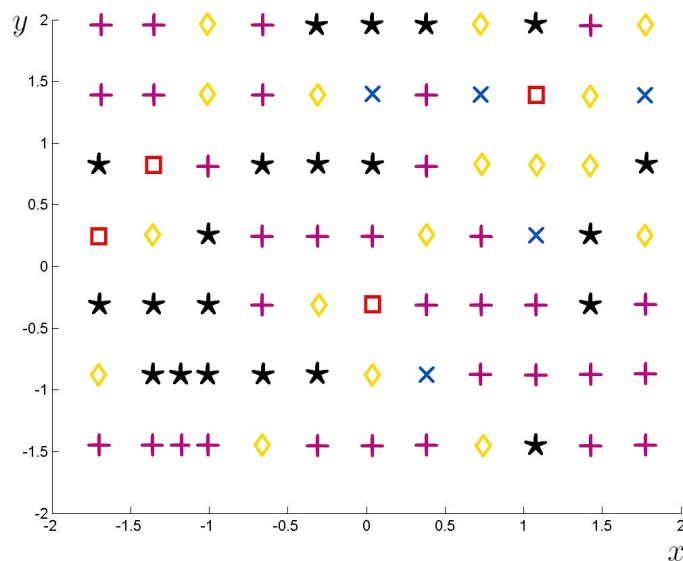
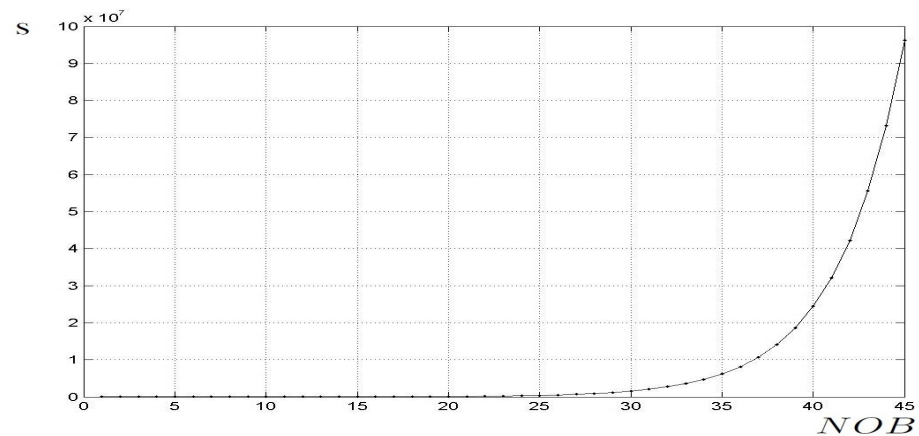


Figure 7. Spatial distribution of the LongRun parameter (within the graph) for 79 groups of 500 ICs located in the plane $z = -17.0625$, with $DC = 30$, and the following markers: ■ LongRun $\in (500, 540]$, × LongRun $\in (540, 580]$, + LongRun $\in (580, 620]$, ◇ LongRun $\in (620, 660]$, and * LongRun $\in (660, 700]$.

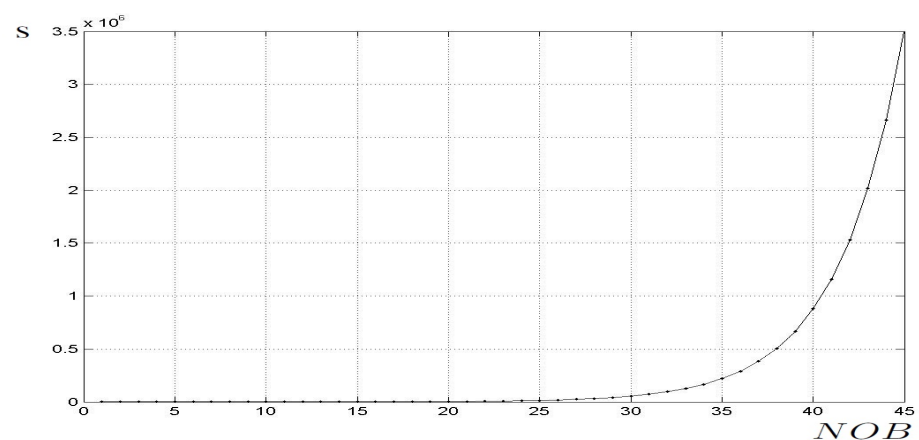
4.3. Security Analysis

Linear cryptanalysis [60] and differential cryptanalysis [61] are the two most prominent attacks applicable to cryptographic schemes [56]. These attacks rely on identifying structural weaknesses in the non-linear components responsible for confusion: the S-boxes (the proposed CB-SBox in this work). The initial step in both techniques involves computing two quantitative parameters dependent on the properties of the S-boxes: the Linear Probability Bias and the Differential Probability Bias.

The effort required to compute the two parameters (Linear and Differential Probability Biases) increases exponentially with the NOB of the S-box, as illustrated in Figure 8, which presents the computation time (in seconds) for each parameter calculation. The values for $\text{NOB} \leq 23$ were obtained through actual computations, while those for $\text{NOB} \geq 24$ were extrapolated based on the exponential trend observed for the lower $\text{NOB} \leq 23$ values. The time required to compute the Linear and Differential Probability Biases represents the initial step in the cryptanalysis of a block cipher and does not directly indicate its resistance to these attacks. Rather, the cipher's vulnerability is determined by the magnitude of the resulting biases—lower bias values correspond to higher resistance to linear and differential cryptanalysis. Nevertheless, performing this computation is a necessary preliminary step in evaluating and strengthening cipher security. As shown in Figure 8a, computing the Linear Probability Bias for $\text{NOB} = 23$ required 10 days, and it is estimated that for $\text{NOB} = 43$, the process would take approximately 2 years. The proposed CB-SBox supports larger NOB values than conventional ROM-based S-boxes, providing a strong initial barrier against cryptanalytic attacks.



(a) Computation time (y-axis) required to evaluate the Linear Probability Bias.



(b) Computation time (y-axis) required to evaluate the Differential Probability Bias.

Figure 8. Security analysis based on the computation time (seconds) required to evaluate the Linear and Differential Probability Biases of S-boxes as a function of their size (NOB).

4.4. Bijectivity Analysis

The proposed Chaos-Based Substitution Box (CB-SBox) maps all possible inputs within V_{IC} (as shown in Figure 1) to their corresponding outputs by iterating the chaotic map defined in Equation (7) over a sufficiently long period (LongRun). This process enables the outputs to spread across the strange attractor \mathcal{A} due to the butterfly effect as illustrated in Figure 6. As a result of the inherent nature of chaotic systems—characterized by sensitive dependence on initial conditions and the fractal structure of strange attractors—iterating Equation (7) for an extended duration causes the outputs to become not only widely dispersed across the attractor but also potentially mapped to locations already assigned to other outputs, leading to an inherently non-bijective behavior.

CB-SBoxes with different numbers of bits (NOB) were simulated, and the results are summarized in Table 3. Their deterministic outputs were stored in matrices referred to as SBOX Chaotic, each of length 2^{NOB} as specified in Table 3. Additional matrices of identical dimensions, named SBOX Chaotic Counter, were employed to record the number of times each output value occurred during the simulations. For a matrix to be considered bijective, each output must appear exactly once, which corresponds to a bijectivity index of 100%, meaning that the number of ones (1s) in the SBOX Bijective Counter matrix is 100%.

Due to their chaotic nature, the proposed CB-SBoxes are inherently non-bijective, exhibiting a mean bijectivity index of 63.57% across the simulated cases presented in Table 3. This indicates that only this proportion of possible outputs was mapped by the CB-

SBox during the simulations, while the remaining 36.43% of outputs were never mapped, corresponding to the proportion of zeros (0s) in the *SBOX Chaotic Counter*. The complete statistical dataset referenced in this analysis is available online in [59].

Table 3. Statistical analysis of the bijectivity of CB-SBoxes, where NOB denotes the number of bits, followed by the corresponding table length that equals 2^{NOB} and the calculated bijectivity index (data available online in [59]).

| NOB | Length | Bijectivity |
|-----|--------|-------------|
| 11 | 2048 | 63.48% |
| 12 | 4096 | 63.16% |
| 13 | 8192 | 63.09% |
| 14 | 16,384 | 62.91% |
| 15 | 32,768 | 63.43% |

The lack of bijectivity is a factor that must be considered from a holistic perspective as discussed in more detail in Section 4.5. S-boxes are integral components of various cryptographic schemes, primarily introducing non-linearity and confusion to enhance security. Traditionally, S-boxes are bijective, which is crucial for invertibility in these cryptographic processes. However, certain cryptographic schemes, such as hash functions, stream ciphers, specific cryptographic protocols, and Feistel networks, can employ non-bijective S-boxes, where the lack of bijectivity does not compromise security or functionality as demonstrated in other scientific works [44,45].

4.5. Joint Analysis

As discussed in Section 4.1, when dealing with large S-boxes ($NOB \gg 8$), it is more advantageous to implement a CB-SBox rather than using a ROM-based approach due to the exponential growth of the latter, resulting in significantly smaller circuits with lower power consumption when using the CB-SBox.

However, as pointed out in Section 4.2, this approach results in slow response times: due to the Butterfly Effect [57], chaotic behavior takes a long time to fully develop as shown in Figures 5 and 6.

The analysis presented in Section 4.3 shows that, despite its slow response time, the CB-SBox is a valuable approach for enhancing the security of block ciphers: the CB-SBox enables the implementation of large S-boxes, which enhances the cipher's resistance to linear and differential cryptanalysis as shown in Figure 8.

The bijectivity of the proposed CB-SBox is analyzed in Section 4.4. The CB-SBox exhibits a mean bijectivity index of 63.57%, indicating non-bijective behavior resulting from the chaotic nature of the Sprott B [5] map used within. Several scientific works have addressed this issue by the offline substitution of repeated S-box outputs. However, this approach was not adopted in the present research, as it would undermine the primary advantage sought in this work: the emulation of a large S-box functioning during execution.

It is worth noting that the CB-SBox design not only facilitates the implementation of large S-boxes ($NOB > 35$) but also achieves this with smaller circuits (lower area) and reduced power consumption, compared to a ROM-based implementation. A realizable 30-bit CB-SBox, with a security level equivalent to that of a 30-bit non-realizable (within the scope of this work) ROM implementation, both requiring 17 days for the Linear Probability Bias calculation, consumes only 0.0004% of the area and power needed by the 30-bit ROM.

Therefore, the joint analysis presented here in Section 4.5 demonstrates that the CB-SBox offers an effective solution when a high level of security is required. The latency issue addressed in Section 4.2 can be mitigated by the compact area and low-power-consumption characteristics of the CB-SBox, with a parallel circuit implementation serving

as an alternative to attenuate the high-latency problem discussed in Section 4.2. Further details regarding this are provided in Section 5, which focuses on the Feistel network implementation using the proposed CB-SBox.

5. CB-SBox in a Feistel Network

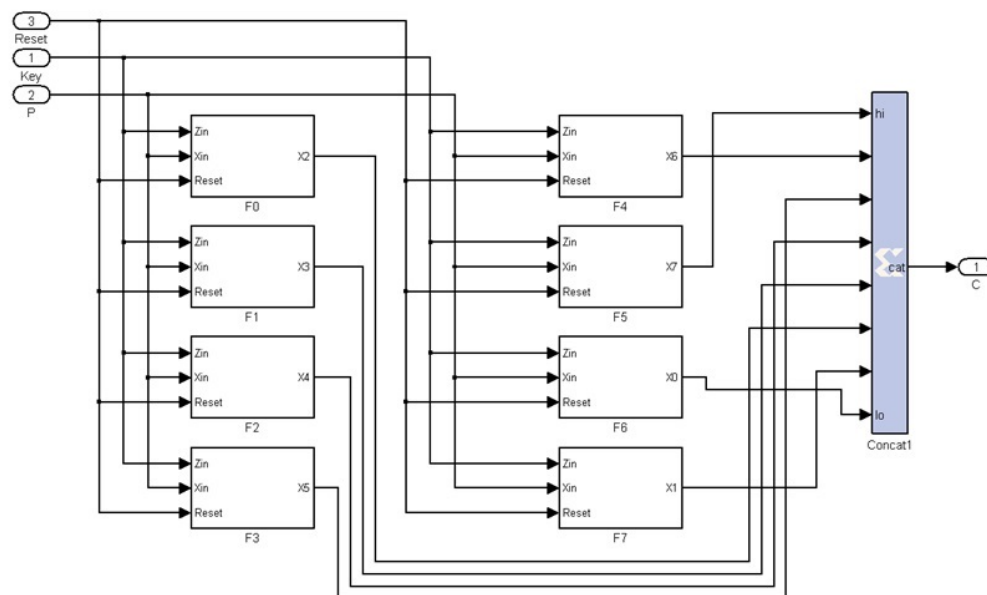
The Chaos-Based Substitution Box (CB-SBox) developed in this work was employed in the construction of a Feistel network based on the architecture proposed by Jakimoski [30]. In the present study, the precomputed ROM-based S-boxes utilized in [30] were substituted by the proposed CB-SBox, which is detailed in Section 3 and analyzed in Section 4.

The schematic diagram of the designed block cipher is shown in Figure 9, which implements Equation (9), obtained from Jakimoski's work [30]. Figure 9a illustrates the complete block cipher with its three inputs: the plaintext B_0 divided into x_k words, where $k = 1, 2, \dots, 8$; the round-related subkeys z_i , where $i = 1, 2, \dots, r$; and the *reset* signal. Figure 9a also shows its output, which is the ciphertext B_r , obtained after r rounds:

$$x_{i,k+1} = x_{i-1,k} \oplus f_{k-1}[x_{i-1,1}, \dots, x_{i-1,k-1}, z_{i-1,k-1}] \quad (9)$$

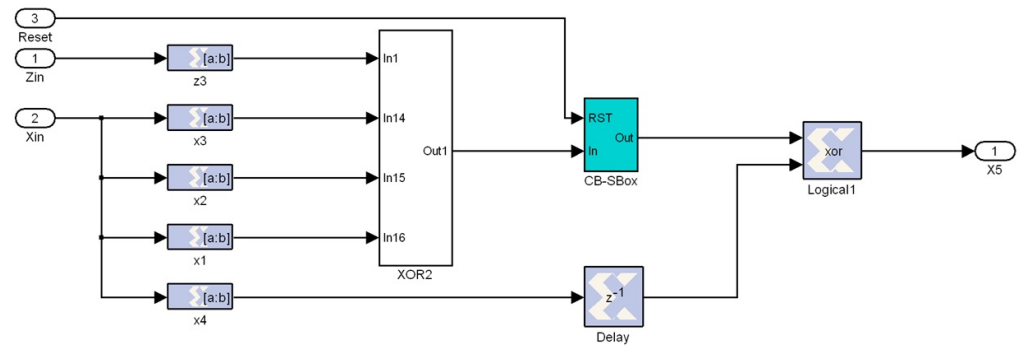
Figure 9b illustrates the internal details of subsystem f_3 in Figure 9a, where the CB-SBox subsystem is highlighted in blue (its schematic diagram is presented in Figure 2). All subsystems f_0 to f_7 in Figure 9a exhibit a similar structure to the one shown in Figure 9b, each containing one CB-SBox but with an XOR port having different inputs as described by Equation (9).

As discussed in Section 3, the CB-SBox is a programmable circuit whose word length ($11 < \text{NOB} < 41$) can be configured through PRM and Slice datasets available online in [59]. Consequently, since the block cipher illustrated in Figure 9 comprises eight CB-SBoxes, the total word length processed by the cipher can be programmed to range from 88 to 328 bits. In the example shown in Figure 9, the cipher is implemented with a total word length of 136 bits.



(a) Feistel network using the proposed CB-SBox.

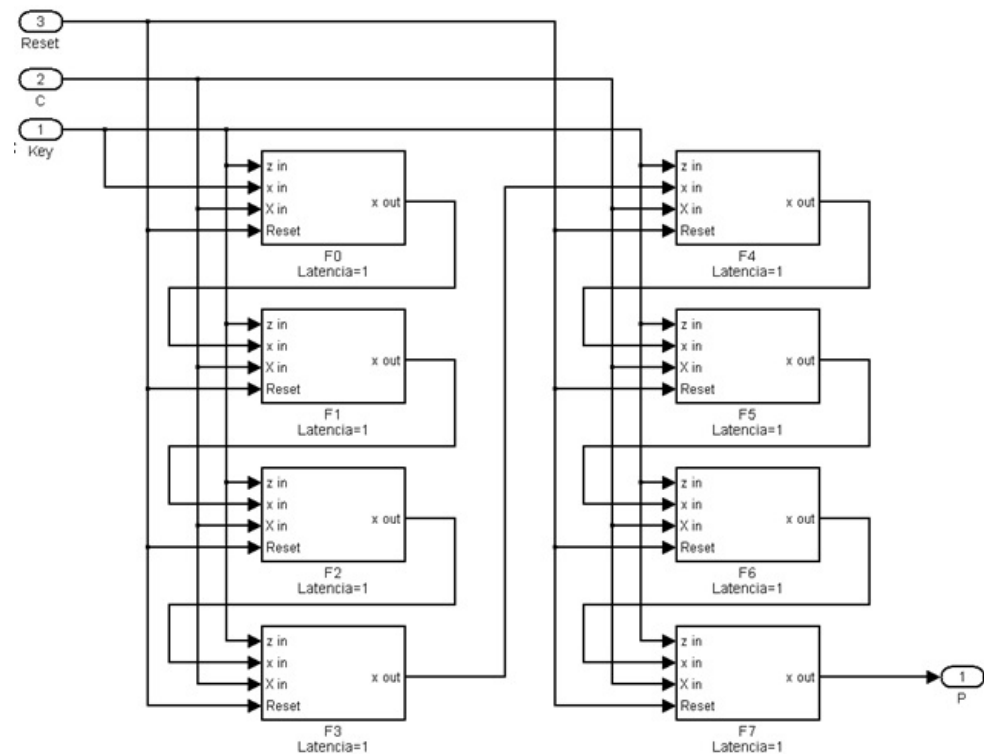
Figure 9. Cont.

(b) Internal details of subsystem f_3 .**Figure 9.** Schematic diagram of the encryption block cipher, where the CB-SBox, embedded within each f_k subsystem and previously detailed in Figure 2, is highlighted in blue.

The block cipher is implemented recursively: only a single round of the cipher is physically realized, and the intermediate encrypted words B_i , where $i = 1, 2, \dots, r - 1$, are fed back to the input $r - 1$ times to complete the required r rounds of encryption.

The decryption architecture is derived from Equation (10), as presented in [30], by applying the inverse transformations defined in Equation (9): the round subkeys z_i are applied in reverse order, and the same number of rounds r are performed on the ciphertext block B_r to recover the original plaintext block B_0 . The decryption operation involves series connections between the f_k blocks, requiring eight times as many clock cycles to complete its tasks compared to encryption, due to the feedback arrangement shown in Figure 10:

$$x_{i-1,k} = x_{i,k+1} \oplus f_{k-1}[x_{i-1,1}, \dots, x_{i-1,k-1}, z_{i-1,k-1}] \quad (10)$$

**Figure 10.** Schematic representation of the decryption block cipher.

The block cipher implementation results in a multi-rate circuit as a result of the presence of up and down samplers in the architecture shown in Figure 2, which are integrated

within the CB-SBox subsystem depicted in Figure 9. These samplers are necessary to compensate for the latency introduced by the multipliers, which equals five clock cycles, and the integration step size, set to 0.01. These parameters are used in the implementation of the Sprott B differential chaotic map defined in Equation (7). The block cipher operation is controlled by a reset signal, which synchronizes the loading of initial conditions with the cycle initialization, governed by the LongRun parameter.

The performance analysis of the FPGA-implemented block cipher is presented in Figure 11. This figure includes four graphs, each plotted as a function of the NOB (x-axis): the size of the proposed programmable CB-SBox. Figure 11a shows the maximum operating frequency of the circuit, measured in MHz (y-axis). As the size of the CB-SBox increases, the maximum frequency decreases as expected. Based on Figure 11a, the interval $11 < \text{NOB} < 20$ was selected as the target design range.

Figure 11b illustrates the data rate, measured in Mbps (y-axis), computed as the product of the maximum circuit frequency and the size of the encrypted word, while accounting for the compensation required by the LongRun parameter. As discussed in Section 4.2, the timing analysis of the CB-SBox is intrinsically linked to the circuit throughput. In ROM-based S-box architectures, the output is available with minimal latency. In contrast, the CB-SBox proposed in this work incurs additional computational delay during execution. As a result, this behavior directly impacts the overall throughput of the cipher. Figure 11b reveals a peak data rate of 2.755 Mbps at $\text{NOB} = 19$, which is significantly lower than the typical 10 Mbps achieved by ROM-based implementations.

The remaining two graphs present the hardware resource utilization, measured in FPGA slices as shown in Figure 11c, and the circuit cost, measured in slices per Kbps as presented in Figure 11d. Both graphs exhibit a local minimum at $\text{NOB} = 19$, justifying the selection of this value as the optimal operating point for the proposed CB-SBox architecture.

The chosen operating point ($\text{NOB} = 19$) lies in the region where the CB-SBox exhibits superior performance in terms of area and power consumption when compared to ROM-based S-box implementations as illustrated in Figures 3 and 4. These figures show that a 19-bit CB-SBox implementation utilizes only 0.0238% of the area and 0.0241% of the power consumed by a securely equivalent, but physically unrealizable, ROM-based counterpart. It is important to highlight that the proposed CB-SBox constitutes the only deviation from the block cipher structure described in [30]. Moreover, it should be emphasized that $\text{NOB} = 13$ corresponds to the largest realizable ROM implementation feasible with the hardware and software resources available in the laboratory.

As discussed in Section 4.5, the main advantage of applying the CB-SBox to the Jakimoski-based cipher [30] is the ability to implement large S-boxes ($\text{NOB} \gg 8$), thereby achieving high levels of resistance against linear and differential cryptanalysis. The block cipher designed in this work, which employs a 19-bit CB-SBox, requires approximately 20 h to compute the Linear Probability Bias and 38 min to compute the Differential Probability Bias as shown in Figure 8. Such a 19-bit implementation would not be feasible using a ROM-based approach. It is also important to emphasize that this analysis represents only the initial step in the process of attacking a cipher through linear and differential cryptanalysis.

The described 152-bit recursive block cipher, employing the proposed CB-SBox and configured with four rounds, was applied to the 'Lena' image as illustrated in Figure 12a. The image was converted into a 512×512 matrix, with each pixel represented using 8 bits. To align with the cipher's 152-bit block size, bit-padding with 18 one bits was applied. The padded data were then encrypted, and the resulting cipher image is presented in Figure 12b. After decryption, the 18 padding bits were removed from the last block, thereby restoring the original 512×512 matrix. The decrypted image is shown in Figure 12c. The FPGA-implemented block cipher required 6.09 s for encryption and 51.76 s for decryption.

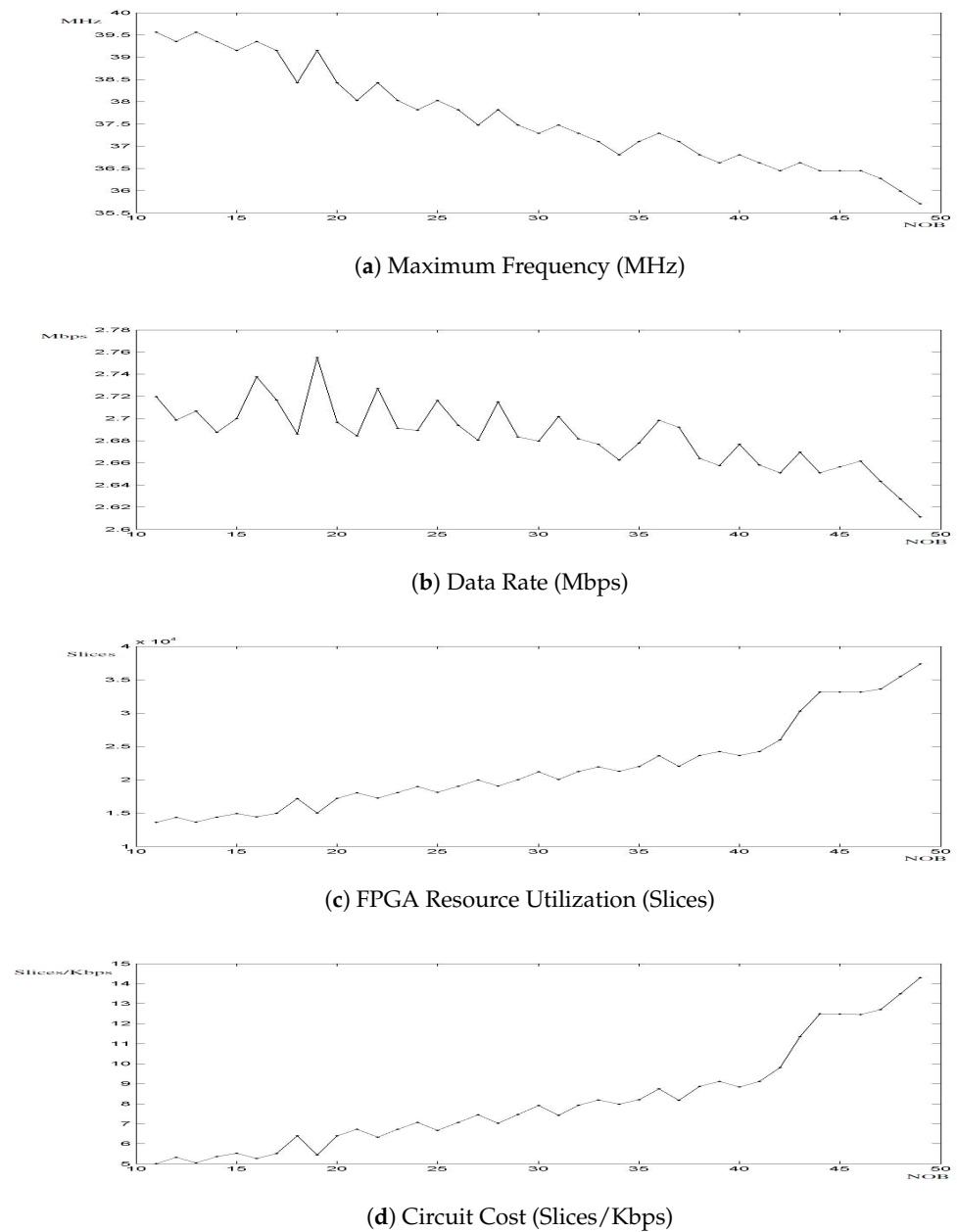


Figure 11. FPGA performance of the block cipher.

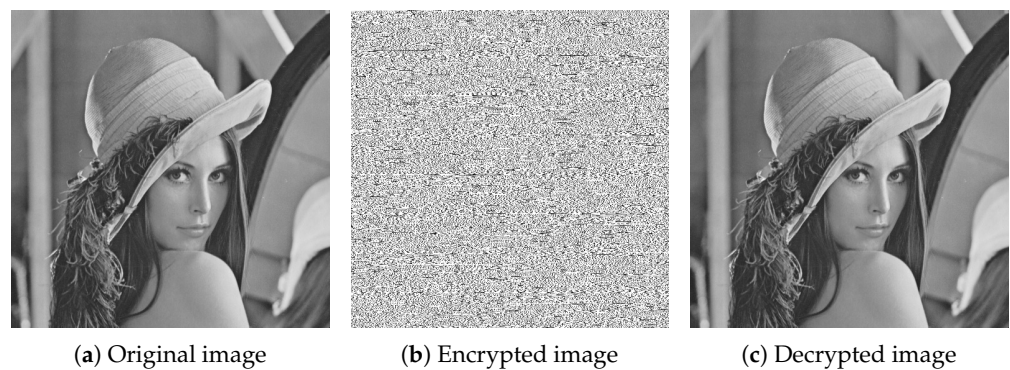


Figure 12. Results of applying the 152-bit recursive block cipher to the 'Lena' image.

Besides image encryption, the 152-bit block cipher described in this section, employing the proposed CB-SBox, was also applied to text encryption. The abstract of this work,

shown in the upper part of Figure 13, was converted into a 1691-bit character string, padded with 14 one bits, and used as the plaintext. The corresponding ciphertext is shown in the lower part of Figure 13. Since the plaintext is based on a LaTeX text fragment, it consists solely of basic ASCII characters, corresponding to codes ranging from 0 to 127, even after one-bit padding, and is represented using 8-bit encoding, which accommodates values from 0 to 255. In contrast, the ciphertext utilizes the full 8-bit ASCII range and, therefore, includes special characters with codes between 128 and 255. As some special characters cannot be properly represented graphically, a hexadecimal representation is employed as illustrated in the lower part of Figure 13. This behavior is further observed in Figure 14, which presents the plaintext histogram (Figure 14a) ranging from 0 to 127, the flattened ciphertext histogram (Figure 14b) ranging from 0 to 255, and the deciphered text histogram (Figure 14c) ranging from 0 to 127 but exhibiting a small peak in the left due to the one-bit padding, which does not appear in the plaintext histogram (Figure 14a) as expected.

Plaintext:

\abstract{In recent years, many chaos-based encryption algorithms have been proposed. Many of these are based on established designs and populate their S-boxes with values derived from chaotic maps, following conventional implementation strategies to enable comparison with their original non-chaotic counterparts. In contrast, this work proposes a novel approach: a \mbox{Chaos-Based} Substitution Box (CB-SBox) implementation, in which conventional \mbox{ROM-based} S-boxes are replaced by a digital circuit that directly executes a selected chaotic map. This method enables the construction of \mbox{S-boxes} with long word lengths through an FPGA-based programmable circuit that allows for variable S-box lengths, facilitating the analysis of S-boxes of varying sizes, and ultimately enhancing security, particularly for larger S-boxes, as demonstrated by increased resistance to linear and differential cryptanalysis. Furthermore, the proposed CB-SBox achieves reductions in both area and power consumption compared to size-comparable \mbox{ROM-based} \mbox{S-boxes}. A 19-bit \mbox{chaos-based} \mbox{S-box} consumes just \mbox{0.0238\%} of the area and \mbox{0.0241\%} of the power required by an equivalent \mbox{ROM-implemented} S-box while providing the same level of security. The inherent unpredictability of non-linear chaotic behavior causes the proposed chaos-based S-boxes to exhibit non-bijective characteristics, making them well suited for application in non-invertible cryptographic primitives, such as hash functions and Feistel networks. The proposed CB-SBox was implemented in a Feistel network as described in the literature, and the results are provided.}

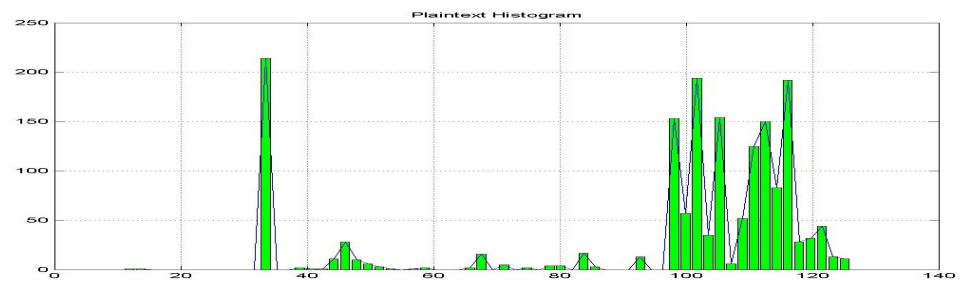
Ciphertext:

19 CD 01 2D 5F 79 EF C9 93 84 C5 FE 7F 7D 6A 99 99 81 D1 09 55 20 70 46 B8 49 73 BA EF 94 08 10 E9 24 D5 61 3F 52 65 B0 67 3F 37 57 59 6E B7 69 ED 4B AC 55 7F CB 70 64 45
12 89 D9 E6 FF 0A B4 53 33 9C EE E9 51 BF 1E DC CC C8 25 FB 54 55 32 B4 B2 87 54 12 29 EC 45 77 DD 88 23 BB 3D A5 DE 50 D8 32 26 69 15 14 55 2D BA EA 00 98 7F 9D
4B BB FB 4C 21 A1 34 B0 82 DC D1 56 C3 6F 17 88 B9 7C 03 C3 E6 D7 63 47 8B 99 11 37 00 B5 51 7C B0 0D BF 3C C6 1F CF 2F 1C EB 28 82 F8 AD F7 53 89 5E 79 26 B2 EA 28 7A
25 E9 08 33 9F 7B EC 4B 35 6F 75 0A 54 55 F2 41 FC EC 24 A7 BF E2 B8 AB 5A 6F 2D 2F 8C 32 44 05 44 E6 96 0D 6C 9D C8 EC 07 98 D2 F4 4C 17 EE 09 03 3B A7 78 03 B7 A1 7A
6C 4D A6 E0 E4 FB 6F 31 64 10 4D FF 86 DF 88 1F F5 5E F5 36 79 03 C1 9B 33 72 B5 F5 67 C3 1A 1C 46 79 09 B8 A3 3C 73 84 B8 48 ED 85 70 6E 36 1F 9B 9B 4E 6D 44 A8 3C 42
B8 E6 8D 48 5B 57 73 EA 62 3E 26 84 91 D6 6B 1D 7F 73 63 89 F9 60 01 C7 42 98 88 05 B7 D5 E1 7C D4 1C A2 14 C1 2C B3 75 31 7F F2 D7 3D 20 B8 68 8C 0F 4E 5B 94 2F
C7 38 10 3D AE D0 24 BE 30 82 01 5D 70 50 E1 8C EA 12 19 15 7B 08 A0 BF BC 47 80 9F 07 C3 BB 58 93 33 51 46 B7 B8 15 E9 66 89 F8 96 C7 F2 C4 87 27 3F EB 36 F3 C9 3D FB 57
2D 8F 34 B7 EC 82 BB EB 15 9C EA 33 7F AC C9 77 52 30 34 70 D3 8B 0A C6 5D 26 7E 92 BD 9F ED 9F F5 33 34 DD 9E A0 BA 0A 0D 35 19 22 D3 0D 7A 18 D0 DD 14 F0 45
36 15 6F 3F 23 C0 DA 2A BD 3A BC EC E7 74 88 C2 3D E6 D0 7A D2 16 27 3E 46 EC 3E E5 81 E8 B3 55 0C 1A 2F A1 72 D8 29 EB 92 26 C8 8C 72 64 EB BE 92 BB C6 B3 EE 16
3C BD FB AB 05 54 D7 1A 4F 5F EF 49 8A F9 3B 61 ED FD 84 48 37 9B 26 99 1D 8B D3 83 B8 D7 48 D9 2A 84 4D 20 D6 08 2D 8B 39 E6 67 25 3E 65 BE 1C AB 7D 78 8E F9 83 B3
2B 55 07 53 92 CB 53 99 46 DC AE D5 D3 DA 14 4E 9E C5 16 04 00 A5 20 67 0B 85 B8 B4 4E CB E6 2C 13 47 5E 00 3E C9 EC 31 43 D4 5B 74 32 54 6A 1D 5D 81 BC D7 85 8A CD
B6 FB 47 7A E8 71 44 34 E8 82 3A 15 CB C1 F6 14 26 81 55 1D 98 0C D3 29 AB C6 7E 5B D6 P9 B5 C9 A9 DC 6E 23 F3 1C 77 DD 2A 8F 79 14 D6 C6 D6 3D F4 DA D1 7C F2 4C
58 7F 48 F2 0C 0D 8A 17 DE 92 A0 6F FC 21 43 CC C1 3E 95 BD 15 D1 76 B4 78 20 62 40 4E 8D 11 8C 43 5C 2B 50 41 04 9F 40 FF B6 A1 5F 58 93 92 5B 43 87 FB 2D 64 B0 56 68 E2
BB 68 48 BA A4 90 48 92 0C 00 A8 54 86 29 4F F7 E4 56 72 F3 FB 55 C9 E4 84 5A 1F AE DE 20 B9 71 1D 1D E2 A2 63 D0 0F 3A E4 E9 F7 CC 4C C6 67 71 4A 34 5D AF 9C
D8 8C B5 C6 FC P9 88 99 25 37 FF 26 1E 49 29 10 D4 D4 70 5D 67 79 C5 10 1D 61 5A C8 1F 08 1E 31 80 0F 07 7A FD 10 79 F3 D8 C7 58 E5 6B D3 42 45 97 7E C2 6F 60 CB 20 DB
69 ED CE 0B 7A 64 71 D4 62 21 86 03 7D 16 5E 91 89 CB 44 DE C1 69 91 3D A3 15 3E D8 2D E8 71 3B B8 09 36 F3 07 22 24 BE A6 70 CE AF 96 3B DE 65 B8 FE FE 77 51 79 30
99 4C 3E 24 F6 A8 3D B5 C0 DC 91 64 70 B4 0F 57 06 60 B4 AE EE A9 1A BB B5 2A D2 7E E3 F1 5C C2 D5 FE 23 11 A4 C1 E7 83 99 46 84 01 B2 FC AE 59 0E 7D 22 08 09 44 99
76 26 A1 0B B5 F5 C2 B7 04 9D D3 5B A4 72 60 FC 4A 18 A0 94 F3 66 21 B0 3B 46 2B 03 03 11 70 A3 6E 03 9B 17 21 51 A1 A4 89 F4 EB FA D4 A8 B5 1B 18 77 98 8F 3A 2F FF 12
BF 9D 10 C2 E3 93 22 DE 1A 85 26 EA 03 BE C0 A5 D7 D2 EC DB A1 22 F6 88 8A 33 85 A0 F4 B8 C3 6B AF CD 61 18 9C 40 82 90 EC 77 46 C7 CC 05 A3 05 A4 2A F9 9A 07
68 74 0C 5E 44 39 0A 46 FF 9B BD CD 38 49 5A 75 D0 2C 31 10 1C 10 61 D6 38 35 60 81 87 CC E6 5E 13 B0 2E D5 EC 7 B6 BE 3B 20 27 F9 80 B9 29 8B 4F FC 44 F5 5D B9 11 05
B5 DD 21 DD FF 0A 43 B5 5A 52 8C 0F 99 87 C1 CA 79 D9 B5 3D 5A 3A 16 51 F6 5E 5C 2A C9 A8 13 99 FE 14 A7 2E DD 31 32 38 4A D8 8A A4 B1 AE 13 6A 2F 88 7C E4 3B 65
62 39 9C 25 8B DD 13 A4 0C 63 07 22 3B 3D 7C A1 7C AA 85 B0 98 22 99 43 10 66 F7 16 72 57 2F E7 45 A0 E0 03 85 1E 40 40 CC 3A B1 5A 86 10 14 D6 4B 1E 04 D0 6D 3F 6F
A7 CF 98 4D C5 86 E6 CA EA 4E E2 5A 46 B3 1B 70 6D FE 09 2D DE 32 45 95 CF F1 88 A8 64 07 87 EA 37 ED 8D 46 1A 5D CE 42 94 4F 16 F5 18 92 9C A9 92 BC C8 F3 93 80 E5
4F 56 93 08 EB F2 25 94 9C C4 6B 87 F7 99 FB 72 C4 BA D8 A8 C3 4E 60 ED B2 9C F3 F7 20 73 E6 BB 6F 87 7D 4A AE C7 A3 C6 25 02 EE 67 FC 7F 1F 78 32 AB 4A 78 53 0C 98
7C C7 A8 27 79 E2 70 33 31 FD 57 66 00 85 AB F3 CC A7 4F 13 3A 9C D1 78 01 13 C0 A1 F0 63 C4 98 0A 3B 56 CC 7F B3 48 85 F1 44 04 05 0E F0 83 8C 6A AF 78 B4 1F 4C DC 61
4C BA D2 7B 01 0D E5 1A FB C7 D9 0F 90 13 B6 E8 21 02 DA 28 FC 22 EC C2 CA 36 6F 69 C9 D7 FE 98 F5 06 25 EA 92 CF 4C 0A 10 95 02 C8 1D 75 D8 D9 FB C9 4D 1C 1A 43
C6 C1 86 21 85 7A 80 6C 36 DB 7F 34 E5 B9 8F 62 08 6E 76 BB 45 53 CD 62 71 46 8E 6E C4 1A D6 52 9D 80 65 67 11 36 23 FD D1 8B 9A 01 62 72 7D C7 31 F7 E6 52 F7 02 F3 12
A5 F6 E4 7F A3 B6 BE 1D C7 6B 2D 1B 39 D5 EC C6 F6 23 BB D9 0B E0 04 AB 8D 15 BC 46 21 42 FC 1F 09 3A 28 02 42 D4 FC 55 3A B8 E9 C2 84 66 7F CF 4E F6 07 F2 13 1E
65 12 D4 19 A7 30 AD 63 2E 92 13 3A 1F 9F B7 94 03 41 28 9A 6A CA 9F B7 92 F0 F7 4D 1C 1F 25 16 7B 3E 3D 60 18 27 F1 69 B9 11 C3 13 29 C2 92 38 01 F6 E3 33 D4 37 83 92 B5
EA 6E 71 6F 62 28 D8 41 BE C5 A8 13 97 6F 68 32 AB 4C 44 95 6C 2A 33 3C 7B EA C0 AE E1 2C B6 60 B3 11 B9 68 F2 72 36 F3 D5 9E A7 16 2C DB 1B EB DE EA 36 63 19 16 7B
8A FE 20 6A 5E 0B 7E FC 2E 8C 56 FD 90 B8 59 4E BA EA 86 D5 2B 95 09 7C A2

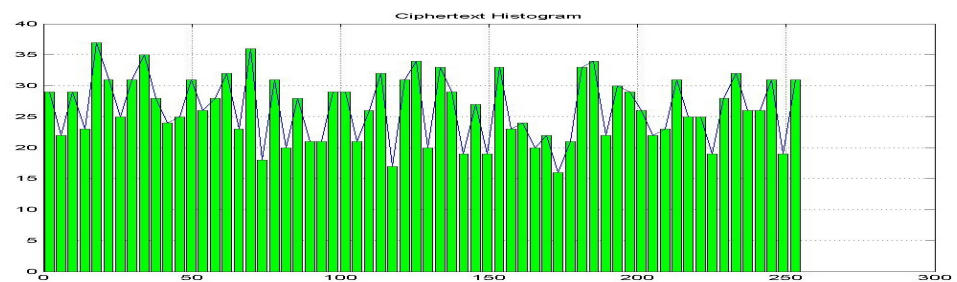
Figure 13. Original and encrypted versions of the LaTeX text presenting the abstract of this work, encrypted using the 152-bit recursive block cipher based on the proposed CB-SBox. The corresponding dataset is available online [59].

The following hardware and software resources were used in the present research: a ThinkPad E14 Gen2 computer manufactured by Lenovo in Brazil, with an 11th Generation Intel® Core™ i5-1135G7 @ 2.40 GHz processor (base frequency 2.42 GHz), featuring 4 physical cores and 8 logical processors, 8 GB of RAM, and 128 MB of video memory, running Windows 11 Pro 64-bit, version 22H2. A virtual machine was also used with the following specifications: Oracle VM VirtualBox 7.0 (version 7.0.18 r162988), with 2 GB of RAM allocated from the 8 GB available on the host system, 32 MB of video memory allocated from the 128 MB available, and 2 CPU cores allocated from the 8 logical cores

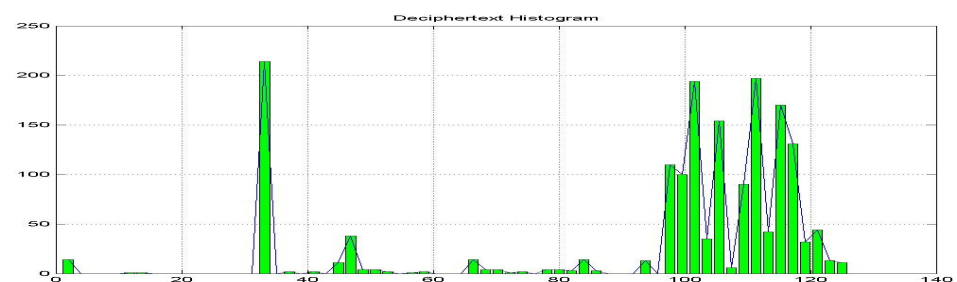
available. The virtual machine runs the following software: Windows XP (32-bit), v5.1, build 2600, Service Pack 1 as the operating system; MATLAB version 6.5.0.180913a (Release 13); Xilinx ISE Design Suite v12, P/N 1050504-01, Xilinx Project Navigator v6.2i; Xilinx System Generator Blockset version v6.1(c); and ModelSim XE 6.2i. A Xilinx Spartan-3 development board manufactured by Avnet in USA, P/N ADS-XLX-SP3-DEV2000, was also used featuring the following main components: Xilinx XC3S1500/2000-FG676 Spartan-3 FPGA, 43 MB Micron DDR SDRAM, 16 MB Flash, 2 MB SRAM, a 2×16 LCD, a 128×64 OSRAM OLED display, and an audio codec.



(a) Plaintext histogram



(b) Ciphertext histogram



(c) Deciphered text histogram

Figure 14. Histograms after applying the 152-bit encryption process, which employs the proposed CB-SBox, to the abstract of this paper, written in LaTeX. The corresponding dataset is available online [59].

6. Conclusions

This research presents a new approach for the design of block ciphers. It is based on the fixed-point realization of a non-linear differential equation exhibiting chaotic behavior. This realization, called Chaos-Based Substitution Box (CB-SBox), enables the implementation of large S-boxes. Such entities bring significant advantages in terms of area and power consumption when compared to a typical ROM-based s-box: a 19-bit long CB-SBox requires only 0.0238% of the area and 0.0241% of the power used by an equivalent but non-feasible ROM-based implementation. They also achieve a high level of security, measured by the

time necessary to compute the first step of linear and differential cryptanalysis, a task that cannot be achieved by their ROM-based counterparts.

As a consequence of the butterfly effect, this approach has the disadvantage of a long iteration time (governed by the LongRun parameter) required to uncorrelate the CB-SBox input signals (the chaotic map's initial condition) from its output (the chaotic map's final states).

The CB-SBox was used as the source of confusion in a block cipher based on the architecture proposed by Jakimoski [30]. This cipher was implemented using a programmable circuit where the CB-SBox can be realized with different sizes, according to the parameters stored in the PRM and Slice datasets (available online in [59]). This enables a block cipher with a programmable key size. We analyzed the performance of the block cipher as a function of the size (NOB) of its CB-SBox, deciding to operate at $\text{NOB} = 19$ as a solution to globally maximize the circuit data rate while keeping the circuit size low and achieving high levels of security.

To conclude this work, six open research areas have been identified and are discussed below. When a given group of initial conditions (ICs), belonging to the volume of valid initial conditions (V_{IC}), is stimulated by a chaotic map, it disperses into the volume of the strange attractor \mathcal{A} due to the butterfly effect. The temporal evolution pattern of the intermediate states may lead to the misleading conclusion that, depending on the relative position of this group of initial conditions, more time would be required for it to fully disperse throughout the attractor. The results presented in Figure 7 suggest an inconclusive relationship, making the correlation between the position of a group of ICs *versus* dispersion time a promising topic for future investigation.

Another potential research direction involves the implementation of a programmable Chaos-Based Substitution Box, in which not only the S-box length (NOB) can be configured—as demonstrated in the present CB-SBox—but also the chaotic map itself can be dynamically altered. This capability would enhance the adaptability and security level of the target encryption primitive.

In this study, the Sprott B chaotic map [5] was selected among six other candidates, based on the criteria provided in [51]. The computation of chaotic properties such as V_{IC} is highly time-consuming but is worthwhile since it not only allows for a better use of the chaos map in the encryption process but also directly addresses quantization errors and dynamical degradation. A valuable extension of the present work would be to replicate the methodology of [51] using other chaotic systems, including hyperchaotic maps, higher-dimensional systems, systems exhibiting multiple attractors, and those with greater ergodicity as found in the literature [29,37,39,62,63]. It is expected that a chaotic map with improved ergodicity could reduce the LongRun parameter, thus increasing encryption throughput, and can also flatten the histogram plot of Figure 14b, combating a potential weakness of the proposed C-SBox: its capability to better spread the final states over all state space.

Non-bijective S-boxes, like the proposed CB-SBox, are suitable only for specific types of cryptographic primitives, such as the Feistel network implemented in this work. An additional future direction includes the usage of the CB-SBox concepts in the design and evaluation of hash functions—non-invertible primitives—along with their implementation in ASICs.

This research also examined the CB-SBox's resistance to linear and differential cryptanalysis, demonstrating that large S-box sizes contribute positively to security. Further work could involve exploring alternative attack vectors found in the literature, as well as proposing novel attack strategies tailored to the developed CB-SBox architecture.

In the present work, the primary focus is the comparison between the proposed chaos-based S-box (CB-SBox) and a ROM-based implementation. Other approaches to implementing S-boxes, such as logic gate designs, are also available. Although such implementations typically focus on small S-boxes, there is no fundamental impediment to implementing larger ones and comparing them with the CB-SBox, which is suggested as future work.

Author Contributions: Conceptualization, É.C.D.e.S.J., C.A.d.M.C., L.S.I., W.A.F. and M.G.; data curation, É.C.D.e.S.J. and I.A.L.S.; formal analysis, É.C.D.e.S.J., C.A.d.M.C. and W.A.F.; funding acquisition, É.C.D.e.S.J., L.S.I. and M.G.; investigation, É.C.D.e.S.J.; methodology, É.C.D.e.S.J., W.A.F. and C.A.d.M.C.; project administration, É.C.D.e.S.J., C.R.P.d.S.J. and C.A.d.M.C.; resources, É.C.D.e.S.J., F.G.S., C.R.P.d.S.J., L.S.I. and I.A.L.S.; software, É.C.D.e.S.J., F.G.S., L.S.I. and I.A.L.S.; supervision, C.A.d.M.C.; validation, É.C.D.e.S.J. and C.A.d.M.C.; visualization, É.C.D.e.S.J.; writing—original draft preparation, É.C.D.e.S.J.; writing—review and editing, É.C.D.e.S.J. and C.A.d.M.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the SENAI Institute of Innovation in Microelectronics (ISI-ME) and the National Service for Industrial Apprenticeship (SENAI), which covered the article processing charges (APC) and other associated research costs. Partial funding was also provided by the Commission for Coordination and Implementation of the Amazon Surveillance System (SIVAM), which supported travel expenses, and by the Microelectronic Systems (MES) Research Group at Technische Universität Darmstadt, which provided software and hardware resources.

Data Availability Statement: The original data presented in the study are openly available in Zenodo research repository at <https://zenodo.org/records/14967851>, (accessed on 7 May 2025) [59].

Acknowledgments: The authors would like to thank the following institutions and their prominent members for supporting this research in a broad and collaborative context: the Federation of Industries of the State of Amazonas (FIEAM); the SENAI Institute of Innovation in Microelectronics (ISI-ME), the Superintendency of Innovation and Technology (SITEC), the National Directorate (DN) and Amazonas Regional Directorate (DR-AM) of SENAI – the National Service for Industrial Training; the Center for R&D in Electronic and Information Technology (CETELI) and the Graduate Program in Electrical Engineering (PPGEE) at the Federal University of Amazonas (UFAM); the Coordination for the Improvement of Higher Education Personnel (CAPES), the National Council for Scientific and Technological Development (CNPq), and the Amazonas State Research Support Foundation (FAPEAM), Brazilian funding agencies for science, technology, and academic development; the Brazilian Society of Microelectronics (SBMicro); the Microelectronic Systems (MES) Research Group at Technische Universität Darmstadt (TUD); Rohde & Schwarz; the University of Leeds (UoL); the Center for Telecommunication Studies (CETUC) at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio); the Commission for the Coordination and Implementation of the Amazon Surveillance System (SIVAM); the Aeronautics Institute of Technology (ITA); and the Institute for Advanced Studies (IEAv).

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

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---------|---|
| ASIC | Application-Specific Integrated Circuit |
| CB-SBox | Chaos-Based Substitution Box |
| DC | Discretization Criterion |
| DTR | Distance-to-Time Ratio |
| FPGA | Field-Programmable Gate Array |
| IC/ICs | Initial Condition/Initial Conditions |
| LUT | Look-Up Table |

| | |
|---------------------|--|
| LongRun | Execution time required for the outputs to fully disperse over \mathcal{A} |
| Mbps | Megabits per second |
| MBps | Megabytes per second |
| MDT | Maximum Dispersion Time |
| NOB | Number of Bits |
| PRM | Parallelepiped Relations Measurements (used as a configuration dataset) |
| ROM | Read Only Memory |
| S-Box | Substitution Box |
| Slice | Parameters for segmentation and selection (used as a configuration dataset) |
| SIVAM | Amazon Surveillance System |
| VHDL | VHSIC Hardware Description Language |
| VHSIC | Very High Speed Integrated Circuit |
| \mathcal{A} | Strange Attractor |
| \mathcal{A}_{all} | All Attractors |
| \mathcal{A}_{cv} | Convergent Attractors |
| \mathcal{A}_{dv} | Divergent Attractors |
| $ave(\cdot)$ | Mean function |
| $BP_{\mathcal{A}}$ | Bounding Parallelepiped |
| $d(\cdot)/dt$ | Temporal derivative function |
| $d_{\mathcal{A}}$ | Attractor Density |
| $d_{V_{IC}}$ | Depth dimension of the largest parallelepiped within V_{IC} |
| ds | Distance Small |
| $g(\cdot)$ | Function that relates ds to the time to spread beyond the V_{Hist} threshold |
| $h_{V_{IC}}$ | Height dimension of the largest parallelepiped within V_{IC} |
| Lat | latency parameter |
| $max(\cdot)$ | Maximum function |
| $min(\cdot)$ | Minimum function |
| sp | Smaller Parallelepiped |
| SS | Set of ICs belonging to a state space |
| st | step size integration parameter |
| $Var(\cdot)$ | Variation function |
| V_{Hist} | Volumetric Histogram |
| V_{IC} | Volume valid of Initial Conditions |
| $Vol(\cdot)$ | Volume function |
| V_{Sub} | Largest Subset Volume |
| $w_{V_{IC}}$ | Width dimension of the largest parallelepiped within V_{IC} |
| $ \cdot $ | Cardinality function of sets |
| $[\cdot]^c$ | Complement operation of sets |
| \cup | Union operator of sets |

References

1. Lorenz, N.E. Deterministic non-periodic flows. *J. Atmos. Sci.* **1963**, *20*, 130–141. [\[CrossRef\]](#)
2. Rössler, O.E. An equation for continuous chaos. *Phys. Lett. A* **1976**, *57*, 397–398. [\[CrossRef\]](#)
3. May, R. Simple mathematical models with very complicated dynamics. *Nature* **1976**, *261*, 459–467. [\[CrossRef\]](#)
4. Sprott, J.; Linz, S. Algebraically simple chaotic flows. *Int. J. Chaos Theory Appl.* **2000**, *5*, 1–20.
5. Sprott, J. Some simple chaotic flows. *Phys. Rev. E* **1994**, *50*, R647–R650. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Chua, L. *The Genesis of Chua's Circuit*; Electronics Research Laboratory, College of Engineering, University of California: Los Angeles, CA, USA, 1992.
7. Linz, S.; Sprott, J. Elementary chaotic flow. *Phys. Lett. A* **1999**, *259*, 240–245. [\[CrossRef\]](#)
8. Li, S.; Zheng, X.; Mou, X.; Cai, Y. Chaotic encryption scheme for real-time digital video. In *Real-Time Imaging VI, Proceedings of SPIE*; Citeseer: Princeton, NJ, USA, 2002; Volume 4666, pp. 149–160.
9. Tenny, R.; Tsimring, L. Additive mixing modulation for public key encryption based on distributed dynamics. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2005**, *52*, 672–679. [\[CrossRef\]](#)

10. Bose, R.; Pathak, S. A novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2006**, *53*, 848–857. [\[CrossRef\]](#)
11. Zou, C.; Zhang, Q.; Wei, X.; Liu, C. Image Encryption Based on Improved Lorenz System. *IEEE Access* **2020**, *8*, 75728–75740. [\[CrossRef\]](#)
12. Khan, M.F.; Ahmed, A.; Saleem, K.; Shah, T. A Novel Design of Cryptographic SP-Network Based on Gold Sequences and Chaotic Logistic Tent System. *IEEE Access* **2019**, *7*, 84980–84991. [\[CrossRef\]](#)
13. Feng, W.; Zhang, J.; Qin, Z. A Secure and Efficient Image Transmission Scheme Based on Two Chaotic Maps. *Complexity* **2021**, *2021*, 1898998. [\[CrossRef\]](#)
14. Feng, W.; Wang, Q.; Liu, H.; Ren, Y.; Zhang, J.; Zhang, S.; Qian, K.; Wen, H. Exploiting Newly Designed Fractional-Order 3D Lorenz Chaotic System and 2D Discrete Polynomial Hyper-Chaotic Map for High-Performance Multi-Image Encryption. *Fractal Fract.* **2023**, *7*, 30. [\[CrossRef\]](#)
15. Feng, W.; Zhang, J.; Chen, Y.; Qin, Z.; Zhang, Y.; Ahmad, M.; Woźniak, M. Exploiting robust quadratic polynomial hyperchaotic map and pixel fusion strategy for efficient image encryption. *Expert Syst. Appl.* **2024**, *246*, 123190. [\[CrossRef\]](#)
16. Feng, W.; Yang, J.; Zhao, X.; Qin, Z.; Zhang, J.; Zhu, Z.; Wen, H.; Qian, K. A Novel Multi-Channel Image Encryption Algorithm Leveraging Pixel Reorganization and Hyperchaotic Maps. *Mathematics* **2024**, *12*, 27. [\[CrossRef\]](#)
17. Ma, X.; Wang, Z.; Wang, C. An Image Encryption Algorithm Based on Tabu Search and Hyperchaos. *Int. J. Bifurc. Chaos* **2024**, *34*, 2450170. [\[CrossRef\]](#)
18. Qian, K.; Xiao, Y.; Wei, Y.; Liu, D.; Wang, Q.; Feng, W. A Robust Memristor-Enhanced Polynomial Hyper-Chaotic Map and Its Multi-Channel Image Encryption Application. *Micromachines* **2023**, *14*, 2090. [\[CrossRef\]](#)
19. Yu, F.; Zhang, S.; Su, D.; Wu, Y.; Gracia, Y.M.; Yin, H. Dynamic Analysis and Implementation of FPGA for a New 4D Fractional-Order Memristive Hopfield Neural Network. *Fractal Fract.* **2025**, *9*, 115. [\[CrossRef\]](#)
20. Baptista, M. Cryptography with chaos. *Phys. Lett. A* **1998**, *240*, 50–54. [\[CrossRef\]](#)
21. Wong, K.; Yuen, C. Embedding compression in chaos-based cryptography. *IEEE Trans. Circuits Syst. II Express Briefs* **2008**, *55*, 1193–1197. [\[CrossRef\]](#)
22. Chen, J.; Zhou, J.; Wong, K. A Modified Chaos-Based Joint Compression and Encryption Scheme. *IEEE Trans. Circuits Syst. II Express Briefs* **2011**, *58*, 110–114. [\[CrossRef\]](#)
23. Alvarez, E.; Fernandez, A.; Garcia, P.; Jiménez, J.; Marcano, A. New approach to chaotic encryption. *Phys. Lett. A* **1999**, *263*, 373–375. [\[CrossRef\]](#)
24. Wong, K. A fast chaotic cryptographic scheme with dynamic look-up table. *Phys. Lett. A* **2002**, *298*, 238–242. [\[CrossRef\]](#)
25. Wong, K.; Ho, S.; Yung, C. A chaotic cryptography scheme for generating short ciphertext. *Phys. Lett. A* **2003**, *310*, 67–73. [\[CrossRef\]](#)
26. Wong, K. A combined chaotic cryptographic and hashing scheme. *Phys. Lett. A* **2003**, *307*, 292–298. [\[CrossRef\]](#)
27. Liao, X.; Wong, K. Improving the security of a dynamic look-up table based chaotic cryptosystem. *IEEE Trans. Circuits Syst. II Express Briefs* **2006**, *53*, 502–506.
28. Xiang, T.; Liao, X.; Tang, G.; Chen, Y.; Wong, K. A novel block cryptosystem based on iterating a chaotic map. *Phys. Lett. A* **2006**, *349*, 109–115. [\[CrossRef\]](#)
29. Zhang, L.; Ma, C.; Zhao, Y.; Zhao, W. A Novel Dynamic S-Box Generation Scheme Based on Quantum Random Walks Controlled by a Hyper-Chaotic Map. *Mathematics* **2023**, *12*, 84. [\[CrossRef\]](#)
30. Jakimoski, G.; Kocarev, L. Chaos and cryptography: Block encryption ciphers based on chaotic maps. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2001**, *48*, 163–169. [\[CrossRef\]](#)
31. Peng, J.; Zhang, D.; Liao, X. A Novel Approach for Designing Dynamical S-Boxes Using Hyperchaotic System. *Int. J. Cogn. Inform. Nat. Intell.* **2012**, *6*, 100–119. [\[CrossRef\]](#)
32. Guesmi, R.; Ben Farah, M.A.; Kachouri, A.; Samet, M. 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), Doha, Qatar, 10–13 November 2014; pp. 678–684. [\[CrossRef\]](#)
33. Tang, G.; Liao, X. A method for designing dynamical S-boxes based on discretized chaotic map. *Chaos Solitons Fractals* **2005**, *23*, 1901–1909. [\[CrossRef\]](#)
34. Ibrahim, S.; Abbas, A.M.; Alharbi, A.A.; Albahar, M.A. A New 12-Bit Chaotic Image Encryption Scheme Using a 12×12 Dynamic S-Box. *IEEE Access* **2024**, *12*, 37631–37642. [\[CrossRef\]](#)
35. Al-Maadeed, T.A.; Hussain, I.; Anees, A.; Mustafa, M.T. A image encryption algorithm based on chaotic Lorenz system and novel primitive polynomial S-boxes. *Multimed. Tools Appl.* **2021**, *80*, 24801–24822. [\[CrossRef\]](#)
36. Ahmad, M.; Al-Solami, E.; Alghamdi, A.M.; Yousaf, M.A. Bijective S-Boxes Method Using Improved Chaotic Map-Based Heuristic Search and Algebraic Group Structures. *IEEE Access* **2020**, *8*, 110397–110411. [\[CrossRef\]](#)
37. Nazir, H.; Bajwa, I.S.; Abdullah, S.; Kazmi, R.; Samiullah, M. A Color Image Encryption Scheme Combining Hyperchaos and Genetic Codes. *IEEE Access* **2022**, *10*, 14480–14495. [\[CrossRef\]](#)

38. Manzoor, A.; Zahid, A.H.; Hassan, M.T. A New Dynamic Substitution Box for Data Security Using an Innovative Chaotic Map. *IEEE Access* **2022**, *10*, 74164–74174. [\[CrossRef\]](#)
39. Alabdullah, B.; Banga, A.; Iqbal, N.; Ikram, A.; Diab, H. Advancing Cryptographic Security With a New Delannoy-Derived Chaotic S-Box. *IEEE Access* **2024**, *12*, 82926–82937. [\[CrossRef\]](#)
40. Goswami, S.S.P.; Trivedi, G. FPGA Implementation of Modified SNOW 3G Stream Ciphers Using Fast and Resource Efficient Substitution Box. *IEEE Embed. Syst. Lett.* **2023**, *15*, 238–241. [\[CrossRef\]](#)
41. Lidong, L.; Jiang, D.; Wang, X.; Zhang, L.; Rong, X. A Dynamic Triple-Image Encryption Scheme Based on Chaos, S-Box and Image Compressing. *IEEE Access* **2020**, *8*, 210382–210399. [\[CrossRef\]](#)
42. Jun, W.J.; Fun, T.S. A New Image Encryption Algorithm Based on Single S-Box and Dynamic Encryption Step. *IEEE Access* **2021**, *9*, 120596–120612. [\[CrossRef\]](#)
43. Ibrahim, S.; Alhumyani, H.; Masud, M.; Alshamrani, S.S.; Cheikhrouhou, O.; Muhammad, G.; Hossain, M.S.; Abbas, A.M. Framework for Efficient Medical Image Encryption Using Dynamic S-Boxes and Chaotic Maps. *IEEE Access* **2020**, *8*, 160433–160449. [\[CrossRef\]](#)
44. Zhang, W.; Pasalic, E. Highly Nonlinear Balanced S-Boxes With Good Differential Properties. *IEEE Trans. Inf. Theory* **2014**, *60*, 7970–7979. [\[CrossRef\]](#)
45. Piret, G.; Roche, T.; Carlet, C. PICARO—A Block Cipher Allowing Efficient Higher-Order Side-Channel Resistance. In *Applied Cryptography and Network Security, 10th International Conference, Singapore, 26–29 June 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 311–328. [\[CrossRef\]](#)
46. Jassim, S.; Farhan, A. Designing a Novel Efficient Substitution-Box by Using a Flower Pollination Algorithm and Chaos System. *Int. J. Intell. Eng. Syst.* **2022**, *15*, 176–187. [\[CrossRef\]](#)
47. Zhu, S.; Wang, G.; Zhu, C. A Secure and Fast Image Encryption Scheme Based on Double Chaotic S-Boxes. *Entropy* **2019**, *21*, 790. [\[CrossRef\]](#)
48. Shannon, C. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [\[CrossRef\]](#)
49. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos Appl. Sci. Eng.* **2006**, *16*, 2129–2151.
50. Dutra e Silva Junior, E.C.; Finamore, W.A.; Glesner, M.; Indrusiak, L.S.; Zipf, P. Chaotic Equations Initial Conditions Analysis for Cryptography Applications. In *Proceedings of the XXVII Simpósio Brasileiro de Telecomunicações (SBrT)*, Blumenau, Brazil, 29 September–2 October 2009. [\[CrossRef\]](#)
51. Dutra, E.; Glesner, M.; Finamore, W.; Indrusiak, L. Novel method of chaotic systems evaluation for implementations of encryption algorithms. In *Proceedings of the 2010 17th International Conference on Telecommunications*, Doha, Qatar, 4–7 April 2010; pp. 89–96. [\[CrossRef\]](#)
52. Li, S. Analyses and New Designs of Digital Chaotic Ciphers. Ph.D. Thesis, Xi'an Jiaotong University, Xi'an, China, 2003.
53. Li, Z.; Li, K.; Wen, C.; Soh, Y. A new chaotic secure communication system. *IEEE Trans. Commun.* **2003**, *51*, 1306–1312.
54. Li, S.; Mou, X.; Ji, Z.; Zhang, J.; Cai, Y. Performance analysis of Jakimoski–Kocarev attack on a class of chaotic cryptosystems. *Phys. Lett. A* **2003**, *309*, 165. [\[CrossRef\]](#)
55. Li, S.; Mou, X.; Cai, Y.; Ji, Z.; Zhang, J. On the security of a chaotic encryption scheme: Problems with computerized chaos in finite computing precision. *Comput. Phys. Commun.* **2003**, *153*, 52–58. [\[CrossRef\]](#)
56. Heys, H.M. A tutorial on linear and differential cryptanalysis. *XXVI Cryptologia* **2002**, *3*, 189–221. [\[CrossRef\]](#)
57. Ruelle, D.; Takens, F. On the nature of turbulence. *Commun. Math. Phys.* **1971**, *20*, 167–192. [\[CrossRef\]](#)
58. Stewart, I. *Does God Play Dice? The Mathematics of Chaos*; Blackwell Publishers: Hoboken, NJ, USA, 1990; p. 418.
59. Dutra, E.C.e.S.J. *Chaos-Based S-Boxes as a Source of Confusion in Cryptographic Primitives*. Dataset; Zenodo: Genève, Switzerland, 2025. [\[CrossRef\]](#)
60. Matsui, M. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology—EUROCRYPT'93*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 386–397.
61. Biham, E.; Shamir, A. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72. [\[CrossRef\]](#)
62. Leo Kingston, S.; Kapitaniak, T.; Dana, S.K. Transition to hyperchaos: Sudden expansion of attractor and intermittent large-amplitude events in dynamical systems. *Chaos* **2022**, *32*, 081106. [\[CrossRef\]](#) [\[PubMed\]](#)
63. Munyaev, V.O.; Khorkin, D.S.; Bolotov, M.I.; Smirnov, L.A.; Osipov, G.V. Appearance of chaos and hyperchaos in evolving pendulum network. *Chaos Interdiscip. J. Nonlinear Sci.* **2021**, *31*, 063106. [\[CrossRef\]](#) [\[PubMed\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.