



Research Article


A novel pseudo-random number generator for IoT based on a coupled map lattice system using the generalised symmetric map



Unsub Zia¹  · Mark McCartney¹ · Bryan Scotney¹ · Jorge Martinez¹ · Ali Sajjad²

Received: 28 September 2021 / Accepted: 7 December 2021

Published online: 04 January 2022

© The Author(s) 2021 

Abstract

Pseudo-random number generators (PRNGs) are one of the building blocks of cryptographic methods and therefore, new and improved PRNGs are continuously developed. In this study, a novel method to generate pseudo-random sequences using coupled map lattices is presented. Chaotic maps only show their chaotic behaviour for a specified range of control parameters, what can restrict their application in cryptography. In this work, generalised symmetric maps with adaptive control parameter are presented. This novel idea allows the user to choose any symmetric chaotic map, while ensuring that the output is a stream of independent and random sequences. Furthermore, to increase the complexity of the generated sequences, a lattice-based structure where every local map is linked to its neighbouring node via coupling factor has been used. The dynamic behaviour and randomness of the proposed system has been studied using Kolmogorov–Sinai entropy, bifurcation diagrams and the NIST statistical suite for randomness. Experimental results show that the proposed PRNG provides a large key space, generates pseudo-random sequences and is computationally suitable for IoT devices.

Highlights

- Contribution in the field of chaos theory by the introduction of generalised symmetric maps and adaptive control parameter to retain chaotic range of the system automatically.
- Proposal of a pseudo-random number generator capable of producing unique and independent random sequences.
- Investigation and analysis of the proposed system using standard benchmarks to validate the suitability for cryptographic applications.

Keywords Coupled map lattice · Generalised symmetric map · Pseudo-random number generator · Internet of Things

1 Introduction

Computers have evolved from large calculating machines to smart handheld devices that have revolutionized human lives [1]. The concept of Internet of Things (IoT) has dramatically changed the way things used to work, since smart computers are replacing humans in the loop. In an IoT framework, large scale decisions are based on the data generated

by the IoT sensors; for instance we could cite temperature control in a factory, intruder alarm systems in a smart home or emergency doctor call for patients in smart care between others [2]. Keeping in mind the importance of the IoT data, it is a crucial requirement to ensure data integrity and security [3]. The most common practice for securing data is to deploy cryptographic systems such as private key encryption, public key encryption and digital signatures etc. [4]. One of the

✉ Unsub Zia, zia-smu@ulster.ac.uk; Mark McCartney, m.mccartney@ulster.ac.uk; Bryan Scotney, bw.scotney@ulster.ac.uk; Jorge Martinez, j.martinez-carracedo@ulster.ac.uk; Ali Sajjad, ali.sajjad@bt.com | ¹School of Computing, Ulster University, Belfast, Northern Ireland, UK. ²Applied Research, British Telecom, Ipswich, UK.



building blocks on which cryptographic primitives depend are the seed values or in simple words the random number sequences that would make the keys difficult to predict [5].

Random numbers can be broadly categorized into true-random numbers (TRNs) [6] and pseudo-random numbers (PRNs) [7]. TRNs are usually considered to be the result of physical processes that produce unpredictable patterns, for instance noise (atmospheric or thermal), and other phenomena like (electromagnetic and quantum) are the sources of true random sequences. Whereas, PRNs are generated via mathematical algorithms that depend on some initial conditions and output a sequence that exhibits similar properties to TRNs [8]. There are several benchmarks available to test and validate the random nature of a number sequence. In literature, several Pseudo-random number generators (PRNGs) have been proposed, and still researchers are coming up with new ideas for lightweight PRNGs that would be a better fit for IoT based applications [9]. The limitations of using PRNGs for IoT data are: (1) the IoT sensor devices are resource constrained and (2) the sensor devices generate large amount of data constantly. Keeping in mind the nature of IoT data, there is a need for PRNGs that could offer wider range of initial seed values to choose from.

Chaos theory is a branch of mathematics that has been extensively studied due to its unique properties of apparent disorder and randomness. The chaotic systems can generate diverging and disorderly sequences, which appear to be random, being in fact deterministic but highly sensitive to initial conditions [10]. These particular characteristics make chaotic systems a good candidate for PRNGs. Chaos based PRNGs have been an active area of research but unfortunately, relatively few chaotic maps have been explored for this purpose [11]. In this paper, a family of symmetric chaotic maps have been considered as the local maps for coupled map lattices (CMLs) and an adaptive approach is proposed for control parameter selection. The proposed model introduces additional control parameters which contribute to the generation of highly chaotic sequences with large key space for cryptographic applications.

The remaining paper is organised as follows: The recent literature and related advances will be discussed in Sect. 2. The proposed model and its analysis are presented in Sect. 3. Testing and validation of proposed model for IoT applications is shown in Sect. 4. Section 5 concludes the study and discusses future research aims.

2 Related work

There is substantial amount of published research on chaos based PRNGs. In this paper, the reviewed literature explores chaos based PRNGs and mainly focuses on CML based PRNGs.

There is a variety of chaotic maps available for pseudo-random number generation. A famous choice of many researchers is to use the logistic map, which is one of the most widely studied chaotic maps [12]. Various versions of modified logistic maps have been used for PRNGs including pseudo-randomly enhanced logistic map (PELM) [13], floating point based modified logistic chaotic system [14], hyper-chaotic modified robust logistic map (HC-MRLM) [15], modified logistic map for increased key-space range [16], nonlinear digitalized modified logistic map [17], and optimised logistic map using perturbation operation [18]. The logistic map exhibits chaotic behaviour for a very small range of control parameter values between [3.57, 4]. However, there have been studies in attempt to increase the parameter range for the chaotic behaviour in logistic map. In a recent study on pseudo-random bit generator, logistic maps were used to generate multimodal maps and the parameter interval showed an increase in the chaotic range [19]. In a few other studies, researchers have used different chaotic maps such as the Lorenz map [20], the Tinkerbell map [21] and the Chirikov map [22].

Several electronic and hardware based implementations for chaos based PRNG have been presented. Matheus et al. proposed a gate level hardware implementation for a PRNG based on an exponential chaotic map [23]. The hardware implementation was designed using Field Programmable Gate Array (FPGA) device for the proposed PRNG system. There is a significant number of hardware implementations for chaotic based PRNGs using FPGAs including FPGA based PRNG using four-wing memristive hyperchaotic system and bernoulli map [24], reconfigurable chaotic PRNG based on FPGA [25], chaos based bitwise dynamical PRNG on FPGA [26], and FPGA implementation of chaos based PRNG for secure communication [27]. FPGAs are preferred by algorithm designers for prototyping as they can execute the code on gate level using hardware descriptive languages. Since, in real world applications microcontrollers are used more commonly, thus several proposals on chaos based PRNGs have been made using microcontrollers. For instance, an analog circuit and microcontroller based PRNG application of a new easy realizable 4D chaotic system has been recently proposed [28], in another study a chaotic random bit generator has been realized with a microcontroller [29], and hardware implementation of initials-boosted coexisting chaos in a two dimensional sine map [30]. The IoT technology that is rapidly gaining popularity in everyday applications depend on resource constrained devices, and there are not many solutions available for IoT sensors. In a recently published work, the authors claimed to design encryption solution for IoT devices based on novel chaotic map but it had several shortcomings [31].

They decided to use non-chaotic algorithms like PRESENT and LED to encrypt initial conditions, this would add overhead to the remaining chaos based encryption process. The time analysis performed for the proposed algorithm was conducted using Intel(R) Core(TM) i5-4210U 2.40 GHz CPU, 4GB RAM computer which is equipped with strong computing power and still takes minimum 6.7 s for 1000 iterations and 72.5 s for 14,373 iterations. The suggested encryption algorithm would fail completely if it was being deployed on a real IoT temperature sensor, which has a single core processor and sends temperature feed multiple times during one second time. Considering, the importance of IoT data and the challenges in terms of limited power and computing resources, there is a dire need for chaos based cryptosystem that is lightweight in terms of computation and also provide robust security against attacks.

Chaos based PRNGs are used in several security applications, for instance stream ciphers [32], block ciphers [33], secure communications [34], image encryption [35], and video encryption [36] amongst others. Despite their popularity and successful applications, there are several shortcomings that chaos based PRNGs possess. Ramazan Yeniçeri et al. published a study in which they showed how a random number generator based on a time delay differential equation could be attacked [37]. To generate an attack, they predicted the values before hand and coupled them with future states of a time delay based chaotic system. Once the system was synchronized with it, the generated signals were similar to the original ones. In another study, the cryptanalysis of chaos based PRNGs was conducted and several improvements were suggested to rectify system weaknesses [38]. Youling et al. conducted cryptanalysis on chaos based cryptosystems with the aspect of hardware based attacks [39]. They first implemented chaos based cryptosystem onto a microcontroller device and conducted side channel attack analysis and correlation power analysis attack, which can be performed by studying the execution time and power consumption of the cryptographic algorithm. Furthermore, in various other studies, vulnerability and security analysis has been performed for chaos based PRNGs in the context of cryptographic applications. In a recent study, researchers conducted security analysis of the efficient chaos based PRNG for video encryption application [40], and in another study a thorough vulnerability analysis has been performed for chaos based PRNG [41].

A remedy to the weaknesses discussed in chaos based PRNGs could be to use complex dynamical systems to generate chaos rather than using simple one (1D) or two dimensional (2D) chaotic maps, this would make the prediction of generated patterns quite strenuous to crack. A

multidimensional chaotic system with discrete time and discrete spatial extension was introduced by Kaneko in the 1980's [42]. These type of spatio-temporal chaotic systems are known as CMLs, that produce a wide array of behaviours. CMLs are capable of producing complex chaotic sequences as they form lattice structure which comprises of local chaotic maps. Despite the complex chaotic behaviour, there are some inherent shortcomings in CML systems which limit their application as PRNG. One major drawback in the CML systems is the limited range of parameter space for the spatio-temporal chaos and another limitation is the variation in local chaotic behaviors i.e some lattices might not reach chaotic zone [43]. There have been several improvements to the Kaneko's model of CML system for PRNG applications. Ping et al. recommended a pseudo-random bit generator (PRBG) based on mixing the state variables of a CML to generate more complex pseudo-random sequences [44]. In another study, a variable time delay has been suggested for the CML based PRNG. The results show that the proposed system is not hefty with computations and also passed all randomness tests provided by NIST [45]. Recently, a CML system with asymmetric coupling has been used as PRBG with a sawtooth map as the local map [46]. The authors also suggested the use of specific systemic parameters, that could convert complex floating calculations to bit operations. In [47], the authors proposed a CML based on discrete chaotic iteration (CMLDCI) that combine the CML and chaotic iteration to produce random sequences for PRNG. By chaotic iteration, they refer to a convergence theorem for chaos coined by Devany et al. in early literature [48]. The idea of a Multi Bit Random Number Generator (MBRNG) published recently, proposed a CML based random number generator based on logistic map [49]. The proposed system does not present any novelty, as it uses logistic map as the local map similar to the conventional CML system. In another article, the authors used skew tent map as local map [50]. The idea is further extended to analyse the behaviour of system by cross coupling skew tent maps.

In the light of reviewed literature, majority of the latest research proposals use a very few well known maps like the logistic map [12] or tent map [51] as local maps for the CML systems. Logistic and tent maps belong to the family of symmetric chaotic maps which contain other higher order maps as well. To our best knowledge from the literature review, there has not been any attempt to explore the family of symmetric maps in general for the CMLs. In this paper, we introduce the concept of Generalised Symmetric Map (GSM) as local map for the CML system and also propose the concept of adaptive control parameter values that results in highly chaotic and complex PRNG.

3 Proposed model

In this paper, a unique PRNG model has been proposed for IoT devices, that generates random sequences with large key space and could be used in various cryptographic applications. The proposed model comprises of a spatiotemporal CML system using GSM as local map. The concept of adaptive β values, based on accumulation points of local GSM map has also been introduced. This ensures that the local maps stays in the chaotic range for majority of the time and they outcome pseudo-random sequences for any initial seed values. Detailed working of the proposed system is explained in the subsequent subsections.

3.1 Coupled map lattice (CML)

This idea of generating spatiotemporal chaos via CMLs was published in a series of studies during the 1980's ranging from the exploration of spatio-temporal intermittency of coupled map lattices [52] to spatio-temporal coupled nonlinear oscillators [53], and space-time dynamics in video feedback [54], with Kaneko being the most active contributor in this domain [55]. The simplest way to visualise a CML system is to consider the diffusive CML model as shown in Eq. (1) [56].

$$x_{n+1}(i) = (1 - \epsilon)f(x_n(i)) + \frac{\epsilon}{2}[f(x_n(i + 1)) + f(x_n(i - 1))] \tag{1}$$

This is a diffusive two-way CML model, where the parameter i gives the i th lattice point, which in this case is coupled with two of its neighbouring nodes i.e. right node $(i + 1)$ and left node $(i - 1)$. i varies between $[1, L]$ where L is the total number of lattice points. The coupling factor is

represented by ϵ , which is a real number in $[0, 1]$. Finally, n is the discrete time step for the CML system. Here, $f(x)$ denotes the local map, which is usually a logistic or tent map but in the proposed model, $f(x)$ could be any chosen chaotic map from symmetric maps family.

3.2 Generalised symmetric map (GSM)

Symmetric maps belong to family of chaotic maps, which when plotted exhibit mirror image of the plot itself (maintain symmetry of its shape). Figure 1a and b show the mapping and psuedo phase trajectory to plot a 2D and 3D image of the tent map. The tent map is known to be a first order symmetric map as it is a straight line map and is symmetric across the x-axis. Figure 2a and b show the plotting for second order logistic map. Logistic maps are one of the most widely studied chaotic maps in the literature and show a parabolic shape when plotted. Figures 3 and 4 show the visualisation for third order and fourth order chaotic symmetric maps. To visualise the properties of symmetric maps, consider Figs. 1, 2, 3 and 4. In this paper, we introduce the concept of (GSM) as defined in Eq. (2).

$$f(x) = \beta(1 - |1 - 2x|^\alpha) \tag{2}$$

where alpha (α) and beta (β) are the control parameters for the GSM. $\alpha \in [1, 4]$ represents the map order and β is the control parameter that decides chaotic or non-chaotic zone of the chosen map being a real number $[0, 1]$. The core working behind GSM is to utilise any map from the symmetric chaotic maps using a single mathematical function i.e Eq. (2). This can be observed that by increasing the value of parameter α in the range $[1, 4]$, transitions the

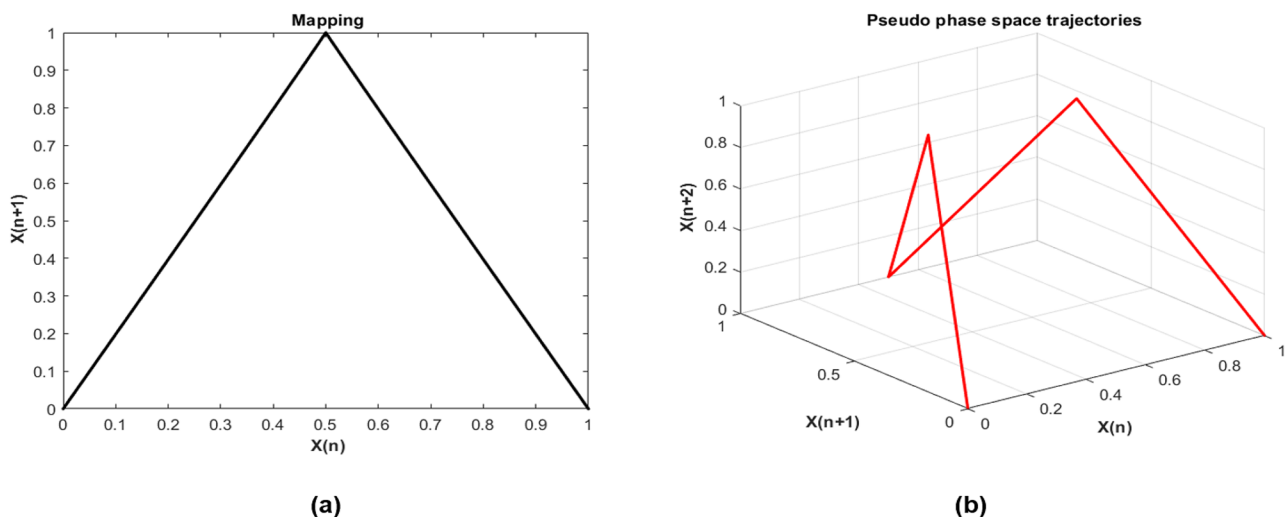


Fig. 1 Shows the a mapping and b pseudo phase space trajectory for first order symmetric chaotic map, also known as Tent map

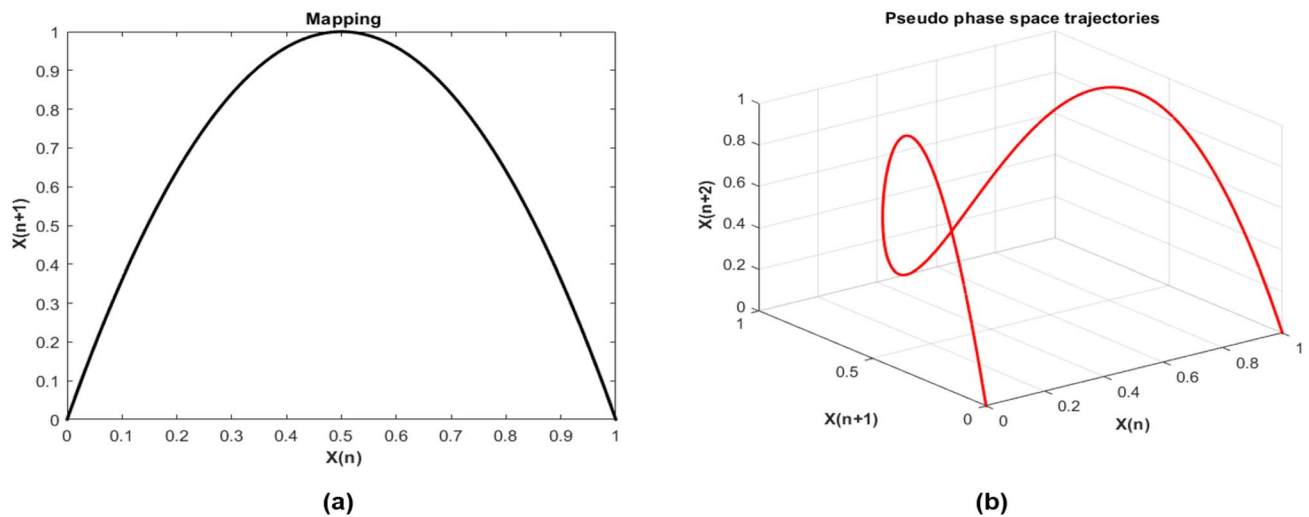


Fig. 2 Shows the **a** mapping and **b** pseudo phase space trajectory for the second order symmetric chaotic map, also known as logistic map

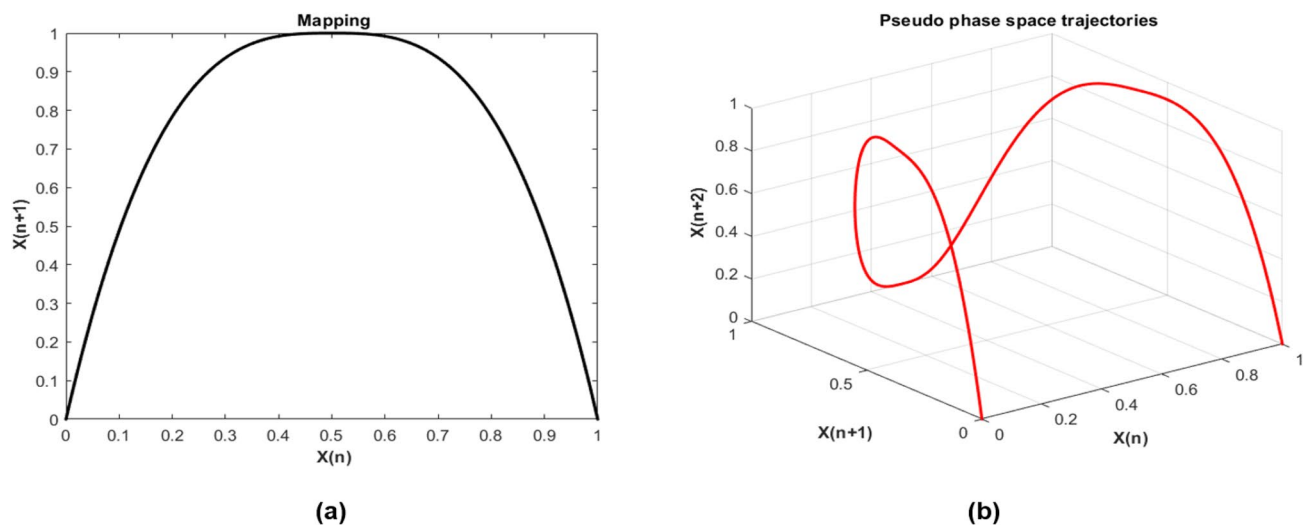


Fig. 3 Shows the **a** mapping and **b** pseudo phase space trajectory for third order symmetric chaotic map

map order from first to fourth order map (i.e from Figs. 1, 2, 3 and 4).

Using GSM as local maps widens the range of maps to choose from, by providing the liberty to choose local map as any symmetric chaotic map. The selection of α and the specific choice of corresponding control parameters would become extremely difficult for an adversary to guess the exact order and type of map used.

3.3 Adaptive β values

Chaotic maps present chaotic behaviour for specific values of control parameter and initial conditions. The

control parameter value at which a 1D map enters in the chaotic zone is known as accumulation point of the map. The accumulation point can be computed via the Lyapunov exponent analysis [57]. The point where the Lyapunov exponent switches from negative to positive values, can be considered as the accumulation point. This can be easily explained by visualizing the Lyapunov spectrum of the Logistic map, as shown in Fig. 5. For the logistic map, α is set to 2 in GSM Eq. (2), and the accumulation point (a) can be observed between 0.89 and 0.90 (0.89995 ignoring the high precision). Therefore, the concept of accumulation point allows us to formulate a system, where we built an adaptive scheme

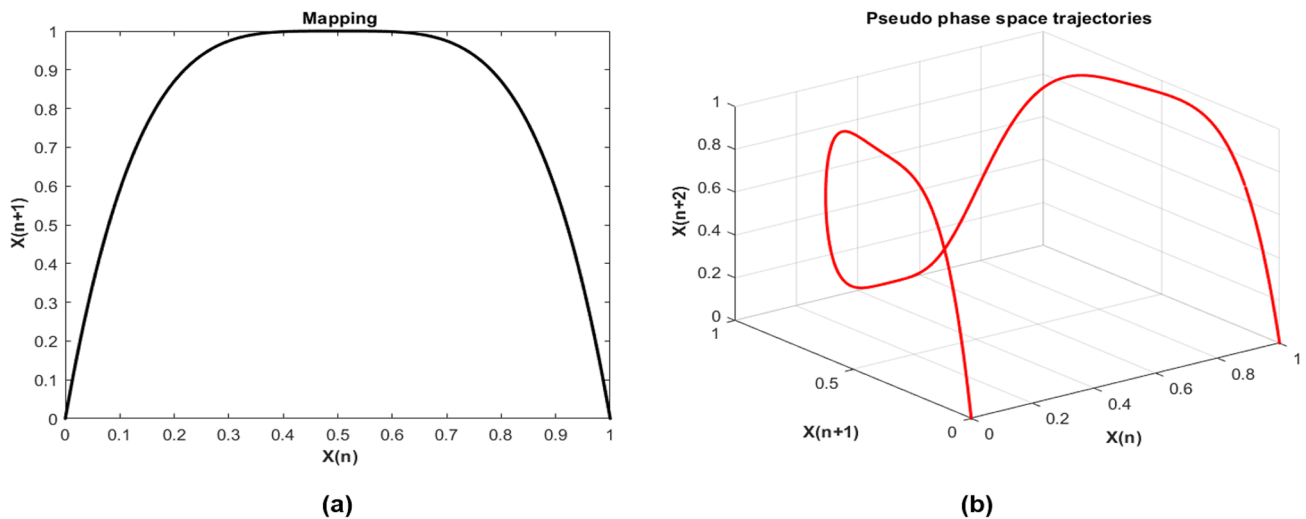


Fig. 4 Shows the **a** mapping and **b** pseudo phase space trajectory for fourth order symmetric chaotic map, also known as quadratic map

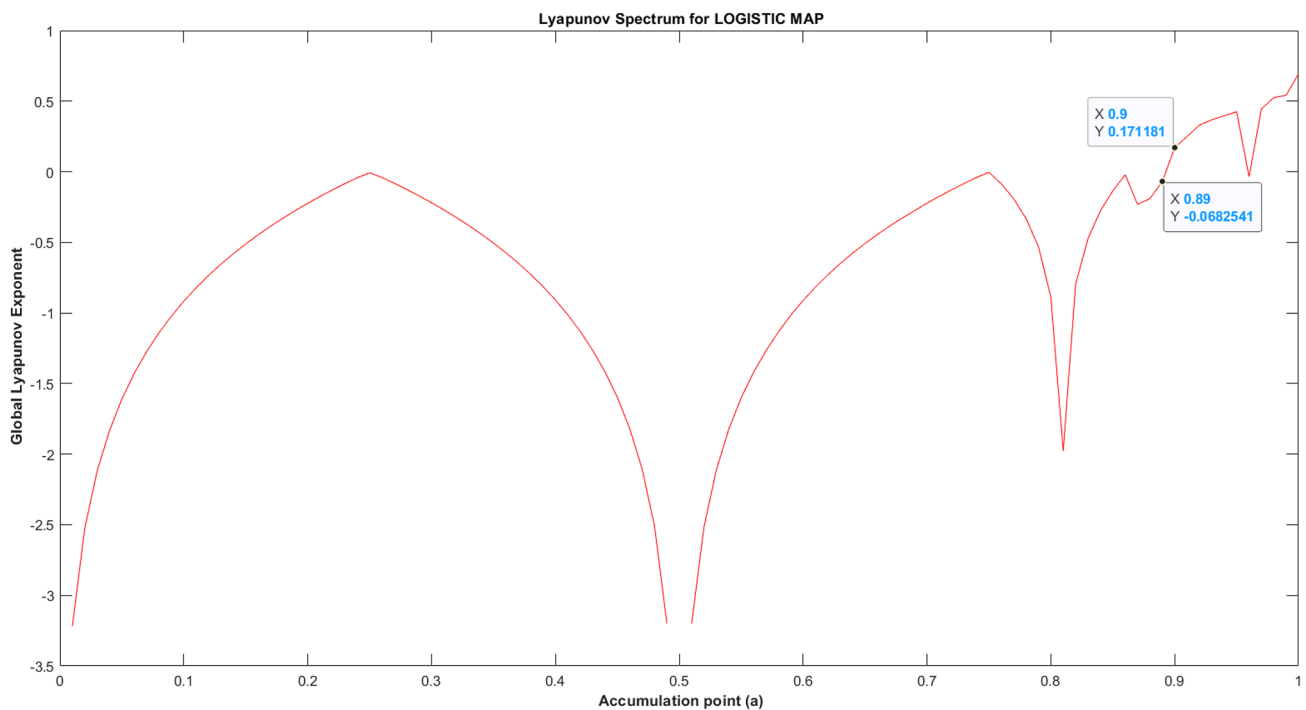


Fig. 5 Shows the Lyapunov spectrum and accumulation point for the logistic map

that calculates β values for any value of α such that they always remain above the accumulation point maintaining a chaotic output for the system.

Furthermore, to investigate the trend of accumulation points and control parameter α , we calculate Lyapunov spectrum for GSMs by varying the values of α in the range $[-4, 4]$. Table 1 summarizes the range of α values and their respective accumulation points. This can be observed from

Table 1 data, that the GSM function is non-chaotic for the interval when α is in the range $[-0.5, 0.5]$. To visualise the trend of accumulation points for GSM, Table 1 data is interpolated as shown in Fig. 6. As in this study, we are more focused onto exploring symmetric maps and having constant range of parameter α for cryptographic applications, we ignore the negative range of α and consider

Table 1 Shows the α values and corresponding accumulation points correct to 2 decimal places for the GSM

Control parameter (α)	Accumulation point (a)
-4.00	1.00
-3.50	1.00
-3.00	1.00
-2.50	1.00
-2.00	0.99
-1.50	0.97
-1.00	0.51
-0.75	0.67
-0.50	Non-chaotic
-0.35	Non-chaotic
-0.25	Non-chaotic
0.00	Non-chaotic
0.25	Non-chaotic
0.35	Non-chaotic
0.50	1.00
0.75	0.67
1.00	0.50
1.50	0.82
2.00	0.90
2.50	0.93
3.00	0.95
3.50	0.97
4.00	1.00

α range between [1,4] which corresponds to symmetric maps of order one to four.

From the above discussion, it can be concluded that there is a higher chance for the system to exhibit chaotic behaviour, if the control parameters are chosen such that which would result in system values to remain above the accumulation points. Therefore, to assure that the system remains in chaotic zone majority of the time, we introduce a concept of adaptive β (ad_β) as shown in Eq. (3).

$$ad_{\beta i} = a + (1 - a)(1 - e^{-\gamma i}) \tag{3}$$

where ad_β is the adaptive β value, calculated for a specific accumulation point. The present lattice number is presented as i . A new parameter Gamma (γ) is introduced to control the spread of ad_β values across all the lattice points. γ is defined as a positive real number i.e. ($\gamma > 0$). The main functionality of Eq. (3) is to keep the ad_β values always in range above the accumulation point $a \leq ad_\beta \leq 1$. The ad_β function when supplied with accumulation points for GSM, returns ad_β values such that, the CML system stays above the accumulation points and therefore exhibits chaotic output depending on the type of local map chosen.

3.4 KS entropy analysis

The Lyapunov spectra of a system [58] is an important tool to classify its chaotic behaviour. A CML system with N

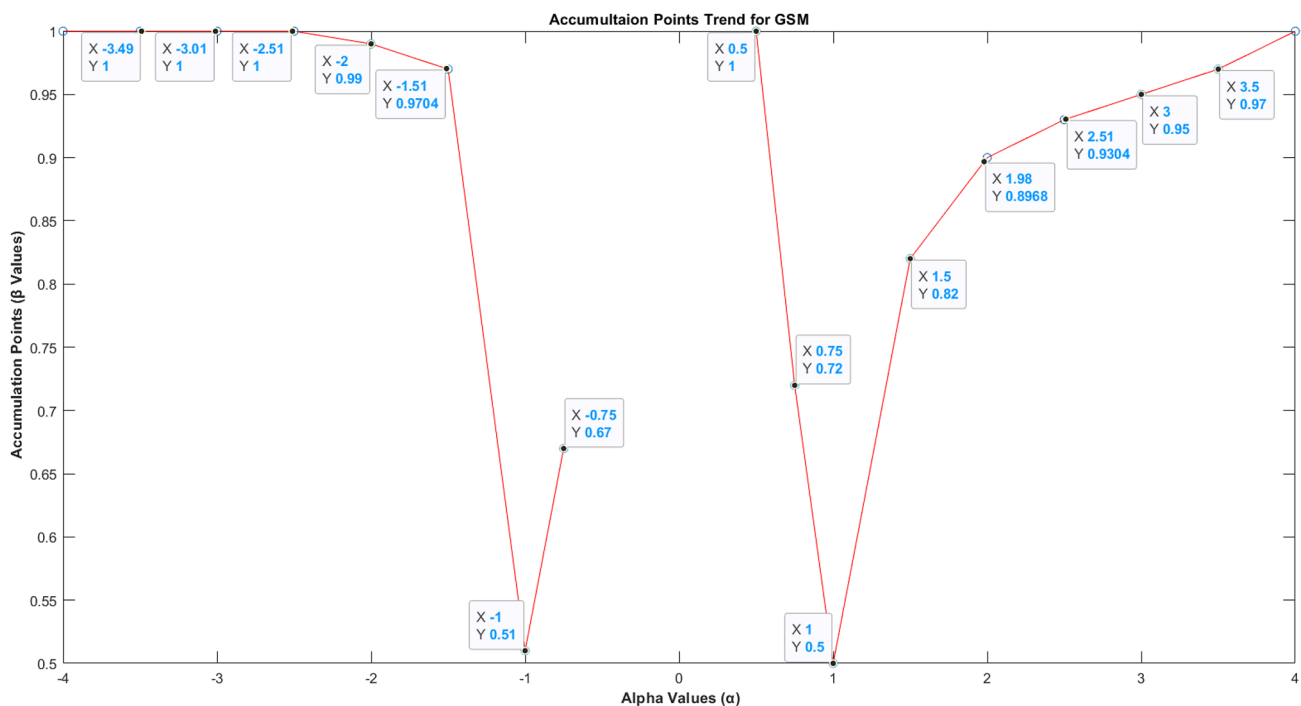


Fig. 6 Shows the trend line of α values and respective accumulation points for GSM

-dimensional phase space gives rise to N Lyapunov exponents. To calculate the Lyapunov spectra in a CML system, one needs the product of the Jacobian matrices (J_p) of the map as shown in Eq. (4).

$$J_p = \lim_{n \rightarrow \infty} \left(\frac{1}{n} \right) J_n J_{n-1} J_{n-2} \dots J_2 J_1 \tag{4}$$

where J_n is the individual Jacobian calculated at time step n . J_p is the final product of all Jacobian matrices divided by the time step n . Lyapunov exponents λ_i of the CML system can be calculated via Eq. (5).

$$\lambda_i = \ln(\text{eig}(J_{pi})) \tag{5}$$

where the logarithms of eigenvalues of J_p are retrieved that are finally divided by the time steps n . Once all the Lyapunov exponents are found, the Lyapunov Spectra is defined as the ordered set $\{\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N\}$. The sum of all positive Lyapunov exponents leads to the Kolmogorov–Sinai (KS) entropy [59]. KS entropy density h_μ can be represented using Eq. (6).

$$h_\mu = \frac{1}{N} \sum_{\lambda_i > 0} \lambda_i \tag{6}$$

where λ_i are the Lyapunov exponents calculated in the previous steps. N is the number of lattice points, or

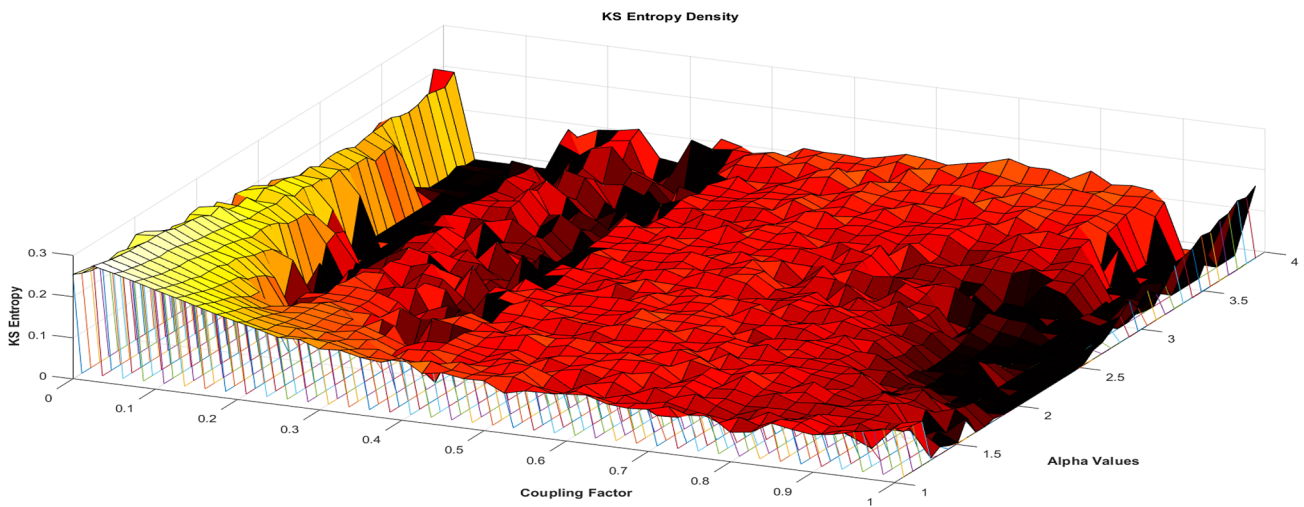


Fig. 7 Shows KS entropy density plot for CML system with GSM and adaptive β for the set parameters ($1 \leq \alpha \leq 4, 0 \leq \epsilon \leq 1, \beta = \text{adaptive}, N = 10$ and $\gamma = 0.25$)

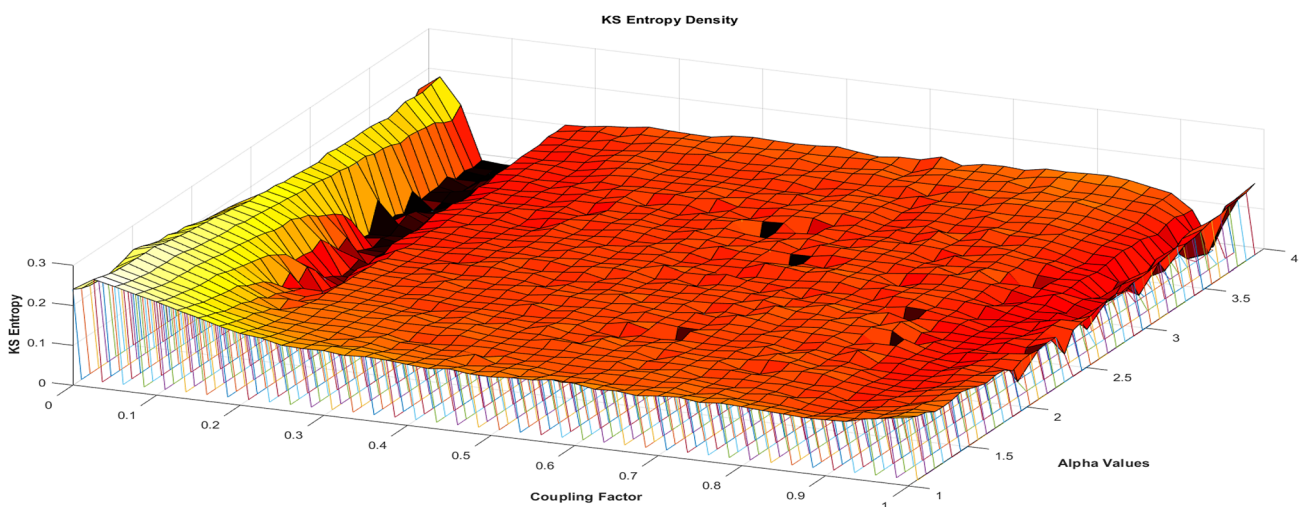


Fig. 8 Shows KS entropy density plot for CML system with GSM and adaptive β for the set parameters ($1 \leq \alpha \leq 4, 0 \leq \epsilon \leq 1, \beta = \text{adaptive}, N = 100$ and $\gamma = 0.025$)

dimension of the CML lattice. To further analyse the chaotic behaviour of the CML system based on GSM with ad_β , KS entropy density analysis has been visualised by varying control parameters. Figure 7 shows the KS entropy density analysis of a system with 10 lattice points, 1000 calculated time steps and 100 discarded iterative values to account for transient behaviour, α ranges between [1,4], e spread between [0,1], γ set to 0.25 and β is adaptive. Since γ is a tuning parameter for ad_β values, it should be tuned with respect to the system. The value for γ can be chosen by observing the ad_β value calculated for each lattice point. In our case we observed that γ gave a good spread of ad_β values for ($\gamma = 0.25$ for $N = 10$) and ($\gamma = 0.025$ for $N = 100$).

Figure 7 shows the KS entropy density analysis of the CML system with GSM mapping and ad_β values for 10 lattice points, whereas Fig. 8 shows the result for the system with a 100 lattice points. It can be observed from Fig. 7 that there were a number of regions of (ϵ, α) parameter, space where the entropy density can be assumed to be zero, whereas, in Fig. 8 the system shows a uniformly populated KS-entropy plot with very few regions of low density. Therefore from Figs. 7 and 8, it can be speculated that increasing the number of lattices improves the overall chaotic behaviour of the CML system with GSM and ad_β .

3.5 Bifurcation analysis

Bifurcation diagrams aid in visualising overall period-doubling progression of a system with increase in control parameters. The KS entropy analysis performed in the previous subsection shows that the coupling factor (ϵ) had great influence on the overall behaviour of the system. Therefore, bifurcation diagrams of the proposed system for first, middle and last lattice point of the proposed CML system have been visualised, for varying values of ϵ (i.e. 0.2, 0.5, and 0.8). The bifurcation diagrams were plotted for increasing α values on x-axis in range [1,4], across the values x values generated by CML system on y-axis ranging between [0,1].

Figure 9 shows the bifurcation analysis of the proposed system with 10 lattice points, 1000 initial discarded iterations, 2000 calculated iterations, and γ set to 0.25. To investigate the lattice behaviour in detail, in Fig. 9 columns present lattice number and rows denote the specific ϵ values. The results have been realised for first, middle and last lattice of the proposed system to interpret overall behaviour of the system. The results can be visually interpreted as several periodic regions appearing for ϵ at 0.2 and 0.8 whereas,

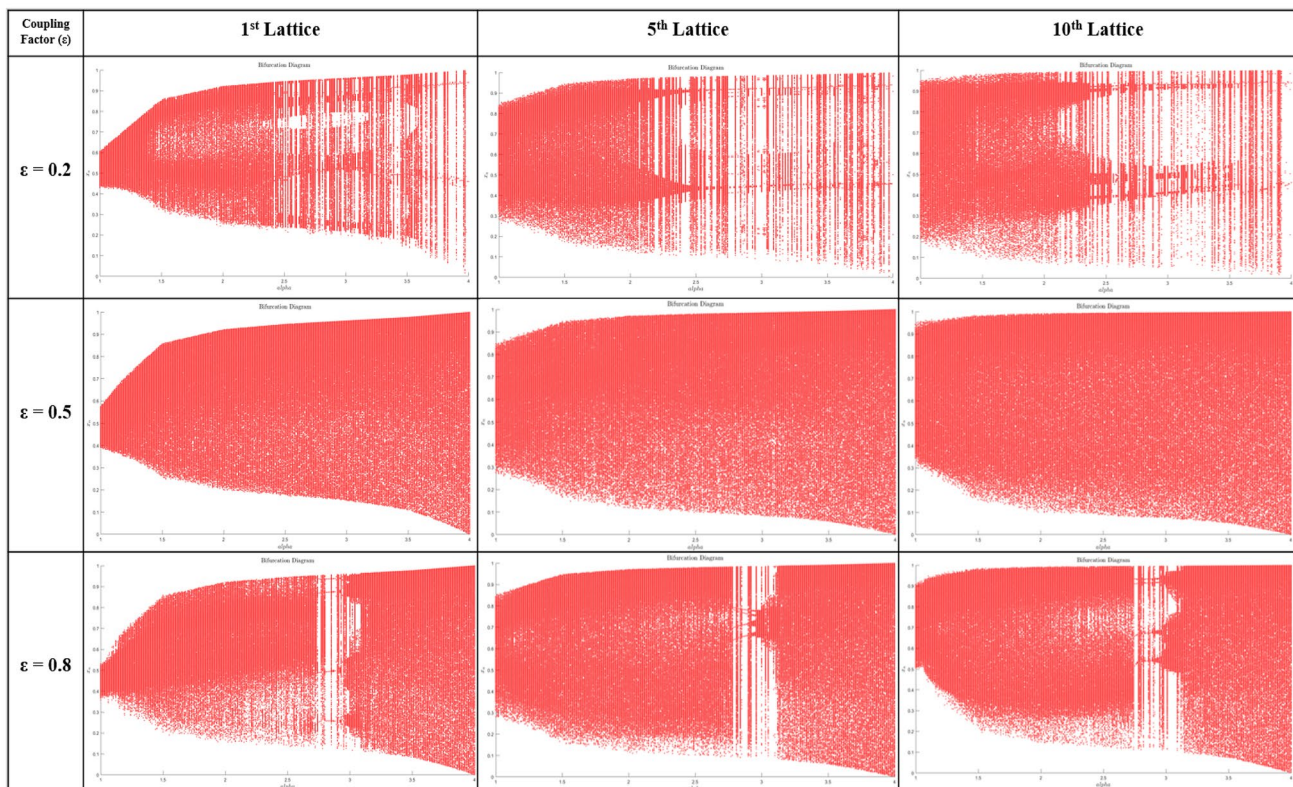


Fig. 9 Shows the bifurcation diagrams for CML system with 10 lattice points for increasing ϵ values and different lattice numbers

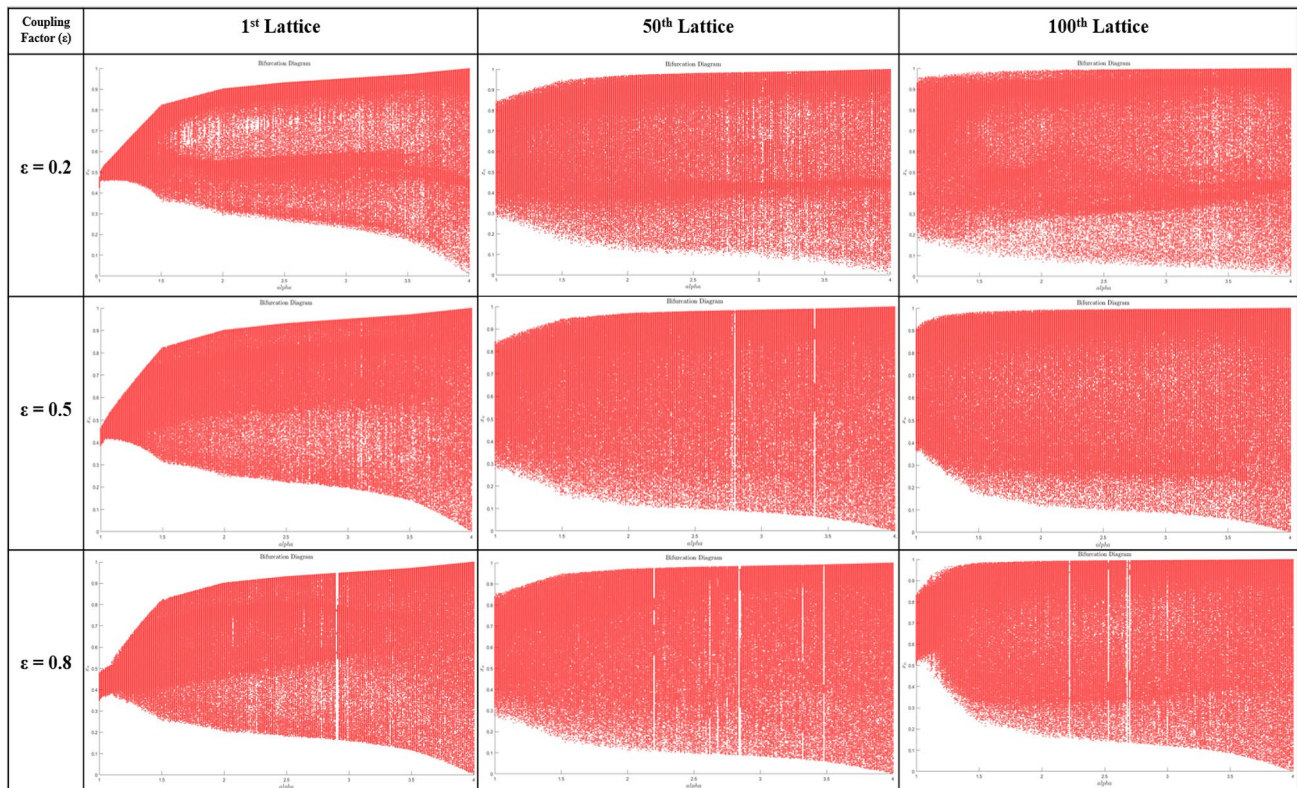


Fig. 10 Shows the bifurcation diagrams for CML system with 100 lattice points for increasing ϵ values and different lattice numbers

for coupling factor of 0.5, the chaotic pattern is evenly spread. Similar experiments were performed for the lattice number increased to 100 lattice points, 1000 initial discarded iterations, 2000 calculated iterations and γ value used as 0.025. Figure 10 shows the detailed trend of bifurcation plots for first, fiftieth, hundredth lattice point presented in columns, whereas the rows of Fig. 10 show the increasing values of coupling factor ($\epsilon = 0.2, 0.5$ and 0.8). Overall, the system with 100 lattice points appears more consistently chaotic for all coupling factors as compared to the system with 10 lattice points. Considering KS entropy experiments and bifurcation diagram analysis, this can be deduced that increasing the lattice number of the proposed system results generation of highly chaotic sequences.

4 Random number generator using proposed system

In this section, we define the steps to create a PRNG based on a CML system with GSM mapping and adaptive β . We also tested the proposed method for randomness by benchmarking it to the available standards and applying it to a real world example of IoT sensors. A pseudo random number must hold some properties close to true random numbers. Another requirement for PRNGs used with IoT sensors devices is high throughput and low latency, thus we assumed 64-bit precision representation using fixed point arithmetic. The detailed steps for generating pseudo-random sequence (64-bit) are explained in Algorithm 1.

Algorithm 1 Generate a 64-bit pseudo-random sequence using CML system with GSM mapping and adaptive β values

Function: $f(x) = \beta(1 - |1 - 2x|^\alpha)$

Output: 64-bit random number sequence

Initialization:

Set Control parameters: α (alpha), a (accumulation point), γ (gamma), ϵ (coupling), t (transient cut)

Set Initial conditions: x_0

Step 1: Select the sequence length required (Calculated as $n \times N$).

- Set the number of time steps = n
- Select the number of lattice points = N

Step 2: Calculate adaptive β for every lattice point i .

```

for  $i$  in range  $N$  do
     $ad_{\beta_i} = a + (1 - a)(1 - \epsilon^{-\gamma\sqrt{i^2}})$ 
end for

```

Step 3: Discard t number of iterations to avoid transient response.

```

for  $n$  in range  $t$  do
     $x_{n+1}(i) = (1 - \epsilon)(f(x_n(i))) + (\frac{\epsilon}{2})(f(x_n)(i - 1)) + f(x_n(i + 1))$ 
end for

```

Step 4: Iterate the CML system for n number of iterations.

```

for  $n$  in range  $n$  do
     $x_{n+1}(i) = (1 - \epsilon)(f(x_n(i))) + (\frac{\epsilon}{2})(f(x_n)(i - 1)) + f(x_n(i + 1))$ 
end for

```

Step 5: Convert the floating-point x values to 64-bit integers.

64-bit random number sequence = $\lfloor x_{n+1}(2^{64} - 1) \rfloor$

```

if sequence length from Step 4 not appropriate then
    Repeat from Step 1 to Step 5 with new  $n$  and  $N$  value.
else
    Quit
end if

```

4.1 Key space analysis

Random number generators play basic role in cryptographic applications, generating strong encryption and decryption keys. By the term strong, we mean that the keys with higher key space are harder to guess, thus the higher the key space is, the stronger is the cryptographic key. More specifically, a cryptographic key smaller than 128-bit is not considered strong enough to be used for data encryption purposes [60]. The proposed method

depends on several control parameters and initial conditions which collectively contribute towards the key space. The experiments were performed using Python 3.7 as programming language with floating point as data type with 53 bits representation (16 digits precision i.e 10^{-16}). Therefore, the key space can be calculated from parameters: $x \in [0, 1]$, $\alpha \in [0, 4]$, $\beta \in [0, 1]$, $\epsilon \in [0, 1]$, $\gamma \in [0, 10^3]$, $n \in [0, 10^{10}]$, $N \in [0, 10^6]$ which is $(10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16}) = 1 \times 10^{112} \approx 2^{372}$. Table 2 shows a comparison of key space and

Table 2 Comparison of key space analysis of various algorithms

Algorithms	Key space	Control parameters
[24]	2^{240}	11
[35]	2^{372}	8
[61]	2^{162}	4
[21]	2^{183}	6
[62]	2^{279}	6
Proposed	2^{372}	7

control parameters with other recently published studies. The key space for the proposed algorithm is $\approx 2^{372}$, which is significantly large to survive brute force attacks.

4.2 Key randomness analysis

To validate the randomness of the system, a thorough testing has been performed using the Statistical Randomness Test Suite (SP 800-22) made available by National Institute of Standards and Technology (NIST) [63]. The NIST randomness test suite comprises of 15 randomness tests as shown in Table 3. Table 3 also shows the parameters that were used during the randomness testing experiments. To best analyse the behaviour of the system, we performed all 15 randomness tests for varying values of α and the ϵ . The heat map chart is plotted such that, for every chosen α and ϵ , the system was tested for the 15 NIST randomness tests, and the number of passed tests is shown in the respective block. The legend on the right side of heat map can also be used to categorize the maximum and minimum number of

NIST randomness tests passed using color codes. Considering that α could be any number in range [1,4], ϵ in between [0,1] and γ variable can make up to numerous possibilities to generate pseudo-random sequences, but for fair analysis to test randomness of the system, NIST randomness tests have been conducted on increasing values of α in range ($0.5 \leq \alpha \leq 4$) at an interval of 0.25, and selected coupling values ($\epsilon = 0, 0.1, 0.3, 0.5, 0.7, 0.9, 1$).

Figure 11 shows the heat map chart for the CML system with 10 lattice points, the overall number of pass tests were more than 10 except for some chunks between $1.75 \leq \alpha \leq 3.75$. The color legend on right of Fig. 11 shows the number of NIST tests passed out of a total fifteen randomness tests. Whereas, Fig. 12 shows the heat map chart for the proposed system with 100 lattice points. As expected, the overall behaviour of the system with 100 lattice points appears to be better in randomness as compared to the system with 10 lattice points. This can be observed that for majority of α and coupling values, the number of NIST tests 'passed' is greater than 10 randomness tests. This can be concluded that increasing the number of lattices, increases the randomness of output generated from CML based PRNG. The use-case considered for the proposed method is for IoT applications, where IoT sensors generate a huge amount of data at high frequency, which needs to be encrypted on the journey to its destination. Encrypting constant streams of data having critical timelines is a very challenging task, as fresh supply of encryption keys are required consistently. The proposed pseudo-random number generator serves the purpose, as the user can generate a wide range of pseudo-random sequences that are independent of any previous sequences. Figures 11 and 12 depict that for majority

Table 3 Fifteen randomness tests from NIST 800-22 Test Suite (version 2.1.2) and the parameters used in this paper

No.	Test type	Parameters
1	Frequency (monobit) test	
2	Frequency test within a block	Block size $m = 128$
3	Runs test	
4	Test for the longest run of ones in a block	$M = 100,000$
5	Binary matrix rank test	
6	Discrete Fourier transform (spectral) test	
7	Non-overlapping template matching test	$m = 9$
8	Overlapping template matching test	$m = 9$
9	Maurer's "universal statistical" test	
10	Linear complexity test	$M = 500$
11	Serial test	$m = 16$
12	Approximate entropy test	$m = 10$
13	Cumulative sums (cusum) test	
14	Random excursions test	
15	Random excursions variant test	sample size = 10 bit streams

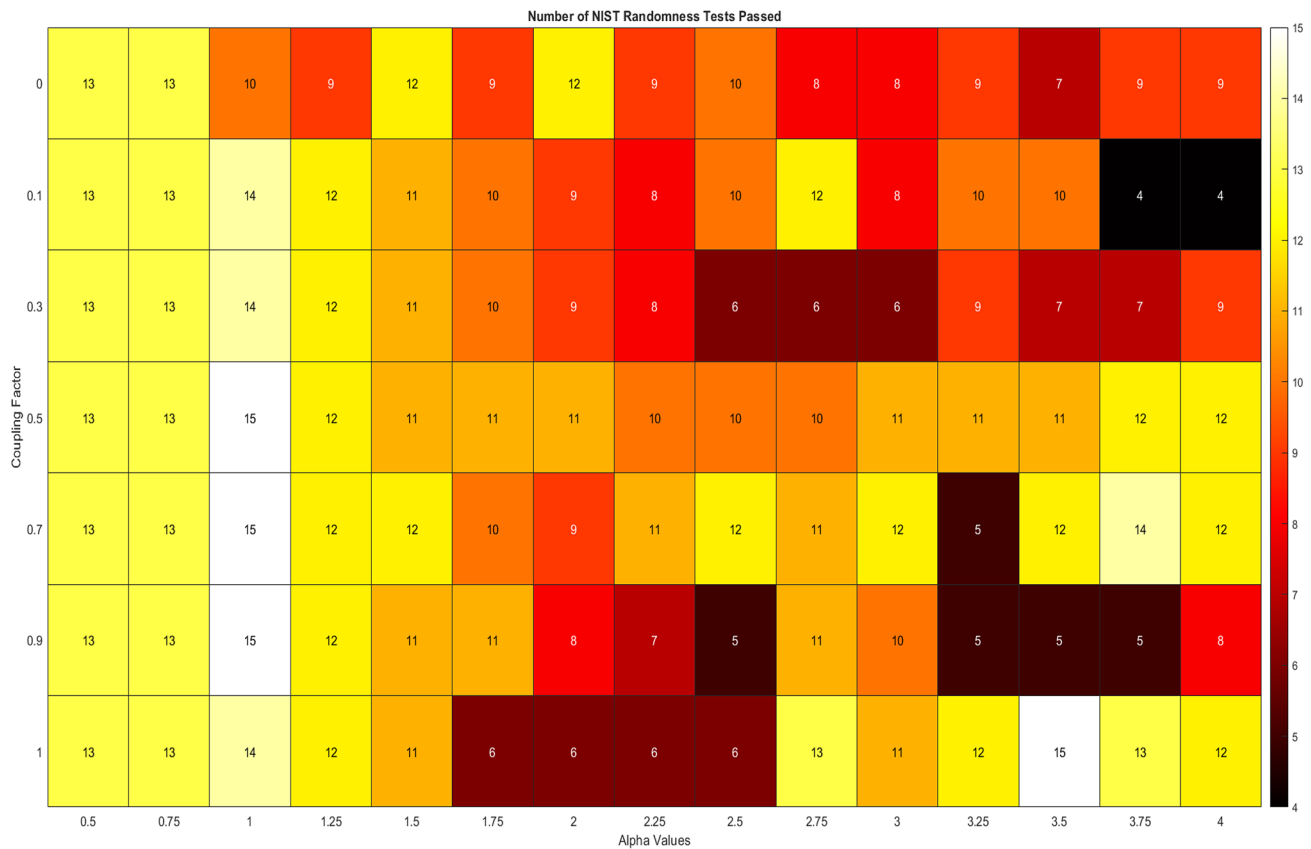


Fig. 11 Shows the number of NIST randomness tests passed by the proposed system with ten lattice points and the parameters ($n = 100,000$, $t = 1000$, $N = 10$, $\gamma = 0.25$, $\alpha = [0.5, 4]$, $\epsilon = [0, 1]$)

of the randomness tests, the pass rate is above 10 tests which means the sequences generated are fairly pseudo-random in nature. Thus, unlimited number of keys can be confidently generated for different devices and sessions, slightly varying the control parameters for the proposed system generates new pseudo-random sequences that are different and independent from the previously generated sequences.

4.3 Performance analysis on IoT devices

The ubiquity of IoT devices is increasing and thus it is crucial to secure the data generated by such hardware. The limitation that hinders the deployment of security primitives on IoT devices is that they are scarce in terms of computation, memory and power resources. Thus, when designing the security algorithms, one should keep the resource limitations of IoT devices under consideration. Even though, IoT devices have evolved and sensors currently available on the market are quite efficient and reliable. To test the proposed algorithm, we chose highly resource constrained systems on chip, Raspberry pi zero (single-core processor) and Raspberry pi 3B+ (quad-core

processor). Table 4 summarises the execution time of the proposed PRNG on both sensor devices with increasing number of iterations and lattice points. It should be noted that increasing lattice number and number of iterations together has double impact on the computation time but still the proposed PRNG is capable of generating pseudo-random sequences in less than a second.

4.4 Comparison with existing techniques

One important stage in proposal of new algorithm is to perform a comparative analysis of proposed method with existing state of the art techniques. This type of comparison allows the researchers to deduce a fair opinion on what evaluation matrices have been considered during the proposal of new algorithm. Table 5 shows the summary of comparison performed between proposed pseudo-random number generator with the existing PRNG techniques based on CML systems.

This can be observed from Table 5, that in majority of existing techniques, the evaluation of proposed ideas has not been performed using all standard measures. In fact, only some of the evaluation techniques have been

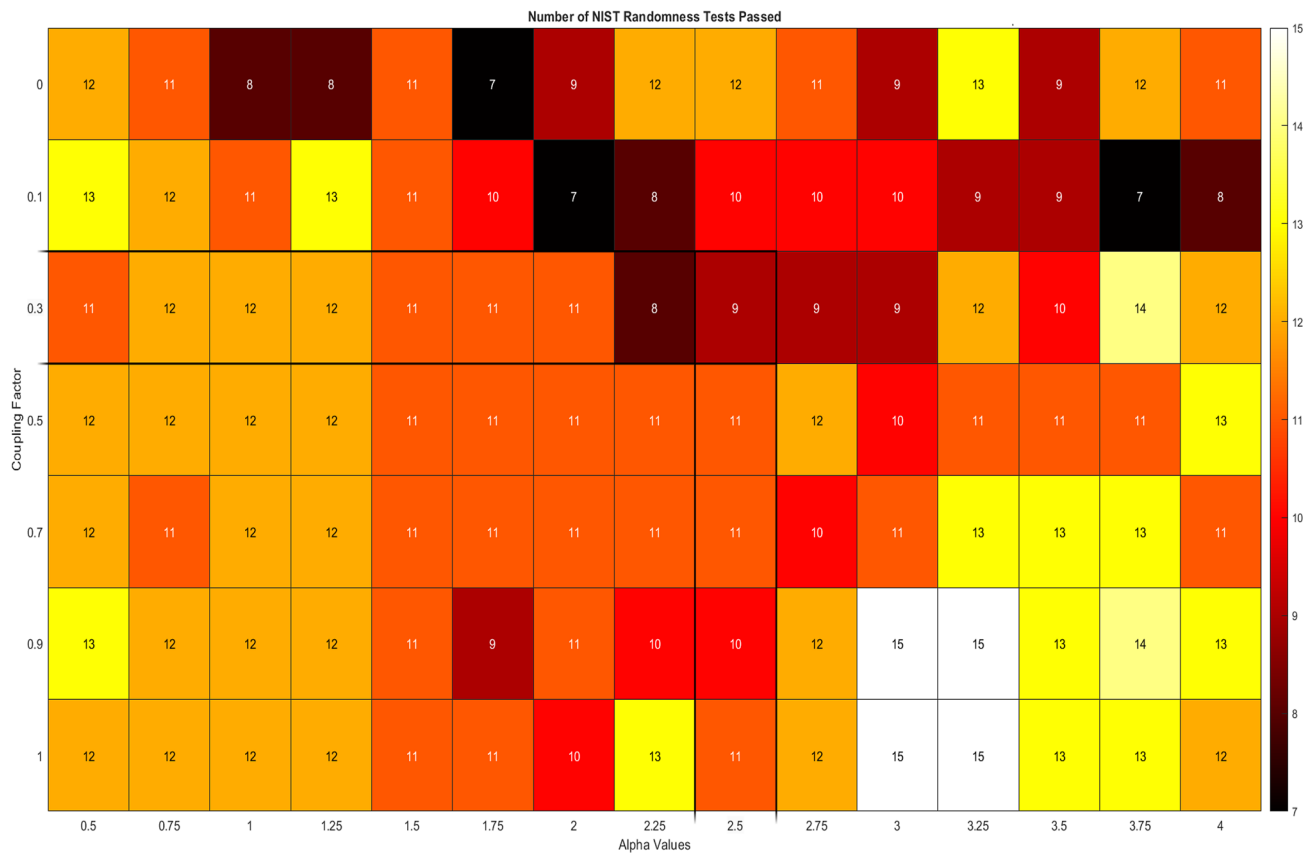


Fig. 12 Shows the number of NIST randomness tests passed by the proposed system with hundred lattice points and the parameters ($n = 100,000, t = 1000, N = 100, \gamma = 0.025, \alpha = [0.5, 4], \epsilon = [0, 1]$)

Table 4 Time taken by proposed PRNG to generate pseudo-random sequences on resource constrained IoT devices based on increasing number of iterations and lattice points

Raspberry Pi zero W (Single core-processor)			Raspberry Pi 3B+ (Four processing cores)		
Lattice points	Number of iterations	Time taken	Lattice points	Number of iterations	Time taken
10	10	0.0093	10	10	0.0014
10	100	0.0930	10	100	0.0136
10	1000	0.9270	10	1000	0.1359
100	10	0.0114	100	10	0.00189
100	100	0.1135	100	100	0.0178
100	1000	1.1350	100	1000	0.1773

considered for validation of proposed method. In this study, for the proposed CML based PRNG, we realised detailed proposal of GSM using CML and visualised their pseudo phase trajectories and Lyapunov spectrum. To analyse the entropy analysis of the system, KS entropy plots were created. Furthermore, for detailed examination of lattice behaviour, bifurcation diagrams were plotted for first, middle and last lattice point of the proposed system. In other existing techniques, only few studies like [44] have used KS entropy or bifurcation analysis to study their proposed CML systems. Whilst designing a PRNG for

cryptosystem, it is crucial to consider the key space and execution speed of the proposed algorithm. Again a very few researchers consider validating their proposed algorithm against the key space size. Majority of the discussed PRNG schemes in Table 5 in have not used keyspace analysis to measure the strength of their proposed algorithm in terms of security robustness.

Most important test for a new PRNG algorithm is to test for the degree of randomness using standard benchmarks. There are variety of randomness tests available but the standard test recognised by research community is NIST

Table 5 Comparison of proposed method with existing pseudo-random number generation techniques on basis of Chaos Properties, Security Analysis and Statistical Tests

	Proposed Method	Ref. [44]	Ref. [45]	Ref. [46]	Ref. [47]	Ref. [49]	Ref. [50]
<i>Chaos analysis</i>							
Local chaotic map	Generalised symmetric map	Logistic map	Logistic map	Sawtooth map	Chaotic iteration	Logistic map	Skew tent map
Pseudo phase trajectories	✓	✗	✓	✗	✗	✗	✓
Lyapunov exponent	✓	✗	✗	✓	✗	✓	✓
Kolmogorov Sinai entropy	✓	✗	✓	✗	✗	✗	✗
Bifurcation diagram	✓	✗	✗	✗	✗	✗	✗
<i>Security analysis</i>							
Key space	✓	✓	✗	✗	✗	✗	✗
Speed	✓	✗	✓	✗	✗	✗	✓
<i>Statistical analysis</i>							
NIST randomness tests	✓	✓	✓	✓	✓	✗	✓
Other randomness tests	✗	✗	✓	✗	✗	✗	✗
Comparison with other PRNG techniques	✓	✗	✓	✗	✗	✗	✗

statistical test suite [63]. In Table 5, this can be observed that almost all the discussed existing PRNG techniques have been tested using NIST randomness test suite. Few studies like [45] have tested their algorithm against other randomness testing benchmarks such as *TESTU01* [64]. This can be deduced from the comparative analysis that the proposed method has been evaluated on the basis of standard benchmarking and testing standards available.

5 Conclusion

In this study, generalised symmetric chaotic maps have been explored. A method to generalise the symmetric maps has been introduced, which allows different chaotic maps to be chosen by changing single a parameter α . Chaotic maps do not always generate chaotic behaviour, rather they show such behaviour only in a specific range of control parameters and initial conditions. We proposed the concept of adaptive control parameter ad_{β} , which based on the accumulation points, strongly increases the likelihood of chaotic behaviour of the map, irrespective of what α value is chosen. A tuning parameter γ has also been proposed, which helps in manually adjusting the spread of β values. The additional control parameters contribute towards large key space for PRNG. The KS entropy analysis, bifurcation diagrams and randomness tests show that the proposed system tends to show random behaviour for majority of α values with exception to few periodic chunks.

The proposed PRNG has also been tested on real time IoT sensors to validate the computational efficiency. It is concluded that the proposed PRNG system can generate cryptographically strong keys, and is computationally suitable for lightweight devices such as IoT sensors. In future, this work would be extended to develop novel cryptosystems, such as image and video encryption schemes.

Acknowledgements This research is supported by the BTIC (British Telecom Ireland Innovation Centre) project, funded by British Telecom and Invest Northern Ireland.

Author Contributions All authors contributed equally.

Funding This research is supported by the BTIC (British Telecom Ireland Innovation Centre) project, funded by British Telecom and Invest Northern Ireland.

Availability of data and materials The data supporting the findings of this study is included in this article.

Code availability The softwares and tools used in this study are open source and available online.

Declarations

Conflict of interest The authors have no conflicts of interest to declare. All co-authors have seen and agreed with the contents of the manuscript. We certify that the submission is original work and is not under review at any other publication.

Ethics approval This article does not contain any studies with human participants or animals per-formed by any of the authors.

Consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Want R (2009) When cell phones become computers. *IEEE Pervasive Comput* 8(2):2–5. <https://doi.org/10.1109/MPRV.2009.40>
2. Wortmann F, Flüchter K (2015) Internet of things. *Bus Inf Syst Eng* 57:221–224. <https://doi.org/10.1007/s12599-015-0383-3>
3. Andrea I, Chrysostomou C, Hadjichristofi G (2015) Internet of Things: security vulnerabilities and challenges. In: 2015 IEEE symposium on computers and communication (ISCC). pp 180–187. <https://doi.org/10.1109/ISCC.2015.7405513>
4. Alaba FA, Othman M, Hashem IAT, Alotaibi F (2017) Internet of Things security. *J Netw Comput Appl* 88(C):10–28. <https://doi.org/10.1016/j.jnca.2017.04.002>
5. Schindler W (2009) Random number generators for cryptographic applications. In: Koç ÇK (ed) *Cryptographic engineering*. Springer, Boston
6. Sunar B (2009) True random number generators for cryptography. In: Koç ÇK (ed) *Cryptographic engineering*. Springer, Boston
7. Senkerik R, Pluhacek M, Zelinka I, Davendra D, Oplatkova ZK (2015) A brief survey on the chaotic systems as the pseudo random number generators. In: Sanayei A, Rössler EO, Zelinka I (eds) *ISCS 2014: interdisciplinary symposium on complex systems. Emergence, complexity and computation*, vol 14. Springer, Cham
8. Loginov SS, Zuev MY (2018) Testing of generators of pseudo-random signals based on a Lorenz system, realized over a Galois finite field. In: 2018 Systems of signal synchronization, generating and processing in telecommunications (SYNCHROINFO). pp 1–4. <https://doi.org/10.1109/SYNCHROINFO.2018.8457039>
9. Bhattacharjee K, Maity K, Das S (2018) A search for good pseudo-random number generators: survey and empirical studies. *arXiv preprint arXiv:1811.04035*
10. Oishi S, Inoue H (1982) Pseudo-random number generators and chaos. *IEICE Trans* (1976–1990) 65(9):534–541
11. Chazottes JR, Fernandez B (eds) (2005) *Dynamics of coupled map lattices and of related spatially extended systems*, vol 671. Springer, Berlin
12. Ausloos M (2006) *The logistic map and the route to chaos: from the beginnings to modern applications*. Springer, Berlin
13. Murillo-Escobar MA, Cruz-Hernández C, Cardoza-Avendaño L et al (2017) A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dyn* 87:407–425. <https://doi.org/10.1007/s11071-016-3051-3>
14. Wang L, Cheng H (2019) Pseudo-random number generator based on logistic chaotic system. *Entropy* 21:960. <https://doi.org/10.3390/e21100960>
15. Irfan M, Ali A, Khan MA, Ehatisham-ul-Haq M, Mehmood Shah SN, Saboor A, Ahmad W (2020) Pseudorandom number generator (PRNG) design using hyper-chaotic modified robust logistic map (HC-MRLM). *Electronics* 9:104. <https://doi.org/10.3390/electronics9010104>
16. Hemdan AM, Faragallah OS, Elshakankiry O et al (2019) A fast hybrid image cryptosystem based on random generator and modified logistic map. *Multimed Tools Appl* 78:16177–16193. <https://doi.org/10.1007/s11042-018-6948-7>
17. Chen S, Hwang T, Lin W (2010) Randomness enhancement using digitalized modified logistic map. *IEEE Trans Circuits Syst II Express Briefs* 57(12):996–1000. <https://doi.org/10.1109/TCSII.2010.2083170>
18. Liu J, Liang Z, Luo Y, Cao L, Zhang S, Wang Y, Yang S (2021) A hardware pseudo-random number generator using stochastic computing and logistic map. *Micromachines* 12:31. <https://doi.org/10.3390/mi12010031>
19. García-Martínez M, Campos-Cantón E (2015) Pseudo-random bit generator based on multi-modal maps. *Nonlinear Dyn* 82:2119–2131. <https://doi.org/10.1007/s11071-015-2303-y>
20. García-Martínez M et al (2015) Hyperchaotic encryption based on multi-scroll piecewise linear systems. *Appl Math Comput* 270:413–424. <https://doi.org/10.1016/j.amc.2015.08.037>
21. Stoyanov B, Kordov K (2015) Novel secure pseudo-random number generation scheme based on two tinkerbells maps. *Adv Stud Theor Phys* 9(9):411–421. <https://doi.org/10.12988/astp.2015.5342>
22. Tutueva A, Pesterev D, Karimov A, Butusov D, Ostrovskii V (2019) Adaptive Chirikov map for pseudo-random number generation in chaos-based stream encryption. In: 2019 25th conference of open innovations association (FRUCT). pp 333–338. <https://doi.org/10.23919/FRUCT48121.2019.8981516>
23. Cardoso MBR, da Silva SS, Nardo LG, Passos RM, Nepomuceno EG, Arias-Garcia J (2021) A new PRNG hardware architecture based on an exponential chaotic map. In: IEEE international symposium on circuits and systems (ISCAS). pp 1–5. <https://doi.org/10.1109/ISCAS51556.2021.9401653>
24. Yu F et al (2019) Design and FPGA implementation of a pseudorandom number generator based on a four-wing memristive hyperchaotic system and bernoulli map. *IEEE Access* 7:181884–181898. <https://doi.org/10.1109/ACCESS.2019.2956573>
25. Rezk AA et al (2019) Reconfigurable chaotic pseudo random number generator based on FPGA. *AEU-Int J Electron Commun* 98:174–180. <https://doi.org/10.1016/j.aeue.2018.10.024>
26. Garcia-Bosque M, Pérez-Resca A, Sánchez-Azqueta C, Aldea C, Celma S (2019) Chaos-based bitwise dynamical pseudorandom number generator on FPGA. *IEEE Trans Instrum Meas* 68(1):291–293. <https://doi.org/10.1109/TIM.2018.2877859>
27. Hobincu R, Datcu O (2018) FPGA implementation of a chaos based PRNG targeting secret communication. In: International symposium on electronics and telecommunications (ISETC). pp 1–4. <https://doi.org/10.1109/ISETC.2018.8583863>
28. Kaçar S (2016) Analog circuit and microcontroller based RNG application of a new easy realizable 4D chaotic system. *Optik* 127(20):9551–9561. <https://doi.org/10.1016/j.jlleo.2016.07.044>
29. Volos CK (2013) Chaotic random bit generator realized with a microcontroller. *J Comput Model* 3(4):115–136

30. Bao H, Hua Z, Wang N, Zhu L, Chen M, Bao B (2021) Initials-boosted coexisting chaos in a 2-D sine map and its hardware implementation. *IEEE Trans Ind Inform* 17(2):1132–1140. <https://doi.org/10.1109/TII.2020.2992438>
31. Nesa N, Ghosh T, Banerjee I (2019) Design of a chaos-based encryption scheme for sensor data using a novel logarithmic chaotic map. *J Inf Secur Appl* 47:320–328
32. Liu Z et al (2020) A stream cipher algorithm based on 2D coupled map lattice and partitioned cellular automata. *Nonlinear Dyn* 101(2):1383–1396
33. Wang X, Bao X (2013) A novel block cryptosystem based on the coupled chaotic map lattice. *Nonlinear Dyn* 72(4):707–715
34. Peng Z et al (2021) Secure communication based on microcontroller unit with a novel five-dimensional hyperchaotic system. *Arab J Sci Eng* 2021:1–16
35. Som S et al (2015) Confusion and diffusion of color images with multiple chaotic maps and chaos-based pseudorandom binary number generator. *Nonlinear Dyn* 80(1):615–627
36. Xu H, Tong X, Meng X (2016) An efficient chaos pseudo-random number generator applied to video encryption. *Optik* 127(20):9305–9319
37. Yeniçeri R, Kiliç S, Yalçın ME (2015) Attack on a chaos-based random number generator using anticipating synchronization. *Int J Bifurc Chaos* 25(02):1550021. <https://doi.org/10.1142/S0218127415500212>
38. Ergün S (2016) Cryptanalysis and improvement of a chaos based random number generator. In: *International symposium on electronics and smart devices (ISESD)*. pp 199–202. <https://doi.org/10.1109/ISESD.2016.7886719>
39. Luo Y, Zhang D, Liu J, Liu Y, Cao Y, Ding X (2018) Cryptanalysis of chaos-based cryptosystem from the hardware perspective. *Int J Bifurc Chaos* 28(09):1850114
40. Lambić D, Janković A, Ahmad M (2018) Security analysis of the efficient chaos pseudo-random number generator applied to video encryption. *J Electron Test* 34:709–715. <https://doi.org/10.1007/s10836-018-5767-0>
41. Ergun S (2018) Vulnerability analysis of a chaos-based random number generator. In: *2018 IEEE international conference on systems, man, and cybernetics (SMC)*. pp 3331–3334. <https://doi.org/10.1109/SMC.2018.00564>
42. Kaneko K (1989) Spatiotemporal chaos in one-and two-dimensional coupled map lattices. *Physica D Nonlinear Phenom* 37(1–3):60–82. [https://doi.org/10.1016/0167-2789\(89\)90117-6](https://doi.org/10.1016/0167-2789(89)90117-6)
43. Huang R, Han F, Liao X, Wang Z, Dong A (2021) A novel intermittent jumping coupled map lattice based on multiple chaotic maps. *Appl Sci* 11(9):3797. <https://doi.org/10.3390/app11093797>
44. Wang P, Qiu J (2017) A pseudorandom bit generator based on mixing of state variable of CML. In: *2017 IEEE 2nd information technology, networking, electronic and automation control conference (ITNEC)*. pp 331–335. <https://doi.org/10.1109/ITNEC.2017.8285000>
45. Lv X, Liao X, Yang B (2018) A novel pseudo-random number generator from coupled map lattice with time-varying delay. *Nonlinear Dyn* 94:325–341. <https://doi.org/10.1007/s11071-018-4361-4>
46. Liang R, Tan X, Zhou H et al (2015) An efficient parallel pseudorandom bit generator based on an asymmetric coupled chaotic map lattice. *Pramana J Phys* 85:617–627. <https://doi.org/10.1007/s12043-014-0905-4>
47. Wang X, Qin X (2012) A new pseudo-random number generator based on CML and chaotic iteration. *Nonlinear Dyn* 70:1589–1592. <https://doi.org/10.1007/s11071-012-0558-0>
48. Bahi JM, Guyeux C (2010) Hash functions using chaotic iterations. *J Algorithms Comput Technol*. <https://doi.org/10.1260/1748-3018.4.2.167>
49. Li P, Li Z, Halang WA, Chen G (2005) A novel multiple pseudo random bits generator based on spatiotemporal chaos. *IFAC Proc* 38(1):1085–1089. <https://doi.org/10.3182/20050703-6-CZ-1902.00837>
50. Elmanfaloty RA, Abou-Bakr E (2019) Random property enhancement of a 1D chaotic PRNG with finite precision implementation. *Chaos Solitons Fractals* 118:134–144. <https://doi.org/10.1016/j.chaos.2018.11.019>
51. Shan L, Qiang H, Li J, Wang ZQ (2005) Chaotic optimization algorithm based on Tent map. *Control Decis* 20(2):179–182
52. Chaté H, Manneville P (1988) Spatio-temporal intermittency in coupled map lattices. *Physica D Nonlinear Phenom* 32(3):409–422. <https://doi.org/10.1143/PTP.74.1033>
53. Waller I, Kapral R (1984) Spatial and temporal structure in systems of coupled nonlinear oscillators. *Phys Rev A* 30(4):2047. <https://doi.org/10.1103/PhysRevA.30.2047>
54. Crutchfield JP (1984) Space-time dynamics in video feedback. *Physica D Nonlinear Phenom* 10(1–2):229–245. [https://doi.org/10.1016/0167-2789\(84\)90264-1](https://doi.org/10.1016/0167-2789(84)90264-1)
55. Kaneko K (1984) Period-doubling of kink-antikink patterns, quasiperiodicity in antiferro-like structures and spatial intermittency in coupled logistic lattice: towards a prelude of a “field theory of chaos”. *Prog Theor Phys* 72(3):480–486. <https://doi.org/10.1143/PTP.72.480>
56. Kaneko K (1989) Pattern dynamics in spatiotemporal chaos: pattern selection, diffusion of defect and pattern competition intermittency. *Physica D Nonlinear Phenom* 34(1–2):1–41. [https://doi.org/10.1016/0167-2789\(89\)90227-3](https://doi.org/10.1016/0167-2789(89)90227-3)
57. Young LS (1982) Dimension, entropy and Lyapunov exponents. *Ergod Theory Dyn Syst* 2(1):109–124
58. Kaneko K (1986) Lyapunov analysis and information flow in coupled map lattices. *Physica D Nonlinear Phenom* 23(1–3):436–447
59. Kaneko K (1993) The coupled map lattice: introduction, phenomenology, lyapunov analysis, thermodynamics. In: *Theory and applications*
60. Smart N et al (2012) ECRYPT II yearly report on algorithms and key sizes (2011–2012). In: *European network of excellence in cryptology (ECRYPT II)*
61. Wang Y, Liu Z, Ma J et al (2016) A pseudorandom number generator based on piecewise logistic map. *Nonlinear Dyn* 83:2373–2391. <https://doi.org/10.1007/s11071-015-2488-0>
62. Hamza R (2017) A novel pseudo random sequence generator for image-cryptographic applications. *J Inf Secur Appl* 35:119–127. <https://doi.org/10.1016/j.jisa.2017.06.005>
63. Rukhin A, Soto J, Nechvatal J, Smid M, Barker E (2001) A statistical test suite for random and pseudorandom number generators for cryptographic applications. *Booz-allen and hamilton inc mclean va*
64. Lécuyer P, Simard R (2007) TestU01: AC library for empirical testing of random number generators. *ACM Trans Math Softw (TOMS)* 33(4):1–40

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.