Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19093-z

ICSPRNG: Ikeda assisted Cryptographically Secure Pseudo Random Number Generator

Subhajit Adhikari[1] · **Anirban Panja**[2] · **Sunil Karforma**[2]

Received: 11 October 2023 / Revised: 21 February 2024 / Accepted: 27 March 2024

Abstract
Pseudorandom numbers have several uses in cryptography, including digital signatures, one-time passwords, hashing, and encryption techniques. The randomness of pseudo random number generators improves significantly when dynamical systems such as chaos maps are used. The two-dimensional ikeda map is an excellent option for exhibiting this random behavior. In this study, the notion of a chaos based Ikeda map and the random number are used to create a pseudo random number sequence that achieves cryptographic-level security, which is referred to as the Ikeda-assisted Cryptographically Secure Pseudo Random Number Generator. Different statistical analyses demonstrate the effectiveness and strength of the random sequence in terms of less control parameters, less amount of generation time, high entropy value, almost zero value for auto correlation and correlation coefficient. Our random sequence demonstrates that it is resilient and cryptographically safe by passing all of the Diehard battery tests with the required value. Also, our method has qualified the majority of the test results of the randomness test recognized by the National Institute of Standards and Technology with the required value. This pseudo random number sequence can be used to generate a single password, as an initialization vector, a replacement or permutation step in a cryptosystem, or as a secret key in encryption methods for picture, text, audio, and video data.

Keywords Pseudo random number · Cryptographically secure · Chaos map · Ikeda map · Security · NIST test · Diehard test

✉ Subhajit Adhikari
subhajit15dec@gmail.com

Anirban Panja
anirbanwithyou@gmail.com

Sunil Karforma
sunilkarforma@yahoo.co.in

[1] BSH Department, IEM(UEM), Kolkata, India

[2] Department of Computer Science, The University of Burdwan, Burdwan, India

Published online: 09 April 2024

Highlights

In this section, we have listed the main contrinutions below.

- Application of chaos theory in the field of cryptography.
- A design for a cryptographically secure pseudo-random number generator generating unique random sequences.
- Evaluate and examine the suggested approach using standard tests to evaluate its applicability for cryptographic purposes.

1 Introduction

An arbitrary number is called as random number when it is selected from a set of all possible potential outcomes, satisfying the condition that all the generated numbers are uniformly distributed in the state space and statistically independent from each other [1].

The concept of number generators that are random in nature, is further subdivided into two categories which are "True Random Number Generators" and "Pseudo-Random Number Generators" (called "PRNG"). The generated output of the "PRNG" is totally based on the initial seed values, which would be extremely hard to predict and guess. The initial seed value must be kept secret; otherwise, if someone gains access to the seed or can manipulate its generation, they can easily predict the PRNG's output, which leads to the breakdown of the entire system. Therefore, it is very hard for the researchers to create cryptographically secure PRNGs. The domain of PRNGs is further classified into two main groups: basic PRNGs and Computationally Secure Pseudo-Random Number Generators(CSPRNG). Simulation is the key area where basic PRNGs are applied, while CSPRNGs are heavily used for cryptographic applications. A random number is fall under the category of CSPRNGs it satisfies two main properties: firstly the number must successfully pass tests for statistical randomness; secondly, the generated number must exhibit robust security measurements against severe attacks, even if the initial parameters values or ongoing state of the generator are exposed to potential attackers. The unique features of the random numbers produced by any CSPRNG are defined by three main properties like high value of entropy, no string repetition, and zero correlation values [2].The pseudo-random number is an essential concept that has already demonstrated its relevance in several fields like one-time passwords (OTP), concept of digital signatures, concept of hashing, process of encryption, generation of seed vector and are the key areas where pseudo-random number generators [3] are broadly used.

Pseudo random numbers can be generated using various methods [1–11]. Among other methods, chaotic "PRN" is gaining popularity because of randomness and unpredictability. This concept is adopted in many cryptosystems for many security reasons. Also, the random values are generated in a very short time. In different resources limited areas, many researchers proved the chaotic pseudo random number sequence is very essential and gives a high level of security when applied with encryption and decryption processes. The term "different resources limited areas" means that the hardware resource and battery power in the field of sensors that are applicable for Internet Of Things(IOT) devices. If the encryption and decryption time is high, the battery power will decrease.

The control parameters of chaotic maps play a vital role in generating a large key space, so the attacks launched by a method that is brute force in nature can be prevented. The values in the histogram generated from the pseudo random numbers are uniform in most of the cases. This will make it the cryptosystem hard to guess the values by the attacker. There is a very

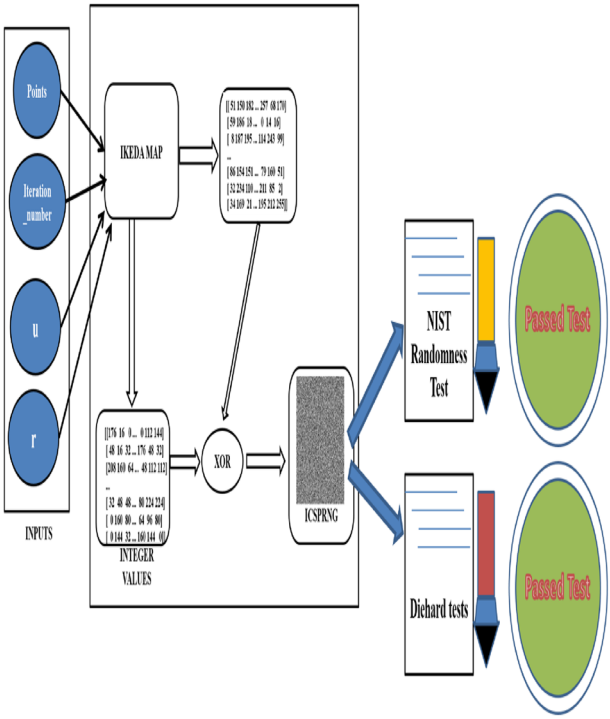

Fig. 1 Representation of Graphical Abstract

low or negative correlation between values can be found, as a consequence, the relationship between values will be destroyed. This will make all the attempts futile for the attacker. Different statistical properties should be tested to verify the randomness of the generator. In this research article, we develop a brand-new PRNGs approach based on the ideas of chaotic maps (Fig. 1). This can also be treated as cryptographically secure pseudo random number generators (known as "CSPRNG") because our generator proves the properties of CSPRNG. The main contributions are listed below.

1. Design of a pseudo random number using an Ikeda chaos map that offers cryptographic-level security.

2. It has passed majority of NIST randomness test and Diehard battery test , So, it shows good randomness behavior.

3. With only few control parameters of ikeda chaotic map, the random sequence is generated, extending the applicability of our PRNG to the resource restricted environment.

4. Generation time of the pseudo random sequence is very less,which proves its efficiency in terms of computational power.

5. The equal spread of histogram plot satisfies that it is very hard to detect the sequence values for attacker.

6. The high data value of the metric entropy shows a huge amount of uncertainty that makes it hard to guess the values from the histogram.

7. The auto correlation values are almost zero.

8. The study of correlation coefficient give the values are either close to zero or completely negative in three configurations , horizontal,vertical and diagonal respectively.

9. With the properties of high value of the metric entropy, no repeat of values in the sequence and zero correlation values, our pseudo random number proves it is Cryptographically Secure Pseudo Random Number

The organization of the paper is as follows, section "2 Background Study" describes the pre-requisites concepts regarding chaos theory. Next section "3 Literature survey" depicts the previous research contributions. Then the section "4 Algorithm and Flowchart for ICSPRNG" will present the proposed algorithm with flowchart. The next section "5 NIST test for ICSPRNG" and section "6 Diehard Battery test for ICSPRNG" will give the randomness test result of the generated pseudo random numbers. The section "7 Implementation example and Security Analysis of ICSPRNG" will give the details about the proposed method with example and security analysis. The section "8 Discussion and Future Scope" will present the usabiity and integrity of the proposed method and advancement possible in future.The section "9 Conclusion" describes the conclusion made from the result and analysis of the proposed method.

2 Background study

2.1 Chaotic theory

Dynamic systems [3] are the areas in the mathematical functions that use time to describe a point within geometric space. One of the common descriptions of the dynamic system is a clock pendulum. The concept of dynamic systems can be divided into two primary categories: dynamic systems linear in nature and dynamic systems with non-linear characteristics. A linear pattern is strictly followed by the assessment in linear dynamic systems; in other words, we can say that a proportional linear relationship to the minimal changes in the user's input is reflected in the generated output. On the other hand, a non-linear dynamic system is characterized by an output that does not reveal a proportional linear relationship to the minimal changes in the user's input. In the domain of non-linear dynamic systems, chaotic systems are one of the most popular non-linear dynamic systems. Chaotic mapping is an area of study in the mathematics domain that explores dynamic systems that generate a completely disordered random state. This state appears irregular, yet it is controlled by the initial seed parameters.

In the above Fig. 2, we try to represent the application areas of chaos maps in terms of nonlinearity.

Depending on the underlying patterns, chaotic maps produce different random states and the relationship between the underlying pattern and the output is the key area of chaos theory. Depending upon the underlying structure and output, chaotic maps are categorized into several types, which are Ikeda, Hitzl-zele, Henon, Tinkerbell, Logistic, Arnold Cat map, Zigzag, Rossler, Trigonometry based, Barnsley-Fern, Zaslavskey, Lemniscate, Quantum, Saw-tooth, Renyi, Tent and Lozi [3]. Most of the chaotic maps are 3-dimensional and work on complex or real numbers. The logistical chaos map can be 2D or Cubic in nature. The trigonometry-based chaotic map is further subdivided into four categories based on their applications, which are Cosine, Sine, Chebyshev and Cubic. In our research work, 2D Ikeda map is used. In the next section , the basic concept of 2D Ikeda map is elaborated.

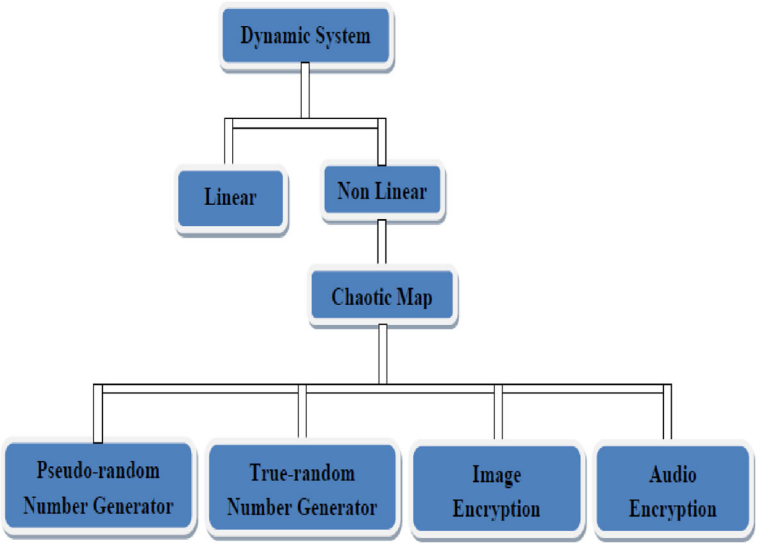

Fig. 2 Classification of Chaos Maps Applications

2.2 Ikeda chaotic map

Scientist Kensuke Ikeda introduced the concept of the Ikeda Map as the initial map serves as a representation of light's characteristics moving within a nonlinear optical cavity resonator [18]. The 2D Ikeda map is defined by equation [19] given below.

$$x_1' = 1 + \mu.[x_1.\cos(t) - x_2.\sin(t)]$$

$$x_2' = \mu.[x_1.\sin(t) + x_2.\cos(t)]$$

$$t = 0.4 - \frac{6}{1+x_1^2+x_2^2}$$

3 Literature survey

In a research work, the 1-D logistic map serves as the foundation for the proposed cPRNG [5], which generates random numbers by varying the value of the control parameter lambda. The test known as Wald-Wolfowitz test runs with six observations has been utilized in this study as a test of randomness. There are no statistical tests like histogram analysis, correlation analysis found in this manuscript.

In other research findings, following the resolution of issue created by the synchronization in the chaotic systems with the architecture of master and slave, [6], we provide a unique TRNG architecture that combines the chaos signals, which are synchronized and random, with the El-Gamal algorithm. It succeeded in every NIST test. But there is no comparative study made extensively to justify the effectiveness of the proposed random number generator algorithm with existing research studies.

Previously, one study found the improvement is based on applying a straightforward operation that is xor between the mantissa of a number in bits that makes up the sequence to further randomize the chaotic map-based sequence [7]. To improve the efficiency of both the

chaotic map like 1D logistic map and the 2D Henon map, an xor operation is performed on the mantissa of the state variables of both chaotic maps. A bit generator with pseudo-random behavior is a revolutionary concept that can be used on a wide range of ICT-enabled devices.

According to the article [8], a lightweight PRNG for Internet of Things(IOT) devices has been suggested. Famous chaotic maps like the Sawtooth never achieve a chaotic orbit in real-world applications because the data is finite in nature. In a lattice-based construction known as Coupled Map Lattices (termed "CML"), where all of the lattices are linked together one after another, we evaluated the chaotic Sawtooth map to be utilized as a local map. When used with IoT prototype sensors that have limited resources, the suggested PRNG has also been evaluated with respect to resource consumption and execution time. The outcomes demonstrate that the suggested PRNG was capable of producing random numbers very firstly, which is in microseconds, with input parameters such as a greater number of lattice points and iterations. As a limitation it can be stated that there is no mechanism given for key (seed value of the PRNG) exchange for cryptographic operations.

In some other study stated, in order to increase the chaotic system's unpredictability and increase its key space, a perturbation procedure is provided [9]. This surjective mapping fulfils the Li-Yorke chaos properties according to the unit area. The outcomes of the suggested technique is improved using a method for the composition of two systems, which finally results in a compound chaotic system by mixing a series obtained from quantum random walks combined with the outcomes of the chaotic system. Utilizing test methodologies including distribution of output frequency, autocorrelation, and time series complexity, the new system is assessed. The test findings demonstrated that the novel system has a high degree of unpredictability and can function effectively in a setting with poor precision equipment. The applicability of the PRNG in the field of resource constrained areas is not specified. The application areas are not completely defined in this manuscript.

In some other study, they have highlighted the prospect of using chaos based theories to create pseudo-random numbers [10]. More specifically, we choose to develop a chaotic pseudo-random number generator in hardware using a gate(field-programmable in nature) array of the Xilinx Spartan-XC6LX16 type. The randomness effect has been enhanced using a perturbation approach for implementation with finite accuracy. Because of this, our generated bit streams are ideal for use in the field of cryptography due to their high performance and low interception likelihood.

According to another study [11], they first developed a hyper-chaotic system with a dimension equal to four, after which we use the keystream associated with the original value of the text produced by the system to encode the original picture with the help of Deoxyribonucleic acid(DNA) coding dynamically. In the end, after the transformation of Ribonucleic acid(RNA) coding and substitution box generation of amino acids, at the time of replacement operations, the final ciphertext picture is produced with the help of an enhanced replacement sequence generator, producing pseudo-random sequences. Theoretical study and simulation findings demonstrate that the suggested method performs very well in terms of metrics of security such as the value of key space, the rate at which pixels get changed, the value of intensity at which they change on average, entropy, clipping, noise, and selected plaintext attacks. As a result, the algorithm is more secure.

One of the research attempts recommends a special discrete-time feedback 2D hyperchaotic map using the Henon and Sine maps [12] . Employing the remainder obtained after the division operation , where statistical properties with more random are gained, further enhances the dynamics of the hyperchaotic map. By employing histograms, sensitivity at launch, Lyapunov exponent analysis,and attractor trajectory, a comparison based analysis is drawn. The first seed of the generator of pseudo random number taking 8 bit , which

is based on the suggested hyperchaotic map, is then generated with the help a secret key composed of 60 numbers of characters in hexadecimal form. For experimental results, it is computed in MATLAB using an Arduino Mega microcontroller. From the perspective of cryptography, a thorough security analysis is provided, encompassing the value of key space, floating frequency, plot of histograms, and information entropy. The randomization is further supported by tests performed under NIST 800-22. The suggested method may be employed in embedded chaotic cryptography applications based on the security results. The secret key generation process is very complex in terms of different operations like addition, division and remainder calculation and not completely defined in the manuscript. This will lead to a very serious problem at the time of key exchange.

A PRNG [13] for usage in Field Programmable Gate Arrays-based design is suggested in the current study and relies on a non-equilibrium system, which is four-wing memristive hypercahotic in nature. The dual entropy sources architecture used by the innovative PRNG, together with the architecture and Bernoulli map, significantly improves the throughput and statistical measurements of the produced bit sequences. The bit series produced by the suggested pseudo-random number generator (PRNG) efficiently clears all statistical measurements from NIST 800.22, ENT, and AIS.31 test collections and produced the ultimate output bit rate of 62.5 Mbps. Finally, the security of the proposed model is measured using several statistical metrics such as dynamic deterioration analysis, study of key space, study of key sensitivity, analysis correlation, and information entropy. In this manuscript, there are several seed values like system parameters, control parameters and threshold parameter are used to generate the PRNG, which are to be transmitted as secret key and this leads to a problem of key exchange. Also, how the key exchange can be possible, is not mentioned in the manuscript.

Previously, it has been proved that the original image has been successfully reconstructed with the help of generalized Arnold cat map in a image encryption system [14]. Four 1-D chaos based mapping like one dimensional tent map, sine map, logistic map, and cubic map, have been employed to encode the scrambled picture. The experimental measurements such as mean number of pixels change rate, peak signal-to-noise ratio, squared error, and unified average change rate has been employed for the purpose of statistical evaluation of the encryption standard to demonstrate the effectiveness and precision of the suggested technique as compared to established models. There is no randomness test suit implemented to check the randomness of the pseudorandom binary numbers generated.

The PRNG, which is based on the Piece-wise Logistic Map, an improved logistic map, is suggested in a research study [15]. Some specific operations, such as substitution operation and feedback operation, are used in their method to destroy the relationship between generated chaotic sequences and pseudo-random numbers. Both theory analysis and simulation experiments show that the suggested PRNG is easy to use, safe, and effective. In this paper, there are 18 numbers of basic operations like addition,subtraction,multiplication,division, xor operations performed to generate an 8-bit number, that is also huge for resource constrained environment.

In this study [16], a novel pseudo-random number generator called a Tinkerbell map is proposed. The widely used the statistical tests like NIST, DIEHARD, and ENT are done to assess the proposed approach. The analysis's findings indicate that the novel derivative bit stream approach is ideal for integration into crucial cryptographic applications. But there is no compartive research study found in this article and also no specific application areas are mentioned.

In an research article [17], an innovative method for generating pseudo-random numbers based on three coordinates of the chaotic Chen maps. Among other cryptography applications, the suggested approach may be used to produce cryptographic keys for digital photographs. Furthermore, the probability distribution that is non-uniform in nature of sequences directly produced by the Chen chaotic system is resolved by our pseudo-random number generator. The results of numerous statistical tests, including the famous Diehard battery test and the NIST test, demonstrate the strong statistical properties of the sequences produced by our suggested system. The security analysis's testing findings confirm the suggested method's strong capacity to fend off diverse assaults. There are no security and statistical tests like key sensitivity analysis. key space analysis. histogram analysis, correlation analysis found in this manuscript.

This research proposes a novel PRNG model for random sequences with huge key spaces in IoT devices [4]. Adaptive control parameter generalized symmetric maps are described in this study. Any symmetric chaotic map may be chosen by the user as long as the output is a continuous stream of independent and random sequences. The effectiveness of the suggested strategy is demonstrated by experimental findings in terms of entropy, key space, bifurcation diagrams, and NIST randomness test. There is no computational complexity analysis in terms of time and space is found in the manuscript,which is very important metric for effective algorithm design.

The very close relationship between chaos maps and pseudo random numbers has been observed in different research attempts preserving good randomness and cryptographically security property.

4 Algorithm and flowchart for ICSPRNG

Algorithm 1 ICSPRNG.

input : num_points=32768, $iter\_n$=10, u=0.78, r=256,rnd_rng=65536,$k = 265$
output: "npts" as pseudo random numbers.

Initialize a list totpt as $totpt = []$
Create a random list of initial 32768 points L using with range -r to r with $random.uniform(-r, r)$.
Loop till $iter\_n$
  Loop cord in length(L)
    Compute L[cord]=ikeda_map(u, L[cord][0], L[cord][1])
  Loop j in range(len(L))
    Compute x.append(L[j][0]) and y.append(L[j][1]).
  End Loop
  End Loop
Compute totpt.append(x) and totpt.append(y).
End Loop
Convert totpt in interger value and save $totpt = totpt \times 10000$
Save pts=$abs(totpt)\%256$
Reshape pts to 256×256
Return pts
Compute ($random.sample(range(0,$rnd_rng),rnd_rng)) and save it to a variable rnum
Reshape the variable rnum using reshape(rnum,(256,256))
Compute $rnum\%k$ and save it to rnum
Compute bitwise XOR between pts and rnum and save it to variable "npts".
Write the "npts" as pseudo random numbers.
Stop.

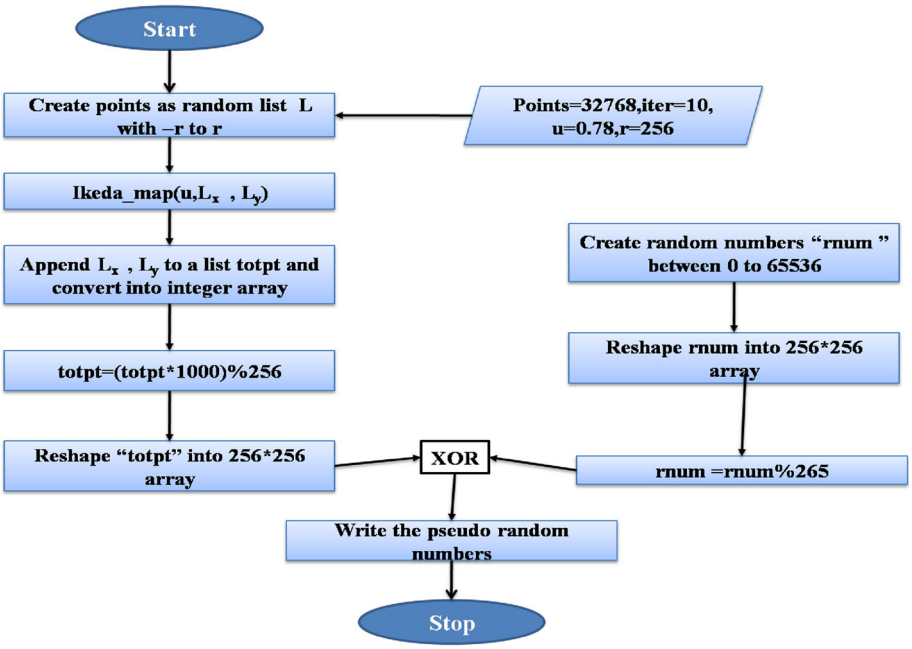

Fig. 3 Flowchart of our algorithm (ICSPRNG)

In the above Fig. 3, the flowchart of our proposed ICSPRNG is presented with detail. Our proposed algorithm runs on three nested loops in terms of iterations. Among them two nested loops iterates over the two same input values L, so it can be concluded the complexity is O(L^2). In other words, it is O(n^2). The outermost loop iterates over another input value $iter\_n$ and iterates over it. So, the overall complexity is O(n^3). Our proposed method takes 0.016% of the total RAM's capacity to create the pseudo random number sequence. As a result, our proposed method can be taken for implementation in a resource-constrained environment which requires very low memory.

5 NIST test for ICSPRNG

The NIST test which assesses the property of randomness is included in this section. The python code for NIST has already in the git-hub collection (https://github.com/dj-on-github/sp800_22_tests/) created by other researcher and taken for our experiment for test the randomness. In the evaluation, if the P_value of the sequence is higher than a certain threshold that is 0.01 (the significance level), therefore the sequences pass the test [9]. From the table given below, the values are greater than or equal to P_value which is 0.01 specified in the NIST test. So, our pseudo random number generator, namely ICSPRNG has proved its randomness qualifying all the majority randomness test specified by the NIST test. The report is shown in the Table 1.

Table 1 Test of Randomness property using NIST tests

SL No	Name of the test	P_value	Status
1	approximate_entropy_test	0.305607164584	Success
2	binary_matrix_rank_test	0.05263527207073906	Success
3	cumulative_sums_test	0.1499656167348773, 0.02768114762569507	Success
4	dft_test	0.8128036624387162	Success
5	frequency_within_block_test	0.608862845443169	Success
6	longest_run_ones_in_a_block_test	0.3541946648486878	Success
7	maurers_universal_test	0.850183398188	Success
8	monobit_test	0.1355587617140609	Success
9	overlapping_template_matching_test	0.9999999637133592	Success
10	random_excursion_test	0.6752956091519277	Success
11	random_excursion_variant_test	0.5354680469287809	Success
12	runs_test	0.38331956754713403	Success
13	serial_test	0.5608663276137132	Success

6 Diehard battery test for ICSPRNG

This statistical test consists of different metrics like birthday spacings metric to craps metric. The diehard test requires the range of P-values must spread over the value range [0, 1) [17]. In the Table 2, the detailed result is given, where our ICSPRNG successfully passed all the test with desired value.

7 Implementation example and security analysis of ICSPRNG

In this paper, the source code of the algorithm is built in Python programming with a processor manufactured by Intel and version i3. The capacity of the RAM is 8.00 GB with a 250GB SSD. The statistical analysis part has been done with Matlab 2016b software.In our experiment, with the help of $u = 0.78$,$num\_points = 32768$, r=256 and $random.uniform(-r, r)$, we create a random list of 32768 initial points as L . In our paper, 2D ikeda map is utilized to generate pseudo random number along with a unique random number sequence. The simple bitwise xor operation is performed the ikeda assisted PRN sequence. We have generated 65536 random numbers with 8 bit each. Total 524288 bits are taken for randomness and statistical test. Not only ,our pseudo random numbers passed the majority of NIST randomness test given in Table 1, but also successfully qualified in all the test specified in Diehard test. The result is given in Table 2. We have converted the 65536 pseudo random numbers into a 256×256 images known as random image. The use of reseeding is very important for PRNG. In our proposed method, after the iteration number that is initiated as 10, we use a random number generator that will help us for reseeding our ICSPRNG generating 0 to 65536 random numbers. So, after every 10th iteration our ICSPRNG will drastically change leading to huge unpredictability.

Table 2 Test of Randomness property using Diehard Test

Test Name	P_value	Status
Birthday spacings	0.45657969	Passed
Overlapping 5-permutation	0.63890414	Passed
Binary rank (32 x 32)	0.04091810	Passed
Binary rank (6 x 8)	0.84245103	Passed
Bitstream	0.22166780	Passed
OPSO	0.03165918	Passed
OQSO	0.03165918	Passed
DNA	0.77194404	Passed
Stream count-the-ones	0.57924121	Passed
Byte count-the-ones	0.11820626	Passed
Parking lot	0.99259700	Passed
Minimum distance	0.55033897	Passed
3D spheres	0.54898465	Passed
Squeeze	0.14701771	Passed
Overlapping sums	0.15301408	Passed
Runs up	0.68233454	Passed
Runs down	0.46924314	Passed
Craps	0.80449343	Passed

With x and y coordinates of points L and display only 10 points in the above Fig. 4, left image. Then, after 10 iterations, our algorithm will produce the 65536 random numbers taking the x and y coordinates of ikdea chaotic map, displayed in the right segment of Fig. 4.

Next, in the algorithm in Section 4, the values of $x\_coordinates$ as $L[cord][0]$ and $y\_coordiates$ as $L[cord][1]$of random "num_points" generated in the range $random$.

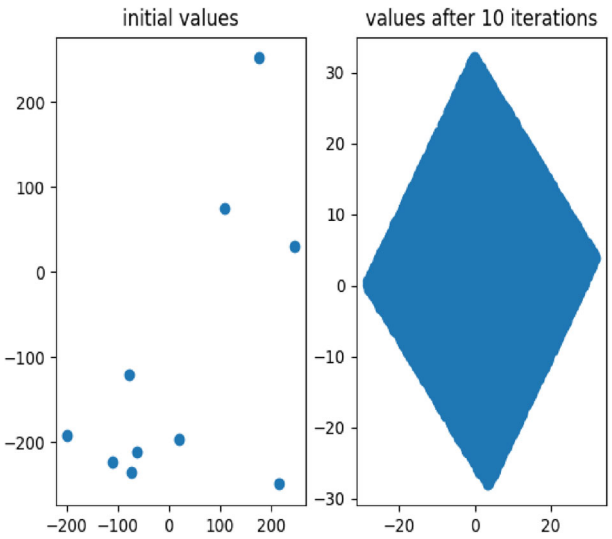

Fig. 4 Representation of initial 10 values and values after 10 iteration

$uniform$(-256, 256) fed into the function "$ikeda\_map(u, L[cord][0], L[cord][1])$" with $u = 0.78$ and iterates over 10 times to generate the modified coordinates as $L[j][0]$ and $L[j][1]$ respectively. Then the plot of modified coordinates $L[j][0]$ and $L[j][1]$ is given in the diagram varying $num\_points = 32768$ is generated as in Fig. 5.

Then, values of $L[j][0]$ and $L[j][1]$ are appended one after another to form a new list known as $totpt$. Then perform the operation $totpt = totpt \times 10000$ and $abs(totpt)\%256$ respectively saving pts as integer array. Next, the values pts is reshaped to 256×256 array. At last, the set of another generated 65536 random numbers are reshaped to 256×256 array known as rnum. After that perform $rnum\%k$, where $k = 265$. Then, we have performed bitwise xor between pts and rnum, and the Fig. 6 is totally random in nature. For most of the security analysis, we arranged the 65536 data values into 256×256 image.

The simple ikeda() can be implemented in any programming language in any platform. Also, there is no restriction of the used of the proposed algorithm with respect to software and hardware specifications using simple addition, multiplication and xor operations

7.1 Key sensitivity test

The key sensitivity metric tells us that a little bit of modification in the key value can generate two completely disjoint sets of pseudo-random numbers. In other words, it represents that only one bit change in the value of the key, will result into two different random number sets. We arrange the 65536 data values into a 256×256 image. In the Fig. 7, there are two images generated with parameter values u = 0.78 and u = 0.79, respectively.

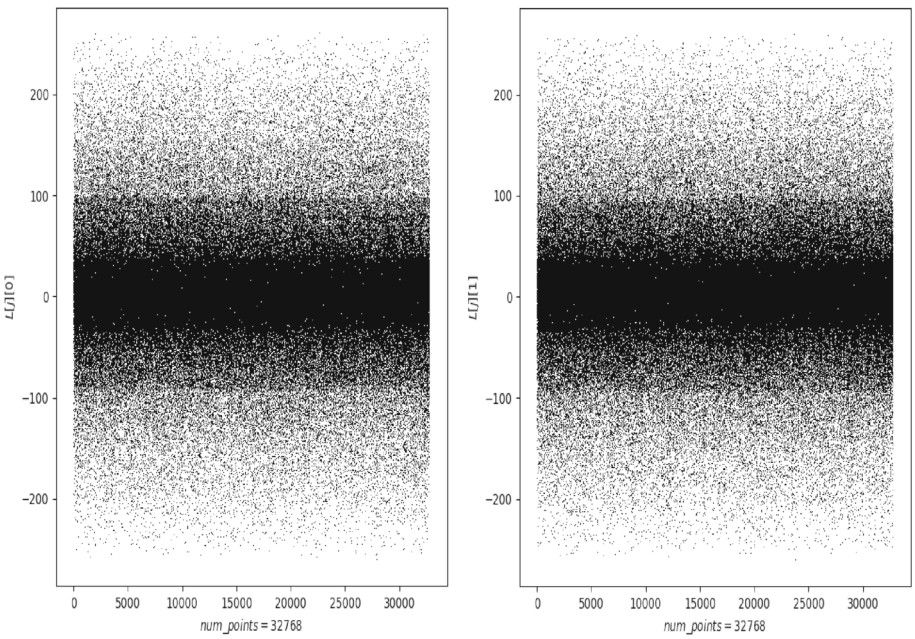

Fig. 5 Representation $L[j][0]$ and $L[j][1]$ with respect to num_points

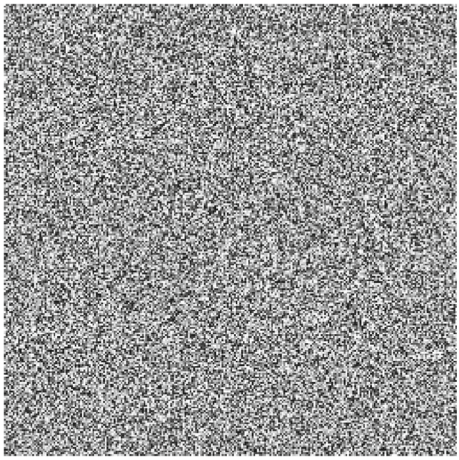

Fig. 6 Representation of ICSPRNG

From the below Fig. 7 image, it can be stated that if only a bit is changed in the parameter "u" that leads to a completely different set of pseudo-random numbers represented as two images in top-left and top-right. The difference image calculated from the images top-left and top-right is given below in for the justification of the fact found.

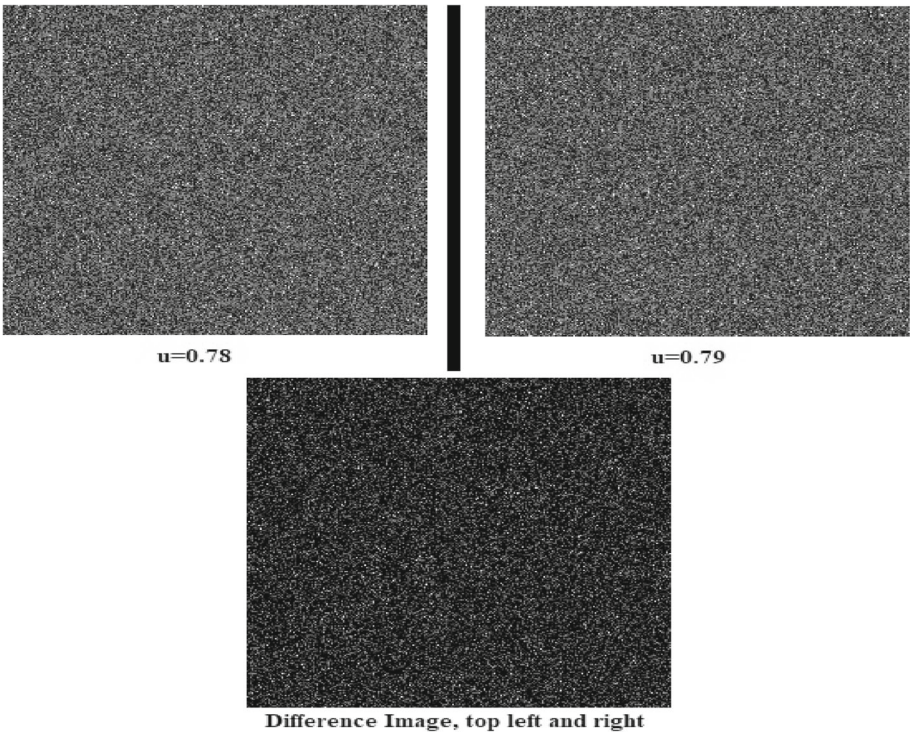

Fig. 7 Representation of Key sensitivity

7.2 Key space test

The analysis of key space reports the number of changing variables used for the experiment. Various types of attacks done by the trail-and-error mechanism can be prevented with the help of the large value indicated by this statistical parameter. The documented IEEE floating-point value is considered in terms of the accuracy of double variables as 10^{-15} with a bit capacity of 64. We have six double variables as points: num_iter, u,r, t,rnd_rng,k. As a result, the value of keys pace is approximately 10^{90}, or $2^{298.97353}$. We have also compared the different key space values with existing research studies in Table 3. So, our pseudo random number sequence with large key space value can be used in any system, protecting it from any type of attack performed using the trial-and-error method ("brute force method").

7.3 Histogram Plot

Histogram depicts the spread of values in a pictorial representation. If the spread of values is uniform throughout the plot, it is very challenging for the hacker to guess the values. The histogram plot should be as evenly distributed as is humanly feasible, meaning that the likelihood of any value existing is the same [21]. We have presented the histogram so that we can understand the spread of the values. From subfigures of the Fig. 8, no one can guess the values because the images are identical. If the histogram plot is non uniform in nature, one can guess the particular region where some of the values occurrs more number of times. Then it is easier to analyze the portion of values for further processing to know the original values of the pseudo random numbers. From the plot of uniform histograms in Fig. 8, with two different values of u in the ikeda map, it can be concluded that the statistical correlation is not prevalent between data values, giving it the properties of being resistant to differential and statistical threats [22].

7.4 Entropy test

The phrase "Entropy" was first coined by Shannon, a famous mathematician, as a metric of uncertainty. In the field of information processing, it has been widely utilized [23]. This shows how much information it has [24]. The entropy equation [25] can be written as .

$$H(P) = \sum_{i=0}^{255} [Prob(X_i) \times \log(\frac{1}{Prob(X_i)})] \tag{1}$$

Table 3 Keys pace analysis

Method	Value of keyspace	No of control parameters
our method	$2^{298.97353}$	6
[9]	2^{208}	4
[13]	2^{240}	11
[14]	2^{372}	8
[15]	2^{162}	4
[16]	2^{183}	6
[17]	2^{279}	6
[4]	2^{372}	7

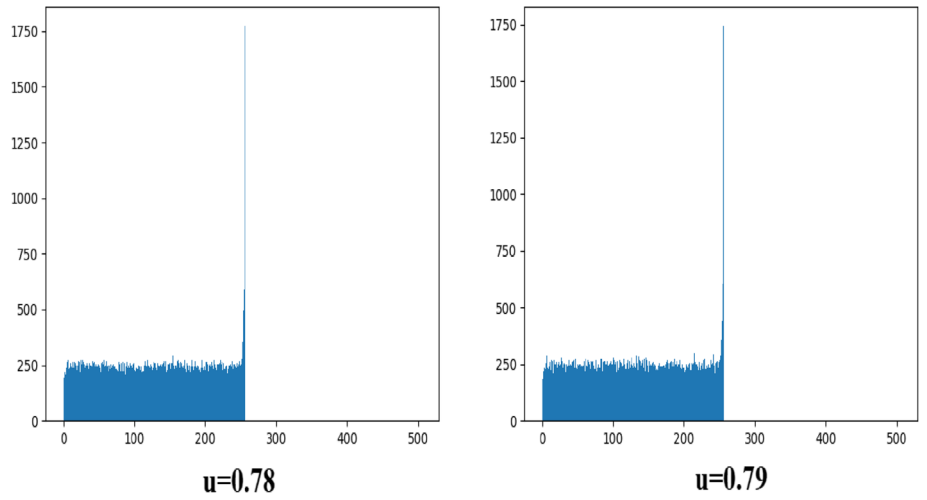

Fig. 8 Representation of Histogram

The analysis of entropy with existing methods is given in the Table 4 below.

7.5 Correlation test

Autocorrelation is used to describe the correlation between values at different times in a sequence [26]. The representation of the autocorrelation of the two random number sequences generated by ICSPRNG with u = 0.78 and u = 0.79 is displayed in the Fig. 9. From the Fig. 9, it can be concluded that our ICSPRN sequences have autocorrelation values nearer to zero, and that satisfies no relation between the values in the random sequences. Also to test the effect of the statistical analysis attack,the correlation value is important. For the minimized effect, the value must be very low and may be close to zero.The computation of the correlation coefficient value is defined using the equation 2.

$$\rho(seq\_a, seq\_b) = \frac{cov(seq\_a, seq\_b)}{\sigma_{seq\_a}\sigma_{seq\_b}} \tag{2}$$

Table 4 Study of Entropy values

Methods	Entropy value
Ref. proposed method	7.2010
Ref. [9]	7.9994779223
Ref. [13]	7.999979
Ref. [14]	7.9998
Ref. [15]	NA
Ref. [16]	7.999999
Ref. [17]	NA
Ref. [4]	NA

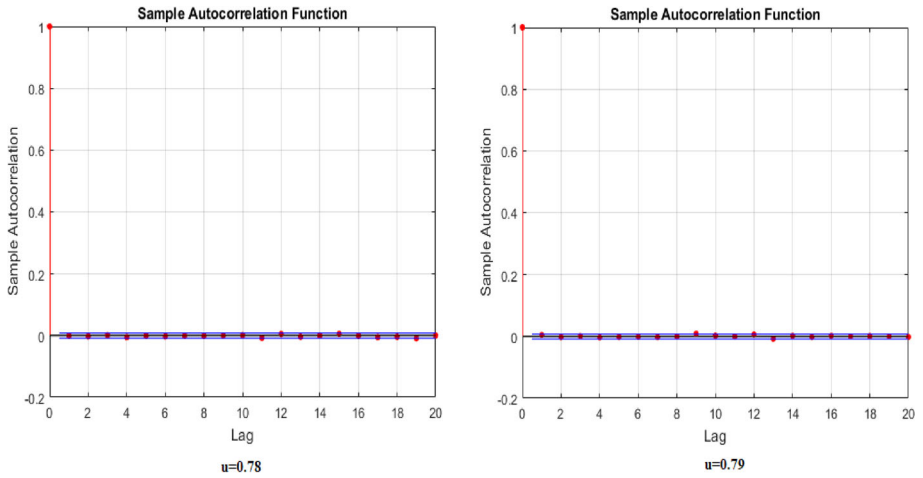

Fig. 9 Auto correlation of two sequences

In the equation (2) , the $cov(seq\_a, seq\_b)$ indicates the covariance value between two sequences seq_a and seq_b, σ_{seq_a} and σ_{seq_b} are the values S.D.(known as "standard deviation") of the sequences with parameters of ikeda map u=0.78 and u=0.79 respectively. The seq_a,seq_b are represented as two images files. The value of the correlation coefficients of seq_a,seq_b is -0.0003. In our paper, based on three directions (horizontal, diagonal, and vertical), the calculation of values of correlation is done and tabulated in Table 5.

As per the obtained value from the Table 5, it can be submitted that our proposed algorithm generates lower correlation values (negative value also) than other existing methods horizontally, vertically and diagonally. So, our pseudo random number generation method can protect any statistical intrusion.

7.6 Generation time analysis

The time taken to generate the pseudo random numbers is essential for their use in a variety of applications, including cryptographic methods such as encryption, decryption, substitution, permutation, hashing, seed vector creation, and initialization vector formation. The comparison of generating time in seconds per Mb is shown in Table 6. It may be said that our ICSPRNG requires significantly less time than other pseudo random number generators. Though our CSRNG passed the randomness tests, security analysis and statistical tests with little generation time, the problem of exchanging initial parameter values as secret key can be there. This problem can be solved employing a good key exchange algorithm. In future

Table 5 Study of Correlation values

Methods	Ref. [14]	Ref. [13]	[9]	[16]	[15]	[17]	Proposed Method
Horizontal	-9.1×10^{-5}	1.98×10^{4}	−0.00290	−0.0014	0.026	NA	−0.0006
Vertical	0.000343	NA	NA	NA	NA	NA	0.0026
Diagonal	NA	NA	NA	NA	NA	NA	0.0030

Table 6 Study of Generation time

Methods	Generation Time (Sec)
Ref. proposed method	0.07456Mbit/S
Ref. [14]	NA
Ref. [15]	20.53 Mbit/S
Ref. [16]	0.4901 Mbit/S
Ref. [17]	NA
Ref. [4]	1.1350(1000 iterations)

we will incorporate one key exchange algorithm with our newly proposed CSPRNG to make it more robust and secure.

8 Discussion and future scope

In many study endeavours, a very good association between chaos maps and pseudo-random numbers has been found while maintaining good randomness and cryptographic security. So we have chosen a simple chaotic map, the Ikeda map, to build a cryptographically secure pseudorandom number generator. With a very small number of mathematical operations, we have succeeded in producing 65536 pseudo-random numbers. Also, randomness tests, various security tests, and statistical tests have been conducted to verify the effectiveness of our proposed algorithm. As a usability feature of our CSPRNG, it can be used to encrypt multimedia data. It can be used to generate secret keys for different applications. The initialization vectors used in cryptographic schemes can be generated from our CSPRNG. It may be employed in the replacement and permutation stages of cryptosystems.

9 Conclusion

In this research work, the design of an Ikeda-assisted Cryptographically Secure Pseudo Random Number Generator(ICSPRNG) is introduced. The chaos map Ikeda uses the concept of random numbers to generate the sequence. Considering all the security analysis of our algorithm, it has been found that our pseudo random number sequence has high entropy values, a uniform histogram plot, and very low correlation values. It also guarantees huge randomness because it has passed the majority of NIST test suites. The generation time is very low with very little computation overhead, so it can be applied in resource-restricted environments. Our pseudo random number sequence can be used in encryption of multimedia data, secret key generation, one-time password generation, and initialization vector generation. Also, our pseudo random numbers can be used in substitution and permutation phases in a cryptosystem. In future we will include one key exchange algorithm with our newly proposed CSPRNG to make it more robust and secure solving the problem of key exchange.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest

References

1. Elmanfaloty RA, Abou-Bakr E (2019) Random property enhancement of a 1D chaotic PRNG with finite precision implementation. Chaos Solit Fractals 118:134–144
2. Ryan C, Kshirsagar M, Vaidya G, Cunningham A, Sivaraman R (2022) Design of a cryptographically secure pseudo random number generator with grammatical evolution. Sci Rep 12(1):8602
3. Naik RB, Singh U (2022) A review on applications of chaotic maps in pseudo-random number generators and encryption. Annals Data Sci 1–26
4. Zia U, McCartney M, Scotney B, Martinez J, Sajjad A (2022) A novel pseudo-random number generator for IoT based on a coupled map lattice system using the generalised symmetric map. SN Appl Sci 4:1–17
5. Mukherjee A, Mallick PK, Mishra D (2022) Chaotic Pseudo Random Number Generator (cPRNG) Using One-Dimensional Logistic Map. In Biologically Inspired Techniques in Many Criteria Decision Making: Proceedings of BITMDM 2021 pp 697-708. Singapore: Springer Nature Singapore
6. Liao TL, Wan PY, Yan JJ (2022) Design and synchronization of chaos-based true random number generators and its FPGA implementation. IEEE Access 10:8279–8286
7. Sharma M, Ranjan RK, Bharti V (2022) A pseudo-random bit generator based on chaotic maps enhanced with a bit-XOR operation. J Inf Secur Appl 69:103299
8. Zia U, McCartney M, Scotney B, Martinez J, Sajjad A (2023) A resource efficient pseudo random number generator based on sawtooth maps for Internet of Things. Security and Privacy e304
9. Zhao W, Chang Z, Ma C, Shen Z (2023) A Pseudorandom Number Generator Based on the Chaotic Map and Quantum Random Walks. Entropy 25(1):166
10. Abderrahim NW, Benmansour FZ, Seddiki O (2023) FPGA Implementation of a Chaotic Pseudo-random Numbers Generator. SN Comput Sci 4(4):410
11. Cun Q, Tong X, Wang Z, Zhang M (2023) A new chaotic image encryption algorithm based on dynamic DNA coding and RNA computing. Vis Comput 1–20
12. Murillo-Escobar D, Murillo-Escobar MÁ, Cruz-Hernández C, Arellano-Delgado A, López-Gutiérrez RM (2023) Pseudorandom number generator based on novel 2D Hénon-Sine hyperchaotic map with microcontroller implementation. Nonlinear Dyn 111(7):6773–6789
13. 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
14. 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
15. 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
16. Stoyanov B, Kordov K (2015) Novel secure pseudo-random number generation scheme based on two tinkerbell maps. Adv Stud Theor Phys 9(9):411–421. https://doi.org/10.12988/astp.2015.5342
17. 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
18. Laskaridis L, Volos C, Munoz-Pacheco J, Stouboulos I (2023) Study of the dynamical behavior of an Ikeda-based map with a discrete memristor. Integration 89:168–177
19. Dinu A, Frunzete M (2023) Singularity, Observability and Statistical Independence in the Context of Chaotic Systems. Mathematics 11(2):305
20. Sriram V, Kearney D (2009) An FPGA implementation of a parallelized MT19937 uniform random number generator. EURASIP J Embed Syst 2009:1–6
21. Noor NS, Hammood DA, Al-Naji A, Chahl J (2022) A Fast Text-to-Image Encryption-Decryption Algorithm for Secure Network Communication. Computers 11(3):39
22. Naskar PK, Bhattacharyya S, Nandy D, Chaudhuri A (2020) A robust image encryption scheme using chaotic tent map and cellular automata. Nonlinear Dyn 100(3):2877–2898. https://doi.org/10.1007/s11071-020-05625-3
23. Xu W, Pan Y, Chen X, Ding W, Qian Y (2022) A Novel Dynamic Fusion Approach Using Information Entropy for Interval-Valued Ordered Datasets. IEEE Trans Big Data
24. Khurana A, Bhatnagar V (2022) Investigating entropy for extractive document summarization. Expert Syst Appl 187:115820
25. Lin H, Wang C, Cui L, Sun Y, Zhang X, Yao W (2022) Hyperchaotic memristive ring neural network and application in medical image encryption. Nonlinear Dyn 110(1):841–855

26. Liu J, Liang Z, Luo Y, Cao L, Zhang S, Wang Y, Yang S (2020) A hardware pseudo-random number generator using stochastic computing and logistic map. Micromachines 12(1):31