Design, Hardware Implementation on FPGA and Performance Analysis of Three Chaos-Based Stream Ciphers

Fethi Dridi, Safwan El Assad, Wajih El Hadj Youssef, Mohsen Machhout

fractal and fractional

MDPI

Article

Design, Hardware Implementation on FPGA and Performance Analysis of Three Chaos-Based Stream Ciphers

Fethi Dridi [1,2], Safwan El Assad [2,*], Wajih El Hadj Youssef [1] and Mohsen Machhout [1]

[1] Electronics and Microelectronics Laboratory (EμE), Faculty of Sciences of Monastir, University of Monastir, Monastir 5019, Tunisia
[2] IETR UMR 6164, CNRS, University Nantes, F-44000 Nantes, France
* Correspondence: safwan.elassad@univ-nantes.fr

Abstract: In this paper, we come up with three secure chaos-based stream ciphers, implemented on an FPGA board, for data confidentiality and integrity. To do so, first, we performed the statistical security and hardware metrics of certain discrete chaotic map models, such as the Logistic, Skew-Tent, PWLCM, 3D-Chebyshev map, and 32-bit LFSR, which are the main components of the proposed chaotic generators. Based on the performance analysis collected from the discrete chaotic maps, we then designed, implemented, and analyzed the performance of three proposed robust pseudo-random number generators of chaotic sequences (PRNGs-CS) and their corresponding stream ciphers. The proposed PRNGs-CS are based on the predefined coupling matrix M. The latter achieves a weak mixing of the chaotic maps and a chaotic multiplexing technique or XOR operator for the output function. Therefore, the randomness of the sequences generated is expanded as well as their lengths, and divide-and-conquer attacks on chaotic systems are avoided. In addition, the proposed PRNGs-CS contain polynomial mappings of at least degree 2 or 3 to make algebraic attacks very difficult. Various experimental results obtained and analysis of performance in opposition to different kinds of numerical and cryptographic attacks determine the high level of security and good hardware metrics achieved by the proposed chaos system. The proposed system outperformed the state-of-the-art works in terms of high-security level and a high throughput which can be considered an alternative to the standard methods.

Keywords: chaos-based stream ciphers; PRNGs-CS; FPGA; security analysis; hardware metrics

Citation: Dridi, F.; El Assad, S.; El Hadj Youssef, W.; Machhout, M. Design, Hardware Implementation on FPGA and Performance Analysis of Three Chaos-Based Stream Ciphers. *Fractal Fract.* **2023**, *7*, 197. https://doi.org/10.3390/fractalfract7020197

Academic Editors: Maria S. Papadopoulou and Christos Volos

Received: 15 January 2023
Revised: 9 February 2023
Accepted: 15 February 2023
Published: 17 February 2023

1. Introduction

As we increasingly rely on intelligent and interconnected devices in every aspect of our lives (residences, transports, hospitals, factories, ...), how do we protect potentially billions of these connected devices from passive and active attacks that could compromise personal privacy or threaten public safety? We all live today in a cyber world. For that, most of the world's web traffic (digital multimedia content such as images and videos, emails) is encrypted because of security threats.

The revised eSTREAM project, September 2008, contains seven stream ciphers that fall into two profiles. Profile 1 contains HC-128, Rabbit, Salsa20/12, and SOSEMANUK stream ciphers, which are best suited for software applications requiring high throughput. Profile 2 contains Grain, MICKEY 2.0, and Trivium stream ciphers, which are particularly suitable for hardware applications with constrained resources and limited power consumption [1,2]. However, these days, most of these stream ciphers are weak against various attacks and then can't be used to secure exchanged and stored data [3]. Likewise, 5G Security uses the NEA3-128 algorithm for privacy and the NIA3-128 algorithm for integrity. Both NEA3-128 and NIA3-128 are based on ZUC stream cipher [4]. However, many published researchers [5,6] noticed that there are some weaknesses and vulnerabilities in the ZUC algorithm for differential power analysis attacks (SCA), Chosen IV attacks, etc.

For this, we use chaos-based stream ciphers, which are very secure against cryptographic attacks because chaotic signals are intrinsically secure and have high nonlinearity [7]. The security of a stream cipher relies heavily on the pseudo-random number generator (PRNG), which acts as the primary component in the system.

There has been a development of various cryptosystems in recent years that employ chaotic systems and maps, including those for the generation of pseudo-random numbers and encryption. Some of them have issues with security or poor computational performance [8–16]. Moreover, some of these cryptosystems are not well-suited for use on smart devices with limited resources, such as low memory, limited processing capabilities, and energy efficiency issues [17–19].

Devices used in the Internet of Things are low-power and resource constrained, so for these devices, hardware implementation of cryptosystems is more suitable than the software implementation. It is noteworthy that currently, there are only a limited number of chaotic systems that have been implemented on Field-Programmable Gate Array (FPGA) boards [20–24]. FPGAs are used for various reasons, including Speed (FPGAs can perform certain operations much faster than microprocessors), customizability, power efficiency, flexibility, and parallelism. A new lightweight chaos-based stream cipher, using two nonlinear feedback shift registers (NLFSR), was proposed by Ding et al. [25] in 2019. The results obtained indicate the good cryptographic characteristics of their stream cipher. To provide security in multimedia applications dealing with high voluminous data, several effective chaos-based encryption systems using four combined chaotic maps (Logistic, Arnold, Lorenz, and Chebyshev) published by Abdelfatah et al. [26] in 2020 and a novel medical image encryption system based on the Logistic-Tent map, Arnold's scrambling, and a special nonlinear function based LFSR was proposed by Subhrajyoti et al. [27] in 2021. Experimental works reveal that the proposed cryptosystems have good prospects for real-time image encryption. In 2022, Jun et al. [28] developed a highly secure stream cipher based on the analog–digital hybrid chaotic system consisting of the Chen chaotic system and a modified three-dimensional Logistic map. The proposed stream cipher has the advantages of huge key space, virtually infinite cycle length, and tight security.

We propose in this research work to design, and implement on an FPGA board, three chaos-based stream ciphers and evaluate their performance, in terms of security against statistical attacks, and cryptographic attacks, as well as in terms of hardware metrics, including throughput and efficiency. The proposed solutions cope with the trade-offs between security, performance, and cost, and have a better level of security compared to standard methods (thanks to the high non-linearity of chaotic systems).

The remainder of this article is organized as follows: In Section 2, we introduce, implement, and evaluate the statistical and hardware performance of some discrete chaotic maps used here, specifically Logistic, Skew-Tent (STM), PWLCM, 3D-Chebyshev (3D-Ch), and a Linear Feedback Shift Register (LFSR) as the basis for the chaos-based stream ciphers design proposed in this work. Based on the performance results of the studied chaotic maps, we designed, implemented, and analyzed in Section 3 three-stream encryption methods that utilize three secure pseudo-chaotic number generators (PRNGs-CS), known as LSP-PRNG, LST-PRNG, and LSPT-PRNG, by combining different chaotic maps such as the Logistic map, STM, PWLCM, and 3D-Ch map. Then, we present the results of the experiments in terms of security evaluation and hardware metrics of the proposed PRNGs-CS and their corresponding stream ciphers. Finally, Section 4 summarizes the whole article.

Notice that, for the proposed chaotic systems, in all statistical experiments, 100 random secret keys are utilized to produce 100 different sequences of 3,125,100 32-bit samples. However, just 3,125,000 samples were utilized per sequence (i.e., 10^8 bits). While the system internally generates the first 100 samples per sequence for each key, they are not employed to deviate from the transitional phase.

2. Study of Used Discrete Chaotic Maps: Statistical Test and Hardware Metrics

In information security, chaotic maps are used in stream ciphers [7,29], block ciphers [30,31], hashing [32], steganography, and digital watermarking [33,34]. They can replace traditional pseudo-random number generators such as maximum length PN sequences, and Gold and Kasami generators. However, the use of these chaotic maps alone in practical data security applications is not secure, as we will see in this paragraph.

In the following, we first consider the discrete equations of the 1-D chaotic maps used (we also give the original equations of these chaotic maps in Appendix A): Logistic, Skew-Tent, PWLCM, and the 3D-Chebyshev map and the primitive polynomial expression of an LFSR of 32-bit. Then we present the statistical security performance (in terms of NIST test, histogram, and Chi-square test) and hardware metrics of these chaotic maps, the 32-bit LFSR, and also for the 3D-Chebyshev chaotic map coupled with the 32-bit LSFR used. The results of these statistical tests indicate the degree of randomness of the generated sequences.

2.1. Discrete Equations of the Chaotic Maps and the 32-Bit LFSR Used

We give below the discrete equations of the chaotic maps, and the 32-bit LFSR used to design the proposed chaotic systems:

Discrete Logistic map

The logistic map, created by Pierre Verhulst in 1845, was initially designed as a model for population growth [35]. Due to the recurrence equation's simplicity, Ulam and Von Neumann in 1947 used the logistic map as a pseudo-random number generator [36]. Since then, it has been widely used in cryptographic applications. When the control parameter value is set to 4, the discrete Logistic map equation is as follows:

$$XL(n) = \begin{cases} \left\lfloor \frac{XL(n-1)\left[2^N - XL(n-1)\right]}{2^{N-2}} \right\rfloor & if\ XL(n-1) \neq \left[3 \times 2^{N-2} - 1,\ 2^{N-1}\right] \\ 3 \times 2^{N-2} - 1 & if\ XL(n-1) = 3 \times 2^{N-2} \\ 2^N - 1 & if\ XL(n-1) = 2^{N-1} \end{cases} \quad (1)$$

where $XL(n)$ is an integer that falls within the range $[1, 2^N - 1]$ with $N = 32$ as the precision used and $\lfloor Z \rfloor$ (*Floor function*) returns the highest integer that is less than or equal to Z.

Discrete Skew-Tent map

The Skew-Tent map is a piecewise linear map, the equation of the discretized Skew-Tent map is defined by:

$$XS(n) = \begin{cases} \left\lfloor \frac{2^N \times XS(n-1)}{P_s} \right\rfloor & if\ 0 < XS(n-1) < P_s \\ \left\lfloor 2^N \times \frac{\left[2^N - XS(n-1)\right]}{2^N - P_s} \right\rfloor & if\ P_s < XS(n-1) < 2^N \\ 2^N - 1 & otherwise \end{cases} \quad (2)$$

where $XS(n)$ is an integer that falls within the range $[1, 2^N - 1]$ and P_s is the Skew-Tent map's control parameter, with a value in the range $[1, 2^N - 1]$.

Discrete PWLCM map

Another piecewise linear chaotic map is the Piecewise Linear Chaotic Maps (PWLCM); due to its good cryptographic characteristics, compared to the other chaotic maps studied

in this section, the PWLCM map is commonly employed in data encryption. The discrete PWLCM map's equation is as follows:

$$XP(n) = \begin{cases} \left\lfloor \frac{2^N \times XP(n-1)}{P_p} \right\rfloor & if\ 0 < XP(n-1) < P_p \\ \left\lfloor 2^N \times \frac{[XP(n-1)-P_p]}{2^{N-1}-P_p} \right\rfloor & if\ P_p < XP(n-1) < 2^{N-1} \\ \left\lfloor 2^N \times \frac{[2^N-XP(n-1)-P_p]}{2^{N-1}-P_p} \right\rfloor & if\ 2^{N-1} < XP(n-1) < 2^N - P_p \\ \left\lfloor 2^N \times \frac{[2^N-XP(n-1)]}{P_p} \right\rfloor & if\ 2^N - P_p < XP(n-1) < 2^N \\ 2^N - 1 & otherwise \end{cases} \quad (3)$$

where P_p is the PWLCM map's control parameter, with a value in the range $[1, 2^{N-1} - 1]$ and XP belongs to the same interval as XS and XL.

Discrete 3D-Chebyshev map

The discrete 3D-Chebyshev (third order) map is given by:

$$XT(n) = \left\lfloor 2^{(-2N+2)} \times \left(4 \times \left(XT - 2^{(N-1)}\right)^3 - 3 \times 2^{(2N-2)} \times \left(XT - 2^{(N-1)}\right)\right) + 2^{(N-1)} \right\rfloor \quad (4)$$

where $1 \leq XT(n) \leq 2^N - 1$.

32-bit LFSR based on primitive polynomial with internal feedback (Galois implementation)

The 32-bit LFSR with maximum length feedback used in our work is defined by the following primitive polynomial:

$$Q(x) = x^{32} + x^{22} + x^2 + x + 1 \quad (5)$$

Its circuit diagram is shown in Figure 1. The sequences generated (random outputs) are periodic of period equal to $2^{32} - 1$ = 4,294,967,295.

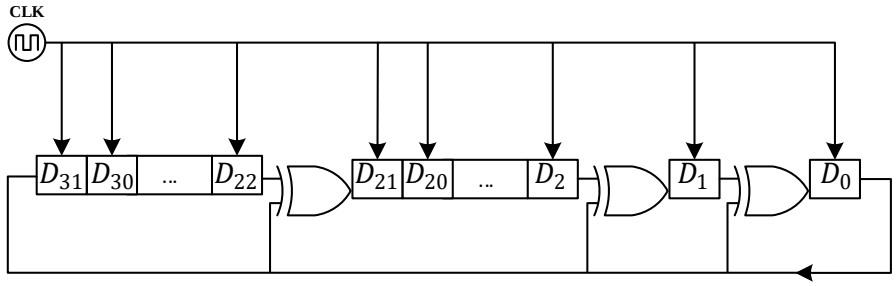


Figure 1. Circuit Diagram of 32-bit (internal feedback) LFSR.

2.2. Statistical Performance: NIST, Histogram, and Chi-Square Test

NIST statistical test

The NIST test is widely considered one of the most effective standards for evaluating the randomness of binary data [37]. This test is a statistical package consisting of 188 proposed tests and sub-tests (15 different tests in total) for evaluating the randomness of binary sequences of arbitrary length. These tests focus on different kinds of non-randomness that can be present in sequences. A p-value is calculated for each test to show the results of the test. A p-value ≥ 0.01 indicates that the sequence was considered random with 99% confidence. A p-value ≤ 0.01 implies the conclusion that the sequence is not random with 99% confidence.

Histogram and Chi-square test analysis

Another key property of any robust pseudo-chaotic number generator is to provide a uniform distribution in the whole phase space. To assert the uniformity of generated sequences, it is necessary to apply on them the chi-square test χ^2 given by the following formula:

$$\chi^2_{exp} = \sum_{i=0}^{N_c-1} \frac{(O_i - E_i)^2}{E_i} \tag{6}$$

where $N_c = 1000$ represents the number of classes, O_i represents the number of calculated samples in the ith class E_i, where the class E_i is the expected number of samples of a uniform distribution, which is equal to Ns/N_c.

To confirm the uniformity of a produced sequence, the experimental chi-square value must be smaller than the theoretical value $\chi^2_{exp} < \chi^2_{th}$. Furthermore, if the experimental chi-square value is less than the theoretical one, the resulting sequence is more uniform.

At the top of Figure 2a–f, from left to right, we present the proportion achieved versus the test for the discrete Logistic map, Skew-Tent map, PWLCM, 3D-Chebyshev map, 32-bit LFSR, and 3D-Chebyshev map coupled with a 32-bit LFSR respectively (see Figure 3). As we can see, the sequences generated by the discrete chaotic maps alone and the 32-bit LFSR do not successfully pass all NIST tests, and some are far from the acceptance criterion of a test indicated by the red line. In addition, we note that the PWLCM map outperforms the other chaotic maps studied, in terms of several tests passed. Moreover, obtained results for the generated sequences by the coupled 3D- Chebyshev map with the 32-bit LFSR show that all NIST tests, except the Block-frequency test, have passed, demonstrating that the coupled technique with the 32-bit LFSR improves the cryptographic properties of the 3D-Chebyshev map (this is true for any type of chaotic map).

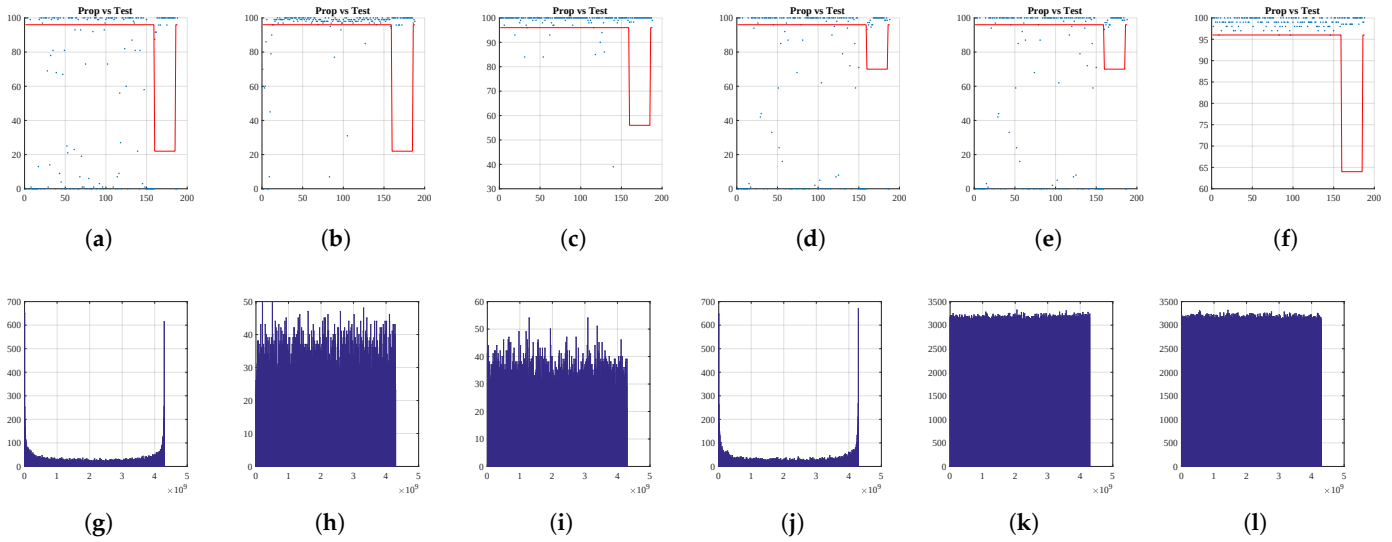


Figure 2. NIST test and histograms of studied chaotic maps, 32-bit LFSR, and the 3D-Chebyshev coupled with the 32-bit LFSR. (**a**) NIST test results of the discrete Logistic map. (**b**) NIST test results of the discrete Skew-Tent map. (**c**) NIST test results of the discrete PWLCM map. (**d**) NIST test results of the discrete 3D-Chebyshev map. (**e**) IST test results of the 32-bit LFSR. (**f**) NIST test results of the discrete 3D-Chebyshev map coupled with LFSR. (**g**) The histogram of a sequence XL. (**h**) The histogram of a sequence XS. (**i**) The histogram of a sequence XP. (**j**) The histogram of a sequence XT. (**k**) The histogram of a sequence Q. (**l**) The histogram of a sequence XTI.

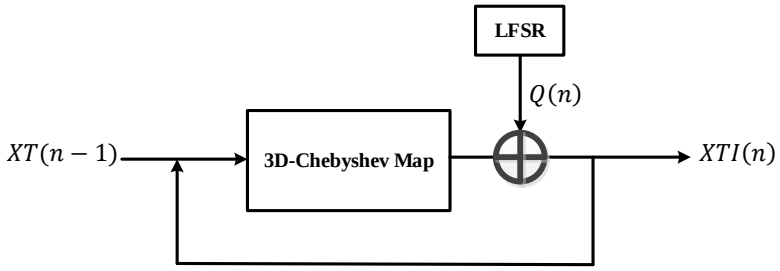


Figure 3. 3D-Chebyshev map coupled with the used LFSR using a XOR operator.

At the bottom of Figure 2g–l, we give the corresponding histogram for each chaotic map, the 32-bit LFSR, and the coupled 3D-Chebyshev map with the 32-bit LFSR. We notice that, visually, the histograms obtained by the chaotic maps are not uniform, except for the 32-bit LFSR and the discrete 3D-Chebyshev coupled with the 32-bit LFSR. This observation is asserted by the obtained experimental and theoretical results of the chi-square tests given in Table 1.

Table 1. Chi-square results on the tested histograms.

Chi-Square Test	Logistic	STM	PWLCM	3D-Ch	LFSR	3D-Ch with LFSR
χ^2_{exp}	38,698.41	1113.45	1146.92	41,865.00	989.48	999.48
χ^2_{th} (1000, 0.05)	1073.64	1073.642	1073.64	1073.64	1073.64	1073.64

2.3. Hardware Metrics

We interpret the implementation of the chaotic maps in regard to the resources utilized (area, DSPs), the speed (WNSi, Max. Freq., throughput), and the efficiency (throughput-to-slices ratio). Furthermore, we give the power consumption.

$$Max.Freq. = \frac{1}{Ti - WNSi}(MHz). \tag{7}$$

where Ti is the target clock period (ns) used during the implementation run "i" and $WNSi$ is the Worst Negative Slack (ns) of the target clock used during the implementation run "i" and must be positive, and very close to zero.

The data Throughput is calculated by Equation (8).

$$Throughput = N \times Max.Freq.(Mbps). \tag{8}$$

The Efficiency is calculated by using Equation (9).

$$Efficiency = \frac{Throughput}{Slices}(Mbps/Slices). \tag{9}$$

The chaotic systems implementation was carried out on the Xilinx-manufactured PYNQ-Z2 FPGA board. To implement the chaotic systems, the code was written in VHDL using 32-bit fixed-point data format, then synthesized and implemented using Xilinx Vivado design suite (V.2017.2). The Vivado design suite tools are a set of tools provided by Xilinx for designing and implementing digital circuits on FPGA boards. These tools enable the complete design flow from creating the RTL design, to implementation, synthesis, and programming of the FPGA. The Vivado design suite includes a powerful integrated development environment (IDE) that allows users to easily design, debug, and test their designs. It also includes various features such as timing analysis, power analysis, and design exploration, which can help in fine-tuning the design for optimal performance and power consumption. As illustrated in Figure 4, we present a summary of the various stages of the conception flow that were executed using Vivado for evaluating the metrics of the discrete

Skew-Tent map as an example of demonstration. Note that the same steps for all studied chaotic systems were repeated since Design entry up to configuration bitstream generation.

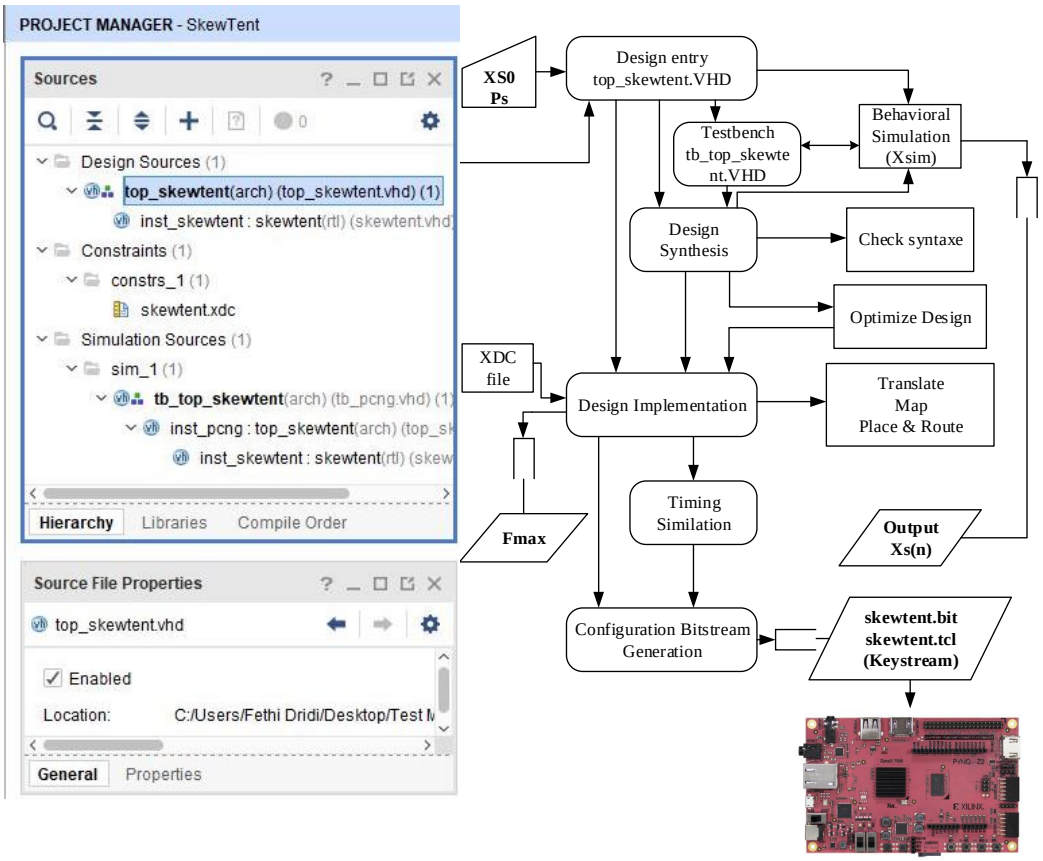


Figure 4. The design flow of the Skew-Tent map on FPGA using Vivado.

The areas of the implementations of the chaotic systems on FPGA are compared using slices, Flip-Flops (FFs), and lookup tables (LUTs), which are the basic logic block of Xilinx FPGAs. Latency, maximum frequency, and throughput together determine execution speed. The efficiency parameter represents the throughput-to-slices ratio and provides a general understanding of the implementation's hardware performance as measured by the hardware metrics.

All designs have been tested after place and route using simulation to ensure the correct functionality and produce one sample at each clock cycle.

At the end of the Place & Route process, we get the hardware metrics of chaotic systems implemented on the PYNQ-Z2 FPGA chip. Table 2, resume the obtained hardware metrics of studied chaotic maps, the 32-bit LFSR, and the 3D-Chebyshev map coupled with the 32-bit LFSR.

Table 2. Hardware metrics of studied chaotic maps, the 32-bit LFSR, and the 3D-Chebyshev map coupled with the 32-bit LFSR.

			Logistic Map	STM	PWLCM	3D-Ch Map	32-Bit LFSR	3D-Ch with LFSR
Resources used	Area	LUTs	77/0.14%	2830/5.32%	7374/13.86%	286/0.05%	2/< 0.01%	319/0.60%
		FFS	49/0.05%	57/0.05%	63/0.06%	47/0.04%	62/0.06%	71/0.07%
		Slices *	33/0.25%	853/6.41%	2171/16.32%	87/0.65%	16/0.12%	99/0.74%
	DSPs		4/1.82%	0/0.00%	0/0.00%	12/5.45%	0/0.00%	12/5.45%
Speed	WNSi (ns)		0.102	0.059	0.108	0.031	6.078	0.096
	Ti (ns)		12	28	31.2	22	8	21.6
	Max. Freq. (MHz)		84.04	35.78	32.16	45.51	520.29	46.50
	Throughput (Mbps)		2689.52	1145.27	1029.20	1456.59	16,649.32	1488.09
Efficiency (Mbps/Slices)			81.500	1.342	0.476	16.742	1040.58	15.031
Power (W)			0.083	0.070	0.105	0.048	0.118	0.055

* Note: Each slice contains four 6-input LUTs and 8 flip-flops.

In terms of computing performance, the PWLCM is slower when compared to other chaotic maps and uses approximately three times more hardware resources than the Skew-Tent map. Moreover, the obtained values for the discrete Logistic map show that a high throughput characterizes the Logistic map compared to the throughput of the other studied chaotic maps and uses fewer hardware resources. Furthermore, the 3D-Chebyshev map coupled with the 32-bit LFSR achieves a throughput of approximately 1.5 Gbps.

3. Proposed Chaos-Based Stream Ciphers Based on Secure PRNGs of Chaotic Sequences

Based on statistical results obtained in the previous Section 2, we can confirm that chaotic maps cannot be used alone as a secure PRNG-CS. The question is how to combine some of them to build PRNGs-CS secure against statistical and cryptanalytic attacks, and efficient, in terms of hardware cost, throughput, and efficiency. A compromise between security and effectiveness is usually made, and it is first about the degree of security required for a given application.

In this section, we created three secure pseudorandom number generators (PRNGs), known as LSP-PRNG, LST-PRNG, and LSPT-PRNG, by combining different mathematical maps such as the Logistic map, Skew-Tent map, PWLCM map, and 3D-Chebyshev map. We also examined the stream cipher systems associated with each PRNG.

3.1. Block Diagram of Stream Ciphers Based on Secure PRNGs-CS

Figure 5 depicts a stream encryption/decryption system's block diagram. The stream cipher method uses the XOR function for both encryption and decryption, by combining the keystream with the plaintext for encryption and with the ciphertext for decryption.

The security of such a system relies entirely on the PRNG-CS keystream. An encryption/decryption system is considered unconditionally secure when the keystream is perfectly random and the period is infinite (a so-called one-time pad).

PRNG-CS takes as input a secret key (K) and an initial value (IV) utilized to defeat a known plaintext attack. The IV changes with each new session and can only be used once. Therefore, generated sequences in different sessions using the same secret key will differ. Note that stream ciphers are used to continuously cipher data, such as selective video encryption or network communications. Below we will describe the proposed secure PRNGs-CS in detail.

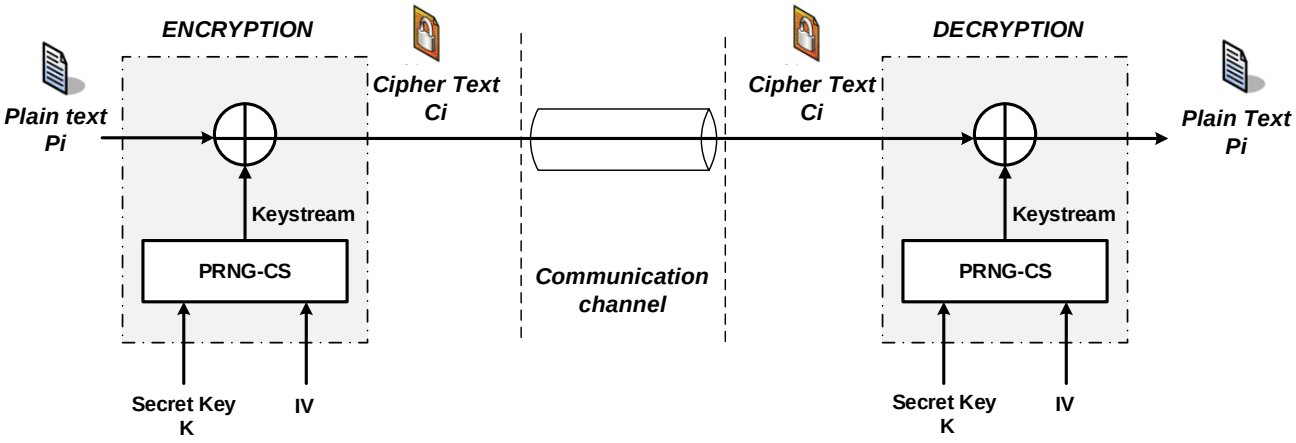


Figure 5. A stream encryption/decryption system's block diagram.

3.2. Architectures Description of the Proposed PRNGs-CS

All the proposed secure PRNGs-CS have the same general design, though different internal states and just slightly different output functions. In this section, we describe in detail the architectures of the proposed secure PRNGs-CS and perform their security analysis and hardware metrics.

3.2.1. The Proposed LSP-PRNG

Figure 6 shows the architecture of the proposed LSP-PRNG. Its internal state is formed by three weakly coupled discrete chaotic maps: Logistic map, Skew-Tent, and Piecewise Linear Chaotic Map (PWLCM). The output function is based on chaotic multiplexing techniques.

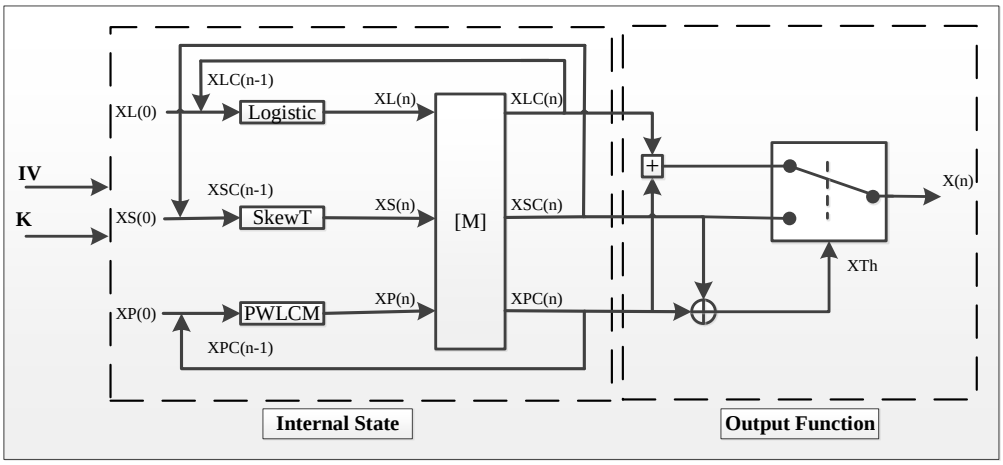


Figure 6. LSP-PRNG design architecture proposed.

The proposed LSP-PRNG takes a secret key K and an initialization vector IV as input and computes initial values $XL(0)$, $XS(0)$ and $XP(0)$ of the three chaotic maps: Logistic, Skew-Tent, and PWLCM respectively. The IV of the system supplies the initial vectors of the three chaotic maps: IVL, IVS, and IVP each are of size N = 32 bits, and the secret key K provides all the initial conditions and parameters of the chaotic maps and the parameters of the coupling matrix M, as summarized in Table 3.

Table 3. The initial conditions and parameters that form the secret key.

Symbol	Definition
$XL0$, $XS0$, and $XP0$	Initial conditions of the chaotic maps: Logistic, Skew-Tent, and PWLCM respectively, ranging from 1 to $2^N - 1$.
P_s	Skew-Tent map's control parameter, in the range $[1,2^N - 1]$.
P_p	PWLCM map's control parameter, in the range $[1,2^{N-1} - 1]$.
ϵ_{ij}	The coupling matrix's parameters, in the range $[1,2^k]$ where $k \leq 5$.

The size of the secret key, noted $|K1|$, of the proposed LSP-PRNG is given by

$$|K1| = |XL0| + |XS0| + |XP0| + |P_s| + |P_p| + \left(6 \times |\epsilon_{ij}|\right) = 189\ bits \tag{10}$$

(see below the form of the Matrix M containing ϵ_{ij}) where $|XL0| = |XS0| = |XP0| = |P_s| = 32\ bits$, $|P_p| = 31\ bits$, and $|\epsilon_{ij}| = 5\ bits$.

The key space of the secret key is 2^{189} different combinations which are large enough to make the brute-force attack infeasible [38]. Indeed, it is commonly agreed upon that a key space of at least 2^{128} is required to make the brute-force attack infeasible.

The secret key K provides initial conditions and parameters for chaotic maps as follows:

$$\begin{cases} XL0 = K(0\ to\ 31) \\ XS0 = K(32\ to\ 63) \\ XP0 = K(64\ to\ 95) \\ P_s = K(96\ to\ 127) \\ P_p = K(128\ to\ 158) \\ \epsilon_{12} = K(159\ to\ 163) \\ \epsilon_{13} = K(164\ to\ 168) \\ \epsilon_{21} = K(169\ to\ 173) \\ \epsilon_{23} = K(174\ to\ 178) \\ \epsilon_{31} = K(179\ to\ 183) \\ \epsilon_{32} = K(184\ to\ 188) \end{cases} \tag{11}$$

The initial values $XL(0)$, $XS(0)$, and $XP(0)$ of the three chaotic maps are calculated as follows:

$$\begin{cases} XL(0) = \text{mod}\ ((XL0 + IVin),\ 2^N) \\ XS(0) = \text{mod}\ ((XS0 + IVin),\ 2^N) \\ XP(0) = \text{mod}\ ((XP0 + IVin),\ 2^N) \end{cases} \tag{12}$$

where:

$$IVin = IVL \oplus IVS \oplus IVP \tag{13}$$

and

$$\begin{cases} IVL = IV(0\ to\ \ 31) \\ IVS = IV(32\ to\ \ 63) \\ IVP = IV(64\ to\ \ 95) \end{cases} \tag{14}$$

The function that controls the internal state creates a loose connection between the chaotic maps, generating future samples of $XLC(n)$, $XSC(n)$, and $XPC(n)$. The output sequence $X(n)$ is then created by applying a chaotic switching method to these samples (see Figure 6).

This equation describes how the coupling system works:

$$\begin{bmatrix} XLC(n) \\ XSC(n) \\ XPC(n) \end{bmatrix} = M \times \begin{bmatrix} XL(n) \\ XS(n) \\ XP(n) \end{bmatrix} \tag{15}$$

where:

$$M = \begin{bmatrix} M_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & M_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & M_{33} \end{bmatrix} \tag{16}$$

with ϵ_{ij} are the weakly coupling parameters, and $M_{11} = (2^N - \epsilon_{12} - \epsilon_{13})$; $M_{22} = (2^N - \epsilon_{21} - \epsilon_{23})$; $M_{33} = (2^N - \epsilon_{31} - \epsilon_{32})$.

$XL(n)$, $XS(n)$, and $XP(n)$ are denoted as the maps output values at instant n of the: Logistic, Skew-Tent, and PWLCM map, respectively, defined as follows:

The first three outputs of the matrix M, $XLC(1)$, $XSC(1)$, $XPC(1)$ are given by the following equation:

$$\begin{bmatrix} XLC(1) \\ XSC(1) \\ XPC(1) \end{bmatrix} = M \times \begin{bmatrix} XL(1) \\ XS(1) \\ XP(1) \end{bmatrix} \tag{17}$$

with:

$$XL(1) = Logistic\left\{\mathrm{mod}\left(XL(0), 2^N\right)\right\} \tag{18}$$

$$XS(1) = STM\left\{\mathrm{mod}\left(XS(0), 2^N\right), P_s\right\} \tag{19}$$

$$XP(1) = PWLCM\left\{\mathrm{mod}\left(XP(0), 2^N\right), P_p\right\} \tag{20}$$

Then, if $2 \leq n \leq N_s$, (N_s represents the number of samples desired), the three outputs of the matrix M are governed by Equation (15), with:

$$XL(n) = Logistic\left\{\mathrm{mod}\left(XLC(n-1), 2^N\right)\right\} \tag{21}$$

$$XS(n) = STM\left\{\mathrm{mod}\left(XSC(n-1), 2^N\right), \ P_s\right\} \tag{22}$$

$$XP(n) = PWLCM\left\{\mathrm{mod}\left(XPC(n-1), 2^N\right), \ P_p\right\} \tag{23}$$

The obtained multiplexed samples of the sequence $X(n)$ are controlled by the chaotic sample $X_{th}(n)$ and a threshold T:

$$X(n) = \begin{cases} mod\left((XPC(n) + XLC(n)), 2^N\right) & if\ 0 < X_{th}(n) < T \\ XSC(n) & otherwise \end{cases} \tag{24}$$

where $X_{th}(n) = XPC(n) \oplus XSC(n)$ and $T = 0.8 \times 2^N$.

3.2.2. The Proposed LST-PRNG

The design of the proposed LST-PRNG is given in Figure 7. The internal state of the system is constructed by weakly coupling three discrete chaotic maps: Logistic map, STM, and 3D-Ch coupled with an LFSR. The latter can improve the 3D Chebyshev periodicity and its uniformity, as shown in Section 2. The output function is based on modulo addition and XOR operations.

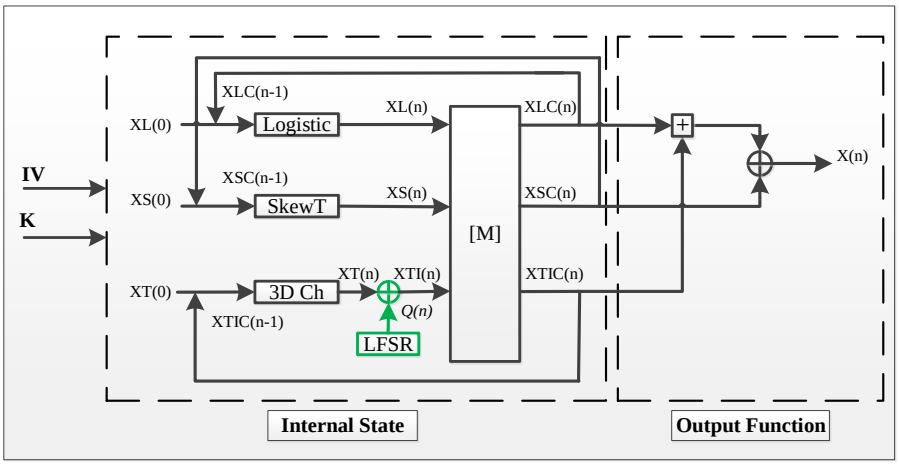


Figure 7. LST-PRNG design architecture proposed.

Compared with the architecture in Figure 6, the proposed LST-PRNG architecture has better hardware performance because the hardware performance of the 3D Ch map is better than those of the PWLCM map. Moreover, using 3D Ch increases the robustness of the system against algebraic attacks.

The size of the secret key of the proposed LST-PRNG is:

$$|K2|=|XL0|+|XS0|+|XT0|+|P_s|+(6\times|\epsilon_{ij}|)+|Q0|=190\ \ bits \tag{25}$$

where $|XT0|=|Q0|=32\ \ bits$; and $XT0$, $Q0$ are the initial conditions of the 3D Ch and the Linear Feedback Shift Register (LFSR), respectively.

The key space contains 2^{190} different values, which is large enough to make brute-force attacks infeasible.

The initial values $XL(0)$, $XS(0)$ and $XT(0)$ of the used chaotic maps are given by:

$$\begin{cases} XL(0)=\text{mod}\ ((XL0+IVin),\ 2^N) \\ XS(0)=\text{mod}\ ((XS0+IVin),\ 2^N) \\ XT(0)=\text{mod}\ ((XT0+IVin),\ 2^N) \end{cases} \tag{26}$$

where:

$$IVin\ =\ IVL\ \oplus\ IVS\ \oplus\ IVT \tag{27}$$

and IVL, IVS and IVT are provided by the initial value IV as follows:

$$\begin{cases} IVL=IV(0\ \ to\ \ 31) \\ IVS=IV(32\ \ to\ \ 63) \\ IVT=IV(64\ \ to\ \ 95) \end{cases} \tag{28}$$

Using the output samples of $XLC(n)$, $XSC(n)$ and $XTIC(n)$, the coupling matrix M (defined in (16)) produces the output sequence $X(n)$ as shown in Figure 7. The internal system is governed by the following equations.

$$\begin{bmatrix} XLC(n) \\ XSC(n) \\ XTIC(n) \end{bmatrix} = M\times \begin{bmatrix} XL(n) \\ XS(n) \\ XTI(n) \end{bmatrix} \tag{29}$$

with:

$$XTI(n)\ =\ XT(n)\oplus Q(n) \tag{30}$$

and $XL(n)$, $XS(n)$, and $XT(n)$ are denoted as the maps output values at instant n of the: Logistic, Skew-Tent, and 3D-Ch map, respectively.

The first three outputs of the matrix M, $XLC(1)$, $XSC(1)$, $XTIC(1)$ are given by:

$$\begin{bmatrix} XLC(1) \\ XSC(1) \\ XTIC(1) \end{bmatrix} = M \times \begin{bmatrix} XL(1) \\ XS(1) \\ XTI(1) \end{bmatrix} \tag{31}$$

with:

$$XTI(1) = XT(1) \oplus Q(1) \tag{32}$$

and:

$$XL(1) = Logistic\left\{\text{mod}\left(XL(0), 2^N\right)\right\} \tag{33}$$

$$XS(1) = STM\left\{\text{mod}\left(XS(0), 2^N\right), P_s\right\} \tag{34}$$

$$XT(1) = 3D - Ch\left\{\text{mod}\left(XT(0), 2^N\right)\right\} \tag{35}$$

Then, for $n \geq 2$ and $n \leq N_s$, the matrix M output is given by Equation (29), with:

$$XL(n) = Logistic\left\{\text{mod}\left(XLC(n-1),\ 2^N\right)\right\} \tag{36}$$

$$XS(n) = STM\left\{\text{mod}\left(XSC(n-1),\ 2^N\right),\ P_s\right\} \tag{37}$$

$$\begin{cases} XT(n) = 3D - Ch\{\text{mod}(XTIC(n-1),\ 2^N)\} \\ XTI(n) = XT(n) \oplus Q(n) \end{cases} \tag{38}$$

Finally the output $X(n)$ is calculated by (see Figure 7):

$$X(n) = \text{mod}\left((XLC(n) + XTIC(n)),\ 2^N\right) \oplus XSC(n) \tag{39}$$

3.2.3. The Proposed LSPT-PRNG

Some applications (military, industrial, etc.) need to sacrifice hardware metrics for maximum security. For this purpose, we propose the LSPT-PRNG architecture shown in Figure 8. The proposed generator was studied, implemented in Matlab code, and published in [30], but it only performs security analysis, currently focusing on hardware metrics.

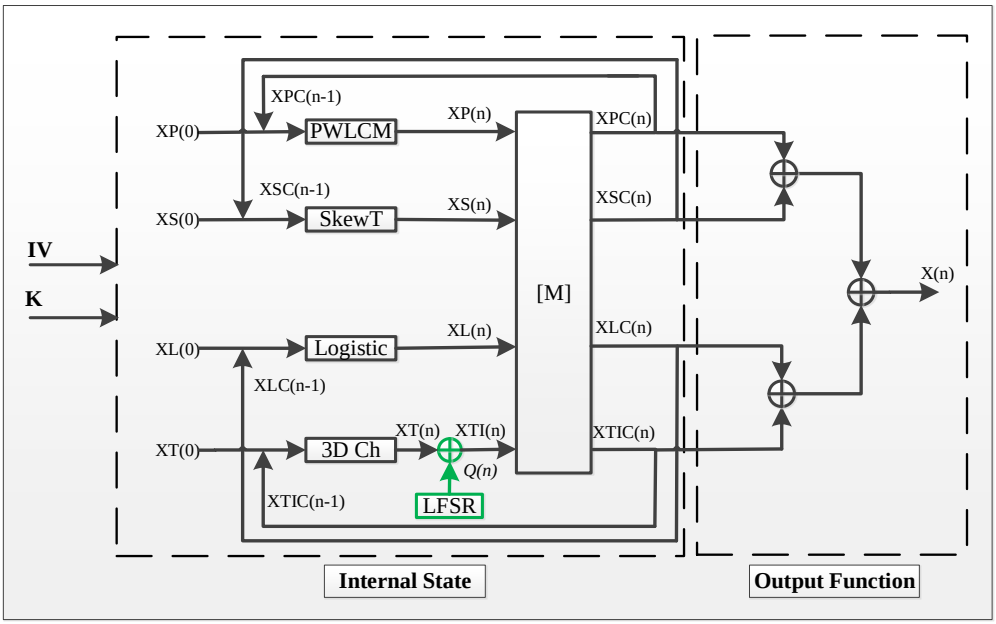


Figure 8. LSPT-PRNG design architecture proposed.

3.3. Security Analysis and Hardware Metrics of the Proposed PRNGs-CS

3.3.1. PRNGs-CS Resilience against Statistical Attacks

A set of tests should be applied to assess the statistical cryptographic characteristic of the generated sequences by the proposed PRNGs-CS. Each test checks for a particular feature, such as the correlation between generated sequences or their uniformity, and the composite results of these tests give an insight into the level of randomness of the generated sequences. The statistical properties of the generated sequences are closely linked to the pseudo-chaotic behavior of these sequences. The National Institute of Standards and Technology (NIST) test, among other tests (TestU01, DieHARD), serves as a reference for quantifying and comparing the statistical effects of pseudo-random sequences.

Phase space test

We draw in Figure 9a–c the phase space or mapping of sequences X1, X2, and X3 generated by the suggested LSP-PRNG, LST-PRNG, and LSPT-PRNG, respectively, and created by 3,125,000 samples. In Figure 9d–f, we present the mapping of 1000 samples (a zoom) selected randomly from X1, X2, and X3, respectively.

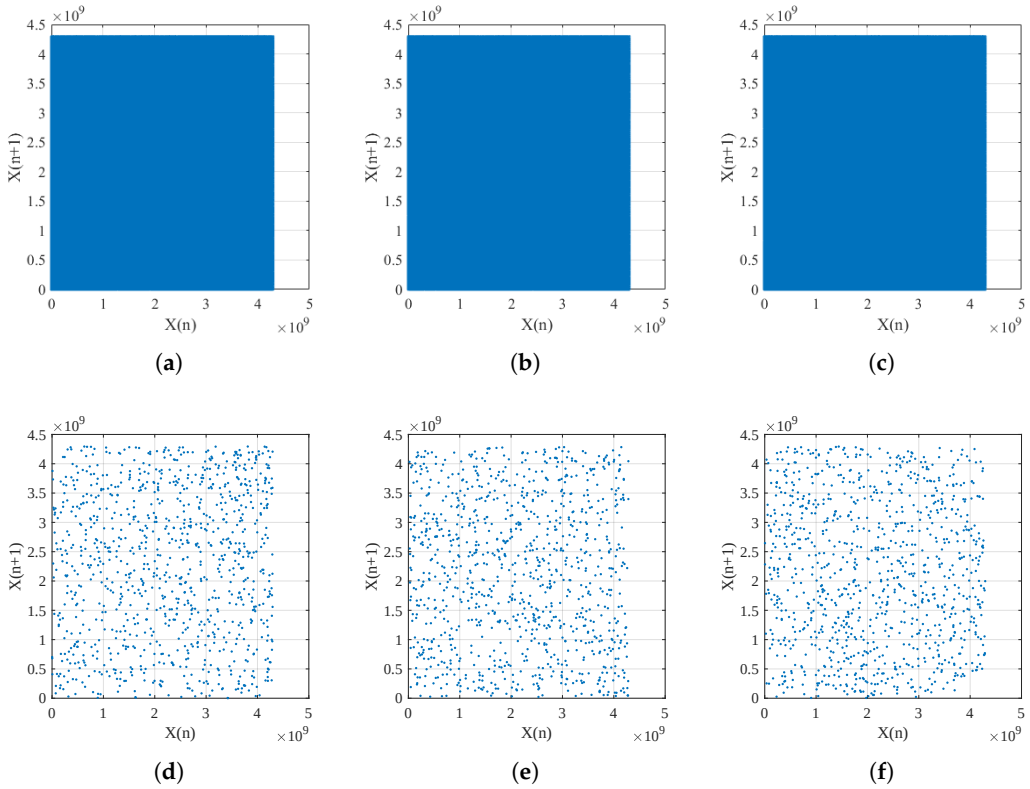


Figure 9. Phase space and zoom of the produced sequences X1, X2, and X3 by LSP-PRNG, LST-PRNG, and LSPT-PRNG, respectively. (**a**) Mapping of sequence X1. (**b**) Mapping of sequence X2. (**c**) Mapping of sequence X3. (**d**) Zoom on the mapping of X1. (**e**) Zoom on the mapping of X2. (**f**) Zoom on the mapping of X3.

Already, from Figure 9d–f, The area seems chaotic, indicating a lack of association between nearby sample results.

Histogram and Chi-square tests

A fundamental characteristic of a secure PRNG-CS is that the generated sequences are equally distributed. Histograms of the produced sequences X1, X2, and X3 are shown in Figure 10. Their uniformities are visually observed.

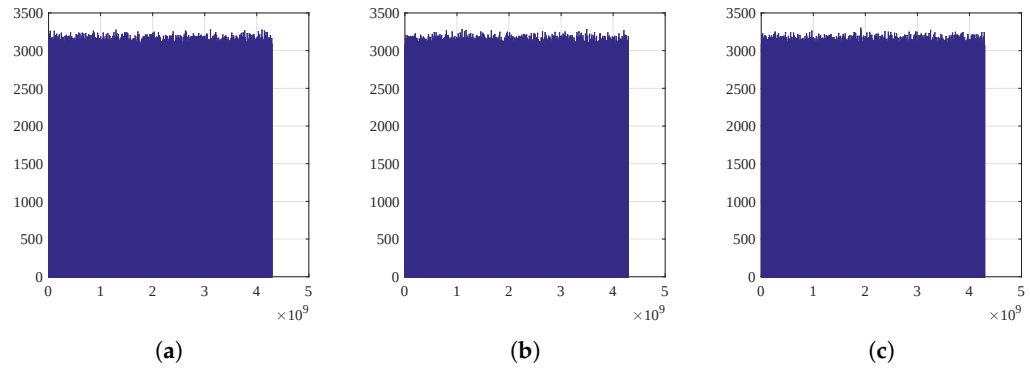


Figure 10. Histograms of the produced sequences X1, X2, and X3 by LSP-PRNG, LST-PRNG, and LSPT-PRNG respectively. (**a**) The histogram of the produced sequence X1. (**b**) The histogram of the produced sequence X2. (**c**) The histogram of the produced sequence X3.

The chi-square test, as defined by Equation (6), should be used to check visual uniformity results. Table 4 shows the experimental and theoretical chi-square tests for sequences X1, X2, and X3 generated by LSP-PRNG, LST-PRNG, and LSPT-PRNG, respectively.

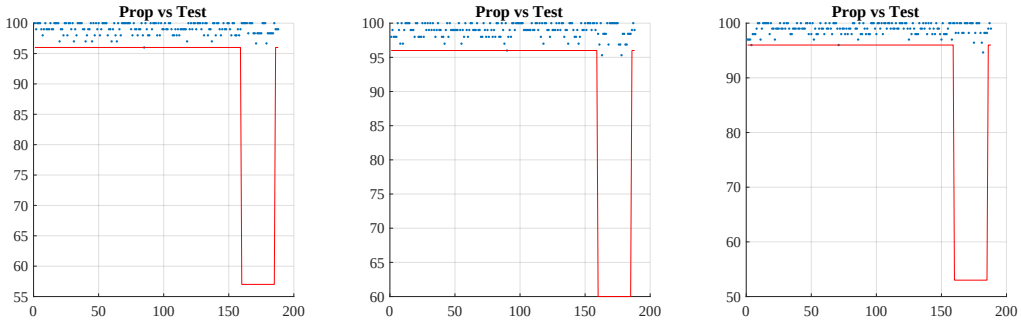
Table 4. Chi-Square test.

Chi-Square Test Value	LSP-PRNG	LST-PRNG	LSPT-PRNG
χ^2_{th}	1073.6426	1073.6426	1073.6426
χ^2_{exp}	941.5878	896.3603	915.5385

The experimental chi-square test values are lower than the theoretical values for all proposed PRNGs-CS, confirming the uniformity of the histograms.

NIST test

A further important effect of a secure PRNG-CS is that the generated sequences must pass the NIST statistical tests (previously defined in Section 2). The results obtained show that the sequences generated by the proposed PRNGs-CS pass all 15 statistical tests, as shown in Figure 11 and Table 5.



(**a**) (**b**) (**c**)

Figure 11. NIST test results of the proposed PRNGs-CS. (**a**) NIST test for LSP-PRNG. (**b**) NIST test for LST-PRNG. (**c**) NIST test for LSPT-PRNG.

Table 5. *p*-values and Proportion results of NIST test for the proposed PRNGs-CS.

Test	LSP-PRNG		LST-PRNG		LSPT-PRNG	
	p-Value	Prop. (%)	*p*-Value	Prop. (%)	*p*-Value	Prop. (%)
Frequency test	0.494	100	0.475	98	0.115	97
Block-frequency test	0.760	99	0.319	99	0.851	97
Cumulative-sums test (2)	0.757	100	0.691	98	0.394	96.500
Runs test	0.367	100	0.883	98	0.988	98
Longest-run test	0.983	99	0.936	100	0.437	98
Rank test	0.720	98	0.972	99	0.658	98
FFT test	0.575	98	0.475	97	0.475	99
Nonperiodic-templates (148)	0.527	99.115	0.489	99.074	0.484	99.054
Overlapping-templates	0.596	99	0.911	100	0.384	100
Universal	0.335	98	0.335	99	0.035	97
Approximty entropie	0.475	99	0.367	100	0.262	100
Random-excursions (8)	0.352	99.792	0.471	98.242	0.371	99.107
Random-excursions-variant (18)	0.468	98.611	0.367	98.438	0.374	98.710
Serial test (2)	0.460	99	0.900	99.500	0.647	99.500
Linear-complexity	0.401	99	0.554	99	0.071	99

Key sensitivity analysis (using Hamming Distance)

Key sensitivity is a critical property for PRNGs. Of course, a slight modification in the secret key should result in a significant difference in the output sequence. To prove this

property, for all proposed PRNGs-CS, we compute the Hamming Distance between two sequences produced (S_1, S_2) that are just the LSB (Least Significant Bit) of parameter P_s different. Compute the mean Hamming Distance HD among the two generated sequences for five random secret keys. $H_D(S_1, S_2)$ is illustrated by the formula below:

$$H_D(S_1, S_2) = \frac{1}{Nb} \sum_{i=1}^{Nb} (S_1(i) \oplus S_2(i)) \tag{40}$$

where Nb is the number of bits in a produced sequence.

Table 6 illustrates the average Hamming distance values for the proposed LSP-PRNG, LST-PRNG, and LSPT-PRNG. These values are relative to the optimal value of 50%. This result demonstrates the high sensibility of the secret key of the proposed PRNGs-CS.

Table 6. Values of HD for the proposed PRNGs-CS.

PRNG-CS	**HD %**
LSP-PRNG	50.0022
LST-PRNG	50.0004
LSPT-PRNG	50.0012

3.3.2. Hardware Metrics of the Proposed PRNGs-CS

We quantify the performance of the proposed PRNGs-CS implementation in regard to the resources utilized, the speed, the efficiency, and the power consumption.

Table 7 lists the results of our three proposed PRNGs-CS designs.

Table 7. Harware metrics comparison of the proposed PRNGs-CS using ZYNQ PYNQ-Z2 FPGA.

			PRNGs-CS		
			LSP-PRNG	**LST-PRNG**	**LSPT-PRNG**
Resources used	Area	LUTs	10,400/19.55%	3525/6.63%	11,391/21.41%
		FFs	349/0.33%	434/0.41%	480/0.45%
		Slices	3096/23.28%	1024/7.70%	3337/25.09%
	DSPs		13/5.91%	25/11.36%	28/12.73%
Speed	WNSi (ns)		0.022	0.056	0.072
	Ti (ns)		31.60	26.20	32.30
	Max. Freq. (MHz)		31.66	38.24	31.09
	Throughput (Mbps)		1013.36	1224	995.14
Efficiency (Mbps/Slices)			0.32	1.20	0.29
Power (W)			0.146	0.083	0.162

We can observe that the LST-PRNG consumes the minimum slice at 1024 slices, and the LSPT-PRNG consumes the maximum slice at 3337 slices. All proposed PRNGs-CS offer high speed. 1013.36 Mbps, 1224 Mbps, and 995.14 Mbps for LSP-PRNG, LST-PRNG, and LSPT-PRNG respectively. The LST-PRNG offers the best throughput. Additionally, the LST-PRNG architecture offers lower power consumption (83 mW) compared to LSP-PRNG and LSPT-PRNG. The LST-PRNG design is ideal for IoT applications, and its efficiency is 1.20 Mbps/Slices.

We perform below a comparison of hardware metrics with several chaotic and non-chaotic systems for various stream ciphers based on the PRNGs-CS proposed above.

3.4. Performance Analysis of Stream Ciphers Based on Proposed PRNGs-CS

In this section, we first evaluate the security of stream ciphers (SC) based on the proposed PRNGs-CS using well-known cryptanalytic analysis. We then provide hardware metrics and compare them to several released systems.

3.4.1. Cryptanalytic Analysis

To evaluate the security of the proposed chaos-based stream cipher against the most common attacks, we perform key sensitivity and statistical analysis on various encrypted images below.

Sensitivity analysis

A robust cryptosystem must also be sensitive to secret keys. In other words, a small change to the secret key will generate a completely different ciphered image. The sensitivity is typically determined by two criteria, NPCR (Number of Pixel Change Rate) and UACI (Unified Average Changing Intensity) [39]. We also apply the Hamming Distance (HD), which is measured in bits. In our view, the HD parameter is more accurate than NPCR and UACI criteria. The formulas for these parameters are given below, with C_1, and C_2 being the two encrypted images of the same original image P.

$$NPCR = \frac{1}{M \times N \times P} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{p=1}^{P} D(i,j,p) \times 100\% \tag{41}$$

$$D(i,j,p) = \begin{cases} 1 & if\ C_1(i,j,p) \neq C_2(i,j,p) \\ 0 & if\ C_1(i,j,p) = C_2(i,j,p) \end{cases} \tag{42}$$

where M, N, and P are the width, height, and plane sizes (for a gray image, $P = 1$; for an RGB color image, $P = 3$) of C_1 and C_2.

The NPCR calculates the proportion of distinct pixel numbers in two encrypted images.

$$UACI = \frac{1}{M \times N \times P \times 255} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{p=1}^{P} |C_1(i,j,p) - C_2(i,j,p)| \times 100\% \tag{43}$$

which computes the average intensity of differences between C_1 and C_2.

For a random image, the expected values of NPCR, UACI, and HD are 99.6094%, 33.4635%, and 50%, respectively. We have tested five color images with the same size $512 \times 512 \times 3$ and different features as shown in Figure 12.

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 12. Five test images with same size $512 \times 512 \times 3$. (**a**) Lena. (**b**) Peppers. (**c**) Baboon. (**d**) Barbara. (**e**) Boats.

Table 8 displays the NPCR, UACI, and HD values obtained for the original images of Lena, Peppers, Baboon, Barbara, and Boats. These findings show that the generated NPCR, UACI, and HD values are extremely near the optimal values. These results exhibit that the proposed chaos-based stream ciphers are highly sensitive to slight changes in the secret key.

Table 8. NPCR, UACI, and HD values.

Chaos-Based Stream Ciphers	Test	Lena	Peppers	Baboon	Barbara	Boats
LSP-SC	NPCR %	99.5966	99.6156	99.6206	99.6206	99.5966
	UACI %	33.4374	33.4900	33.4769	33.4863	33.4377
	HD %	49.9755	50.0010	49.9868	49.9868	49.9755
LST-SC	NPCR %	99.6059	99.6111	99.6181	99.5933	99.6199
	UACI %	33.4565	33.4634	33.4486	33.4971	33.4809
	HD %	50.0162	50.0079	50.0050	49.9926	50.0480
LSPT-SC	NPCR %	99.6159	99.6135	99.6056	99.6135	99.6159
	UACI %	33.4637	33.4453	33.4751	33.4373	33.4606
	HD %	50.0391	49.9834	49.9912	49.9834	49.9912

Statistical analysis

To analyze the resilience of the proposed chaos-based stream cipher against most statistical attacks, we perform the following statistical analysis: histogram, chi-square, entropy, and correlation.

Histogram and chi-square tests

The histogram of the ciphered image plays a crucial role in determining the effectiveness of the encryption process. This indicates the pattern in which the gray levels of pixels in an encrypted image are arranged, and it should be highly similar to a uniform distribution. Figures 13 and 14 show the results for Barbara and Boats of size $512 \times 512 \times 3$ as an example, in (a) the plain images, (b) the cipher images by LSP-SC, (c) the cipher images by LST-SC, (d) the cipher images by LSPT-SC, and in (e–h) their histograms, respectively.

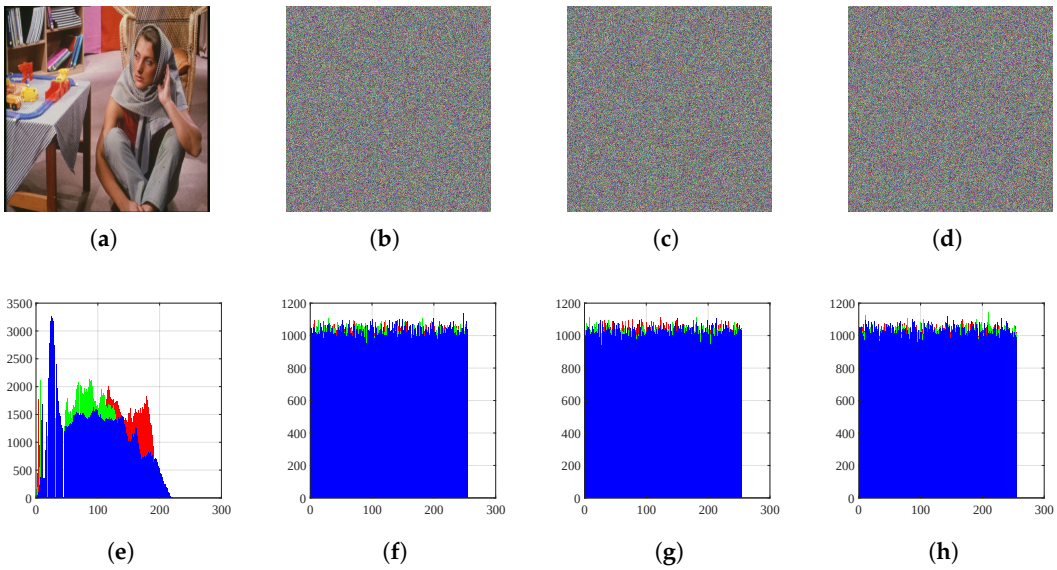


Figure 13. Results of Barbara image. (**a**) Plain image. (**b**) Encrypted image by LSP-SC. (**c**) Encrypted image by LST-SC. (**d**) Encrypted image by LSPT-SC. (**e**) Histogram of plain image. (**f**) Histogram of the ciphered image by LSP-SC. (**g**) Histogram of the ciphered image by LST-SC. (**h**) Histogram of the ciphered image by LSPT-SC.

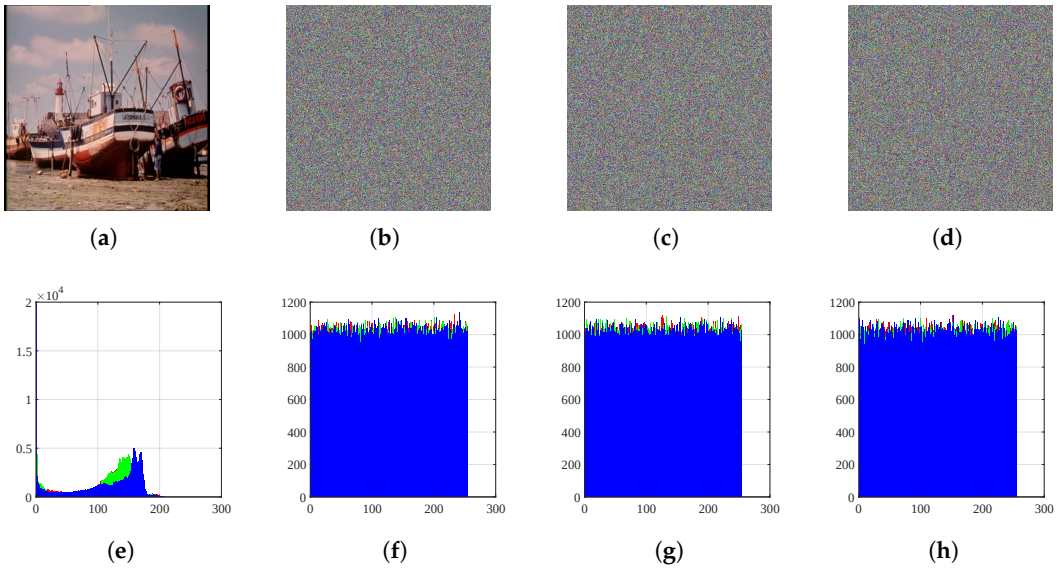


Figure 14. Results of Boats image. (**a**) Plain image. (**b**) Encrypted image by LSP-SC. (**c**) Encrypted image by LST-SC. (**d**) Encrypted image by LSPT-SC. (**e**) Histogram of plain image. (**f**) Histogram of the ciphered image by LSP-SC. (**g**) Histogram of the ciphered image by LST-SC. (**h**) Histogram of the ciphered image by LSPT-SC.

We can see that the histogram of the ciphered image is highly similar to a uniform distribution and vastly different from the original image. Using Equation (6), we perform a chi-square test on ciphered images to establish their uniformity statistically, where here: N_c is the number of levels, which is equal to $2^8 = 256$.

As demonstrated in Table 9, the chi-square test results indicate that the histograms of the encrypted images tested are distributed uniformly, as he theoretical values are higher than the experimental ones.

Table 9. Chi-square test.

Chi-Square Test Value		Lena	Peppers	Baboon	Barbara	Boats
χ^2_{th}		293.2478	293.2478	293.2478	293.2478	293.2478
χ^2_{exp}	LSP-SC	240.8893	251.3581	257.1510	243.0104	240.6367
	LST-SC	221.4082	265.3704	241.5293	266.8132	228.1328
	LSPT-SC	264.9030	244.7487	260.0397	280.6595	254.4102

Entropy analysis

The randomness of the encrypted image can be quantitatively measured through the entropy information provided by Shannon [40]:

$$H(C) = -\sum_{i=0}^{N_c-1} P(c_i) \times log_2(P(c_i)) \tag{44}$$

with $H(C)$ represents the entropy of the ciphered image and $P(c_i)$ denotes the probability of occurrence of each gray level ($c_i = 0, 1, \ldots, 255$). When the probability is the same, the entropy reaches its peak value (=8). The encryption algorithm is more robust when the experimental entropy value is closer to the maximum value. The results of the entropy test on the original and ciphered images are displayed in Table 10.

Table 10. Entropy results obtained.

Entropy		**Lena**	**Peppers**	**Baboon**	**Barbara**	**Boats**
Plain image		5.68222	7.66982	7.72644	7.69869	7.30541
Ciphered image	LSP-SC	7.99977	7.99976	7.99976	7.99977	7.99977
	LST- SC	7.99979	7.99975	7.99977	7.99975	7.99979
	LSPT- SC	7.99975	7.99977	7.99976	7.99973	7.99973

The entropies of the encrypted images are close to the optimal value; therefore, the stream cipher proposed in this study has a high level of robustness, as demonstrated by these results.

Correlation analysis

In an original image, correlation analysis can be used to measure the relationship between adjacent pixels in horizontal, vertical, and diagonal directions. This can be useful in identifying patterns and features in the image, such as edges and textures. An effective encryption process should result in ciphered images with minimal correlation and redundancy between adjacent pixels (as close to zero as possible). We carry out the following steps to evaluate the correlation: First, 8000 pairs of adjacent pixels are chosen at random from the test image, and then the correlation coefficients are calculated using the formula below.

$$\rho_{xy} = \frac{Cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{45}$$

where:

$$Cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)][y_i - E(y)] \tag{46}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{47}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)]^2 \tag{48}$$

The x and y are grayscale values of two pixels that are close to each other in the image under test. The results achieved are presented in Table 11.

Table 11 indicates that the correlation coefficients for the original images are almost one, demonstrating that the pixels are highly related, On the other hand, for the encrypted images, the correlation coefficients are nearly zero, indicating that there is no correlation between the original and encrypted images. As a result, there is no similarity between the original and ciphered images, which demonstrates the high level of confusion achieved by the proposed chaos-based stream ciphers.

Based on the analysis of histogram, entropy, and correlation, the proposed chaos-based stream ciphers have a high capability to defend against statistical attacks.

Table 11. Correlation coefficients of two adjacent pixels in the plain and ciphered images.

Image			**Horizontal**	**Vertical**	**Diagonal**
Lena	Plain image		0.97444	0.98468	0.96601
	Ciphered image	LSP−SC	0.01204	0.00549	0.00739
		LST−SC	−0.00414	0.00020	−0.00595
		LSPT−SC	−0.00770	0.00437	−0.03169

Table 11. *Cont.*

Image			Horizontal	Vertical	Diagonal
Peppers	Plain image		0.96216	0.96712	0.95239
	Ciphered image	LSP−SC	0.01803	−0.01284	0.00973
		LST−SC	−0.00927	−0.00896	−0.00004
		LSPT−SC	−0.00797	0.01538	−0.00102
Baboon	Plain image		0.95444	0.93200	0.91957
	Ciphered image	LSP−SC	−0.00380	−0.00371	−0.00347
		LST−SC	0.00374	−0.00295	−0.01533
		LSPT−SC	−0.01895	−0.00289	−0.00559
Barbara	Plain image		0.92453	0.97042	0.90707
	Ciphered image	LSP−SC	0.00939	−0.02223	0.00264
		LST−SC	−0.01218	−0.00765	0.01441
		LSPT−SC	−0.00748	0.00685	−0.00556
Boats	Plain image		0.96971	0.97280	0.94049
	Ciphered image	LSP−SC	−0.01362	0.00053	0.01487
		LST−SC	−0.01202	−0.00099	0.02195
		LSPT−SC	−0.01370	−0.00055	−0.01461

3.4.2. Hardware Metrics

To quantify the performance of the proposed implementation of the chaos-based stream ciphers in regard to the resources utilized (area, DSPs), the speed (Max. Freq., Throughput), the efficiency, and the power consumption, we cipher just four pixels. The proposed PRNGs-CS produce at each clock cycle a 32-bit sample, and we used the least significant 8 bits for XORing with each pixel (8 bits) to generate the corresponding encrypted pixel. Hardware metrics results for the proposed chaos-based stream ciphers are shown in Table 12 and, as expected, are linked to results for the various proposed PRNGs-CS.

Table 13 provides an overview of the hardware metrics comparison of the chaos-based stream ciphers proposed, the ZUC cipher, and chaotic and non-chaotic systems that are part of the eSTREAM project phase 2-focus hardware profile. It is hard to make sense of this comparison due to the varying characteristics of the used FPGAs. However, considering the clock rate of the FPGA board and the efficiency achieved, this comparison can be made. Therefore, with the exception of the Trivium and ZUC ciphers, the proposed chaos-based stream ciphers have competitive hardware performance when compared to most other chaotic and non-chaotic systems. Nevertheless, since 2007, several kinds of attacks have been launched against the eSTREAM cipher and ZUC stream cipher, revealing several vulnerabilities to cryptanalysis attacks [3,41–44].

Table 12. Hardware metrics results of the proposed chaos-based stream ciphers.

			Chaos-Based Stream Ciphers		
			LSP-SC	**LST-SC**	**LSPT-SC**
Resources used	Area	LUTs	10,420/19.59%	3549/6.67%	11,416/21.46%
		FFs	448/0.42%	531/0.50%	577/0.54%
		Slices	3160/23.71%	1049/7.89%	3385/25.45%
	DSPs		13/5.91%	25/11.36%	28/12.73%
Speed	WNSi (ns)		0.050	0.060	0.149
	Ti (ns)		30.90	27.20	32.40
	Max. Freq. (MHz)		32.41	36.84	31.00
	Throughput (Mbps)		1037.27	1179.07	992.21
Efficiency (Mbps/Slices)			0.32	1.12	0.29
Power (W)			0.152	0.099	0.182

Table 13. Comparison of hardware metrics usage among various chaotic and non-chaotic systems.

Cipher	**Device**	**Frequency (MHz)**		**Slices**	**Throughput (Mbps)**	**Efficiency (Mbps/Slices)**	**Power (W)**
		Clock Frequency	**Max. Freq.**				
LSP-SC	PYNQ-Z2	125	32.41	3160	1037.27	0.32	0.152
LST-SC	PYNQ-Z2	125	36.84	1049	1179.07	1.12	0.099
LSPT-SC	PYNQ-Z2	125	31.00	3385	992.21	0.29	0.182
LST_RC-SC [7]	PYNQ-Z2	125	36.78	1186	1177.20	0.99	0.101
LWCB SC [20]	Zynq-7000	-	18.5	2363 LUTs	565	-	-
Lorenz's chaotic System [21]	Virtex-II	50	15.598	1926	124	0.06	-
Chaos-ring [22]	Virtex-6	125	464.688	1050	464.688	0.44	-
Trivium [45]	Spartan 3	50	190	388	12,160	31.34	-
Grain-128 [46]	Virtex- II	50	181	48	181	3.77	-
Mickey-128 [46]	Virtex- II	50	200	190	200	1.05	-
ZUC stream cipher	PYNQ-Z2	125	79.52	246	2544.53	10.34	0.23

Note: Nowadays, promising new fractional chaotic functions have appeared, which seem very robust against cryptographic attacks. We believe they deserve further study to build chaotic fractional systems for data security. The design of such fractional chaotic systems must deal with the trade-offs between safety, performance, and cost.

4. Conclusions

In this paper, we evaluate the statistical security and hardware metrics on a Xilinx PYNQ-Z2 FPGA board using VHDL of some chaotic maps, which are the basic components of the proposed PRNGs-CS. We first demonstrated that the studied chaotic maps have good hardware metrics but cannot be used alone as a secure PRNG. They are weak against statistical attacks and have recognized attractors. Furthermore, we have shown that XORing the output of a chaotic map with an LFSR improves the cryptographic properties of the coupled system. We then studied and implemented a new chaos-based stream cipher (SC) based on a proposed secure PRNGs-CS. The proposed chaotic systems use a weakly coupled matrix that prevents divide-and-conquer attacks on the initial vector (IV) and improves the randomness of the generated sequences. We then evaluated the cryptographic features of the proposed PRNGs-CS and analyzed their performance on hardware metrics. The achieved results show a high level of security on the one hand and good hardware metrics achieved by the proposed PRNGs-CS on the other hand. We then realized corresponding chaos-based stream ciphers and confirmed their resistance to cryptanalysis attacks. We also evaluated their hardware performance and compared them to several chaotic and non-chaotic systems. All the obtained results demonstrate that the proposed chaos-based stream ciphers are alternative candidates for data confidentiality and integrity.

Author Contributions: Conceptualization, F.D. and S.E.A.; formal analysis, F.D. and S.E.A.; methodology, F.D.; writing—original draft, F.D.; writing—review & editing, F.D. and S.E.A.; validation, S.E.A., M.M., and W.E.H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

Appendix A. Original Equations of Chaotic Maps Used

The logistic map is given by:

$$x(n) = f(x(n-1), r) = r \times x(n-1) \times (1 - x(n-1)) \tag{A1}$$

with $0 < x(n) \leq 1$ and $r \in [1, 4]$ the control parameter.

The Skew-Tent map is given by:

$$x(n) = f(x(n-1), p) = \begin{cases} \frac{x(n-1)}{p} & if\ 0 \leq x(n-1) \leq p \\ \frac{1-x(n-1)}{1-p} & if\ p < x(n-1) \leq 1 \end{cases} \tag{A2}$$

where $0 < x(n) \leq 1$ and p is the control parameter of the chaotic map, which varies in the following interval: $0 < p < 1$.

The PWLCM is described by:

$$x(n) = f(x(n-1), p) = \begin{cases} \frac{x(n-1)}{p} & if\ 0 \leq x(n-1) < p \\ \frac{[x(n-1)-p]}{0.5-p} & if\ p \leq x(n-1) < 0.5 \\ f(1 - x(n-1), p) & otherwise \end{cases} \tag{A3}$$

where $0 < x(n) \leq 1$ and p the control parameter: $0 < p < 0.5$.

The standard 3D Chebyshev map is given by:

$$x(n) = f(x(n-1)) = 4[x(n-1)]^3 - 3x(n-1) \tag{A4}$$

with $x(n) \in [-1, 1]$.

References

1. eSTREAM. eSTREAM: The ECRYPT Stream Cipher Project. 2012. Available online: https://www.ecrypt.eu.org/stream/ (accessed on 14 January 2019).
2. Robshaw, M. The eSTREAM project. In *New Stream Cipher Designs*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–6. [CrossRef]
3. Manifavas, C.; Hatzivasilis, G.; Fysarakis, K.; Papaefstathiou, Y. A survey of lightweight stream ciphers for embedded systems. *Secur. Commun. Netw.* **2016**, *9*, 1226–1246. [CrossRef]
4. ETSI/SAGE. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification, Version: 1.5. Technical Report, ETSI. 2011. Available online: http://www.gsmworld.com/documents/EEA3_EIA3_ZUC_v1_5.pdf (accessed on 17 December 2021).
5. Sulaiman, A.G. Overview of ZUC Algorithm and its Contributions on the Security Success and Vulnerabilities of 4G Mobile Communication. *Int. J. Comput. Appl.* **2017**, *975*, 8887.
6. Ming, T.; PingPan, C.; ZhenLong, Q. Differential Power Analysis on ZUC Algorithm. Cryptol. Eprint Arch. 2012. Available online: https://eprint.iacr.org/2012/299 (accessed on 14 January 2023).

7. Dridi, F.; El Assad, S.; El Hadj Youssef, W.; Machhout, M.; Lozi, R. The Design and FPGA-Based Implementation of a Stream Cipher Based on a Secure Chaotic Generator. *Appl. Sci.* **2021**, *11*, 625. [CrossRef]
8. Pareek, N.K.; Patidar, V.; Sud, K.K. Image encryption using chaotic logistic map. *Image Vis. Comput.* **2006**, *24*, 926–934. [CrossRef]
9. François, M.; Grosges, T.; Barchiesi, D.; Erra, R. Pseudo-random number generator based on mixing of three chaotic maps. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 887–895. [CrossRef]
10. Wang, X.Y.; Zhang, J.J.; Zhang, F.C.; Cao, G.H. New chaotical image encryption algorithm based on Fisher–Yatess scrambling and DNA coding. *Chin. Phys. B* **2019**, *28*, 040504. [CrossRef]
11. Belazi, A.; Abd El-Latif, A.A.; Belghith, S. A novel image encryption scheme based on substitution-permutation network and chaos. *Signal Process.* **2016**, *128*, 155–170. [CrossRef]
12. Datcu, O.; Macovei, C.; Hobincu, R. Chaos based cryptographic pseudo-random number generator template with dynamic state change. *Appl. Sci.* **2020**, *10*, 451. [CrossRef]
13. Acho, L. A chaotic secure communication system design based on iterative learning control theory. *Appl. Sci.* **2016**, *6*, 311. [CrossRef]
14. Abdoun, N.; El Assad, S.; Manh Hoang, T.; Deforges, O.; Assaf, R.; Khalil, M. Designing Two Secure Keyed Hash Functions Based on Sponge Construction and the Chaotic Neural Network. *Entropy* **2020**, *22*, 1012. [CrossRef]
15. Battikh, D.; El Assad, S.; Hoang, T.M.; Bakhache, B.; Deforges, O.; Khalil, M. Comparative Study of Three Steganographic Methods Using a Chaotic System and Their Universal Steganalysis Based on Three Feature Vectors. *Entropy* **2019**, *21*, 748. [CrossRef] [PubMed]
16. Liao, T.L.; Wan, P.Y.; Yan, J.J. Design of synchronized large-scale chaos random number generators and its application to secure communication. *Appl. Sci.* **2019**, *9*, 185. [CrossRef]
17. François, M.; Grosges, T.; Barchiesi, D.; Erra, R. A new image encryption scheme based on a chaotic function. *Signal Process. Image Commun.* **2012**, *27*, 249–259. [CrossRef]
18. Alippi, C.; Bogdanov, A.; Regazzoni, F. Lightweight cryptography for constrained devices. In Proceedings of the 2014 International Symposium on Integrated Circuits (ISIC), Singapore, 10–12 December 2014; pp. 144–147.
19. Thakor, V.A.; Razzaque, M.A.; Khandaker, M.R. Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities. *IEEE Access* **2021**, *9*, 28177–28193. [CrossRef]
20. Gautier, G.; Le Glatin, M.; El Assad, S.; Hamidouche, W.; Déforges, O.; Guilley, S.; Facon, A. Hardware Implementation of Lightweight Chaos-Based Stream Cipher. In Proceedings of the International Conference on Cyber-Technologies and Cyber-Systems, Porto, Portugal, 22–26 September 2019; p. 5.
21. Tanougast, C. Hardware implementation of chaos based cipher: Design of embedded systems for security applications. In *Chaos-Based Cryptography*; Springer: Berlin, Germany, 2011; pp. 297–330. [CrossRef]
22. Koyuncu, I.; Tuna, M.; Pehlivan, I.; Fidan, C.B.; Alçın, M. Design, FPGA implementation and statistical analysis of chaos-ring based dual entropy core true random number generator. *Analog. Integr. Circuits Signal Process.* **2020**, *102*, 445–456. [CrossRef]
23. Sambas, A.; Vaidyanathan, S.; Bonny, T.; Zhang, S.; Hidayat, Y.; Gundara, G.; Mamat, M. Mathematical model and FPGA realization of a multi-stable chaotic dynamical system with a closed butterfly-like curve of equilibrium points. *Appl. Sci.* **2021**, *11*, 788. [CrossRef]
24. Vaidyanathan, S.; Sambas, A.; Abd-El-Atty, B.; Abd El-Latif, A.A.; Tlelo-Cuautle, E.; Guillén-Fernández, O.; Mamat, M.; Mohamed, M.A.; Alçin, M.; Tuna, M.; et al. A 5-D multi-stable hyperchaotic two-disk dynamo system with no equilibrium point: Circuit design, FPGA realization and applications to TRNGs and image encryption. *IEEE Access* **2021**, *9*, 81352–81369. [CrossRef]
25. Ding, L.; Liu, C.; Zhang, Y.; Ding, Q. A new lightweight stream cipher based on chaos. *Symmetry* **2019**, *11*, 853. [CrossRef]
26. Abdelfatah, R.I.; Nasr, M.E.; Alsharqawy, M.A. Encryption for multimedia based on chaotic map: Several scenarios. *Multimed. Tools Appl.* **2020**, *79*, 19717–19738. [CrossRef]
27. Deb, S.; Bhuyan, B. Chaos-based medical image encryption scheme using special nonlinear filtering function based LFSR. *Multimed. Tools Appl.* **2021**, *80*, 19803–19826. [CrossRef]
28. Zheng, J.; Hu, H. A highly secure stream cipher based on analog-digital hybrid chaotic system. *Inf. Sci.* **2022**, *587*, 226–246. [CrossRef]
29. Shujun, L.; Xuanqin, M.; Yuanlong, C. Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography. In Proceedings of the International conference on cryptology in India, Chennai, India, 16–20 December 2001; pp. 316–329. [CrossRef]
30. Dridi, F.; El Assad, S.; El Hadj Youssef, W.; Machhout, M.; Lozi, R. Design, Implementation, and Analysis of a Block Cipher Based on a Secure Chaotic Generator. *Appl. Sci.* **2022**, *12*, 9952. [CrossRef]
31. Jakimoski, G.; Kocarev, L. Chaos and cryptography: Block encryption ciphers based on chaotic maps. *IEEE Trans. Circuits Syst.* **2001**, *48*, 163–169. [CrossRef]
32. Xiao, D.; Liao, X.; Deng, S. Parallel keyed hash function construction based on chaotic maps. *Phys. Lett. A* **2008**, *372*, 4682–4688. [CrossRef]
33. Wu, X.; Guan, Z.H. A novel digital watermark algorithm based on chaotic maps. *Phys. Lett. A* **2007**, *365*, 403–406. [CrossRef]
34. Mooney, A. Chaos Based digital watermarking. In *Intelligent Computing Based on Chaos*; Springer: Berlin, Germany, 2009; pp. 315–332. [CrossRef]
35. Verhulst, P.F. Recherches mathématiques sur la loi d'accroissement de la population. *J. Écon.* **1845**, *12*, 276.

36. Ulam, S.M. On combination of stochastic and deterministic processes. *Bull. Am. Math. Soc.* **1947**, *53*, 1120.
37. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; Technical Report; Booz-Allen and Hamilton Inc.: Mclean, VA, USA, 2001. Available online: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762 (accessed on 12 September 2019).
38. Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
39. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. *Cyber J.* **2011**, *1*, 31–38.
40. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.P.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. [CrossRef]
41. Maximov, A.; Biryukov, A. Two trivial attacks on Trivium. In Proceedings of the International Workshop on Selected Areas in Cryptography, Ottawa, ON, Canada, 16–17 August 2007; pp. 36–55. [CrossRef]
42. AlMashrafi, M.J. A different algebraic analysis of the ZUC stream cipher. In Proceedings of the 4th International Conference on Security of Information and Networks, Surathkal, India, 16–18 December 2011; pp. 191–198. [CrossRef]
43. Wu, H.; Huang, T.; Nguyen, P.H.; Wang, H.; Ling, S. Differential attacks against stream cipher ZUC. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 5–7 December 2012; pp. 262–277. [CrossRef]
44. Lafitte, F.; Markowitch, O.; Van Heule, D. SAT based analysis of LTE stream cipher ZUC. In Proceedings of the 6th International Conference on Security of Information and Networks, Aksaray, Turkey, 26–28 November 2013; pp. 110–116. [CrossRef]
45. Gaj, K.; Southern, G.; Bachimanchi, R. Comparison of hardware performance of selected Phase II eSTREAM candidates. In Proceedings of the State of the Art of Stream Ciphers Workshop (SASC 2007), Ottawa, ON, Canada, 16–17 August 2007; Volume 26, p. 2007.
46. Bulens, P.; Kalach, K.; Standaert, F.X.; Quisquater, J.J. FPGA implementations of eSTREAM phase-2 focus candidates with hardware profile. In Proceedings of the State of the Art of Stream Ciphers Workshop (SASC 2007), Ottawa, ON, Canada, 16–17 August 2007; Volume 24, p. 2007.