Received: April 11, 2022. Revised: May 25, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

New Pseudo-Random Key Generator for IoT-security Model Based on a Novel 3D Coupled Map Lattice

Mohammed Sh. Oudah[1]* **Abeer Tariq Maolood[1]**

[1]*Computer Science Department, University of Technology, Baghdad, Iraq*
* Corresponding author's Email: cs.20.42@grad.uotechnology.edu.iq

Abstract: In cryptography fields, one concern is the consideration of the capacity of the resource, especially, in environments that run IoT devices. This paper presents a lightweight and comprehensive security system, which involves a novel Pseudo-Random Number Generator (PRNG) based on a novel 3D Coupled Map Lattice system (3D-CML) as a chaotic system that is more practical in cryptography, and encryption algorithms depend on two security levels, firstly, permuting the plain image using a standard 2D Henon chaotic map, then, encrypt it by a One-Time Symmetric Key (OTSK). The bifurcation diagram of the 3D-CML shows that the chaotic parameters have been extended in their ranges, this feature has a positive effect on the key-space size which its size equal to 2^373. From another aspect, with fewer iteration configurations, all random sequences that are produced by the proposed PRNG have passed all statistical tests of the NIST suite, in turn, this configuration would lower the run-time, hence decreasing the computational effort as a response to the requirements of limited-resources environments, such as IoT. Several assessment metrics, such as Mutual Information, Gray Difference Degree, Histogram, Chi-square, Correlation Coefficient, Entropy, and more, confirmed that the proposed algorithms can be robust in dissolving the internal characteristics in the original image for producing an encryption image, resulting in strong resistance against any cyberattacks.

Keywords: Pseudo-random key generation, Coupled map lattice, 3D logistic map, Henon chaotic map, IoT.

1. Introduction

IoT devices become commonly used in different places, from household things to complicated industrial things, these devices help users to exchange critical and real-time information, such as pictures or videos of surveillance and security cameras, weather information, control signals, sensors data, and otherwise [1]. Many cyber-threats are developed to attack IoT information that is transited through Internet; therefore, the efforts of developing protection roles and security mechanisms still have high priority. The Cryptography field is one of these techniques that get special attention, this field is based on four security bases: Data integrity, Confidentiality, Authentication, and Non-repudiation of data resources [2]. In the IoT context, One challenge in such environments is how to establish a lightweight security system that restricts and validates the user's access, to prevent illegitimate access from being capable of accessing, reading, modifying, or destroying information, a lightweight mechanism is required to considerateness the limited resources in IoT's devices [3].

With the easiest implementation, the term "chaotic cryptography" appears as a comprehensive term, comprising all methods that take advantage of dynamic systems exhibiting chaotic behaviour. The major advantages that make the chaotic systems considered suitable to deploy into cryptographic techniques are [4]:

1. The sensitivity to the initial condition, in which no different seeds have the same system's products.
2. A chaotic system is dynamically instability and has unpredictability behavioural.

Unfortunately, the usage of chaotic maps in cryptographic is not straightforward but should study

the chaotic behaviour of maps, to doing some improvements to make it as complex as possible, the improvements are centered on the development of the range of the initial conditions and control parameters, in which these parameters reflect as same as the notion of diffusion principle in cryptographic. Therefore, the range extension would make the system more practice and robust against cyberattacks.

Several works have proposed different chaos-based methods that are used as permutation and pseudo-random key generator models, this came for several reasons, some of them are related to the chaotic-based cryptography's motivations [5], that what explained above. The important issue related to IoT environments, i.e., the chaotic-based encryption methods are considered faster and more fixable to applied in such environments, for instance, the user can specify the number of iterations of the chaotic maps to make the overall running-time is lighter, also, the complex of chaotic map's parameters and initial conditions plays the main role in produce complex and pseudo-randomness results [6].

A pseudo-random number generator PRNG has a pivotal role in any cryptosystem, and the efficiency of the generator is directly proportional to the degree of its randomness. Primarily, a random number generator can be true randomly or pseudo-randomly [7], the former as a result of physical phenomena such as noise or electromagnetic sources, whereas, the latter is a result of deterministic equations that exhibit as same as true random number characteristics with specified initial conditions. There is an interest in how to establish a lightweight and efficient PRNG that can run in limited-resource environments. In this paper, three dimensions Coupled Map Lattice (3D-CML) system was proposed, the proposed PRNG depend on CML that is a kind of spatiotemporal chaotic system, it was developed to overcome the limits and gaps of chaotic map[5, 8], typically, CML used the logistic map as a local chaotic map, in the proposed algorithm, a 3D Generalized Logistic chaotic Map (3D-GLM), that proposed in [9], was used as the local map in CML. The key notion of the proposed PRNG is fewer computational iterations are desired, to achieve an acceptable randomness degree.

Almost, the permutation level provides extended randomness in a cryptosystem. HM is a chaotic nonlinear system, that is frequently used by researchers as a permutation model [13], also, another kind of chaotic system is used for such purpose, is called Arnold cat map, however, several drawbacks should be evaluated by researchers because Arnold Cat map produces a little randomness in a scrambled image and requires a higher iteration to produce an acceptable result, so these cases lead to exploit more resources. To clarify this issue, this paper included a comparison between HM and Arnold Cat map to emphasize that the HM can give an effective scrambled image by applying fewer steps, and it is quickest in recovering the original image, this feature goes with the rest methods that are used in the proposed algorithm, to build a secure and lightweight system.

2. Related work

Several recent research works are leverage chaotic system capabilities to develop the encryption algorithm, especially, the PRNG module, for example, in [10] the encryption model mixes three chaotic systems, logistic, sine, and tent maps into one system CML, the proposed chaotic system was used as a key generator of 256 bits length, furthermore, the encryption model involved a bit-level z-scan scrambler module, this scheme might rein the performance. In [11] the key generator of the encryption model has been implemented using 1D logistic and sine maps and a combination of them. For use in distributed systems, however, this study didn't consider the improvement of the chaotic parameter's range. The same is true for [12] where the proposed encryption algorithm was incorporated CML to perform a low overhead permutation and diffusion encryption, without the parameter's range expanded. Almost, the strength of the encryption algorithm comes from the power of the key generation, so the [13] focused on the improvement of the PRNG of the encryption algorithm when XOR function as an aggregator for three four-wing memristive hyperchaotic systems. The proposed method in [14] generates initial conditions using 3D logistic maps, thus, these conditions are used to scramble the image, then the encryption key is generated by another 3D logistic map, and the XORing operation is done after converting the image into DNA representation. A combination of quasigroups and the chaotic standard map has been proposed in [15] to overcome the shortage of key-space, the algorithm achieved both substitution procedures by quasigroup of order 256, and the permutation procedure is done by the standard chaotic map.

In this paper there is a substantial point, the proposed algorithm is needed to meet the limitations of IoT platforms that are have limited processor power and storage capacity, and such devices have multiple jobs that might run simultaneously together, several studies seek to implement the chaotic-based (PRNG) on general-purpose hardware, such as Field-Programmable Gate Array (FPGA), for instance, in [16] a lower-power lemniscate chaotic map is

proposed to be practical on FPGA, the proposed model has reduced 48.3% of resources using. A modified logistic map was suggested to increase the randomness of PRNG that was implemented in Virtex 7 FPGA [17], thus a 0.989 passing rate of NIST was achieved. In [18], the output of the discrete-time zigzag map is formed in a 32bit chaotic orbit and then transformed to produce a bit-level random number. Unsub Zia et al. [6] has been proposed generalized symmetric maps with a user-chosen chaotic map by changing the adaptive control parameter used to generate a pseudo-random key, this model was tested on raspberry pi 3b+ and raspberry pi zero.

The Contributions of this study can be summarized in the following:

1. Analyse and improve the chaotic range of the initial condition and parameters of the original chaotic system that was used.
2. Proposed a new PRNG using the 3D-GLM as a local map in CML to produce a novel chaotic system called 3D-CML. The new PRNG consider more complex than the original system, and its parameters have a larger chaotic range.
3. The proposed method considers the resources-limitations in IoT devices.
4. Depending on the proposed PRNG, the construction of a multi-level security cryptography model that can be secure as possible and at the same time can be considered a lightweight model.

The last point is the main concern in this study, in addition to the chaotic system development.

The remainder paper's organization is Section 3 presents the general concepts of the methods that have been used in the proposed model. Section 4 gives a view of the security levels design. The security analysis and performance efficiency were discussed in Section 5. Section 6 shows the conclusions.

3. Theoretical background

The proposed involved multiple security levels, 3D-CML with 3D-GLM as a local map, this chaotic map is the core of 3D-PRNG to produce a one-time Symmetric Key $OTSK$. After, a Plain Image $PI_{(N\times N)}$ scrambled by using standard Henon-map (HM), the Scrambled Image $SI_{(N\times N)}$ XORed with $OTSK_{(N\times N)}$ to produce the Encrypted Image $EI_{(N\times N)}$. The Details are in subsequent sections.

Two kinds of chaotic maps are used, following details about them:

3.1 Coupled map lattice (CML):

In 1992 Kaneko [8] Proposed a novel spatiotemporal chaotic nonlinear map with more chaotic parameters and complex behaviour, to become more suitable to apply in different fields, one of them is the cryptographic field. CML equation describes as follows:

$$x_n = (1-\varepsilon)f\big(x_{n-1}(i)\big) + \left(\frac{\varepsilon}{2}\right)f\big(x_{n-1}(i-1)\big) + f\big(x_{n-1}(i+1)\big) \quad (1)$$

where ε: is a coupled map lattice coefficient, that ranges between$[0-1], i \in [1, L]$ is a lattice within a specific system's size. If $i = 1$, then $(i-1) = L$, if $i = L$, then $(i+1) = 1$,n is the discrete-time step, L is the number of lattices.

The standard definition of the $f(x_n)$ is a one-dimensional local map of a chaotic regime and could be any chosen chaotic map function, typically, a logistic map is used:

$$x_n = \mu x_{n-1}(1-x_{n-1}) \quad (2)$$

if $\mu > 3{,}57$, then $f(x_n)$ in chaos.

3.2 3D generalized logistic map (3D-GLM):

An extended version of the 1D logistic map with complex chaotic behaviour and more initial and control parameters could be a better choice to make it at the core of PRNG, this map was presented in [14, 19], described as follows:

$$\begin{cases} x_{n+1} = Ux_n(1-x_n) + By_n^2x_n + Az_n^3 \\ y_{n+1} = Uy_n(1-y_n) + Bz_n^2y_n + Ax_n^3 \\ z_{n+1} = Uz_n(1-z_n) + Bx_n^2z_n + Ay_n^3 \end{cases} \quad (3)$$

This map in chaotic status whenever:$U \in [3.53, 3.81]$, $B \in [0, 0.022]$, $and\ A \in [0, 0.015]$, And the initial states $x, y, z \in (0,1)$.Figure1 shows the Bifur cation of Eq. (3) at $U = 3.6324, B = 0.0193, A = 0.0146, and\ (x, y, z) = (0.905, 0.035, 0.304)$, in respe ctively.

In the proposed algorithm, the Eq. (3) was used as a local map in Eq. (1), to produce a 3D-CML that can be given many chaotic parameters with a wide chaotic range, that can be guaranteed more randomness for PRNG, and enlarge the space of encryption key to become more resistant to a brute-force attack. Fig. 2 shows the bifurcation of the 3D-CML with 10 lattices, the range of ε coefficient and the initial condition x, y, z are extended from [0,1] in original CML, up to (0,2), in the new proposed map

(3D-CML), result in increasing the key-space size, hence, more complex and randomisation in PRNG.

3.3 Henon map (HM):

Another frequently-used chaotic map is a nonlinear 2D dimensional map [20, 21], described in the following equation:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n (mod\ N) \\ y_{n+1} = b\ x_n + c\ (mod\ N) \end{cases} \quad (4)$$

Where $a = 1.4$ and $b = 0.3$, and C is constant. Coordinates of $PI_{(N\times N)}$ are $x_n, and\ y_n \in \{0,1,2, \dots., N-1\}$.

Eq. (4) is exactly invertible at the receiver side with the same iterations and control parameters. The inverse of Eq. (4) is:

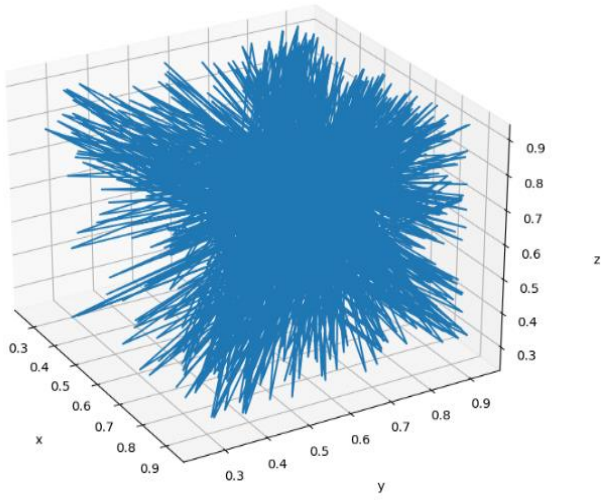


Figure. 1 Bifurcation of 3D-GLM

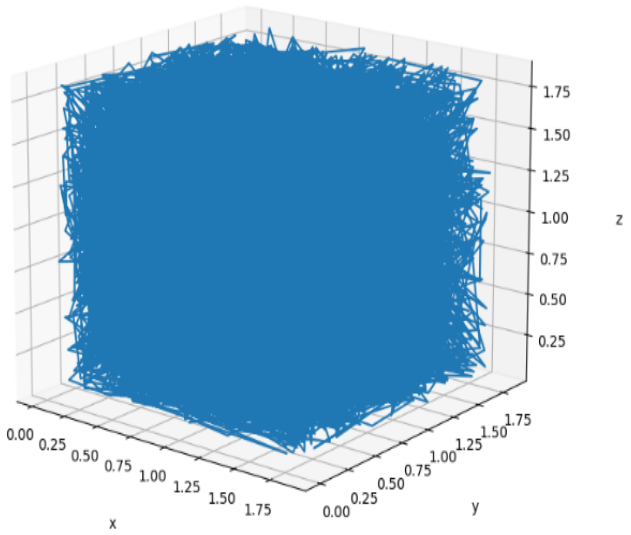


Figure. 2 Bifurcation of 3D-GLM with 10 lattices

$$\begin{cases} x_n = \frac{1}{b}(y_{n+1} - c)\ (mod\ N) \\ y_n = x_{n+1} - 1 + ax_n^2\ (mod\ N) \end{cases} \quad (5)$$

4. Proposed system design

4.1 Pseudo-random number generator using the proposed 3D-CML:

In 3D-PRNG, the session key (SK) can be used as an initial condition for the 3D-CML. whenever the SK has satisfied the perfect forward secrecy protocol, then, the generator can give a unique One-Time Symmetric Key OTSK at a given session.

The algorithm1 explains the main steps of PRNG based on 3D-CML, the output is an 8-bit random sequence, and the initial conditions of the map are initialized by SK's parts $(sk_i, sk_{i+1}, sk_{i+2})$, $x_0 = sk_i$, $y_0 = sk_{i+1}$, $z_0 = sk_{i+2}$.

After generating 3D-L lattices using Eq. (3), feed the lattices into 3D-CML for t iterations.

4.2 Scramble image algorithm:

At this security level, the Plain Image's $PI_{(N\times N)}$ pixels are permuting under multiple levels of a chaotic scrambler, the original coordinates of pixels are fed to the 2D Henon Map system to build a Scrambled Image $SI_{(N\times N)}$. This level increases the

Algorithm 1. Generate a triple 8-bit sequence PRN key

Input: L no. of lattices, initial chaotic parameters (ε, U, A, B), T no. of iterations, SK.

STEP1: specify the SK as initial conditions:

$$x_0 = sk_i, \quad y_0 = sk_{i+1}, \quad z_0 = sk_{i+2}$$

STEP2: generate initial Lattice using Eq. (3):

for i = 0 to L:

$$x_{seqi} = Ux_n(1 - x_n) + By_n^2 x_n + Az_n^3$$
$$y_{seqi} = Uy_n(1 - y_n) + Bz_n^2 y_n + Ax_n^3$$
$$z_{seqi} = Uz_n(1 - z_n) + Bx_n^2 z_n + Ay_n^3$$

STEP3: pass n iterations (transient response).

STEP4: for j = 0 to T:

for i = 0 to L: {

$$X_{seqj} = (1-\varepsilon)f(x_{n-1}(i)) + \left(\frac{\varepsilon}{2}\right) f(x_{n-1}(i-1)) + f(x_{n-1}(i+1))$$

$$Y_{seqj} = (1-\varepsilon)f(y_{n-1}(i)) + \left(\frac{\varepsilon}{2}\right) f(y_{n-1}(i-1)) + f(y_{n-1}(i+1))$$

$$Z_{seqj} = (1-\varepsilon)f(z_{n-1}(i)) + \left(\frac{\varepsilon}{2}\right) f(z_{n-1}(i-1)) + f(z_{n-1}(i+1))\}$$

STEP5: concatenate the pair $(X_{seqj}, Y_{seqj}, Z_{seqj})$.

Output: L triple 8-bit key $k_i \in \{k_1, k_2, \dots \dots, k_L\}$.

Algorithm 2. Scramble image

I/P: $PI_{(N\times N)}$, T = number of iterations.
STEP1: convert the $P_{(M\times N)}$ into $P_{(N\times N)}$or $P_{(M\times M)}$.
STEP2: set a buffer $SI_{(N\times N)}$ size.
STEP3: feed each pixel's coordinate into $Eq(4)$:

$$for\ x = 0\ to\ N:$$
$$for\ y = 0\ to\ N:\{$$
$$x_{n+1} = 1 - ax_n^2 + by_n\ (mod\ N)$$
$$y_{n+1} = x_n + c\ (mod\ N)$$
$$SI[x_{n+1}, y_{n+1}] = PI[i,j]\}$$

STEP4: Repeat STEP3 for $T\ times$
Output: $(T - times)$Scrambled Image $SI_{(N\times N)}$.

Algorithm 3. Encryption of scramble image

I/P: Scrambled Image $SI_{(N\times N)}$, $OTSK_{(N\times N)}$
STEP1: create empty $EI_{(N\times N)}$ buffer.
STEP2: for each pixel's coordinate of $SI_{(x,y)}$:

$$for\ x = 0\ to\ N:$$
$$for\ y = 0\ to\ N:$$
$$EI_{(x,y)} = SI_{(x,y)} \oplus OTSK_{(x,y)}, \quad (6)$$

Output: Encrypted image $EI_{(N\times N)}$.

randomness degree and enhances the efficiency of an encryption algorithm. At the same time, the scrambling method is exactly reversible, thus the receiver side will not concern about the information lost. Algorithm 2 explains the steps at this level. The Re-Scramble Image algorithm is the reverse of the Scramble image algorithm using Eq. (5), just when computed y_n the result should be rounded up to the nearest integer using ceil function in python v3.10.

4.3 Encryption algorithm:

This algorithm requires two main inputs, the Scrambled Image $SI_{(N\times N)}$ and One Time symmetric Key $OTSK_{(N\times N)}$ to perform XORing bitwise operation between each $SI's$ pixel with triple 8-bit-$OTSK$, to produce the encrypted pixel of the Encryption Image $EI_{(N\times N)}$, that will be sent to the appropriate destination.

The main steps of the Encryption Algorithm are shown in algorithm 3, the decryption algorithm is the inverse of the encryption using the same $OTSK_{(N\times N)}$ that generated by an IoT device, where the sender and receiver sides exchanged the same initial conditions $(sk_i, sk_{i+1}, sk_{i+2})$.

5. Experimental results and analyses

Several statistical and analysis measurements were used to assess the security performance of the proposed model. In addition to the initial conditions and chaotic parameters of the 3D-CML, there are two things in this context that should be evaluated, the encryption key that is used in encryption and the encryption algorithm itself.

Table 1. Resolutions, execution time of scramble

Image name	Resolution	Scramble time
Cameraman	512X512	0.059s
Balloon	1024X1024	0.22s
Lena	513X513	0.076s

Three images are used and evaluate the algorithm performance according to, these images are: cameraman.png, lena.png, and Balloon.png, table1 shows the plain images and the corresponding sizes, and the run-time of the Scramble procedure.

5.1 Performance analysis on IoT devices:

The 3D-CML PRNG was tested on Raspberry pi 3b+ (CPU: 1.2 GHz quad-core ARM Cortex-A53, RAM: 1GB SRAM, OS: Raspbian), this testing is essential, since the proposed algorithm was implemented for applied in limited resources platforms, so all specifications of them should be taken into consideration, table2 shows runtime results of key generation using 3D-CML model, under a different number of lattices and iterations.

On another hand, the run-time of this algorithm has been improved by using parallel programming. In python v3.10, Ray [22] library was used to parallelize the XOR bitwise execution of the encryption algorithm, this library distributes multiple processes on multiple cores in a local machine or multiple global machines to decrease the run-time execution.

In the Parallel Encryption Algorithm, a set of CPU cores is specified to perform the XOR operations between every pair of ($SI_{(x,y)}$, triple 8-bit-$OTSK_{(N\times N)}$) by sharing the input data among several cores. Table3 shows the difference in run-time among the execution of parallel Encryption Algorithm, and serial Encryption Algorithm for two images differs in their sizes, the results indicated that (40-65%) of the run time was reduced.

5.2 Key-space:

Defined as all possible and valid keys that initialize the PRNG system, key-space indicates whether the PRNG system can be considered strong enough against brute-force attach or not, at this point, typically, a cryptography system with a key-space more than 128-bit is considered strong [23].

The 3D-CML algorithm depends on several parameters that initialize the map, using 53bits representation (16 digits precision) the key-space is

Table 2. the run-time of PRNG under different No. of lattice

No. of lattices	No. of Iterations	Run-time
10	100	0.157 sec
10	1000	0.583 sec
30	100	0.463 sec
30	1000	1.786

Table 3. Execution time of encryption algorithm (serial execution, parallel execution)

Image size	Serial	Parallel
512X512	1. 17s	0.018s
1024X1024	1.998s	0.05s

Table 4. Comparison of the $MI(I_1, I_2)$ with related works

The Proposed	[15]	[12]
0.0625	0.4496	0.13

equal to (2^{373}), since there are three initial condition $(x, y, z) \in [0,2]$, and four chaotic parameters ($\varepsilon \in [0, 2], 3.53 \leq U \leq 3.81, 0 < B < 0.022, 0 < A < 0.0.015$). In addition, the secret T iterations that specified by the user. So, the 3D-CML consider strong enough in this case.

5.3 Key sensitivity test:

An important thing to test the sensitivity of the encryption key is to ensure that the cryptosystem can withstand Differential Cryptanalysis, encryption key sensitivity means a whole new ciphertext could be produced by a bit change in an original key [12]. So, in this section, a calculation of Mutual Information (MI) was presented between two images, encrypted by using two encryption keys, that differ in a specific bit among them. The MI is defined by:

$$MI(I_1, I_2) = E(I_1) + E(I_2) - JE(I_1, I_2) \quad (7)$$

Where I_1 ,and I_2 are two encrypted images, $E(I_1)\ and\ E(I_2)$ are Information Entropy of both images, respectively, $JE(I_1, I_2)$ is the joint information entropy of them. When the MI is small that refers to a completely different encrypted image produced by a slight change in an encryption key. As shown in Table 4, the proposed algorithm achieved better key sensitivity compared with other proposed algorithms.

5.4 Scramble algorithm analysis:

Gray Difference Degree (GDD) metric show how the proposed algorithm is resistant to passive attacks. The scramble algorithm performance depends on the effectiveness of diffusion effects that are produced, this performance can be expressed as numerical representation through calculating the GDD, as following steps:

Calculate the Gray Difference (GD) between a current pixel with neighbour pixels:

$$GD(i,j) = \frac{1}{12}\sum_{b=1}^{3}\sum_{i`,j`}\left[P_{(i,j)} - P_{(i`,j`)}\right]^2 \quad (8)$$

Where b is the number of pixel bands (RGB image has three 8-bits bands), and $P_{(i`,j`)}$ is a set of neighbour pixels of a given pixel $P_{(i,j)}$, included = $\{P_{(i-1,j)}, P_{(i+1,j)}, P_{(i,j-1)}, P_{(i,j+1)}\}$. This step should exclude the edge pixels.

Find the Average Gray Difference (AGD) for the whole image, as:

$$AGD = \frac{\sum_{i=0}^{M-2}\sum_{j=0}^{N-2} GD(i,j)}{(M-1)\times(N-2)} \quad (9)$$

Finally, the GDD defined by:

$$GDD = \frac{AGD` - AGD}{AGD` + AGD} \quad (10)$$

AGD is the average gray difference for Plain Image, $AGD`$ is the average gray difference for the Scrambled Image.

GDD values are varying between [-1,1], and the target value is near 1, which means a high scramble performance.

Table 5 included the experimental results of the proposed method that was implemented upon the Henon-map and the results that were generated from implementation upon the Arnold Cat-map, obviously, the scrambling by Henon-map can produce a more random scrambled image by fewer iterations, this feature is more important in IoT environments where the need to deploy high scrambling performance in fewer iterations. On another side, Henon-map is an invertible map, which means the original image can be re-scrambled by the equivalent scramble's iteration number.

Table 5. GDD of 10-iterations-scrambled cameraman by using Henon-map (HM), and Cat-map (CM)

Iter.	GDD HM	GDD CM	Iter.	GDDHM	GDD CM
1	0.95	0.46	6	0.96	0.96
2	0.96	0.74	7	0.96	0.96
3	0.96	0.84	8	0.96	0.95
4	0.96	0.90	9	0.96	0.96
5	0.96	0.93	10	0.96	0.96

5.5 Key randomness test:

Testing the randomness of the 3D-PRNG system has been implemented using NIST's statistical suite test (sts-2.1.2) [22], the NIST included 16 randomness tests, and the outcome results of the tester can be interpreted in a probabilistic term, in which consider the randomness pattern in each sequence. Hence, for each test compute a probability value p-value, that interprets the strength of evidence against the null hypothesis α, so if p-value ≥ α then the sequence is random, otherwise, if p-value < α then the sequence is non-random. α can vary in range [0.001, 0.01].

Ten sequences have been generated using 3D-PRNG under different initial conditions, each sequence has 10^6 bits as a length, and these sequences are iterated by 3D-CML as three lattice sets for 10 times, 50 times, and 100 times. As observed in table6, the sequences are passed all tests under the different iteration numbers, where the p-value columns for all iterations were greater than the significance level α, and the proportion columns refer that the sequences are random with a confidence of 100% and 99%.

Accordingly, the results reflect the response of the proposed algorithm to the requirements of a limited-resources platform since the key generator doesn't require more lattices and iterations to be pseudorandom.

5.6 Histogram and chi-square χ^2 analysis:

Both are measurements of representing the image's pixels distribution and uniformity of gray-scale values, the former is graphical Information, while the latter is a numerical metric defined as[10]:

$$\chi^2 = \sum_0^{255} \frac{(O_i - e)^2}{e} \qquad (11)$$

O_i is an occurrence of a gray value that is observed. e is an expected occurrence of the gray value.

The χ^2 results of encrypted images that are shown in table7 are below the significant level: $\alpha = 0.05$ and $\chi^2_{0.05} = 293.25$, so, the proposed algorithm can be robust against statistical attacks.

The histogram refers to how the gray level occurrence in pixel counts context, such information should not be available in the encrypted image. Fig. 3(a) shows the histogram of the original image, and Fig. 3(b) demonstrates the uniform gray-scale intensity distribution.

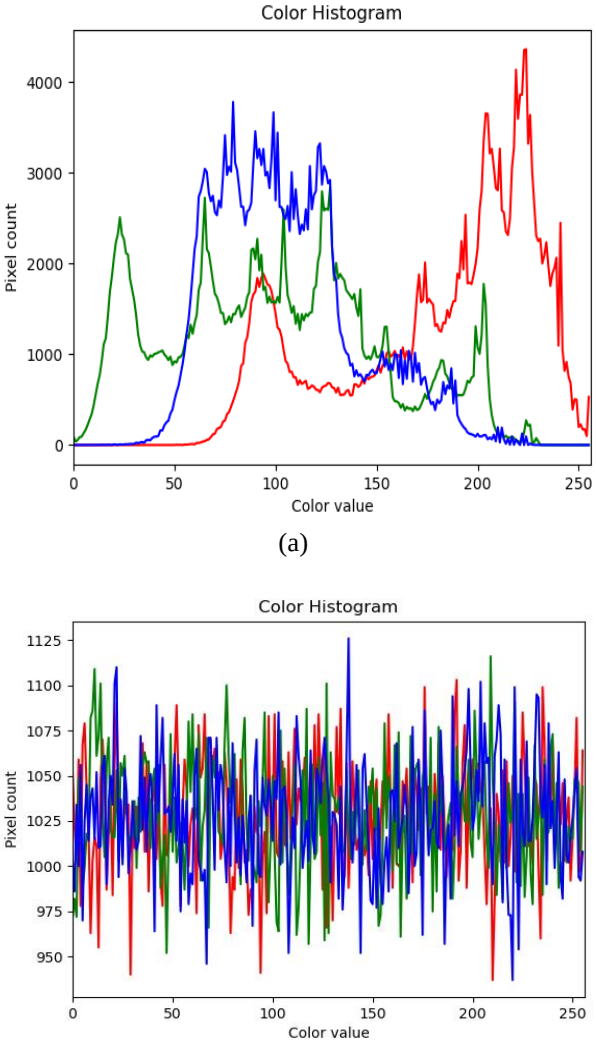


Figure. 3 (a) Histogram of original Lena image and (b) Histogram of Lena's encrypted image

5.7 Correlation analysis:

a more correlation between two adjacent pixels, naturally, appears in the Plain Image, but any image encryption algorithm aims to resolve such relations. In three directions, horizontally, vertically, and diagonally, selected random adjacent pixels to evaluate the relations between them, that what shown in Fig. 4. The correlation coefficient is denoted by $r_{(x,y)}$ described in Eq. (12), whenever the result is closest to zero then the encrypted image can be resisted different statistical attacks.

$$r_{(x,y)} = \frac{\sum(x_i - x`)(y_i - y`)}{\sqrt{\sum(x_i - x`)^2 \sum(y_i - y`)^2}} \qquad (12)$$

x and y are two neighbour pixel vectors.

$x`$ and $y`$ are the mean of vectors x and y respectively.

Table 6. NIST suite tests results (PROPORTION = Prop, Result = RES)

STATISTICAL TEST	10 iterations			50 iterations			100 iterations		
	P-VALUE	Prop	RES	P-VALUE	Prop	RES	P-VALUE	Prop	RES
Frequency	0.739918	100%	Pass	0.911413	100%	Pass	0.534146	99%	Pass
Block Frequency	0.739918	100%	Pass	0.066882	100%	Pass	0.911413	100%	Pass
Cumulative Sums (Forward)	0.579749	100%	Pass	0.534146	100%	Pass	0.739918	99%	Pass
Cumulative Sums (Reverse)	0.529735	100%	Pass	0.122325	100%	Pass	0.350485	99%	Pass
Runs	0.350485	100%	Pass	0.350485	100%	Pass	0.739918	100%	Pass
Longest Run	0.534146	100%	Pass	0.350485	100%	Pass	0.350485	100%	Pass
Rank	0.350485	100%	Pass	0.213309	100%	Pass	0.534146	100%	Pass
FFT	0.213309	100%	Pass	0.534146	100%	Pass	0.739918	100%	Pass
Nonoverlapping Template	0.350485	100%	Pass	0.739918	100%	Pass	0.122325	100%	Pass
Overlapping Template	0.350485	99%	Pass	0.739918	100%	Pass	0.213309	100%	Pass
Universal	0.534146	100%	Pass	0.534146	99%	Pass	0.534146	100%	Pass
Approximate Entropy	0.739918	99%	Pass	0.213309	100%	Pass	0.122325	100%	Pass
Random Excursions	0.583628	100%	Pass	0.776223	100%	Pass	0.554793	100%	Pass
Random Excursions Variant	0.522431	100%	Pass	0.17475	100%	Pass	0.537657	100%	Pass
Serial	0.350485	100%	Pass	0.350485	100%	Pass	0.534146	100%	Pass
Linear Complexity	0.739918	100%	Pass	0.350485	99%	Pass	0.213309	99%	Pass

5.8 Entropy:

A describer measurement indicates the information quantities in an image. According to Shannon's theory [24], the Entropy E represents how many bits are required to encode each pixel in the image. let's assume that the data source I, the $E(I)$ formula is:

$$E(I) = -\sum_{k}^{L-1} p(k) \log_2 p(k) \quad (13)$$

Where I is an original image, $L\ is\ equal\ to\ 256$, i.e., gray level values in an image, and $p(k)$ Is the probability of occurrence k gray value in an image. In the case of an 8-bit encrypted image, all probabilities of occurrence k values are desired to be equal, which means close to 8. Table 8 compares the entropy results for several plain images and corresponding encrypted images, with the other related works.

Table 7. χ^2 results of each RGB band

Image Name	Expected value (e)	χ^2
Lena	1028	255.5
Baboon	1028	256
Pepper	1024	268
Butterfly	1024	272.4
Cameraman	1036	228.8

5.9 The number of changing pixel rate (NPCR) and unified average changing intensity (UACI):

Measurements of encryption algorithm sensitivity, which can assess the algorithm resistance against chosen-plaintext attack[25], where a higher sensitivity encryption algorithm is desired. The NPCR and UACI formula is defined by Eq. (14) and Eq. (16):

$$NPCR = \frac{\sum_{i=1,j=1}^{m,n} D(i,j)}{M \times N} \quad (14)$$

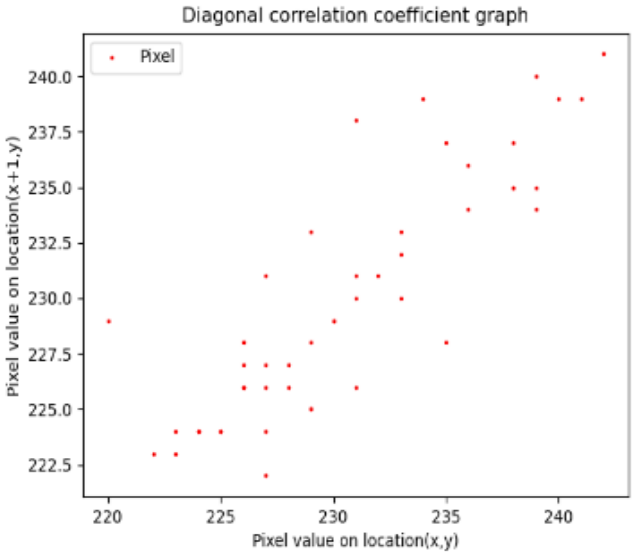
$$\text{Where: } D(i,j) = \begin{cases} 0, c_1(i,j) = c_2(i,j) \\ 1, c_1(i,j) \neq c_2(i,j) \end{cases} \quad (15)$$

$$UACI = \frac{\sum_{i=1,j=1}^{m.n} \frac{|c_1(i,j) - c_2(i,j)|}{255}}{M \times N} \quad (16)$$

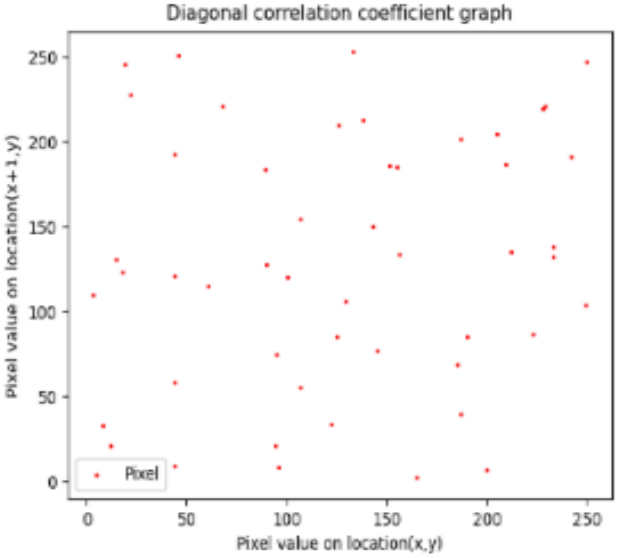
The results of Table 9 ensure that the proposed algorithm is more strength and provides an effective confusion effect when a slight change occurs in an input of the algorithm, these results are calculated after altered LSB in $OTSK$, hence the $NPCR\ and\ UACI$ are calculated between two image ciphers (C_1, C_2) for the same Plain Image.

5.10 Quality of decrypted image:

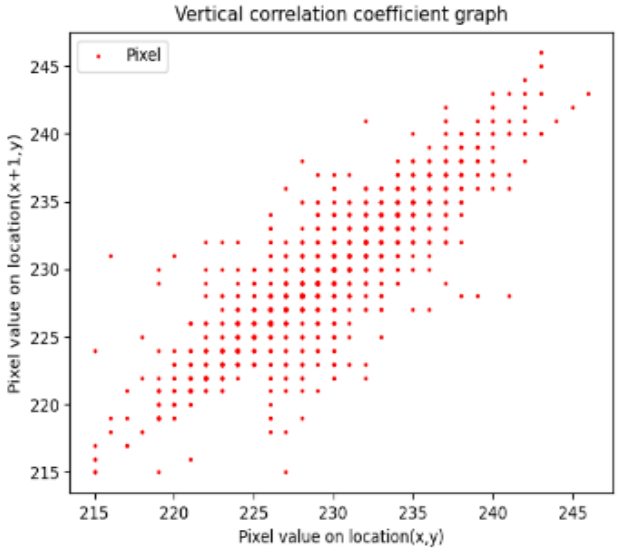
is important to know whether there is a loss in image quality after decrypted or not, the quality is



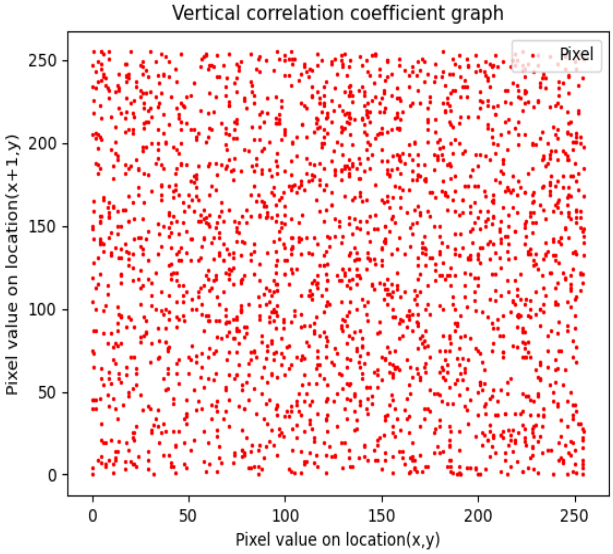
(a)



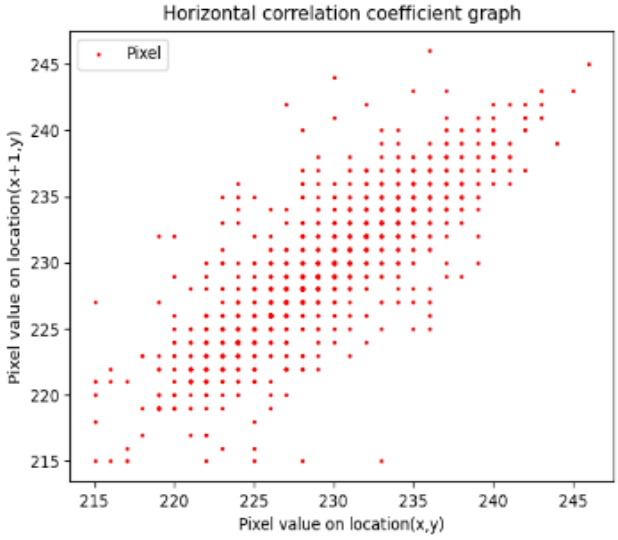
(d)



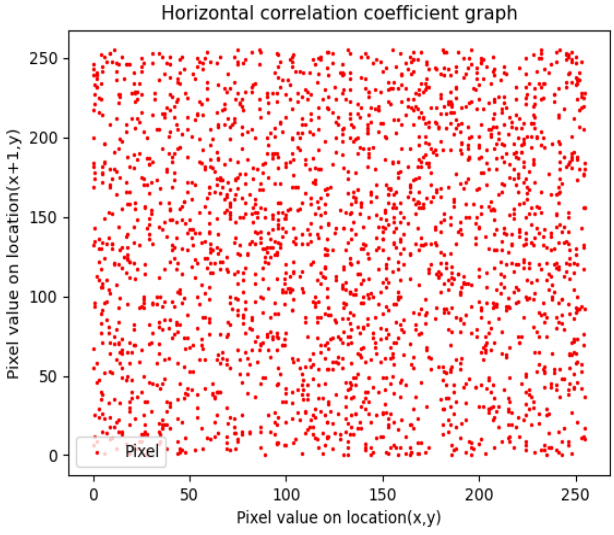
(b)



(e)



(c)



(f)

Figure. 4: (a-c) correlation analysis of Lena's encrypted image in three directions and (d-f) correlation analysis of original Lena image in three directions

Table 8. Entropy results for different original images and corresponding encrypted images

Image name	PI	Proposed	[14]	[13]	[11]	[10]	[12]
Cameraman	7.017	**7.9994**	-	-	-	7.9971	-
Baboon	7.7393	**7.9998**	7.9893	7.9971	7.9993	7.9993	7.9993
Lena	7.739	**7.9995**	7.9899	7.998	7.9993	-	7.9993
Butterfly	7.577	**7.9998**	-	-	-	-	-
Boat	7.363	**7.947**	-	-	-	-	7.9994
Pepper	7.7115	**7.9998**	7. .9890	-	7.9993	7.9991	-

Table 9. $NPCR\ and\ UACI$ cipher's randomness

Plain Image	$NPCR$	$UACI$
Lena	99.558	33.48
Baboon	99.557	33.42
Boat	99.562	33.54
Cameraman	99.558	33.44
Balloon	99.579	33.44
Pepper	99.556	33.41

the most crucial issue when dealing with multimedia objects. So, three metrics have been used to evaluate the quality of the decoded image compared to the plain image; Mean Square Error (MSQ), Peak Signal-to-Noise Ratio (PSNR), and Structure Similarity Index Measure (SSIM), which is defined as the following:

$$MSE = \sum_{i=1,j=0}^{M,N} \frac{(P(i,j)-C(i,j))^2}{MXN} \quad (17)$$

Where $P(i,j), and\ C(i,j)$ represent the Plain Image and Encrypted Image, respectively. Both are sized $N \times N$. The $MSE's$ values closer to 0 denotes to those inputs are more identical, hence better quality.

$$PSNR = 10\ log_{10}\left(\frac{(I_{max})^2}{\sqrt{MSE}}\right) \quad (18)$$

I_{max} is the greatest intensity value. the $PSNR's$ results are varying between $[30-50]dB$ for 8-bits data representation, the higher value is targeted here.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (19)$$

$\mu_x\mu_y$ are referring to the averages of x and y, respectively. σ_x and σ_y are the variance of both x and y. SSIM′s values closer to 1 are better, which means the similarity between Decryption and Plain Image is higher, hence, no loss in quality after decrypting the Encrypted Image.

Table 10. MSE, PSNR, and SSIM

Name	MSE	PSNR	SSIM
Lena	0.0	48.14	1.0
Pepper	0.55	50	0.99
Baboon	0.5	50	0.99

The resultants, between Decrypted and Plain Images, appear high value of PSNR, approximately zero values for MSE, and SSIM closer to 1, as shown in Table 10, as a result, these resultants demonstrated the proposed algorithm preserved the original image quality.

6. Conclusion

In a limited-resources environment, like IoT, a lot of restrictions make the drafting of a cryptographic algorithm challenging to accommodate such an environment's requirements. This paper proposed a comprehensive cryptographic system that involved multiple cryptography mechanisms that would be secure, and IoT-adaptive performance. At first, a 3D-PRNG worked on the lightweight configurations to approve the resources limitations, the chaotic behavior was evaluated by the bifurcation diagram, and it illustrated that there is an expansion in the parameter's domain, in which the domain was expanded from [0.0-1.0] up to (0.0-2.0), this expansion immune the encryption key against brute force attack by strengthening the key-space size, making it equal to 2^{373}. On another hand, the generated keys under different parameters were tested by NIST-suite, this testing proved that the 3D-PRNG can produce random sequences in a lightweight configuration. The first stage in encryption is 2D-Henon map scramble model, the comparison with the Cat map demonstrates that the HM could produce a better-scrambled image in fewer iterations and can be exactly invertible without losing information, consequently, can conclude that the employment of the HM in a system that has been run in a restricted environment, wouldn't take up many resources. About (40-65%) of the run-time was reduced by parallelized encryption and decryption

algorithm. The outputs of encryption and decryption algorithm were assessed under several assessments, that can give a clear picture of the security of scramble-encrypted images, original image recovery quality, and degree of resistance against statistical, differential, brute-force attacks, hence, all above indicators were acceptable regarding the specified criteria and achieved better results relative to other related works. So, the significance of this study indicates that the proposed system can be high-secure and lightweight to suit the resources-capabilities of IoT devices.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing-original preparation, writing review, editing, and visualization have been implemented by the first author. Supervision and project administration have been implemented by the second author.

References

[1] K. S. Roy and H. K. Kalita, "A Survey on Authentication Schemes in IoT", In: *Proc. of 2017 Int. Conf. Inf. Technol. ICIT 2017*, No. November 2018, pp. 202-207, 2018, doi: 10.1109/ICIT.2017.56.

[2] K. Juvva, "Security.", 1998. [Online] Available: https://users.ece.cmu.edu/~koopman/des_s99/security/.

[3] M. S. Fadhil, A. K. Farhan, and M. N. Fadhil, "A lightweight aes algorithm implementation for secure iot environment", *Iraqi J. Sci.*, Vol. 62, No. 8, pp. 2759-2770, 2021, doi: 10.24996/ijs.2021.62.8.29.

[4] P. L. Carmen and L. R. Ricardo, "Notions of Chaotic Cryptography: Sketch of a Chaos Based Cryptosystem", *Appl. Cryptogr. Netw. Secur.*, 2012, doi: 10.5772/36419.

[5] J. S. Muthu and P. Murali, "Review of Chaos Detection Techniques Performed on Chaotic Maps and Systems in Image Encryption", *SN Comput. Sci.*, Vol. 2, No. 5, 2021, doi: 10.1007/s42979-021-00778-3.

[6] U. Zia, M. McCartney, B. Scotney, J. Martinez, and A. Sajjad, "A novel pseudo-random number generator for IoT based on a coupled map lattice system using the generalised symmetric map", *SN Appl. Sci.*, Vol. 4, No. 2, 2022, doi: 10.1007/s42452-021-04919-4.

[7] A. Cronwright, "Validation of Pseudo Random Number Generators through Graphical Analysis", 2005. [Online] Available: http://www.cs.ru.ac.za/research/g02c2954/Final.

[8] K. Kaneko, "Overview of coupled map lattices", *Chaos*, Vol. 2, No. 3, pp. 279-282, 1992, doi: 10.1063/1.165869.

[9] M. B. Hossain, M. T. Rahman, A. B. M. S. Rahman, and S. Islam, "A new approach of image encryption using 3D chaotic map to enhance security of multimedia component", In: *Proc. of 2014 Int. Conf. Informatics, Electron. Vision, ICIEV 2014*, pp. 6-11, 2014, doi: 10.1109/ICIEV.2014.6850856.

[10] X. Wang, N. Guan, H. Zhao, S. Wang, and Y. Zhang, "A new image encryption scheme based on coupling map lattices with mixed multi-chaos", *Sci. Rep.*, Vol. 10, No. 1, pp. 1-15, 2020, doi: 10.1038/s41598-020-66486-9.

[11] I. A. Taqi and S. M. Hameed, "A new Color image encryption based on multi chaotic maps", *Iraqi J. Sci.*, Vol. 59, No. 4, pp. 2117-2127, 2018, doi: 10.24996/IJS.2018.59.4B.17.

[12] X. Wang, L. Feng, and H. Zhao, "Fast image encryption algorithm based on parallel computing system", *Inf. Sci. (Ny).*, Vol. 486, pp. 340-358, 2019, doi: 10.1016/j.ins.2019.02.049.

[13] X. Chen, "Pseudorandom Number Generator Based on Three Kinds of Four-Wing Memristive Hyperchaotic System and Its Application in Image Encryption", *Complexity*, Vol. 2020, 2020, doi: 10.1155/2020/8274685.

[14] S. Patel, K. P. Bharath, and M. R. Kumar, "Symmetric keys image encryption and decryption using 3D chaotic maps with DNA encoding technique", *Multimed. Tools Appl.*, Vol. 79, No. 43-44, pp. 31739-31757, 2020, doi: 10.1007/s11042-020-09551-9.

[15] V. Patidar, N. K. Pareek, and G. Purohit, "A Novel Quasigroup substitution scheme for Chaos based image encryption", *J. Appl. Nonlinear Dyn.*, Vol. 7, No. 4, pp. 393-412, 2018, doi: 10.5890/JAND.2018.12.007.

[16] M. Saber and M. M. Eid, "Low power pseudo-random number generator based on lemniscate chaotic map", *Int. J. Electr. Comput. Eng.*, Vol. 11, No. 1, pp. 863-871, 2021, doi: 10.11591/ijece.v11i1.pp863-871.

[17] M. G. Bosque, A. P. Resa, C. S. Azqueta, C. Aldea, and S. Celma, "Chaos-Based Bitwise Dynamical Pseudorandom Number Generator on FPGA", *IEEE Trans. Instrum. Meas.*, Vol. 68, No. 1, pp. 291-293, 2019, doi:

10.1109/TIM.2018.2877859.
[18] E. Erdem and A. M. Garipcan, "Hardware Implementation of Chaotic Zigzag Number Generator on Field-Programmable Gate Array psevdo-random generatorja števil v FPGA na", Vol. 50, No. 4, pp. 243-253, 2020.
[19] G. Ye, K. Jiao, C. Pan, and X. Huang, "An effective framework for chaotic image encryption based on 3D logistic map", *Secur. Commun. Networks*, Vol. 2018, 2018, doi: 10.1155/2018/8402578.
[20] H. Wen, "A review of the Henon map and its physical interpretations", *Sch. Phys. Georg. Inst. Technol. Atlanta, GA 30332-0430*, pp. 1-9, 2014.
[21] P. Ping, Y. Mao, X. Lv, F. Xu, and G. Xu, "An image scrambling algorithm using discrete Henon map", In: *Proc. of 2015 IEEE Int. Conf. Inf. Autom. ICIA 2015 - Conjunction with 2015 IEEE Int. Conf. Autom. Logist.*, No. August, pp. 429-432, 2015, doi: 10.1109/ICInfA.2015.7279326.
[22] R. Team, "Ray documentation", *arXiv*, p. 420, 2020.
[23] V. Patidar, K. K. Sud, and N. K. Pareek, "A pseudo random bit generator based on chaotic logistic map and its statistical testing", *Inform.*, Vol. 33, No. 4, pp. 441-452, 2009.
[24] C. E. Shannon, "A Mathematical Theory of Communication", *Bell Syst. Tech. J.*, Vol. 27, No. 4, pp. 623-656, 1948, doi: 10.1002/j.1538-7305.1948.tb00917.x.
[25] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI Randomness Tests for Image Encryption", *Cyberjournals.Com*, 2011. [Online] Available: http://www.cyberjournals.com/Papers/Apr2011/05.pdf.