Research Article

Beyond Statistical Analysis in Chaos-Based CSPRNG Design

John Prakash Arockiasamy ¹, **Lydia Elizabeth Benjamin** ²,
and **Rhymend Uthariaraj Vaidyanathan**¹

¹*Ramanujan Computing Centre, Anna University, Chennai 600025, India*

²*Department of IT, Anna University, Chennai 600044, India*

Correspondence should be addressed to Lydia Elizabeth Benjamin; lydiabenjamin@gmail.com

Received 25 February 2021; Revised 11 May 2021; Accepted 26 May 2021; Published 10 June 2021

Academic Editor: Vincenzo Conti

Copyright © 2021 John Prakash Arockiasamy et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The design of cryptographically secure pseudorandom number generator (CSPRNG) producing unpredictable pseudorandom sequences robustly and credibly has been a nontrivial task. Almost all the chaos-based CSPRNG design approaches invariably depend only on statistical analysis. Such schemes designed to be secure are being proven to be predictable and insecure day by day. This paper proposes a design and instantiation approach to chaos-based CSPRNG using proven generic constructions of modern cryptography. The proposed design approach with proper instantiation of such generic constructions eventually results in providing best of both worlds that is the provable security guarantees of modern cryptography and passing of necessary statistical tests as that of chaos-based schemes. Also, we introduce a new coupled map lattice based on logistic-sine map for the construction of CSPRNG. The proposed pseudorandom number generator is proven using rigorous security analysis as that of modern cryptography and tested using the standard statistical testing suites. It is observed that the generated sequences pass all stringent statistical tests such as NIST, Dieharder, ENT, and TestU01 randomness test suites.

1. Introduction

Cryptographically secure pseudorandom number generator (CSPRNG) efficiently generates sequences that cannot be distinguished from random sequences by (computationally) efficient adversaries. The number of hardware and software implementations of CSPRNG based on chaotic maps has increased recently along with chaos-based cryptosystems. Chaotic maps are mathematical functions that exhibit random or chaotic behaviour that is hard to predict. The idea is to use the behaviour of such chaotic maps to produce chaotic sequences that are disordered, unpredictable, and sensitive to the initial conditions. However, the designers of such chaos-based cryptosystems claim security by statistical analysis of the topological properties of disorder as defined by the mathematical theory of chaos. More often, there is no rigorous provable security methodology that bridges the gap between statistical analysis of chaotic maps topological properties and security guarantees unlike modern cryptography. As a result, the use of chaos-based cryptosystems is

disputed, especially for cryptographic applications, and they are often shown to be flawed, with such failures often attributed to the use of nonrigorous empirical-only methodology in the design process [1]. Modern cryptography instead creates cryptographic primitives by instantiating tried and tested over time standard generic constructions (such as Goldreich–Levin construction, Feistel structure, SPN, counter mode, and sponge construction) with new computationally efficient mathematical functions. We know that such designs drawing strength and credibility from the foundational generic constructions have been resilient to attacks for over sufficient period of time. However, in chaos-based cryptosystems, more often bespoke generic constructions are designed with standard candidate (mathematical) chaotic functions leading to faulty designs. This is largely due to lack of rigorous analysis treatment to bespoke generic constructions used to design cryptographic primitives unlike new instantiations of time-tested generic construction approach of modern cryptography. Therefore, the need to bridge the gap between such chaos-based cryptosystem designs and modern

cryptography to provide credible and robust cryptographic primitive designs is pertinent. This paper attempts to bridge this gap through demonstration of provably secure chaos-based CSPRNG design denoted G . We show in this paper that the sequences generated are computationally indistinguishable and hard to predict in the presence of efficient adversaries using modern cryptography design tools. Also, we show that such sequences pass all necessary statistical analysis tests that are required to a chaos-based cryptosystem design. Therefore, the design approach advocates to consider statistical analysis as a necessary condition and mathematical simulation-based proofs as sufficient condition for credible chaos-based cryptographic primitive designs. Provable CSPRNG constructions instantiated with chaotic maps that they will pass all efficient statistical tests are hardly proposed to the best of our knowledge.

Generally, pseudorandom sequences generated from a potential CSPRNG G passes a subset of all statistical tests. The question of whether there exists a mechanism to determine the existence of a potential pseudorandom number generator G passes all efficient statistical tests is provided by the abstract notion of computationally indistinguishable (denoted $\approx_c$) formalized in [2]. The abstract notion of computationally indistinguishable is proven in [2] to be equivalent to the abstract notion of unpredictability (denoted $\approx_p$). The notion of computationally indistinguishable requires nonexistence of efficient (computationally bounded) distinguisher (D) which can distinguish samples taken from two different distributions. In the same way, the notion of unpredictability requires nonexistence of efficient (computationally bounded) predictor (P) which can predict the distribution from which a sample was taken given two sample distributions.

Based on the notion of computationally indistinguishable, one-way functions (OWFs) [3, 4] or unpredictable functions (UFs) are designed based on assumptions that $P \neq NP$ and intractable problems. Furthermore, based on the assumption that OWFs or UFs exist, generic pseudorandom generator constructions are proposed [5–7]. The generic constructions provide a construction framework when instantiated with provable one-way functions or unpredictable functions can be composed to design pseudorandom number generator G [7]. These abstract notions provides provability that such instantiated constructions will pass all efficient statistical tests and provably unpredictable. The chaotic maps when operated in the region of chaos or sensitive dependence on initial conditions (SDIC) exhibit function solutions or trajectories which are unpredictable and uniformly distributed in the state space. The design of PRNGs using chaotic maps have provided many efficient candidates for G . Moreover, such G using chaotic maps are entropy sources and generate sequences which look random and provide guarantees for conducted experiments but no guarantees for passing of all efficient statistical tests.

Yao [2] showed that the notion of unpredictability implies pseudorandomness. It was then shown that unpredictability ($\approx_p$) for bit sequences is equivalent to pseudorandomness with the notion of computationally indistinguishable ($\approx_c$). It is under the premise that

existence of efficient predictor (P) to predict bit sequences from unpredictable functions can be used to construct efficient distinguisher (D) to look apart from uniformly random sequences U . Also, unpredictable functions imply one-way functions from which pseudorandom functions can be constructed. It is known that modern cryptography constructions from one-way functions to PRNGs are provably secure but take a toll on computational efficiency. This fact makes pseudorandom generators constructed from unpredictable functions to PRNGs using chaotic maps as unpredictable functions relatively efficient [8].

This kind of modern cryptography design approach in chaos cryptography can lead to credible and robust designs. The proven modern cryptographic constructions such as Merkle–Damgard, sponge construction, and block cipher modes can be used by instantiating such proven constructions with suitable chaos-based functions. Such design approach will reduce the reliance of security assessment methods on statistical analysis. Moreover, statistical analysis is performed only on the produced output of cryptographic algorithms such as on ciphertexts obtained from an encryption mechanism, hash or message digest obtained from a hash algorithm, and pseudorandom bits obtained from a pseudorandom number generator. Also, statistical tests conducted only on the output of cryptographic algorithms generally do not capture the notion of attackers' potential, attackers' knowledge of the algorithm (Kerckhoffs' principle), and attackers' capabilities to interact or query the algorithm. Cryptographic algorithms are often vulnerable in real-world situations due to such attackers' characteristics. Therefore, beyond statistical analysis, the design approach for chaos-based cryptographic design should be based more on instantiating proven constructions with chaotic maps as unpredictable functions rather designing new constructions on each proposed chaotic cryptographic algorithm.

Considering all the above factors, we demonstrate a design approach by instantiating a proven modern cryptographic PRNG construction with a new chaotic map based on coupled map lattices, prove its security using modern cryptographic attack models, and perform statistical analysis on the output. The proposed design approach can also be used for designing other chaos-based cryptographic primitives with proper instantiation of the chaotic maps. The chaotic maps should satisfy the conditions or properties as required by the choice of construction and security model to prove the security. For example, the choice of construction in this work requires that the instantiated chaotic map to be a unpredictable function, and hence, a map known to be noninvertible is used in the composition. Similarly, other chaotic maps can be used in the proposed approach depending on the choice of construction. PRNGs are required to produce output sequence of bits which are distributed almost uniformly. However, for a CSPRNG, the output sequence is required to have negligible probability to be distinguished by an adversary not knowing the initial seed from random sequence besides passing all polynomial-time (in respect of size of the seed) restricted statistical tests. Such CSPRNG can be used in stream ciphers and block ciphers as a submodule as required by the design. The main contributions of this paper are as follows:

- (1) We instantiate Goldreich–Levin generic construction with a new deterministic discrete coupled map lattice (δ)
- (2) We prove through theoretical security analysis methodology using modern cryptography tools that δ is an unpredictable function and subsequently, we prove the pseudorandomness of construction $\hat{G}$ as required by modern cryptography
- (3) The sequences generated using the proposed $\hat{G}$ are tested using standard statistical test suites and show that the sequences indeed pass all statistical analysis tests as required by chaos cryptography

1.1. Organization. In Section 2, we summarize the related works. The preliminaries and definitions are presented in Section 3. The construction and instantiation of the proposed CSPRNG $\hat{G}$ is presented in Section 4. The theoretical security analysis of the proposed construction using computational indistinguishability properties leading to security of the CSPRNG is presented in Section 5. The experimental tests for randomness using various statistical testing suites are given in Section 6. Section 7 discusses about various implementation issues. We conclude with final remarks in Section 8.

2. Related Works

Entropy of nondeterministic source is used as input to the true random number generator (TRNG) that induces complex and unpredictability properties. For instance, mouse movements or electrical signal noise and bioelectrical signals can be utilized for TRNG process [9–13]. On the other hand, PRNG computed by a deterministic algorithm makes use of a complex mathematical formula in order to remain deterministic and yet unpredictable [14].

In recent years, chaos has been used extensively in computing cryptographically secure PRNGs as it is computationally infeasible to predict the preceding bits despite the visible part of the output dynamics. In spite of the hype surrounding the chaos-based PRNGs, there are many shortcomings of this technique which make them insecure to be used in cryptographic systems. The selection of the chaotic system is a indispensable problem in the generation of the pseudorandom bits. Various single-dimensional chaotic maps such as logistic map, multimodal maps, Tinkerbell map, quantum logistic map, and piecewise logistic map are used in the design of PRNGs [14–17]. García-Martínez and Campos-Cantón [14] employed a unidimensional multimodal discrete map in computing the pseudorandom bits. The positive and negative values of the multimodal map yield a complex sequence with long periods. Furthermore, the complex sequences are tested using the NIST statistical testing suite for its credibility. However, the statistical test showcased a number of weak keys causing improper functioning of the generator [18]. Wang et al. [19] present a PRNG based on a piecewise logistic map. Furthermore, the sequence generated from the map is

transformed into PRNGs based on the S-box of AES. The author claims that the generated PRNGs have good statistical properties and no shorter periods. In spite of the claims, the PRNG was found to be insecure where the auxiliary mechanism used in computing the control parameter of the system is exploited [20].

Xu et al. [16] proposed an algorithm to produce binary sequence that has three layers. The top layer consists of linear feedback shift register (LFSR) of 16 cells, the bit-reorganization (BR) at the middle, and a nonlinear function F in the bottom. The two chaotic maps are used as a nonlinear element to avoid the finite precision problem. However, PRNG does not enhance its security since the attacker is able to reconstruct the secret key after six iterations [21]. Hu et al. [11] proposed PRNGs based on quantum logistic map, whose randomness was merely based on the chaotic equation involved. The PRNG is subjected to various statistical tests on randomness using testing suites such as ENT, NIST, Dieharder, and TestU01. Although the author claims sufficient randomness, the PRNG has serious security implications. Degradation of security has emerged due to the poor selection of control parameter and the resulting secret key that leads to nonchaotic behaviour of the quantum logistic map [22]. Sahari and Boukemara proposed a 3D chaotic map by coupling two maps of piecewise and logistic maps in order to overcome the weaknesses of one-dimensional map. The PRNG has cryptographic statistical properties such as the simplicity, ergodicity, and higher sensitivity. Additionally, as generation process involves only multiplications and additions, realization by both hardware and software was made easy [23]. Fractional-order chaotic systems such as [24] are dynamical systems which rely on functional calculus and modelled as fractional differential equations. They have the potential to exhibit chaos with low order as low as 0.3. Montero-Canela et al. [25] proposed a PRNG as part of their cryptosystem design using fractional-order chaotic system specifically designed for Internet of Things. Though the work proves the properties of dynamics supported with standard statistical analysis, formal security proof capturing the attacker and attacks is not provided.

Another issue which is profoundly identified in the chaotic PRNG is the implementation of generators with finite precision [21, 26–29]. A number of studies have been conducted to solve the problem of dynamical degradation of the chaotic system caused by the data sequence with shorter period. Thus, the security of the cryptographic applications may become completely insecure. Flores-Vergara et al. [30] proposed a novel chaotic PRNG as an alternative to diminish the dynamical degradation of the chaotic dynamics. This method generates pseudorandom numbers with the double precision specified by the IEEE 754 standard for floating-point arithmetic. Nevertheless, the computational speed of the system is significantly influenced. François et al. [31] proposed a solution with the use of more than one chaotic map in cascade. The PRNG consists of mixing three chaotic maps generated from an input initial vector; however, it increases the overhead cost. Perturbation using a nonlinear element was one of the prospective ideas deployed to weed out the influence of shorter period. Dastgheib and Farhang

[32] developed a multiple recursive generator based on a digitized sawtooth map. It is demonstrated that this map in a recursive structure and a tiny perturbation can potentially cause unpredictable longer periods in a finite precision. However, the aforementioned solutions such as combining multiple chaotic maps, holding higher precision, and perturbation methods can improve the average length of the period and short trajectories cannot be avoided indefinitely. It can be inferred from the above discussion that unlike cryptographic system, the chaos-based PRNGs lack instantiations based on mathematically proven constructions guarantee satisfiability of all required cryptographic statistical properties. However, chaos provides many potential functions for proven generic construction instantiations, but only the process of exploiting the right function with right properties has been nontrivial.

3. Preliminaries and Notations

The choice of the dynamical system in this paper is limited to discrete dynamical functions as in [33]. A parametric discrete time dynamical system is a tuple denoted by $\mathcal{D} = (\mathcal{X}, \mathcal{P}, \delta)$ where $\mathcal{X} \subseteq \mathbb{R}^n$ is the state space, $\mathcal{P} \subseteq \mathbb{R}^m$ is the parameter space, and $\delta: \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{X}$ is a function. The orbit or trajectory formed by such a discrete dynamical functions is denoted by $\mathcal{O}$. For $\delta: \mathcal{X} \rightarrow \mathcal{X}$ a map, a forward orbit is defined by the set $\mathcal{O}_x^+ = \{y \in \mathcal{X} | y = \delta^n(x), n \in \mathbb{N}\}$. If δ is invertible, then a backward orbit is defined by the set $\mathcal{O}_x^- = \{y \in \mathcal{X} | y = \delta^{-n}(x), n \in \mathbb{N}\}$. A full orbit is defined by $\mathcal{O}_x = \{y \in \mathcal{X} | y = \delta^n(x), n \in \mathbb{N}\}$. For $t \geq 0$, the time $-t$ map of a continuous dynamical system is the transformation of state space which takes $x_0, x_1 \leftarrow \delta(x_0), x_2 \leftarrow \delta^2(x_1), \dots, x_n \leftarrow \delta^n(x_{n-1})$. The study of qualitative behaviour of the dynamical system is thus the analysis of the orbits formed by δ . The orbits could be periodic, eventually periodic, asymptotic, and more. To get rich complex behaviour, functions such as δ are formed as an array of cells called coupled map lattices (CMLs) similar to cellular automata arrays. The cells or array elements are called lattices. The CML exhibits spatiotemporal dynamics in discrete time in which each lattice executes a δ and $x \leftarrow \delta(x)$ is coupled to its nearest lattice neighbours or coupled dynamically based on a configuration, denoted in the form $\delta^{n+1,i} = x_{n+1}^{(i)} = (1 - \epsilon)\delta(x^{(i)}) + (\epsilon/2)[\delta(x_n^{(i-1)}) + \delta(x_n^{(i+1)})]$.

In such a CML, the lattices are ordered on an $l \times n$ grid and each lattice can be considered as a function $\delta: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$. A configuration of the CML denoted as $\mathcal{D}$ consists of real assignments to each of its lattices, ϵ and parameters for δ if any. A coupled map lattice (CML) as in [34–37] is denoted by a pair $(\mathcal{X}, \delta)$, where $\mathcal{X}$ is a continuous set of states, and $\delta: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$ is a transition function. A configuration of a CML, at time $t \in \mathbb{Z}$, $\mathcal{D}^{(t)}: \mathbb{Z} \rightarrow \mathcal{X}$, is an assignment of a unique state to each lattice, where a lattice is represented by an integer. $\mathcal{D}^{(i,t)}$ is the state of lattice i at time t is defined by $\mathcal{D}^t = \{t \in \mathbb{Z}, x \in \mathcal{X}, p \in \mathcal{P}, \delta\}$, and a configuration at time t uniquely determines a configuration at time $t+1$ as $\mathcal{D}^{(t+1)} = \{(i, q): q = \delta(\mathcal{D}^{(t-1,t)}, \mathcal{D}^{(i,t)}, \mathcal{D}^{(i+1,t)}), i \in \mathbb{Z}\}$ represented succinctly as $\mathcal{D}^{(t+1)} = \delta(\mathcal{D}^{(t)})$ while $\mathcal{D}^{(0)}$ denotes an initial configuration. The dynamical

system $\mathcal{D}$ with an initial configuration $\mathcal{D}^0$ computes to reach a destination configuration $\mathcal{D}^t$, creating an orbit of points given by $x_0, x_1 \leftarrow \delta(x_0), x_2 \leftarrow \delta^2(x_1), \dots, x_n \leftarrow \delta^n(x_{n-1})$, and this notion of computability is formalized using [37] as follows.

Definition 1. Computable: a computation of a CML of length n is a sequence of configurations $\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \dots, \mathcal{D}^{(n-1)}$. If $\mathcal{D}^i$ and $\mathcal{D}^j$, $i \leq j$ are two configurations in a computation, then $\mathcal{D}^i \vdash^* \mathcal{D}^j$ or $\mathcal{D}^j = \delta^t(\mathcal{D}^i)$ represents to say that $\mathcal{D}^i$ computes $\mathcal{D}^j$. Also, we say that $\mathcal{D}^i \vdash^t \mathcal{D}^j$ when a configuration $\mathcal{D}^i$ computes $\mathcal{D}^j$ in t time steps. The probability that it is computable is then $\Pr[\mathcal{D}^j \leftarrow \delta^t(\mathcal{D}^i)] = 1$.

We often for brevity denote $x^t \vdash^P x^{t+P}$, if $\mathcal{D}^t \vdash^P \mathcal{D}^{t+P}$ by $x_a \vdash^t x_b$ in the context when configuration can be inferred and computable. In a complex dynamical system, the problem of determining the preimage is formalized [37] as follows.

Definition 2. Preimage problem: given a fixed dynamical system $\mathcal{D}$ and a configuration substring x determine whether there is a configuration substring x_0 such that $x_0 \vdash^k x$ where k is the time steps and $x_0 \leftarrow \{x_0: \mathcal{D}, k, x, x_0 \vdash^k x\}$.

A deterministic dynamical system does not always exhibit deterministic behaviour but rather exhibit chaotic or unpredictable behaviour or sensitive dependence to certain initial conditions or parameters. From [33], a deterministic dynamical system is said to be chaotic if δ has sensitive dependence on initial conditions, δ is topologically transitive, and periodic points are dense. From the multitude of definitions, Devaney [33] formalizes chaos using theory of topology and the qualitative behaviour of orbits in region of chaos for a $\mathcal{D}$ given that conditions for chaos exists, such as $x_0 \vdash^k x_t$ is unpredictable. This implication of chaos is formalized by the notion of approximately probabilistically irrelevant using measure theory defined in [38] is given below.

Definition 3. Approx. probabilistically irrelevant: X^{t_n} is approximately probabilistically irrelevant for predicting Y^{t_m} , ($t_m \geq t_n$) at level $\epsilon > 0$ iff $\Pr[Y^{t_m} | X^{t_n}] - \Pr[Y^{t_m}] < \epsilon$.

The intuition behind the above definition under the assumption that $\mathcal{D}$ is operating in its region of chaos, given x_0 predicting x_t is negligible. The condition for predictability can be defined in terms of Lyapunov stability. A discrete deterministic dynamical system is said to be Lyapunov stable if two different orbits whose initial conditions are sufficiently small and progress in time arbitrarily close to each other towards infinity. The system as in [39] is given by

$$\forall \epsilon_t > 0, \exists \epsilon_0, \forall x, x' \in \mathcal{X}, \forall t \geq t_0: |x_0 - x'_0| < \epsilon_0 \rightarrow |x_t - x'_t| < \epsilon_t, \quad (1)$$

where ϵ_0 is the lower bound on the level of measurement accuracy error of initial conditions and ϵ_t is the practical lower bound on the tolerable error in the measurement accuracy of the predicted destination state. Given the preliminaries, the next section describes the constructions of a

pseudorandom number generator applying these results or notions of dynamical systems and chaos.

4. Construction and Instantiation

A general approach towards construction of efficient encryption mechanism is used to generate a short random key k and then expand k using a pseudorandom generator $G(k)$, into a longer sequence g that looks random, and use g as the key in one-time pad encryption as $c = x \oplus k$ since key generated from a truly random source such that $|x| = |k|$ is not practical. The existence of such PRNGs $g \leftarrow G(k)$ is a contradiction to Shannon's theorem [40] assuming a computationally unbounded malicious adversary. However, the probability of a computationally unbounded adversary is almost negligible. Therefore, without contradicting Shannon's theorem, it is surmised that the adversary is computationally bounded or efficient and also it is conjectured that $G(k)$ exists if $P \neq NP$ [2]. Assuming $P \neq NP$, the class of hard problems is used in cryptographic applications such as PRNG for generating sufficiently random looking sequences for cryptographic applications.

Let $\hat{G}$ be a pseudorandom generator as formalized in [41] that generates pseudorandom sequences which are sufficiently random. Let δ be a unpredictable function and let hc be a hard-core predicate of δ defined as $hc(x, r) = \bigoplus_{i=1}^n x_i \cdot r_i$, where $x = x_1 \dots x_n$ and $r = r_1 \dots r_n$. Then, $G(s) = \delta(s) \parallel hc(s)$ is a pseudorandom generator with expansion factor $l(n) = n + 1$. The construction of $\hat{G}$ involves parameter generation and construction which uses a coupled map lattice.

4.1. Parameter Generation. The deterministic discrete dynamical system $\mathcal{D} = \{\mathcal{X}, \mathcal{P}, \delta\}$ is chosen as follows:

- (1) Let the state space be $\mathcal{X} \leftarrow \{x: x \in \mathbb{R}, x \in (0, 1)\}$.
- (2) The function δ is the proposed one-dimensional coupled map lattice (CML) based on logistic-sine map given by

$$\delta_1^{n+1}(x) = \left(\alpha x_n (1 - x_n) + (4 - \alpha) \frac{\sin(\pi x_n)}{4} \right) \bmod 1,$$

$$\delta_2^{n+1}(x) = \delta_1^{n+1}(x),$$

$$\delta^{n+1} = \delta_1(x_n^j) + \frac{\epsilon}{2} [\delta_2(x_n^{j+1}) + \delta_2(x_n^{j-1}) - 2\delta_2(x_n^j)],$$

where $\alpha \in \mathbb{R}$ is control parameter, x_0^j is the initial state for lattice $j \in \mathbb{Z}$, and ϵ is coupling coefficient. The choice of CML being composed of logistic-sine map lattices is to have a well-known noninvertible map which can be scaled and be able to scale the number of pseudorandom bits.

- (3) Let hc denote a hard-core predicate of δ^j of the j^{th} lattice given by

$$hc(x^j) = \bigoplus_{i=1}^n x_i^j \cdot r_i, \quad (4)$$

where $x = x_1 \dots x_n, r = r_1 \dots r_n, j = 1 \dots m$ and m being the number of lattices in the δ_3 .

- (4) If δ is an unpredictable function and hc a hard-core predicate of δ , then a pseudorandom number generator can be constructed for each lattice. We prove in Section 5 that δ is indeed unpredictable.
- (5) The choice of $p \in \mathcal{P}$ is given by $p = \{x_0, k, \alpha, r\}$ which are chosen such that

$$\forall t \geq t_p \forall x, x' \in \mathcal{X}: |x_0 - x'_0| < \epsilon_0 \wedge |x_t - x'_t| \geq \epsilon_t. \quad (5)$$

The values of p are chosen to be $\alpha \in (0, 4)$ and $\epsilon = 0.1$ based on observation from the experimental results which is shown in Figure 1. The bifurcation diagram, KS entropy diagram, and scatter plot depicted in Figure 1 show chaos and uniform distribution of function values for δ map defined in equation (3). The Lyapunov exponent is often used to quantify the amount of separation of nearby orbits. A positive Lyapunov value indicates that the system is chaotic. A KS entropy density normalizes the amount of chaos in a coupled map lattice where the combined measure of chaos across all lattices is required for assessing the overall chaos. A positive KS entropy value is indicative of δ being unpredictable and in chaos state as a whole or almost all lattices exhibiting a positive Lyapunov value. The higher percentage of lattices exhibiting chaotic behaviour and less periodic windows in bifurcation diagram make it suitable for cryptography. The KS entropy (h) and universality (hu) were determined as in [42] by varying α between $(0, 4)$ for 25 lattices and 1000 iterations as shown in Figure 1.

4.2. $\hat{G}$ Construction. We construct pseudorandom generator $\hat{G}$ with expansion factor $l + p(l)$, for some polynomial p . On input p from parameter generation described in Section 4.1, the $\hat{G}$ does the following:

- (1) Set $\delta_1^{n+1}(x) = (\alpha x_n (1 - x_n) + (4 - \alpha) (\sin(\pi x_n)/4)) \bmod 1$
- (2) Set $\delta_2^{n+1}(x) = \delta_1^{n+1}(x)$
- (3) Set $\delta^{n+1} = \delta_1(x_n^j) + (\epsilon/2)[\delta_2(x_n^{j+1}) + \delta_2(x_n^{j-1}) - 2\delta_2(x_n^j)]$
- (4) Set $hc(x^j) = \bigoplus_{i=1}^n x_i^j \cdot r_i^j$
- (5) For $i = 1, \dots, k$ do
 - (a) $\forall j, 1 \leq j \leq m, x_0^j = \delta(x_0^j)$
- (6) Set $G(x^j) = \delta(x^j) \parallel hc(x^j)$
- (7) Set $s^j = G(x_0^j)$
- (8) Set $t_0^j = s$. For $i = 1, \dots, p(l)$ do
 - (a) For $j = 1, \dots, m$ do

FIGURE 1: δ characteristics. (a) $\delta^k(x^0)$ bifurcation diagram. (b) $\delta^k(x)$ KS entropy. (c) $\delta^k(x^0)$ distribution.

- (i) Let s_{i-1}^j be the first n bits of t_{i-1}^j , and let σ_{i-1}^j denote the remaining $i-1$ (when $i=1, s_0^j = t_0^j$ and σ_0^j is empty string)
- (ii) Set $t_i^j = G(s_{i-1}^j) \parallel \sigma_{i-1}^j$

(9) Output $t_{p(l)}$

Figure 2 shows the working of the construction $\hat{G}$ with m lattices and will output m bits. The claim that $\hat{G}$ construction is a cryptographically secure PRNG is proven in Section 5.

5. Theoretical Security Analysis

The construction $\hat{G}$ is based on the abstract notions of unpredictability. We analyse the security of the construction through a series of theorem using modern cryptography. First, we prove δ 's unpredictability and then its one-way function property. We then show that the construction $\hat{G}$ instantiated with δ is next-bit unpredictable secure PRNG with polynomial expansion factor. We also show that construction $\hat{G}$ is secure against adversaries given adaptive queries. These proof techniques are a design strategy which eventually helps in identification and removal of design weaknesses if not completely but to a large extend.

Theorem 1. *Unpredictable function: the function $\delta: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$ is an unpredictable function in the chaotic interval*

Proof. The theorem is proven through two claims, namely, the unpredictability of the δ and hardness of preimage problem using the inherent properties of δ and choice of parameters made in the construction. Though the dynamic system is deterministic, it exhibits chaotic behaviour which makes predictions hard when they operate in their region of chaos. Given a deterministic dynamic system with an initial configuration $\mathcal{D} = \{\delta, \alpha, x_0, k, l\}$ where x_0 is the initial state, δ is the function, k is the number of iterations, and l is the number of bits used for representation, two things are necessary for making predictions:

- (1) $\mathcal{D}_0$ should be known and particularly x_0 , and the initial state should be measured with reasonable accuracy such that the errors in the prediction are small say ε
- (2) Given the initial state x_0 and k time steps, compute $x_0 \vdash^k x_t$ in polynomial time

Measuring the initial state x_0 with the required accuracy to keep the prediction errors small and within ε is difficult even if we consider that the initial configuration $\mathcal{D}_0$ is known [39]. Therefore, meeting the two necessary conditions is nontrivial to make predictions in $\mathcal{D}$. Let us assume that the computations are carried out in discrete time and continuous state space which are efficiently made discrete with finite precision arithmetic. $\square$

Claim 1. If a system $\mathcal{D} = \{\delta, \alpha, x_0, k, l\}$ with parameters $x_0 \in [0, 1], \alpha \in [a, b]$ as initial conditions chosen uniformly random such that

$$\forall t \geq t_p \forall x, x' \in \mathcal{X}: |x_0 - x'_0| < \varepsilon_0 \wedge |x_t - x'_t| \geq \varepsilon_t, \quad (6)$$

then the system is pragmatically unpredictable where $[a, b]$ is the chaotic interval and the following holds:

$$\Pr[x'_t: \mathcal{D} \setminus \{\alpha, x_0, k\}, |x_t - x'_t| \leq \varepsilon_t, |x_0 - x'_0| < \varepsilon_0, x'_t \leftarrow \delta^k(x'_0)] \leq \text{negl}(l). \quad (7)$$

Proof. Consider the deterministic dynamical system $\mathcal{D} = \{\delta, \alpha, x_0, k\}$, and it is said to be strongly pragmatically predictable if and only if the following holds [39]:

$$\forall t \geq t_p \forall x, x' \in \mathcal{X}: |x_t - x'_t| \geq \varepsilon_t \longrightarrow |x_0 - x'_0| \geq \varepsilon_0, \quad (8)$$

where t_p represents interested prediction time x, x' denotes state space, ε_0 denotes the smallest measurement accuracy of initial conditions, and ε_t represents practically tolerable prediction error. The above equation intuitively gives the implication that the inability to predict x_t with reasonable

FIGURE 2: Construction $\hat{G}$ for $n > 1$.

accuracy level below ε_t at all times implies system's initial conditions could not be measured with accuracy less than ε_0 . In other words, if the system's initial conditions cannot be measured with at least the measurement accuracy ε_0 , the system is not strongly pragmatically predictable. Hence, strong pragmatic predictability condition cannot hold if the initial conditions are kept secret as in construction $\hat{G}$, and therefore, $|x_0 - x'_0| \gg \varepsilon_0$. Also, if the systems are carefully chosen such that the system is in chaotic regime, as described in Section 4 (parameter generation Section 4.1), it makes predictability hard even when the measurement of initial conditions is within ε_0 . If the system satisfies equation (6), then the equation for strong predictability cannot hold. The condition for weak pragmatic predictability is given by

$$\exists t \geq t_p \forall x, x' \in \mathcal{X}: |x_t - x'_t| \geq \varepsilon_t \longrightarrow |x_0 - x'_0| \geq \varepsilon_0. \quad (9)$$

The above condition for weak pragmatic predictability can neither hold for the system $\mathcal{D}$ if $|x_0 - x'_0| \gg \varepsilon_0$ holds. The intuitive meaning of the above equation is that for all times and states in the chaotic region, if the initial state prediction error is less than ε_0 and final state prediction error is more than ε_t , then the system is unpredictable. This also means sensitive dependence to initial conditions or chaos. Moreover, it is known that boundedness of the trajectories produces nonlinearity, being nonperiodic, simulates statistical random experiment, and fills the entire state space [39]. The system $\mathcal{D}$ when bounded in the interval $(0, 1)$ by parameter generation will exhibit the following characteristics consequently:

- (1) Nonlinearity due to addition of nonlinear term sine to the linear differential equation
- (2) Trajectory divergence or positive Lyapunov implies that the time-independent orbits will oscillate without being periodic in a finite region of the state space periodic making it unpredictable as observed from Figure 1
- (3) Orbits will contain all possible state values and thus simulate a statistical random experiment

- (4) Set of orbits starting from one finite state space will fill the entire state space as observed from Figure 1 after some time

Hence, the system $\mathcal{D}$ is unpredictable if equation (6) holds. Moreover, equation (6) holds by assumption and given $\mathcal{D} \setminus \{x_0, k, \alpha\}$, that is, all the dynamical system parameters except for $\{x_0, k, \alpha\}$, an approximation x'_0 such that $|x_0 - x'_0| \gg \varepsilon_0$ also holds, and then it implies that it is hard to compute $x'_0 \vdash^k x'_t$ such that $|x_t - x'_t| \leq \varepsilon_t$. Any adversary trying to predict x'_t with practically relevant accuracy ε_t has to exhaustively search all possible values of initial conditions over $|\mathcal{X}|$, and then the probability of determining x'_t from x'_0 assuming x_0 is chosen uniformly random which is given by

$$\begin{aligned} & \Pr[x'_t: \mathcal{D}, |x_t - x'_t| \leq \varepsilon_t, |x_0 - x'_0| < \varepsilon_0, x'_t \leftarrow \delta^k(x'_0), x'_0] \\ & \leq \frac{\varepsilon_t}{|\mathcal{X}|} \leq \frac{\varepsilon_t}{2^l}. \end{aligned} \quad (10)$$

The second inequality follows from the fact that the state space is represented in finite arithmetic with l bits. The quantity $\varepsilon_t |\mathcal{X}|^{-1} = \varepsilon_t 2^{-l} \leq \text{negl}(l)$ is negligible, and the above equation can be rewritten as

$$\begin{aligned} & \Pr[x'_t: \mathcal{D}, |x_t - x'_t| \leq \varepsilon_t, |x_0 - x'_0| < \varepsilon_0, x'_t \leftarrow \delta^k(x'_0), x'_0] \\ & \leq \text{negl}(l). \end{aligned} \quad (11)$$

Considering that $sk = p = \{\alpha, k, x_0\}$ being kept secret by design and inference from the above equation that the probability of computing $x_0 \vdash^k x_t$ such that $|x_t - x'_t| \geq \varepsilon_t$ given that $|x_0 - x'_0| < \varepsilon_0$ is negligible, the above equation can thus be written as

$$\begin{aligned} & \Pr[x'_t: \mathcal{D} \setminus \{\alpha, x_0, k\}, |x_t - x'_t| \leq \varepsilon_t, |x_0 - x'_0| < \varepsilon_0, x'_t \leftarrow \delta^k(x'_0)] \\ & \leq \text{negl}(l). \end{aligned} \quad (12)$$

Hence, Claim 1 is proven. $\square$

Claim 2. Let $\mathcal{D}$ be such that $\Pr[x'_t: \mathcal{D} \setminus \{\alpha, x_0, k\}, |x_t - x'_t| \leq \varepsilon_t, |x_0 - x'_0| < \varepsilon_0, x'_t \leftarrow \delta^k(x'_0)] \leq \text{negl}(l)$, then it holds that

$$\Pr\left[\mathcal{O}_{x'_0}^+: \mathcal{D} \setminus \{\alpha, x_0, k\}, |\mathcal{O}_{x'_0}^+ - \mathcal{O}_{x'_0}^+| \leq \varepsilon_t, |\mathcal{O}_{x'_t}^- - \mathcal{O}_{x'_t}^-| \leq \varepsilon_0\right] \leq \text{negl}(l). \quad (13)$$

Proof. It is given that for the system $\mathcal{D}$, the following holds:

$$\Pr\left[x'_t: \mathcal{D} \setminus \{\alpha, x_0, k\}, |x_t - x'_t| \leq \varepsilon_t, |x_0 - x'_0| < \varepsilon_0, x'_t \leftarrow \delta^k(x'_0)\right] \leq \text{negl}(l), \quad (14)$$

and then for all the points in the forward predicted orbit $\mathcal{O}_{x'_0}^+$ for x'_0 implies that given a x'_0 such that $|x_0 - x'_0| < \varepsilon_0$, predicting the points in the forward orbit $\mathcal{O}_{x'_0}^+ = \{x'_i: (x'_i, x_i) \in \delta_1^k(x_0), |x_i - x'_i| \leq \varepsilon_t\}$ is negligible, and then it can be written without loss of generality:

$$\Pr\left[\mathcal{O}_{x'_0}^+: \mathcal{D} \setminus \{\alpha, x_0, k\}, |\mathcal{O}_{x'_0}^+ - \mathcal{O}_{x'_0}^+| \leq \varepsilon_t, x'_0\right] \leq \text{negl}(l). \quad (15)$$

Given that δ is a noninvertible map and probability of computing $\mathcal{O}_{x'_0}^+$ is negligible, it implies that given a destination point in an orbit x_t and predicting all the points backward towards the initial point x_0 in the backward orbit $\mathcal{O}_{x_t}^- = \{x'_i: (x'_i, x_i) \in \delta_1^k(x_0), |x_i - x'_i| \leq \varepsilon_t, |x_0 - x'_0| \geq \varepsilon_0, x'_0\}$ is also negligible, gives

$$\Pr\left[\mathcal{O}_{x'_t}^-: \mathcal{D} \setminus \{\alpha, x_0, k\}, |\mathcal{O}_{x'_t}^- - \mathcal{O}_{x'_t}^-| \leq \varepsilon_0, x_t\right] \leq \text{negl}(l). \quad (16)$$

Combining forward and backward orbit computability probability from equations (15) and (16), we get

$$\Pr\left[\mathcal{O}_{x'_0}^+: \mathcal{D} \setminus \{\alpha, x_0, k\}, |\mathcal{O}_{x'_0}^+ - \mathcal{O}_{x'_0}^+| \leq \varepsilon_t, |\mathcal{O}_{x'_t}^- - \mathcal{O}_{x'_t}^-| \leq \varepsilon_0\right] \leq \text{negl}(l). \quad (17)$$

Hence, Claim 2 is proven.

It follows from Claims 1 and 2 that the function δ is unpredictable. $\square$

Corollary 1. *Preimage $_{\delta^k}$ problem: let $\mathcal{D}$ be a system such that $\Pr[\mathcal{O}_{x'_t}^-: \mathcal{D} \setminus \{\alpha, x_0, k\}, |\mathcal{O}_{x'_t}^- - \mathcal{O}_{x'_t}^-| \leq \varepsilon_0, x_t] \leq \text{negl}(l)$, then the following holds in the chaotic interval:*

$$\Pr\left[x'_0: \mathcal{D} \setminus \{\alpha, x_0, k\}, |x_0 - x'_0| \geq \varepsilon_0, x_t \leftarrow \delta^k(x_0), x'_0, x'_t\right] \leq \text{negl}(l). \quad (18)$$

Theorem 2. *If $\delta \in \mathcal{D}$ is unpredictable, then it is a one-way function in the chaotic interval*

Proof. A function is one-way if the following two conditions hold [41]:

- (1) There exists a polynomial-time algorithm M_δ computing δ , that is, $M(\delta) = \delta^k(x_0, l)$ for all $x_0 \in (0, 1)$ and $k, l \in \text{poly}(n)$
- (2) For every probabilistic polynomial-time algorithm $\mathcal{A}$, there is a negligible function negl such that

$$\Pr[\text{Invert}_{\mathcal{A}, \delta}(x_t) = 1] \leq \text{negl}(l). \quad (19)$$

The above equation can be also represented succinctly as

$$\Pr_{x_0 \in_R (0, 1)}[\mathcal{A}(1^n, \delta^k(x_0)) = \delta^{-k}(x_t)] \leq \text{negl}(l). \quad (20)$$

The proof of the theorem is structured by means of proving each condition for one-way function described above holds for δ through claims as given below. $\square$

Claim 3. Given $\mathcal{D}_0$ and $\{\alpha, k, x_0\}$ with finite precision, the function δ computes $x_0 \vdash^k x_t$ deterministically in polynomial time

Proof. It is assumed that given $\mathcal{D}_0$ is of finite precision and $x_0 \in (0, 1)$. The δ^k function computes $x_0 \vdash^k x_t$ at each lattice given by

$$\begin{aligned} \delta^k(x_0) &= \delta^k(x_0) \bmod 1 = \delta^{k-1}(\delta^{k-2}(\dots(\delta^1(x_0) \bmod 1))) \\ &= \delta^{k-1}(x_{k-2}) \bmod 1 \circ \dots \circ \delta^1(x_0) \bmod 1 = x_t \in (0, 1). \end{aligned} \quad (21)$$

We can see the abovementioned way of computing recursively and using modular arithmetic which keeps the complexity of finite arithmetic computing deterministic within polynomial time as long as k is polynomial. $\square$

Claim 4. Given $\mathcal{D} \setminus \{\alpha, x_0, k\}$, for every probabilistic-time algorithm $\mathcal{A}$, there is a negligible function negl such that $\Pr[\text{Invert}_{\mathcal{A}, \delta}(x_t)] \leq \text{negl}(l)$

Proof. We construct the experiment $\text{Invert}_{\mathcal{A}, \delta}$ as follows. $\square$

Construction 1. $(\delta - \text{Invert}_{\mathcal{A}, \delta})$

- (1) Choose uniform $x_0 \in (0, 1)$, and compute $x_t = \delta^k(x_0)$
- (2) $\mathcal{A}$ is given 1^n and $y = x_t$ as input, and outputs x'_0
- (3) The output of the experiment is defined to be 1 if $\delta(x'_0) = y$, and 0 otherwise

A closer observation of the definition of the experiment will reveal the fact that the experiment Invert is just a conceptual redefinition of the preimage problem defined in Section 3 and Definition 2. Rewriting the equation $\Pr[\text{Invert}_{\mathcal{A}, \delta}(x_t)]$ in terms of the preimage problem, we get

$$\begin{aligned}
\Pr[\text{Invert}_{\mathcal{A},\delta}(x_t)] &= \Pr_{x_0 \in_R(0,1)}[\mathcal{A}(1^n, \delta^k(x_0)) = \delta^{-k}(x_t)] \\
&= \Pr\left[\mathcal{O}_{x_t}^-: \mathcal{D} \setminus \{\alpha, x_0, k\}, |\mathcal{O}_{x_t}^- - \mathcal{O}_{x_t}^-| \leq \varepsilon_0, x_t\right] \\
&= \Pr[x'_0: \mathcal{D} \setminus \{\alpha, x_0, k\}, |x_0 - x'_0| \geq \varepsilon_0, x_t \leftarrow \delta^k(x'_0), x_t] \leq \text{negl}(l).
\end{aligned} \tag{22}$$

The second and third equality follows from equation (16) and Corollary 1. The fourth inequality proves Claim 4.

Now, from Claims 3 and 4, it follows that δ is one way. From Theorems 1, 2, and [41], the corollaries are as follows.

Corollary 2. *If δ is a one-way function in the chaotic interval and $G(x, r) = (\delta(x), r)$, then the function $hc(x, r) = \oplus_{i=1}^n x_i \cdot r_i$ where $x = x_1, \dots, x_n$ and $r = r_1, \dots, r_n$ is a hard-core predicate of G .*

Corollary 3. *If δ is an OWP in the chaotic interval, hc is hard-core bit of δ and G is defined by $G(x) = \delta^n(x) \circ G'(x) = \delta^n(x) \circ h(\delta^{n-1}(x)) \circ \dots \circ h(x)$, then G is next-bit unpredictable.*

Corollary 4. *If $\hat{G}$ is a pseudorandom generator with expansion factor $n + 1$, then for any polynomial denoted $\text{poly}(\cdot)$, then there exists a pseudorandom generator $\hat{G}$ with expansion factor $\text{poly}(n)$.*

Theorem 3. *Let predictor be a construction as given below for $\Delta = \{\delta_n\}_{n \in \mathbb{N}}$ an efficient $\mathbb{R} \rightarrow \mathbb{R}$ function ensemble, and let $c \in \mathbb{N}$ be some constant as follows.*

Construction 2. $(\delta - \text{Predictor}_{\mathcal{A},\delta,I}(x))$:

- (1) Assume two parties, the predictor $\mathcal{T}$ and verifier $\mathcal{V}$
- (2) Let $\mathcal{D}_0$ denote the private initial configuration generated during parameter generation
- (3) Let x_0 be the initial state space and iterated k such that $x_t = \delta^k(x_0)$ or $x_0 \vdash^k x_t$
- (4) The protocol runs in $q - 1$ rounds for $q = n^c$. $\mathcal{T}$ sends to $\mathcal{V}$ a point $x_i \in \mathcal{X}$ and in return $\mathcal{V}$ sends to $\mathcal{T}$ at the i^{th} round of the protocol value y_i computed as follows:
 - (a) $x'_i = x_0 + x_i \text{mod} 1$ where x_i takes values in the interval $(0, 1)$

(b) $y_i = \delta(x'_i)$

- (5) $\mathcal{T}$ outputs a point x_q which is not previously queried in $x_1 x_2 \dots x_{q-1}$ and a string y_q which is its guess for $\delta^{k+q}(x_q)$ at the termination of the protocol

Δ is unpredictable against an adaptive sample and an adaptive challenge if for any polynomial-time machine $\mathcal{T}$ and any constant $c \in \mathbb{N}$, then

$$\Pr[y_q = \delta^{k+q}(x_q) | y_{1 \dots q-1}] \leq \text{negl}(l). \tag{23}$$

Proof. The initial configuration parameters are generated using parameter generation described in Section 4.1 such that for $\mathcal{D}$, the following condition for chaos or unpredictability holds:

$$\forall t \geq t_p \forall x, x' \in \mathcal{X}: |x_0 - x'_0| < \varepsilon_0 \wedge |x_t - x'_t| \geq \varepsilon_t. \tag{24}$$

From [38], we know that predicting any point in a orbit at any level of precision $\varepsilon > 0$, all sufficiently past points are approximately probabilistically irrelevant. Hence, if the condition for chaos holds, then by Definition 2, we have for $t_m \leq t_n$ at level $\varepsilon > 0$ where ε is a negligible quantity, without loss of generality ε can be assumed to be equal to a small quantity $\text{negl}(l)$:

$$|P(B_{t_m} | A_{t_n}) - P(B_{t_m})| < \text{negl}(l). \tag{25}$$

Rewriting the above equation, we get

$$|\Pr[y_q = \delta^k_\alpha(x_q) | y_{1 \dots q-1}] - \Pr[y_q = \delta^k_\alpha(x_q)]| < \text{negl}(l). \tag{26}$$

The ε is negligible function of level of precision by definition and its equivalent to $\text{negl}(l)$. Hence, it can be inferred that the probability conditioned on past observations has negligible effect under unpredictability assumption, and then it can be written as

$$\begin{aligned}
\Pr[y_q = \delta^k_1(x_q) | y_{1 \dots q-1}] &< \Pr[y_q = \delta^k_1(x_q)] + \text{negl}(l) \\
&< \Pr[x'_0: \mathcal{D} \setminus \{\alpha, x_0, k\}, |x_0 - x'_0| \geq \varepsilon_0, x_q \leftarrow \delta^{k+q}_1(x'_0), x_q] + \text{negl}(l) \\
&< \text{negl}(l) + \text{negl}(l) \leq \text{negl}(l).
\end{aligned} \tag{27}$$

The first inequality is obtained by rewriting equation (26). The second inequality follows from Theorem 2 and its corollary. The third and fourth inequality follows that the

predictor's advantage is negligible. Hence, the adversary or the predictor $\mathcal{T}$ has negligible advantage in predicting given adaptive sample query access. $\square$

Corollary 5. Let δ be such that $\Pr[\text{Predictor} = 1] \leq \text{negl}(k)$, then from [43], it holds that

$$\Pr[b = g_i: x \in_R (0, 1), g = g_1 \dots g_{p(k)} = G(x), b \leftarrow \mathcal{T}(g_1 \dots g_{i-1})] < \frac{1}{2} + \text{negl}(l). \quad (28)$$

Corollary 6. From Corollaries 2–5, it follows that $\hat{G}$ is a secure next-bit unpredictable PRNG with expansion factor $\text{poly}(n)$.

6. Empirical and Statistical Analysis

The central requirement for any PRNG to be practical is pseudorandomness and unpredictability (both forward and backward). A well-designed generator produces sequences that exhibit good statistical properties and are evenly distributed. To test the statistical strength of the designed PRNG, the sequence generated is subjected to rigorous statistical tests against the null hypothesis. The null hypothesis is that the sequence under test is random, and alternate hypothesis that the sequence is nonrandom. The statistical test suite describes the probability of the tested potential pseudorandom sequence, called test sequence, against a priori known truly random source or reference distribution. The intuition is that when the computed statistical value does not exceed the critical value, it means that a low probability event does not occur naturally, and hence, it must be random.

There are an infinite number of statistical tests describing a methodology to test the existence of a prescribed pattern to prove that the tested sequence is nonrandom. The presence of an infinite number of statistical tests leads to no finite set of tests to quantify randomness. Therefore, the statistical tests are only a necessary condition and not a sufficient condition, but they are truly an indicator of randomness using probability. In this section, we analyze the statistical strength of the binary sequence generated by generators $\hat{G}$ using statistical randomness testing suites, namely, NIST 800-22 randomness test suite [44], Dieharder battery of test [45], ENT [46], and TestU01 [47, 48]. The pseudorandom binary bits are generated using $\hat{G}$ and instantiated with δ with $m = 100$, and $p(l) = 10^6$ is used for both proposed generators. All other parameters are chosen as described in Section 4.1.

6.1. NIST Test Results. NIST SP800-22 statistical test suite consists of 15 independent statistical tests to investigate the randomness of the arbitrary long binary sequence produced by the generators. For each test, with a fixed significant level ($\alpha = 0.01$), the quality of the sequence is given as p values. If the p value is (> 0.01), the generated sequence passes the test; otherwise, it fails. The sample size of 100 binary sequences for each test is used with the bit length of the order 10^6 . The interpretation of the empirical results can be done from the proportion of passing of a test, i.e., the pass rate. The minimum pass rate is calculated from the sample size

and the significant level as given in [44]. The minimum pass rate for 100 binary sequences with 0.01 significance level is 0.9602. The test is statistically successful if at least 96 sequences out of 100 sequences pass the test and the generator $\hat{G}$ successfully passed all the 15 tests as seen in Table 1 confirming the randomness of the generated sequence.

6.2. Dieharder Test Results. The Dieharder test suite developed by Robert G. Brown consists of 31 stringent tests for distinguishability from random. The p value distribution generated by each test for Dieharder in contrast to NIST test suite is analyzed by a Kuiper–Kolmogorov–Smirnov (KKS) test, which has a higher sensitivity to deviations from an equal distribution. A random sequence with size 10^6 bit sequence from generators $\hat{G}$ is used as input for the Dieharder test suite. Table 2 shows the result of the Dieharder battery of tests. The number of trials used in each test is given as t_{sample} . The generators passed all the 31 stringent tests, which indicate that there is evidence of randomness at a high confidence level.

6.3. ENT Test Results. The ENT battery of tests consists of 5 tests for randomness. Table 3 shows the ENT test results. The information density of the sequence is evaluated using the entropy test expressed as number of bits per byte. Entropy quantifies how much expected information is contained in the generated sequence. $\hat{G}$ shows a maximum level of entropy (ideal value is 8) indicating a good measure of randomness. The chi-square test indicates that the generated sequence is random, and it is extremely sensitive to errors in PRNG. The arithmetic mean is the sum of all the bytes in the generated file divided by the file length. $\hat{G}$ is very close to random, i.e., within the mean range of 127 ± 0.01 . The Monte Carlo value converges to π with minimal error 0.01 percent for $\hat{G}$ signifying that the sequence is close to random. The serial correlation coefficient (SCC) test indicates that there is no correlation between each byte with the previous byte in the sequence.

6.4. TestU01 Results. The results for TestU01 are shown in Table 4. TestU01 battery of tests developed by Pierre L'Ecuyer and Richard Simard is one of the most stringent tests for randomness. The binary sequence battery test includes Rabbit (38 tests), Alphabit (17 tests), and Block-alphabit (17 tests) are used to test the randomness of the generators. The test results are shown in Table 4. The generator $\hat{G}$ successfully passed all the tests (p value between $[0.001, 0.999]$) of the TestU01. This indicates that the

TABLE 1: NIST test results.

Test name (test no.)	$\hat{G}$	
	p value	p -rate
$\geq$	0.01	0.9602
Frequency (1)	0.978072	0.9900
BlockFrequency (2)	0.759756	1.0000
CumulativeSums-forward (3)	0.249284	0.9800
CumulativeSums-backward (4)	0.851383	0.9800
Runs (5)	0.574903	0.9900
LongestRun (6)	0.816537	0.9800
Rank (7)	0.616305	1.0000
FFT (8)	0.334538	0.9900
NonOverlappingTemplate* (9–156)	0.497008	0.9900
OverlappingTemplate (157)	0.978072	1.0000
Universal (158)	0.779188	0.9900
ApproximateEntropy (159)	0.534146	0.9900
RandomExcursions* (160–167)	0.473546	0.9848
RandomExcursionsVariant* (168–185)	0.327361	0.9848
Serial 1 (186)	0.739918	0.9800
Serial 2 (187)	0.911413	1.0000
LinearComplexity (188)	0.474986	1.0000
Success	—	15/15

*Mean value.

TABLE 2: Dieharder test results.

Test name (test no.)	$\hat{G}$	
	T sample	p value
Diehard Birthday (0)	100	0.623511
Diehard OPERMS (1)	1000000	0.892017
Diehard 32×32 Binary (2)	40000	0.563151
Diehard 6×8 Binary Rank (3)	100000	0.925417
Diehard Bitstream (4)	2097152	0.851563
Diehard OPSO (5)	2097152	0.867954
Diehard OQSO (6)	2097152	0.415923
Diehard DNA (7)	2097152	0.442299
Diehard Count the 1 s (stream) (8)	256000	0.874544
Diehard Count the 1 s (byte) (9)	256000	0.177324
Diehard Parking Lot (10)	12000	0.909532
Diehard Min. Distance (2d Circle) (11)	8000	0.362603
Diehard 3d Sphere (Min. Distance) (12)	4000	0.616346
Diehard Squeeze (13)	100000	0.140148
Diehard Sums (14)	100	0.040395
Diehard Runs* (15)	100000	0.562278
Diehard Craps* (16)	200000	0.569362
Marsaglia and Tsang GCD* (17)	10000000	0.931485
STS Monobit	100000 (100)	0.716828
STS Runs (101)	100000	0.801802
STS Serial (generalized) (102)	100000	0.422199
RGB Bit Distribution* (200)	100000	0.597944
RGB Generalized Min. Distance* (201)	10000	0.550842
RGB Permutations* (202)	100000	0.497884
RGB Lagged Sum* (203)	1000000	0.494495
RGB Kolmogorov–Smirnov (204)	10000	0.240666
Byte Distribution (205)	51200000	0.665759
DAB DCT (206)	50000	0.441544
DAB Fill Tree* (207)	15000000	0.393346
DAB Fill Tree 2* (208)	5000000	0.385133
DAB Monobit 2 (209)	65000000	0.009707
Success counts	31/31	—

*Mean value.

TABLE 3: ENT test results.

Test name	$\hat{G}$
Sample size (bytes)	246460416
Entropy (bits)	7.999999
Chi-square	272.11
Arithmetic mean	127.4943
Monte Carlo value for pi	3.141934062
SCC	−0.000147

TABLE 4: TestU01 test results.

Length	Rabbit	Alphabit	Blockalphabit
10^6	38/38	17/17	17/17

generated binary sequence has good randomness and hence unpredictable.

Experimentally through a series of stringent battery of the test, it is seen that generator $\hat{G}$ has passed all the randomness tests successfully in agreement to the security analysis presented in Section 5. Therefore, it is evident that the generated sequences exhibit good statistical evidence of randomness and can be used confidently in sensitive applications.

7. Discussion

Implementation of cryptographic primitives for practical usage and for memory-constrained devices requires standardization in implementation. Also, the choice of operations involved in the algorithm directly affects the choice of hardware. In this proposal, we have focused to establish a solid approach to create robust and credible chaos cryptographic primitive design methodology. However, it can be observed from the construction that the design uses logistic-sine map-based lattices composed into a CML, and hence, the design only requires implementing mainly logistic-sine map in hardware. Therefore, the performance of the CML and the proposed CSPRNG in hardware or specific suitability to resource constrained devices can be assumed to be on par with the performance similar to that of LS map implementations as in [49, 50]. Furthermore, in future we would like to study the performance of the construction on hardware implementations.

As dynamical system with chaotic behaviour, its trajectory will never repeat in theory but due to finite precision limitations in digital implementations may give rise to cycles. The robustness to dynamical degradation is thus captured by measuring the cycle length of symbols or bits produced by a chaotic systems. Since the proposed system passed all NIST tests with 1 million bit length output sequence from the proposed CSPRNG, including the random excursion test which captures the notion of cycle lengths in the output sequence, it can be safely assumed that effect of dynamical degradation is negligible for all practical purposes. Moreover, the experiments are carried out using Python 3.7 which adheres to the IEEE Standard for Floating-Point Arithmetic (IEEE 754), thereby restricting the key

space to single and double precision. In this CSPRNG, the only secret is the initial seed or initial vector. Therefore, the key length is proportional to the precision or number of bits used to represent one initial seed in a lattice times the total number of lattices in the CML. For example, if a double precision IEEE 754 representation takes 53 significant and 11 exponent bits, then for 25 lattices, the key length is 275 and the total key space is in the order of 2^{275} without considering the parameters of logistic-sine map which are nondominant factors in the order. Similarly, for a single precision, the total key space is in the order of 2^{200} . In chaos-based cryptography, a key space above 2^{128} is considered secure to be used for encryption, and hence, for the proposed CSPRNG, the probability of a brute force attack is negligible.

Provable security has been demonstrated in chaos cryptography with the Chebyshev chaotic map in a multiparty computation setting using the hardness of computational Diffie–Hellman (CDH) problem’s chaotic equivalent computational chaotic Diffie–Hellman (CCDH) problem in their designs. Such designs use the computational chaotic Diffie–Hellman problem posed by Chebyshev maps to build one-way functions. However, to the best of our knowledge, hardly any provable security is proposed in chaos cryptography for a CSPRNG setting. Some of the works using provable security under multiparty computation setting are [51–53] in which the authors present a formal proof of security in a multiparty computation setting to establish a secret key between parties which do not have a preshared secret key for secure communication. Iwasaki and Umeno [54] improved a vector space cipher by modelling a linear masking attack and proved that the improved algorithm can resist linear masking attack but did not use any standard provable security framework for encryption. The reasoning behind the choice of Chebyshev map used in such designs is due to its inherent property of CCDH which perfectly fits in the provable security framework for multiparty computation setting. Similarly, the proposed CSPRNG uses a CML composed of the LS map which is known to exhibit noninvertible property and fits well in the provable security framework for CSPRNG setting. Hence, any chaotic maps or flows can be used in the provable security framework if it would fit in the choice of provable security framework and setting. The proposed chaos-based CSPRNG design methodology demonstrates such instantiation of construction with a chaotic map for a choice of provable security framework and setting. Moreover, the authors in [55] provide a checklist and the proposed work captures those that are pertinent to a CSPRNG setting. The modern cryptography’s provable security frameworks are indeed designed to capture such notions of attack and attacker capabilities enlisted in [55], and their correlation to this work is as follows:

- (1) Cryptographic primitive’s algorithm should be stated explicitly in the form of mathematical model as described in Section 4.2
- (2) Propositions for algorithms stated and proven explicitly as described from Theorems 1 to 3

- (3) Specification of domain and range as described in Section 4.2 parameter generation
- (4) Models capturing attackers’ knowledge of the algorithm and security depending on just secret key as demonstrated through Theorems 1 to 3
- (5) Proof-driven design approach as demonstrated through in Theorems 1 to 3
- (6) Discussion on hardness assumptions and dynamic degradation as demonstrated through Theorems 1 to 3 and Section 7
- (7) Scope of attacks and robustness as captured through the notion of Theorems 1 to 3

8. Conclusion

The theory that a deterministic system shall produce deterministic output has been proven otherwise with the study of dynamical systems. The construction $\hat{G}$ is based on the results that deterministic dynamical systems can produce chaos over long range. Moreover, when such systems are made to operate in the region of chaos by choosing system parameters appropriately, they tend to become unpredictable. More precisely, the chosen function δ exhibits unpredictability proving computation of the defined preimage problem as hard or unpredictable making them candidate unpredictable functions according to modern cryptography. It is a proven fact that the pseudorandom number generator constructions satisfying the abstract notions of computational indistinguishability will pass all efficient statistical tests. Such notions to the best of our knowledge hardly have been applied in the design of pseudorandom generator constructions using chaos. This paper presents CSPRNG using the proven Goldreich–Levin generic construction of modern cryptography and then instantiated it with a new CML (δ), namely, $\hat{G}$. The security is proven using the abstract notions of unpredictability, a proven equivalent to computational indistinguishability and modern cryptography primitive design tool. The pseudorandom generator $\hat{G}$ is then proven to be computationally indistinguishable. Pseudorandom sequences generated from such design are then tested using standard statistical randomness test suites used by chaos-based cryptography designers, namely, NIST, Dieharder, ENT, and TestU01. The pseudorandom sequences generated from $\hat{G}$ passed all the tests of NIST, Dieharder, ENT, and TestU01 (tests for bits) test suites proving the claims of abstract notions of unpredictability. Hence, the premise that statistical analysis is used as a necessary condition and rigorous mathematical simulation-based proof is used as sufficient condition is demonstrated to eventually produce better designs. Thus, we believe that the demonstrated design approach will provide new directions beyond statistical analysis in designing chaos-based cryptographic primitives by using modern cryptography design tools. Also, it will motivate chaos-based designers to more often instantiate time-tested proven generic constructions with candidate chaotic functions rather than design generic constructions for new cryptographic primitives. This kind of design approach will lead to robust and credible chaotic

cryptographic primitives and prevent depreciating motivation on chaos-based cryptographic primitive design.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2137–2150, 2018.
- [2] A. C. Yao, "Theory and application of trapdoor functions," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pp. 80–91, IEEE, Chicago, IL, USA, November 1982.
- [3] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*, Cambridge University Press, Cambridge, UK, 2007.
- [4] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.
- [5] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *Journal of the ACM*, vol. 33, no. 4, pp. 2137–2150, 1986.
- [6] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudorandom bits," in *Proceedings of the 1982/23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, Chicago, IL, USA, November 1984.
- [7] O. Goldreich and L. A. Levin, "A hard-core predicate for all one-way functions," in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, Seattle, USA, May 1989.
- [8] M. Naor and O. Reingold, "From unpredictability to indistinguishability: a simple construction of pseudo-random functions from macs," in *Proceedings of the Annual International Cryptology Conference*, pp. 267–282, Santa Barbara, CA, USA, August 1998.
- [9] B. Karakaya, V. Çelik, and A. Gülten, "Chaotic cellular neural network-based true random number generator," *International Journal of Circuit Theory and Applications*, vol. 45, no. 6, 2017.
- [10] S. A. Tuncer and T. Kaya, "True random number generation from bioelectrical and physical signals," *Computational and Mathematical Methods in Medicine*, vol. 2018, Article ID 3579275, 11 pages, 2018.
- [11] Y. Hu, X. Liao, K.-W. Wong, and Q. Zhou, "A true random number generator based on mouse movement and chaotic cryptography," *Chaos, Solitons & Fractals*, vol. 40, no. 5, pp. 2286–2293, 2009.
- [12] S. Buchovecká, R. Lórencz, F. Kodýtek, and J. Buček, "True random number generator based on ring oscillator puf circuit," *Microprocessors and Microsystems*, vol. 53, pp. 33–41, 2017.
- [13] I. Cicek, A. Emre Pusane, and G. Dundar, "A novel design method for discrete time chaos based true random number generators," *Integration*, vol. 47, no. 1, pp. 38–47, 2014.
- [14] M. García-Martínez and E. Campos-Cantón, "Pseudo-random bit generator based on multi-modal maps," *Nonlinear Dynamics*, vol. 82, no. 4, pp. 2119–2131, 2015.
- [15] D. Lambić, A. Janković, and M. Ahmad, "Security analysis of the efficient chaos pseudo-random number generator applied to video encryption," *Journal of Electronic Testing*, vol. 34, no. 6, pp. 709–715, 2018.
- [16] H. Xu, X. Tong, and X. Meng, "An efficient chaos pseudo-random number generator applied to video encryption," *Optik*, vol. 127, no. 20, pp. 9305–9319, 2016.
- [17] A. Akhshani, A. Akhavan, A. Mobaraki, S.-C. Lim, and Z. Hassan, "Pseudo random number generator based on quantum chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 1, pp. 101–111, 2014.
- [18] D. Lambić, "Security analysis of the pseudo-random bit generator based on multi-modal maps," *Nonlinear Dynamics*, vol. 91, no. 1, pp. 505–513, 2018.
- [19] Y. Wang, Z. Liu, J. Ma, and H. He, "A pseudorandom number generator based on piecewise logistic map," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2373–2391, 2016.
- [20] D. Lambić, "Security analysis and improvement of the pseudo-random number generator based on piecewise logistic map," *Journal of Electronic Testing*, vol. 35, no. 4, pp. 519–527, 2019.
- [21] V. K. Dmitry, S. I. Kuzmina, and L. P. Bobylev, *Discontinuity, Nonlinearity, and Complexity*, p. 25, L & H Scientific Publishing, Glen Carbon, IL, USA, 2015.
- [22] D. Lambić, "Security analysis and improvement of the pseudo-random number generator based on quantum chaotic map," *Nonlinear Dynamics*, vol. 94, no. 2, pp. 1117–1126, 2018.
- [23] M. L. Sahari and I. Boukemara, "A pseudo-random numbers generator based on a novel 3d chaotic map with an application to color image encryption," *Nonlinear Dynamics*, vol. 94, no. 1, pp. 723–744, 2018.
- [24] J. G. Lu and G. Chen, "A note on the fractional-order Chen system," *Chaos, Solitons & Fractals*, vol. 27, no. 3, pp. 685–688, 2006.
- [25] R. Montero-Canela, E. Zambrano-Serrano, E. I. Tamariz-Flores, J. M. Muñoz-Pacheco, R. Torrealba-Meléndez, and R. Torrealba-Meléndez, "Fractional chaos based-cryptosystem for generating encryption keys in ad hoc networks," *Ad Hoc Networks*, vol. 97, Article ID 102005, 2020.
- [26] Z. Kotulski, J. Szczepański, K. Górski, A. Paszkiewicz, and Z. Anna, "Application of discrete chaotic dynamical systems in cryptography—dcc method," *International Journal of Bifurcation and Chaos*, vol. 9, no. 6, pp. 1121–1135, 1999.
- [27] T. I. Karimov, D. N. Butusov, D. O. Pesterev, D. V. Predtechenskii, and R. S. Tedoradze, "Quasi-chaotic mode detection and prevention in digital chaos generators," in *Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EICon-Rus)*, pp. 303–307, Moscow and St. Petersburg, Russia, January 2018.
- [28] E. G. Nepomuceno, H. M. Rodrigues Jr, S. A. M. Martins, M. Perc, and M. Slavinec, "Interval computing periodic orbits of maps using a piecewise approach," *Applied Mathematics and Computation*, vol. 336, pp. 67–75, 2018.
- [29] E. G. Nepomuceno and E. M. A. M. Mendes, "On the analysis of pseudo-orbits of continuous chaotic nonlinear systems simulated using discretization schemes in a digital computer," *Chaos, Solitons & Fractals*, vol. 95, pp. 21–32, 2017.
- [30] A. Flores-Vergara, E. E. García-Guerrero, E. Inzunza-González et al., "Implementing a chaotic cryptosystem in a 64-bit embedded system by using multiple-precision arithmetic," *Nonlinear Dynamics*, vol. 96, no. 1, pp. 497–516, 2019.

- [31] M. François, T. Grosge, D. Barchiesi, and R. Erra, "Pseudo-random number generator based on mixing of three chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 887–895, 2014.
- [32] M. A. Dastgheib and M. Farhang, "A digital pseudo-random number generator based on sawtooth chaotic map with a guaranteed enhanced period," *Nonlinear Dynamics*, vol. 89, no. 4, pp. 2957–2966, 2017.
- [33] R. Devaney, *An Introduction to Chaotic Dynamical Systems*, CRC Press, Boca Raton, FL, USA, 2018.
- [34] L. A. Bunimovich and Y. G. Sinai, "Spacetime chaos in coupled map lattices," *Nonlinearity*, vol. 1, no. 4, 491 pages, 1988.
- [35] K. Kaneko, "Spatiotemporal intermittency in coupled map lattices," *Progress of Theoretical Physics*, vol. 74, no. 5, pp. 1033–1044, 1985.
- [36] K. Kaneko, "Overview of coupled map lattices," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 2, no. 3, pp. 279–282, 1992.
- [37] F. Green., "Np-complete problems in cellular automata," *Complex Systems*, vol. 1, no. 3, pp. 453–474, 1987.
- [38] C. Werndl, "What are the new implications of chaos for unpredictability?" *The British Journal for the Philosophy of Science*, vol. 60, no. 1, pp. 195–220, 2009.
- [39] G. Schurz, "Kinds of unpredictability in deterministic systems," in *Law and Prediction in the Light of Chaos Research*, pp. 123–141, Springer, New York, NY, USA, 1996.
- [40] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.
- [41] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC Press, Boca Raton, FL, USA, 2014.
- [42] Y.-Q. Zhang and X.-Y. Wang, "Spatiotemporal chaos in mixed linear-nonlinear coupled logistic map lattice," *Physica A: Statistical Mechanics and Its Applications*, vol. 402, pp. 104–118, 2014.
- [43] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, vol. 15, no. 2, pp. 364–383, 1986.
- [44] L. E. Bassham, A. L. Rukhin, J. Soto et al., *Sp 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, National Institute of Standards & Technology, Gaithersburg, MD, USA, 2010.
- [45] M. George, "Diehard: a battery of tests of randomness," 1996.
- [46] J. Walker, "Ent-a pseudorandom number sequence test program," 2003.
- [47] P. L'Ecuyer and R. Simard, "Testu01: A library for empirical testing of random number generators," *ACM Transactions on Mathematical Software (TOMS)*, vol. 33, no. 4, pp. 1–40, 2007.
- [48] P. L'Ecuyer and R. Simard, "Testu01: a software library in ansi c for empirical testing of random number generators—user's guide, compact version," 2013.
- [49] Z. Hua, B. Zhou, and Y. Zhou, "Sine chaotification model for enhancing chaos and its hardware implementation," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1273–1284, 2018.
- [50] Z. Hua, Y. Zhou, and B. Bao, "Two-dimensional sine chaotification system with hardware implementation," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 887–897, 2019.
- [51] M. Luo, Y. Zhang, M. K. Khan, and D. He, "An efficient chaos-based 2-party key agreement protocol with provable security," *International Journal of Communication Systems*, vol. 30, no. 14, Article ID e3288, 2017.
- [52] L. Xiong, J. Niu, S. Kumari et al., "A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security," *Wireless Personal Communications*, vol. 89, no. 2, pp. 569–597, 2016.
- [53] M. Sabzinejad Farash and M. Ahmadian Attari, "An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps," *Nonlinear Dynamics*, vol. 77, no. 1, pp. 399–411, 2014.
- [54] A. Iwasaki and K. Umeno, "Improving security of vector stream cipher," *Nonlinear Theory and Its Applications, IEICE*, vol. 7, no. 1, pp. 30–37, 2016.
- [55] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 305–313, 2018.