Journal of Systems Science and Information
Dec., 2024, Vol. 12, No. 6, pp. 775–789
DOI: 10.21078/JSSI-2024-0054

A New Pseudo-Random Generator Based on Two Chaotic Systems

Abderrahmene HADJ BRAHIM
LACOSI Laboratory, USTO-MB University, Oran 31000*, Algeria*
E-mail: *abderrahmene.hadjbrahim@univ-usto.dz*

Hana ALI PACHA
Quartz Laboratory, ECAM-EPMI, Cergy 95092*, France*
E-mail: *h.ali-pacha@ecam-epmi.com*

Mohammed NAIM
LACOSI Laboratory, USTO-MB University, Oran 31000*, Algeria*
E-mail: *mohammed.naim@univ-usto.dz*

Adda ALI PACHA
LACOSI Laboratory, USTO-MB University, Oran 31000*, Algeria*
E-mail: *a.alipacha@gmail.com*

Abstract In many applications of information processing, such as cryptography, generating random sequences presents many difficulties. In this paper, a new pseudo-random sequence is proposed, based on two chaotic systems, a logistic map and a seven-dimensional (7D) hyperchaotic system. The main process of the proposed generator is that it functions by using the logistic map to control the 7D hyperchaotic system, which exhibits random behavior to produce a pseudo-random sequence. Specifically, the logistic map is used to select one of the variables from the 7D hyperchaotic system. The variable selected at each iteration is used as a controller to fill the pseudo-random sequence, choosing from one of the other variables of the 7D hyperchaotic system. This means that in each iteration, the pseudo-random sequence takes a value from the 7D hyperchaotic system according to the logistic map and the selected variable of the 7D hyperchaotic system. This method allows the creation of a highly efficient pseudo-random generator through simple processes. Experimental and analysis results show that the proposed generator has good random characteristics, making it suitable for cryptography applications such as encryption algorithms.

Keywords hyperchaotic system; random generator; pseudo-random sequence; statistical tests

1 Introduction

With the evolution of technology, random generators, which exhibit random behavior, have become very important in different fields. However, generators that have perfect randomness are not widely used due to their high cost and slow performance[1]. To solve these problems, random

Received April 27, 2024, accepted October 29, 2024

generators are replaced by pseudo-random generators, which produce deterministic sequences based on mathematical formulas, achieving excellent randomness and statistical properties similar to white noise[2–6]. These generators are widely used in many fields such as satellites[7], digital information processing systems[8], and cryptography[9], where pseudo-random sequences are used for generating secret keys, authentication, ensuring confidentiality, and information integrity.

There are many methods to construct pseudo-random sequences, such as the m-sequence and the gold-sequence[10, 11]. However, these methods have low complexity and are not suitable for cryptography, due to their limited speed of creation. In recent years, another method based on chaotic systems has been used to generate a pseudo-random sequence with excellent performance.

Chaotic systems are generated by deterministic equations and have several advantages in generating random sequences, such as sensitivity to small changes in initial conditions, non-periodicity, and unpredictability[12]. There are two types of chaotic systems: continuous chaotic systems and discrete chaotic systems. Recent studies have focused on proposing new hyperchaotic systems, which have better random behavior and can effectively improve the security of pseudo-random sequences compared to classical chaotic systems. In recent studies, several methods have been proposed to generate pseudo-random sequences based on hyperchaotic systems[13–16].

In this paper, a new pseudo-random sequence generation method is proposed based on two chaotic systems, the logistic map, and a seven-dimensional (7D) hyperchaotic system. Firstly, the logistic map generates a variable in each iteration, which is adjusted to ensure its values are between 1 and 7. Then, this variable is used to select one variable from the 7D hyperchaotic system. Secondly, the selected variable is modified to fill the pseudo-random sequence with one of the remaining six variables of the 7D hyperchaotic system. Thirdly, the two steps are repeated N times (depending on the needs of the application) to obtain the pseudo-random sequence.

Based on the above information, the proposed algorithm has the following advantages. 1) The proposed pseudo-random sequence is generated using a 7D hyperchaotic system, which means that the sequence is highly sensitive to the initial conditions, similar to the 7D hyperchaotic system. 2) The processes for generating the pseudo-random sequence from the 7D hyperchaotic system are simple and do not require a lot of hardware. 3) Since the proposed pseudo-random sequence is generated using a 7D hyperchaotic system, the generator exhibits the random behavior of both the 7D hyperchaotic system and the proposed method. 4) The proposed pseudo-random generator does not have a limit on the number of bits it can generate, allowing it to produce bits according to the specific needs of the application it is implemented in.

The remainder of the paper is organized as follows: Section 2 introduces the methods used in this paper. Section 3 describes the empirical research (the proposed pseudo-random generator). Section 4 provides the analysis results and discussion. Finally, conclusions are given in Section 5.

2 Methods

2.1 Logistic Map

The logistic map is one of the simplest discrete, dynamic, non-linear systems[17]. It is defined as follows:

$$L_{n+1} = \mu(1 - L_n), \tag{1}$$

where, L_0 represents the initial condition, μ the parameter coefficient, and n signifies the number of iterations. Figure 1 shows the bifurcation of the logistic map. It demonstrates that the logistic map has chaotic behavior when μ is close to 4.

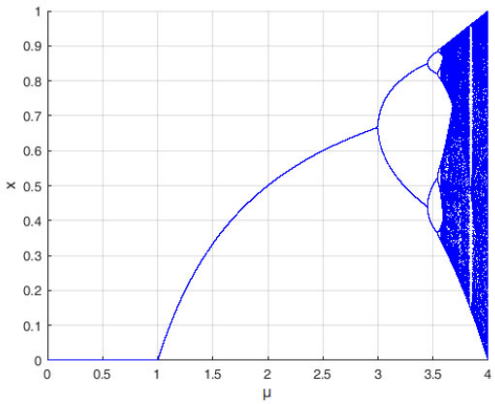

Figure 1 Logistic map bifurcation diagram

Figure 2 shows the difference between two signals when a small change occurs in the initial state. The first signal is obtained with an initial condition $L_0 = 0.1$, while the second signal is obtained with an initial condition $L_0 = 0.100000000001$. It is observed a slight error in the initial condition results in two distinct signals.

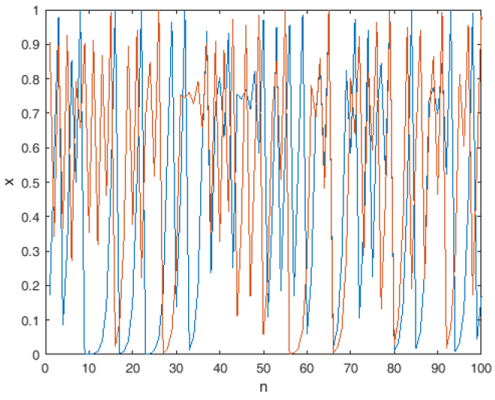

Figure 2 Logistic map with two different initial conditions

2.2 7D Hyperchaotic System

In 2019, Yu, et al.[13] proposed a new seven-dimensional (7D) hyperchaotic system, which is described by

$$\begin{cases} 1) & \dot{x} = a(y-x) + w - u - v, \\ 2) & \dot{y} = cx - y - xz - p, \\ 3) & \dot{z} = -bz + xy, \\ 4) & \dot{w} = dw - yz, \\ 5) & \dot{u} = ev + yz, \\ 6) & \dot{p} = fx + yz, \\ 7) & \dot{v} = rx. \end{cases} \tag{2}$$

where x, y, z, w, u, p, v are the system state variables, $\dot{x} = \mathrm{d}x/\mathrm{d}t$, and a, b, c, d, e, f, r are the system parameters. When $a = 10, b = 8/3, c = 28, d = -1, e = 8, f = 1$, and $r = 5$; the 7D hyperchaotic system is in a chaotic state, and for the initial condition $Q(1, 1, 1, 1, 1, 1, 1)$, the Lyapunov exponents of the hyperchaotic system are: $L_1 = 0.58845$, $L_2 = 0.178183$, $L_3 = 0, L_4 = -0.139483, L_5 = -0.404591, L_6 = -0.80156, L_7 = -14.088129$. The phase diagrams of the 7D hyperchaotic system are shown in Figure 3, where (a), (b), (c), (d), (e) and (f) correspond to the phase diagrams of the attractors for $x - y, z - w, p - v, x - y - z, w - p - v$, and $x - y - v$, respectively.

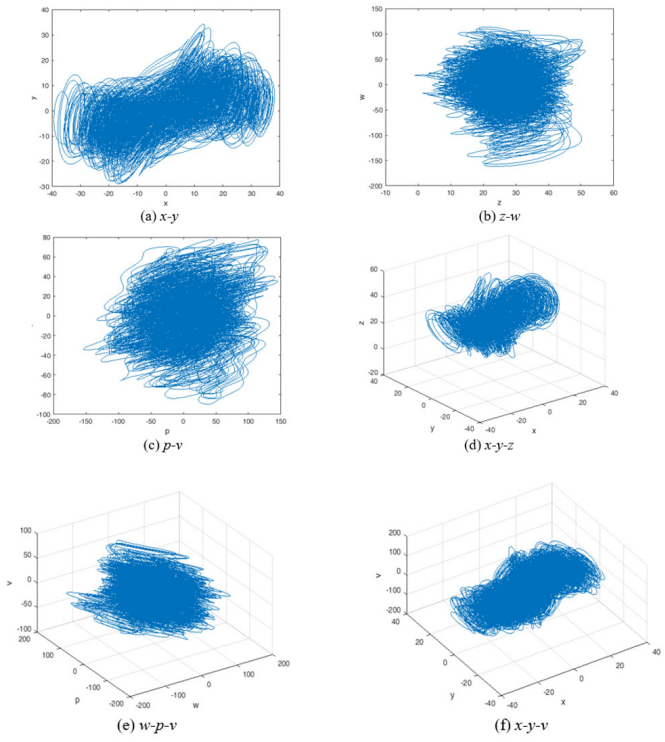

Figure 3 Phase diagram of 7D hyperchaotic system: (a) $x - y$; (b) $z - w$; (c) $p - v$; (d) $x - y - z$; (e) $w - p - v$; (f) $x - y - v$

Based on the above analyses, the 7D hyperchaotic has the following advantages: First, two of the Lyapunov exponents of the hyperchaotic system are greater than zero. Second, the dynamic behavior of the 7D hyperchaotic system (shown in Figure 3) is more complex, with phase trajectories separated in multiple directions. As a result, the 7D hyperchaotic system demonstrates effective random generation capabilities.

3 Empirical Research

The proposed pseudo-random generator is obtained from the logistic map and the 7D hyperchaotic system, and the detailed steps are as follows:

- Step 1: The logistic map is iterated $500 + N$ times (where N is the number of values needed from the pseudo-random generator) with the initial values L_0. Then, to avoid the transient, the first 500 values from the logistic map are abandoned.

- Step 2: The 7D hyperchaotic is iterated $500+N$ times with initial values $x_0, y_0, z_0, w_0, u_0, p_0$, and v_0, Then, the first 500 values from each of the seven sequences are abandoned to obtain the sequences x, y, z, w, u, p, and v with length $1N$.

$$\begin{cases} 1) & x = [x_1, x_2, \cdots, x_N], \\ 2) & y = [y_1, y_2, \cdots, y_N], \\ 3) & z = [z_1, z_2, \cdots, z_N], \\ 4) & w = [w_1, w_2, \cdots, w_N], \\ 5) & u = [u_1, u_2, \cdots, u_N], \\ 6) & p = [p_1, p_2, \cdots, p_N], \\ 7) & v = [v_1, v_2, \cdots, v_N]. \end{cases} \tag{3}$$

- Step 2: The sequence from step 1 (logistic map sequence) is used as the initial controller for the pseudo-random generator. At each iteration, a new parameter $c1$ is calculated using a variable of the logistic map sequence as follows

$$c_1 = mod(flor(L(I) \times 10^7), 7) + 1, \tag{4}$$

  where $i = 1, 2, \cdots, N$, mod is the modular operator, and $floor(x)$ gives the greatest integer less than or equal to x.

- Step 3: The values of the parameter $c1$ range between one and seven, determining the selection of a variable from the 7D hyperchaotic system at each iteration. If $c1$ is equal to one ($c1 = 1$), the selected value is taken from the first sequence x. If $c1$ is equal to two ($c1 = 2$), the selected value is taken from the second sequence y. If $c1$ is equal to three ($c1 = 3$), the selected value is taken from the third sequence$z$. If $c1$ is equal to four ($c1 = 4$), the selected value is taken from the fourth sequence w. If $c1$ is equal to

five ($c1 = 5$), the selected value is taken from the fifth sequence u. If $c1$ is equal to six ($c1 = 6$), the selected value is taken from the sixth sequence p. If $c1$ is equal to three ($c1 = 7$), the selected value is taken from the last sequence v. The resulting values are stored in the parameter P.

$$P = 7D\_hyperchaotic(c1). \tag{5}$$

- Step 4: A new parameter $c2$ is calculated using the parameter P as follows

$$c_2 = mod(floor(P \times 10^{10}), 6) + 1. \tag{6}$$

- Step 5: The values of the parameter $c2$ range between one and six, determining the selection of a variable from the 7D hyperchaotic system at each iteration to fill the pseudo-random sequence.

- Step 6: For example, if the variable selected in step 3 is x, it means that the pseudo-random sequence is filled with a value from the other six variables of the 7D hyperchaotic system (y, z, w, u, p, and v). The value of parameter $c2$ determines which sequence fills the pseudo-random sequence. If $c2$ is equal to one ($c2 = 1$), the pseudo-random sequence is filled by one value from the second sequence y. If $c2$ is equal to two ($c2 = 2$), the pseudo-random sequence is filled by one value from the third sequence z. If $c2$ is equal to three ($c2 = 3$), the pseudo-random sequence is filled by one value from the fourth sequence w. If $c2$ is equal to four ($c2 = 4$), the pseudo-random sequence is filled by one value from the fifth sequence u. If $c2$ is equal to five ($c2 = 5$), the pseudo-random sequence is filled by one value from the sixth sequence p. If $c2$ is equal to seven ($c2 = 7$), the pseudo-random sequence is filled by one value from the last sequence v.

- Step 7: Steps 2 to 6 are repeated N times to obtain the desired pseudo-random sequence.

4 Result Analysis

This section presents the results of simulations implemented using Matlab 2010 on a 64-bit computer with an Intel (R) Core (TM) i3 CPU @2.13 GHZ with 4.00 Go RAM and Microsoft Windows 7 operating system.

4.1 Analysis Results and Discussion of the Proposed Generator

The initial condition of the logistic map is $L_0 = 0.1$, and the initial conditions of the 7D hyperchaotic system used are $x_0 = 0.12, y_0 = 0.38, z_0 = 0.22, w_0 = 0.54, u_0 = 0.43, p_0 = 0.51$, and $v_0 = 0.34$, with $N = 100000$. To evaluate the performance of the proposed pseudo-random generator, several tests proposed by NIST (the National Institute of Standards and Technology) are used[18]. These tests include entropy, frequency, mean, variance, autocorrelation factor, and spectral tests. To apply these tests to our pseudo-random generator, the values of the pseudo-random sequence S are modified as follows

$$S = mod(floor(S \times 10^7), 256) \tag{7}$$

4.1.1 Entropy

The entropy test is an important metric used to measure the amount of information in a data source, and it can be measured by[19]

$$H(x) = \sum_{i=1}^{n} P(x_i) \log_2 \left(\frac{1}{P(x_i)} \right), \tag{8}$$

where n represents the possible number of states in a sequence, $p(x_i) = x_i/N$, and x_i is the frequency of occurrence of each state. This test evaluates the probability of each value occurring in the sequence, and a good random generator should have equal probabilities, which correspond to the ideal value. The entropy test is used in this paper to evaluate the sequences $c1$ (Equation (4)), $c2$ (Equation (6)), and S (Equation (7)) and it is listed in Table 1.

In the parameter $c1$ (the logistic map sequence), n equals 7 (there are 7 possible states from 1 to 7). If all these states occur with equal frequency, the entropy value equals $\log_2(7) = 2.8073549$, which is the ideal value in this case. The entropy value of the parameter c1 in our proposed pseudo-random generator is 2.8073020, which is very close to the ideal value. Therefore, we can conclude that the parameter $c1$ is sampled from the seven variables of the 7D hyperchaotic system with equal frequency.

In the sequence $c2$ (the selected variable), n equals 6 (there are 6 possible states from 1 to 6). If all these states occur with equal frequency, the entropy value equals $\log_2(6) = 2.5849625$, which is the ideal value in this case. The entropy value of the sequence $c2$ in our proposed pseudo-random generator is 2.5849165, which is very close to the ideal value. Therefore, we can conclude that the pseudo-random generator is filled from the six sequences of the 7D hyperchaotic system with equal frequency.

In the sequence S (the pseudo-random sequence), n equals 256 (there are 256 possible states from 0 to 255). If all these states have the same frequency, the entropy value equals $\log_2(256) = 8$ which is the ideal value in this case. The entropy value of the sequence S in our proposed pseudo-random generator is 7.9981, indicating that the sequence S has a ratio of 99.976% of a source that contains an equiprobable frequency. Therefore, the proposed pseudo-random generator passes the entropy test.

Table 1 Entropy values of the proposed generator

Sequences	Logistic map sequence $c1$	Selected variable $c2$	Pseudo-random sequence S
Entropy	2.8073020	2.5849165	7.9981
Ideal values	2.8073549	2.5849625	8
% For ideal value	99.998%	99.998%	99.976%

4.1.2 Frequency

One of the standard tests to evaluate the pseudo-random generator is the frequency test, where a good pseudo-random sequence must have a uniform distribution among its values,

meaning that the values within the sequence occur with nearly equal frequency[20]. Figure 4 shows the frequency distribution of values in the sequence S.

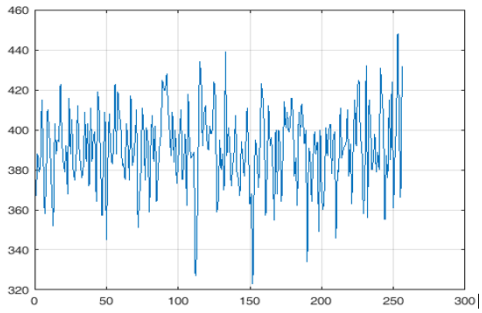

Figure 4 Frequency of the pseudo-random sequence

As can be seen in Figure 4, it is observed that all values (0 to 255) have approximately equal frequency. Therefore, our proposed pseudo-random generator passes the frequency test.

4.1.3 Mean, Variance, and Autocorrelation Factor Tests

The simplest tests to evaluate the distribution of numbers in the pseudo-random generator are the mean, variance, and autocorrelation factors. These tests consist of calculating the mean μ, variance σ^2, and autocorrelation factor of the pseudo-random sequence, defined as follows[20]:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} v_i, \tag{9}$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} v_i^2 - \mu^2, \tag{10}$$

$$R = \frac{1}{N}\sum_{i=1}^{N-1} v_i \times v_{i+1}, \tag{11}$$

where $v_i = x_i/m$, and m is the largest value in the sequence. The ideal values for a good pseudo-random sequence are $0.5, 1/12$, and 0.25, for the mean, variance, and autocorrelation factor, respectively. Figure 5 gives the results of calculating the mean, variance, and autocorrelation

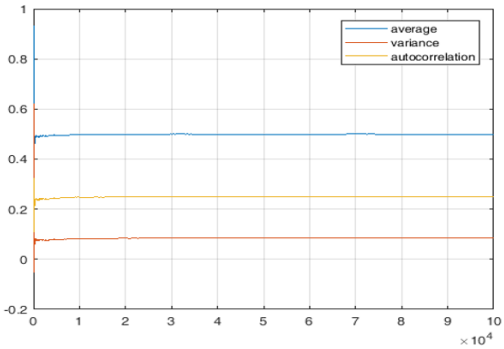

Figure 5 Mean, variance, and autocorrelation factor tests of the pseudo-random sequence

factor of the proposed pseudo-random sequence.

From Figure 5, we can see that the results of the mean, variance, and autocorrelation factor tests of our pseudo-random sequence are close to the ideal values. Therefore, the proposed pseudo-random generator has good random behavior.

4.1.4 Spectral Test

The spectral test is an important metric used to assess the quality of a random generator, and it is considered the most powerful known test. It involves studying the distribution of values created by the pseudo-random generator in different dimensional spaces, such as two dimensions (2D) and three dimensions (3D)[19]. In 2D, each pair of consecutive values represents the coordinates of a point in space, while in 3D, each triplet of consecutive values represents the coordinates of a point in space. For a good random generator, the points should be uniformly distributed in a square in 2D and uniformly distributed in a cube in 3D. Figures 6 and 7 show the 2D and 3D distributions of our pseudo-random sequence, respectively.

From Figures 6 and 7, we can see that the points are uniformly distributed in a square for 2D and a cube for 3D. Therefore, the proposed pseudo-random generator passes the spectral test.

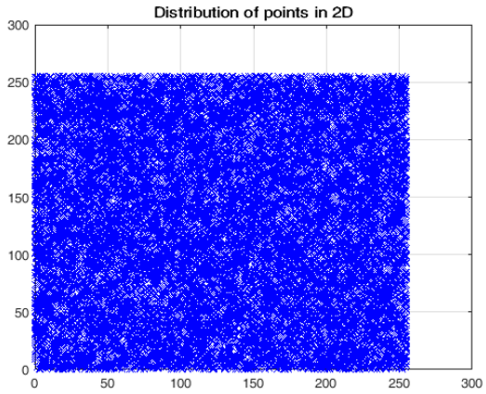

Figure 6 2D distribution of the pseudo-random sequence

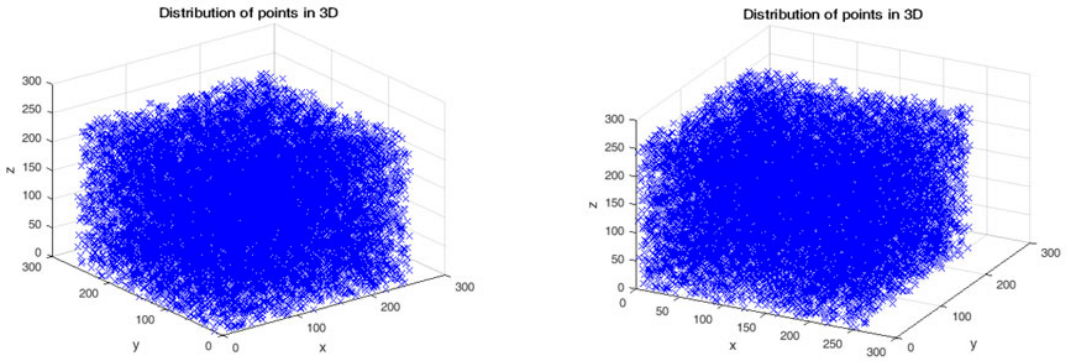

Figure 7 3D distribution of the pseudo-random sequence

4.1.5 Computational Complexity

The speed of bit production is an important metric for any pseudo-random generator. To evaluate the time required to generate the proposed pseudo-random sequence, the quantity of bits generated per second is listed in Table 2 and compared with recent works.

Table 2 Quantity of bits produced per second by different generators (unit: Mbit)

Generator	Proposed	[21]	[22]	[23]
Quantity of Bits	28.43	21.50	14.48	3.89

From Table 2, it can be observed that the proposed generator can achieve a higher bits per second output compared to the recent works.

4.1.6 Analysis of the Choice of Initial Conditions

To determine how the initial conditions of the chaotic system can affect the proposed generator, a modification is applied to the initial condition of the logistic map ($L_0 + 10^{-15}$) to generate the pseudo-random sequence. Moreover, another modification is applied to one initial condition of the 7D hyperchaotic system ($x_0 + 10^{-15}$). The three results (initial conditions without modification, modification in the logistic map, and modification in the 7D hyperchaotic system) are then tested using the spectral test (Figure 8).

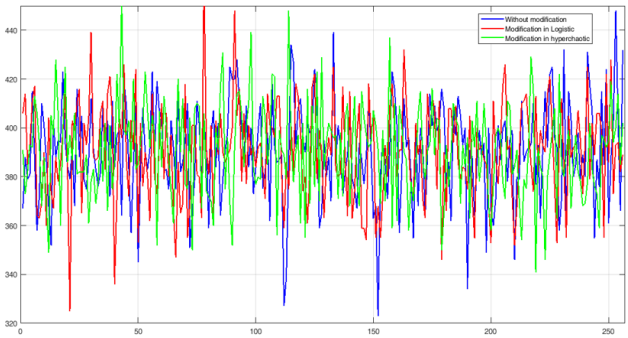

Figure 8 Spectral test using different initial conditions

From Figure 8, it can be observed that small changes in the initial conditions produce completely different sequences, which means that the proposed generator performs well in applications where a sensitive pseudo-random generator, such as cryptography, is needed.

4.2 Application of the Proposed Generator

In this paper, the proposed pseudo-random generator is used for encrypting images using a stream cipher. The proposed generator creates a sequence of values corresponding to the number of pixels in the image, and then each pixel is XORed with a value from the pseudo-random sequence to obtain the encrypted image (Figure 9).

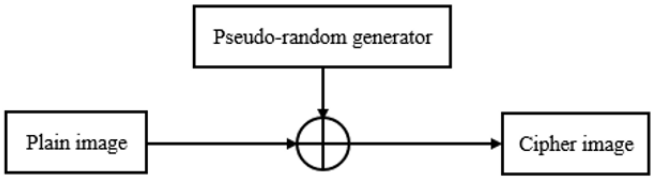

Figure 9 Stream cipher

The encryption key in this paper consists of $L_0 = 0.1, \mu = 3.9999, x_0 = 0.12, y_0 = 0.38, z_0 = 0.22, w_0 = 0.54, u_0 = 0.43, p_0 = 0.51$, and $v_0 = 0.34$. The plain images used are "Lena", and "Cameraman" both sized 256×256 pixels, and the encrypted images are shown in Figure 10.

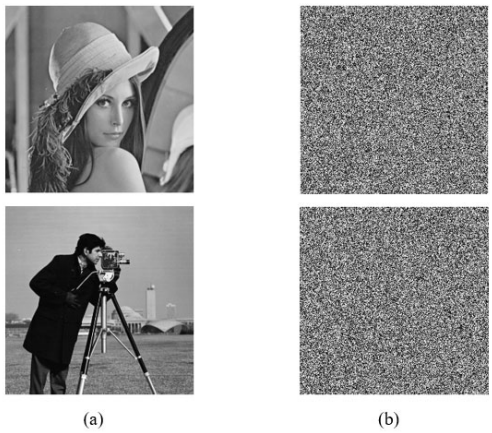

Figure 10 Plain and encrypted images

From Figure 10, it is shown the encrypted images appear visually identical and do not give any visual information about the plain images.

4.3 Histogram

The histogram metric illustrates the distribution of pixel values in an image. The pixel distribution in plain images is uneven. While in cipher images, pixel distribution should be uniform[24]. Figure 11 shows the histograms of both plain and cipher images.

From Figure 11, it is observed that the histograms of the cipher images differ significantly from those of the plain images, and the pixel distribution in the cipher images is almost uniform, indicating that statistical analysis of the histogram in the cipher images cannot give any information about plain images. Therefore, the proposed algorithm exhibits resistance against statistical attacks.

4.3.1 Correlation Coefficients Analysis

The correlation between pixels is an important index in images. In plain images, pixel correlation is strong, and in a good encrypted image, this correlation should be reduced as possible. The correlation coefficient r_xy between adjacent pixels, x and y is defined as follows[25]

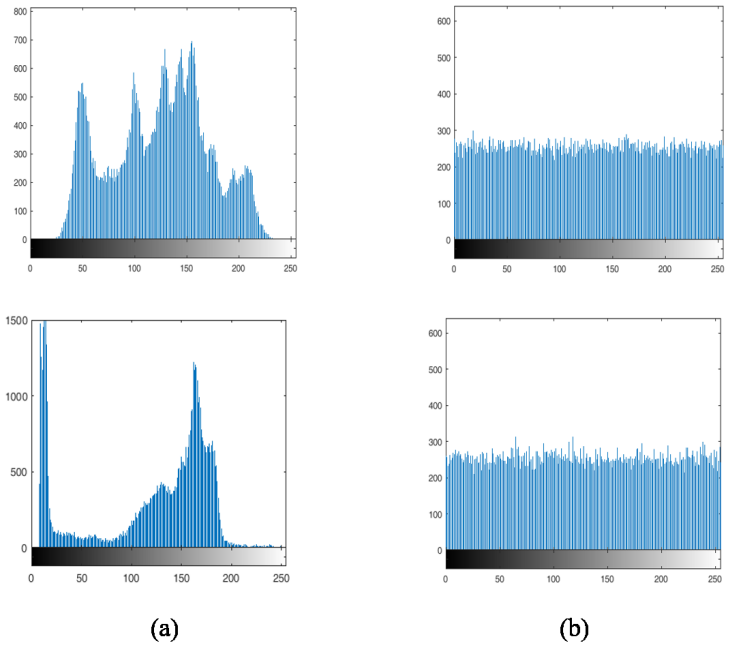

Figure 11 Histogram of (a) plain images, (b) cipher images

$$r_{xy} = \frac{Con(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{12}$$

$$Con(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{13}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{14}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \tag{15}$$

where $E(x)$ and $D(x)$ are the expectation and variance of the variable x, respectively. The correlation value ranges between -1 and 1. A correlation close to -1 or 1 indicates a high correlation between the pixels, while a correlation close to 0 indicates a weak correlation between the pixels. By randomly picking up 5000 pairs of adjacent pixels from plain images and their corresponding encrypted images in the horizontal, vertical, and diagonal directions, the pixel distributions are shown in Figure 12. Moreover, the correlation coefficients are listed in Table 3.

From Figure 12 and Table 3, we can see that most pixels in the plain image lie along the same line, indicating a strong relationship between adjacent pixels. On the other hand, in the encrypted image, the pixels are randomly distributed in space, indicating a weak correlation between adjacent pixels. Moreover, the coefficients of the encrypted images are close to zero, indicating a lower correlation between adjacent pixels in these images.

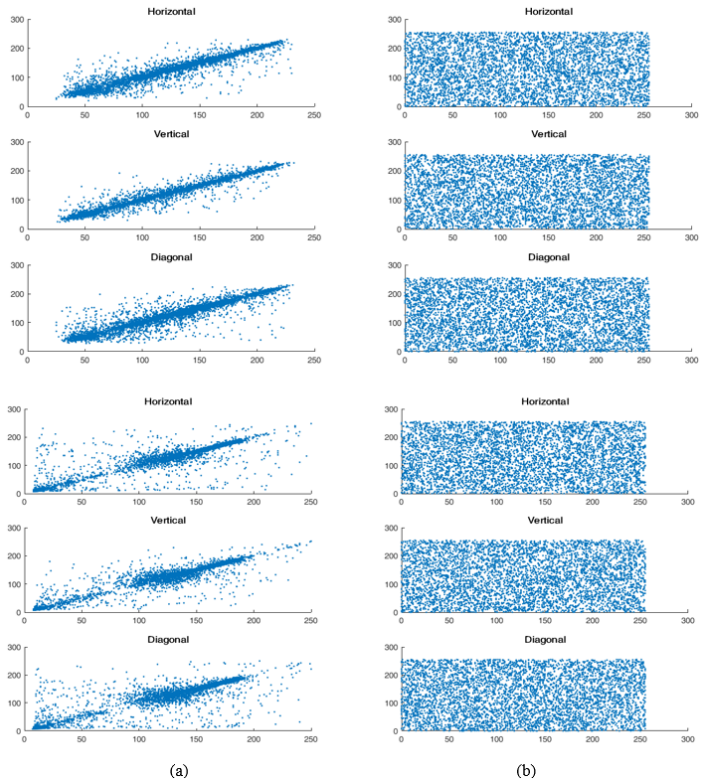

Figure 12 Correlation between pixels in horizontal, vertical, and diagonal directions for (a) Plain image, (b) encrypted image

Table 3 Correlation coefficients of different images

Images	Plain Image			Cipher Image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	0.9412	0.9699	0.9155	−0.0069	−0.0007	−0.0040
Cameraman	0.9335	0.9592	0.9087	0.0016	0.0048	−0.0034

4.3.2 Information Entropy

In grayscale images, the pixels are encoded with 8 bits, which means that the ideal entropy value for images is 8. The entropy values for different plain images and their corresponding encrypted images are given in Table 4.

Table 4 Information entropy of different images

Images	Plain Image Entropy	Encrypted Image Entropy	% For Ideal Value
Lena	7.4442	7.9977	99.971%
Cameraman	7.2972	7.9967	99.959%

From Table 4, we can see that the entropy values are close to the ideal value (8), which means that the encrypted images can resist entropy attacks.

5 Conclusion

This paper introduces a new pseudo-random generator based on two chaotic systems: the logistic map and a 7D hyperchaotic system. The pseudo-random sequence is obtained using two controllers: the first controller is the logistic map, and the second controller is a variable from the 7D hyperchaotic system. The main advantage of the pseudo-random is its ability to generate bits according to the needs of the application without a limit on the number it can produce. The values of the sequence obtained by the logistic map are modified to select one variable from the seven variables. The results show that the seven variables are used approximately with equal frequency. Moreover, the selected values are modified to choose one variable from the remaining six variables of the 7D hyperchaotic system to fill the pseudo-random sequence. The results show that these six variables are used approximately with the same frequency in filling the pseudo-random sequence. The simulation results and performance analyses show that our pseudo-random generator has good random behavior and passes several NIST tests, including entropy, frequency, and distribution of values. Therefore, the proposed pseudo-random generator can be effectively used in different fields, such as cryptography.

References

[1] James F. A review of pseudorandom number generators. Computer Physics Communications, 1990, 60(3): 329–344. DOI: 10.1016/0010-4655(90)90032-V.

[2] Deng L Y, Bowman D. Developments in pseudo-random number generators. Wiley Interdisciplinary Reviews: Computational Statistics, 2017, 9(5): e1404. DOI: 10.1002/wics.1404.

[3] James F, Moneta L. Review of high-quality random number generators. Computing and Software for Big Science, 2020, 4(1): 1-12. DOI: 10.1007/s41781-019-0034-3.

[4] Sathya K, Premalatha J, Rajasekar V. Investigation of strength and security of pseudo random number generators. IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, 1055(1): 012076. DOI: 10.1088/1757-899X/1055/1/012076.

[5] Naik R B, Singh U. A review on applications of chaotic maps in pseudo-random number generators and encryption. Annals of Data Science, 2024, 11(1): 25–50. DOI: 10.1007/s40745-021-00364-7.

[6] Bhattacharjee K, Das S. A search for good pseudo-random number generators: Survey and empirical studies. Computer Science Review, 2022, 45: 100471. DOI: 10.1016/j.cosrev.2022.100471.

[7] Malaviya P, Sarvaiya V, Shah A, et al. A comprehensive review on space solar power satellite: An idiosyncratic approach. Environmental Science and Pollution Research, 2022, 29(28): 42476–42492. DOI: 10.1007/s11356-022-19560-w.

[8] Panchal U H, Srivastava R. A comprehensive survey on digital image watermarking techniques. 2015 Fifth International Conference on Communication Systems and Network Technologies. IEEE, 2015: 591–595. DOI: 10.1109/CSNT.2015.165.

[9] Jumaa N K. Digital image encryption using AES and random number generator. Iraqi Journal of Electrical and Electronic Engineering, 2018, 14(1): 80–89. DOI: 10.33762/eeej.2018.144343.

[10] Wang F, Huang Z, Zhou Y. A new method for m-sequence and gold-sequence generator polynomial esti-

mation. 2007 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications. IEEE, 2007: 1039–1044. DOI: 10.1109/MAPE.2007.4393445.

[11] Zhang X Y. Analysis of m-sequence and gold-sequence in CDMA system. 2011 IEEE 3rd International Conference on Communication Software and Networks. IEEE, 2011: 466–468. DOI: 10.1109/ICCSN.2011.6014765.

[12] Alvarez G, Li S. Some basic cryptographic requirements for chaos-based cryptosystems. International Journal of Bifurcation and Chaos, 2006, 16(8): 2129–2151. DOI: 10.1142/S0218127406015970.

[13] Yu W, Wang J, Wang J, et al. Design of a new seven-dimensional hyperchaotic circuit and its application in secure communication. IEEE Access, 2019, 7: 125586–125608. DOI: 10.1109/ACCESS.2019.2935751.

[14] Abdullah H A, Abdullah H N, Mahmoud Al-Jawher W A. A hybrid chaotic map for communication security applications. International Journal of Communication Systems, 2020, 33(4): e4236. DOI: 10.1002/dac.4236.

[15] Vaidyanathan S, Sambas A, Azar A T, et al. A new 5-D hyperchaotic four-wing system with multistability and hidden attractor, its backstepping control, and circuit simulation. Backstepping Control of Nonlinear Dynamical Systems. Academic Press, 2021: 115–138. DOI: 10.1016/B978-0-12-817582-8.00013-1.

[16] Midoun M A, Wang X, Talhaoui M Z. A sensitive dynamic mutual encryption system based on a new 1D chaotic map. Optics and Lasers in Engineering, 2021, 139: 106485. DOI: 10.1016/j.optlaseng.2020.106485.

[17] Brahim A H, Pacha A A, Said N H. Image encryption based on compressive sensing and chaos systems. Optics & Laser Technology, 2020, 132: 106489. DOI: 10.1016/j.optlastec.2020.106489.

[18] Knuth D E. The art of computer programming. 3rd ed. Reading, Mass: Addison-Wesley, 1997.

[19] Hana A P, Naima H S, Adda A P. Image encryption by using a specific adaptation of Lehmer's algorithm. Journal of Discrete Mathematical Sciences and Cryptography, 2020, 23(5): 949–971. DOI: 10.1080/09720529.2019.1652402.

[20] Hana A P, Naima H S, Adda A P, et al. Cryptographic adaptation of the middle square generator. International Journal of Electrical and Computer Engineering, 2019, 9(6): 5615. DOI: 10.11591/ijece.v9i6.pp5615-5627.

[21] Huang X, Liu L, Li X, et al. A new pseudorandom bit generator based on mixing three-dimensional chen chaotic system with a chaotic tactics. Complexity, 2019, 2019(1): 6567198. DOI: 10.1155/2019/6567198.

[22] Alhadawi H S, Zolkipli M F, Ismail S M, et al. Designing a pseudorandom bit generator based on LFSRs and a discrete chaotic map. Cryptologia, 2019, 43(3): 190–211. DOI: 10.1080/01611194.2018.1548390.

[23] Huang X, Liu L, Li X, et al. A new two-dimensional mutual coupled logistic map and its application for pseudorandom number generator. Mathematical Problems in Engineering, 2019, 2019(1): 7685359. DOI: 10.1155/2019/7685359.

[24] Hadj Brahim A, Ali Pacha A, Hadj Said N. A new image compression-encryption scheme based on compressive sensing & classical AES algorithm. Multimedia Tools and Applications, 2023, 82(27): 42087–42117. DOI: 10.1007/s11042-023-15171-w.

[25] Brahim A H, Pacha A A, Said N H. A new image encryption scheme based on a hyperchaotic system & multi specific S-boxes. Information Security Journal: A Global Perspective, 2023, 32(2): 59–75. DOI: 10.1080/19393555.2021.1943572.