Ibn Al-Haitham Journal for Pure and Applied Sciences
Journal homepage: jih.uobaghdad.edu.iq
PISSN: 1609-4042, EISSN: 2521-3407
IHJPAS. 2025, 38(3)

Using a 3D Chaotic Dynamic System as a Random Key Generator for Image Steganography

Mohammed Abod Hussein[1*], Saad Al-Momen[2]
[1,2]Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq.
*Corresponding Author.

Received: 25 June 2023 Accepted: 1 October 2023 Published: 20 July 2025
doi.org/10.30526/38.3.3619

Abstract

In today's digital era, the importance of securing information has reached critical levels. Steganography is one of the methods used for this purpose by hiding sensitive data within other files. This study introduces an approach utilizing a chaotic dynamic system as a random key generator, governing both the selection of hiding locations within an image and the amount of data concealed in each location. The security of the steganography approach is considerably improved by using this random procedure. A 3D dynamic system with nine parameters influencing its behavior was carefully chosen. For each parameter, suitable interval values were determined to guarantee the system's chaotic behavior. Analysis of chaotic performance is given using the Lyapunov exponents, fractal dimension, and bifurcation diagrams. Furthermore, an algorithm is suggested to generate a random binary key, serving as the controller for the embedding process. And the randomness of the generated key was checked. Moreover, this paper introduces a technique that utilizes the generated random key to govern both the embedding process in the spatial domain and the frequency domain. The results of this study are promising and its potential applications can be extended to various fields that require discreet communication and robust data protection.

Keywords: Random Key Generator, Chaotic Dynamic System, Information Security, Spatial Domain, Frequency Domain, Chaos, Lyapunov Exponents, Fractal Dimension, Bifurcation.

1. Introduction

With the advent of digital communication methods in the modern era, ensuring information security has become crucial for information and communication technology. Two significant techniques in this field are cryptography and steganography (1). Cryptography is employed to safeguard the confidentiality of communication, preserving the message's content and nature. Occasionally, it is necessary to exchange information with a hidden secret message that no one knows about its existence other than the sender and the recipient; this implementation technique is steganography which is refers to the practice of concealing sensitive information within another file known as a cover (2). In other words, Steganography is the art of concealing a confidential message (file, data, image, video, audio) inside another message (file, data, image, video, audio) (3). In this particular study, text was

employed as the secret message to be concealed, while an image served as the cover to hide this message within it (4). This paper presents a steganography technique based on the utilization of a chaotic method. In this approach, a chaotic system is employed to generate a series of random numbers, which are then utilized to modify the cover data. The resulting sequence of numbers, generated through the chaotic method, exhibits a high level of unpredictability, thereby making it challenging for an attacker to detect the presence of the hidden message (5). Therefore, the chaotic sequence is used as a key in the hiding process, enhancing the security of the steganographic method (6).

2. Related Works

There is a considerable number of researchs that explore the fusion of chaos theory and steganography. Noteworthy investigations in this realm encompass the following studies. In (7) introduced a novel steganography technique that leverages a chaotic logistic map. This technique offers simplicity, speed, efficiency, and high imperceptibility. The chaotic logistic map is utilized for encoding and embedding using Discrete Cosine Transform (DCT), enhancing security and incomprehensibility. The sensitivity of the initial state in the logistic map generates diverse sequences with distinct initial values. In 2017 (8) discussed several alternative schemes for hiding the grayscale image in a color cover image. They found that the chaotic, one-dimensional maps took less time to use to select the pixels in which the secret message would be embedded. However, 2D chaotic maps have more parameters to adjust which makes them more robust against attacks if a secret message is suspected. As for (9) proposed a safe LSB technique for hiding images using the concept of a nonlinear dynamic system (chaos). Where image files hosted in the spatial domain were used to hide the presence of sensitive information regardless of its format. An analysis of the performance of the proposed technique after comparison with the 3-3-2 LSB technique was very encouraging. In 2019, (10) introduced a novel algorithm for grayscale image processing. The algorithm incorporated Mandelbrot's chaotic logistic map and eclipse clusters, along with Huffman coding compression. The results of this study demonstrated that the proposed algorithm outperforms existing methods in terms of both image quality (as measured by the perception deficiency quality index) and data hiding capabilities. Moreover, the algorithm enhances security measures. In 2021, (11) introduced a technique that employs a 3D chaotic cat map to conceal images. The resulting chaotic map is utilized to determine the pixel locations and color properties, thereby facilitating the embedding of a confidential message into the original image.

This research paper introduces a significant advancement in the field of chaos-based steganography techniques. The proposed approach enables the hiding of a secret message within an image by utilizing a randomly generated key derived from a three-dimensional chaotic dynamic system. The system's sensitivity to initial conditions ensures a strong and secure fusion process. Furthermore, the proposed method used both spatial and transformation domains, making it more versatile and increasing its overall effectiveness.

3. Materials and Methods

In order to obtain a random binary key for the purpose of randomly embedding data within images, the chaotic nature of a dynamic system can be leveraged. This involves identifying a function that converts the system's solution from continuous values to a binary string.

3.1. 3D Chaotic Dynamic System

Maghool and Naji (12), in their investigation the impact of fear on a three-species food chain, they suggest the following 3D dynamic system:

$$\frac{dx}{dt} = x\left[\frac{1}{1+u_1 y} - x - \frac{y}{1+u_2 x^2} * \left(\frac{1}{1+u_3 z}\right)\right]$$
$$\frac{dy}{dt} = y\left[\frac{u_4 x}{1+u_2 x^2}\left(\frac{1}{1+u_3 z}\right) - \frac{u_5 z}{1+u_6 y^2} - u_7\right] \quad (1)$$
$$\frac{dz}{dt} = z\left[\frac{u_8 y}{1+u_6 y^2} - u_9\right]$$

System (1) is controlled by nine parameters, each of which plays a crucial role in determining its behavior. For Example, Starting from the initial point $(0.8, 0.7, 0.6)$; system (1) had a rich chaotic behavior with the following parameter's values

$$U = [0.25, 2, 0.25, 1, 1, 2, 0.1, 1, 0.1] \quad (2)$$

Figure (1 a) shows the strange attractor for the 3D dynamic system, while **Figure** (**1b-d)** show the projection in the xy-, xz-, and yz-plane, respectively.

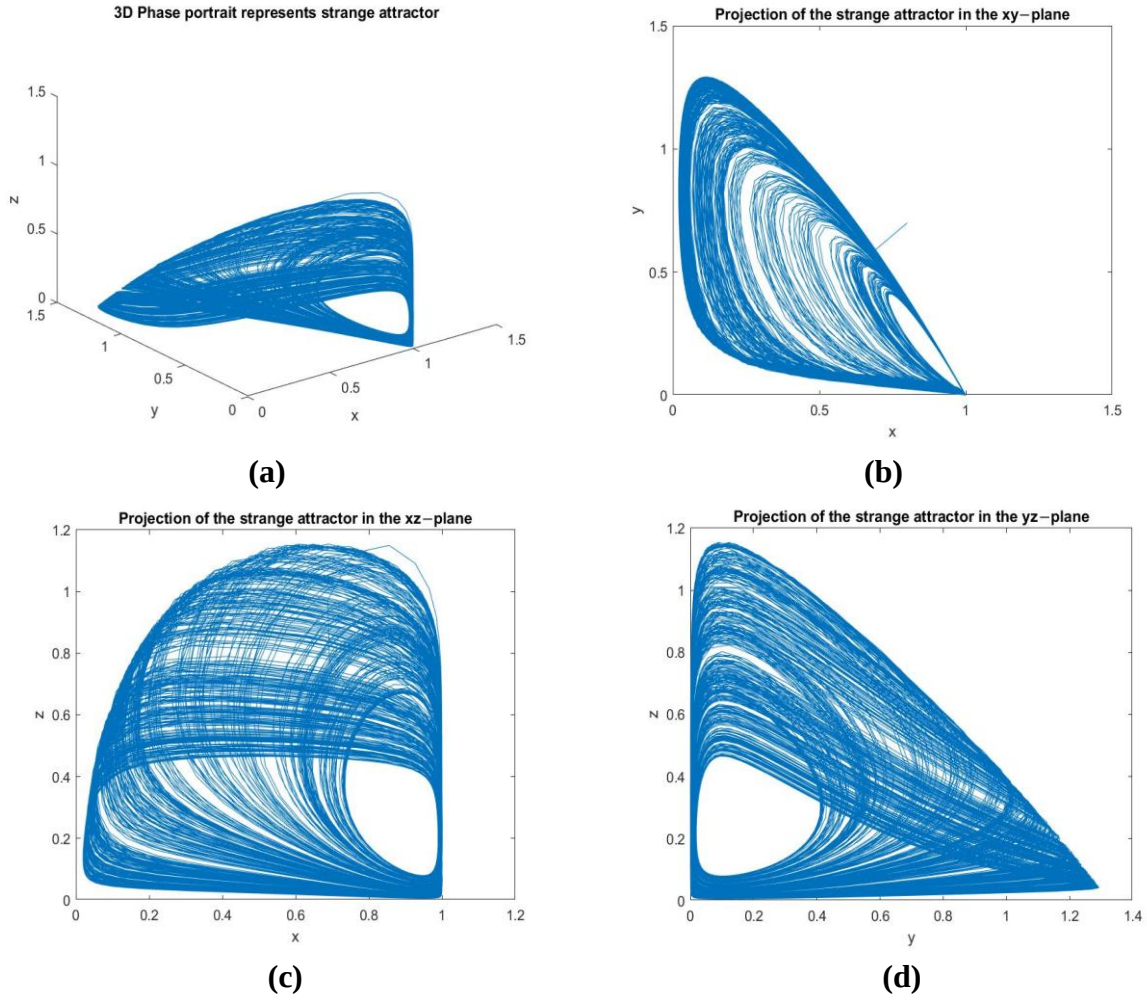


Figure 1. The trajectories of system (1) using parameters values (2).

3.2. Investigating the Chaotic Nature of the System

The calculation of the Lyapunov exponent provides a quantitative approach to measure the sensitive dependence on initial conditions (SDIC) in accordance with nonlinear dynamical theory. It represents the average rate of divergence or convergence between two neighboring trajectories. The three Lyapunov exponents with parameter's values in (2) are:

$L_1 = 0.008741$, $L_2 = -0.00018$, and $L_3 = -1.103348$

The positive value of the largest Lyapunov exponent demonstrates the presence of chaotic behavior for this system.

One of the typical features of chaos is the Fractal dimension which determined through Lyapunov exponents by Kaplan-Yorke dimension, where D_{KY} can be form as (13):

$D_{KY} = j + \frac{\sum_{i=1}^{j} L_i}{|L_{j+1}|}$ where j is the largest integer for which $\sum_{i=1}^{j} L_i > 0$

In this study, for the suggested 3D dynamic system, the value of $j = 2$, hence

$$D_{KY} = 2 + \frac{\sum_{i=1}^{2} L_i}{|L_3|} = 2 + \frac{0.008741 - 0.00018}{1.103348} = 2.00776$$

The fractional Lyapunov dimension shows that the system exhibits a fractal nature. Due to this fractal nature, the proposed system displays non-periodic orbits, and the trajectories of neighboring points diverge. Thus, it can be concluded that the proposed system is in a state of chaos (14).

The chaotic behavior of system (1) is apparent, not only at the specified values (2), but also within a range of values for each parameter. The vectors LB and UB represent the lower and upper bounds of this parameter range, respectively. By keeping the remaining parameter values constant as specified in (2), the system exhibits chaotic behavior when the parameters fall within the corresponding range.

$LB = [0, 1.516, 0.02, 0.7866, 0.2, 1.58, 0.066, 0.2875, 0.0536]$ (3)

$UB = [0.19, 2, 0.8, 1, 1.5, 2, 0.15, 0.734, 0.266]$ (4)

In **Figures 2-10**, the bifurcation diagrams depict the variation of Max(x), Max(y), and Max(z) with respect to the parameter values u_i, where $i = 1,2, \ldots 9$. The parameter values u_i are constrained within the intervals $[LB_i, UB_i]$, as indicated.

3.3. The Proposed Algorithm to Generate a Random Binary Key

The generation of a random binary key involves several steps. First, a keyword is selected and converted into its corresponding ASCII code. This ASCII code is then transformed into a binary sequence. From this sequence, the initial 8 bits are extracted and converted into a decimal number, referred to as d_1, which should fall within the range $[0,255]$.

Using d_1, the parameter to be modified is determined. The parameter number, denoted as i, is calculated through the equation:

$i = (d1 \ mod \ 9) \ + \ 1$ (5)

where $1 \leq i \leq 9$.

Additionally, the bits 9 to 12 of the binary sequence are converted into another decimal number, d_2, which should be in the range $[0,15]$. The range of parameter $\#i$ is divided into d_2 subintervals of length h, where h is calculated using the equation

$h = \frac{UB_i - LB_i}{d_2}$ (6)

The vectors UB and LB respectively contain the upper and lower bounds of all parameter values. Furthermore, bits 13 to 16 of the binary sequence are converted into a decimal number, d_3, which falls within $[0,15]$. To determine the serialization of the value within the specified range of the parameter, j is calculated as

$j = d_3 \ mod \ (d_2 + 1)$ (7)

with $0 \leq j \leq d_2$. The value of parameter $\#i$ is then modified according to

$U_i = LB_i + j * h$ (8)

Utilizing the vector U that contains the initial specified values of all the nine parameters.

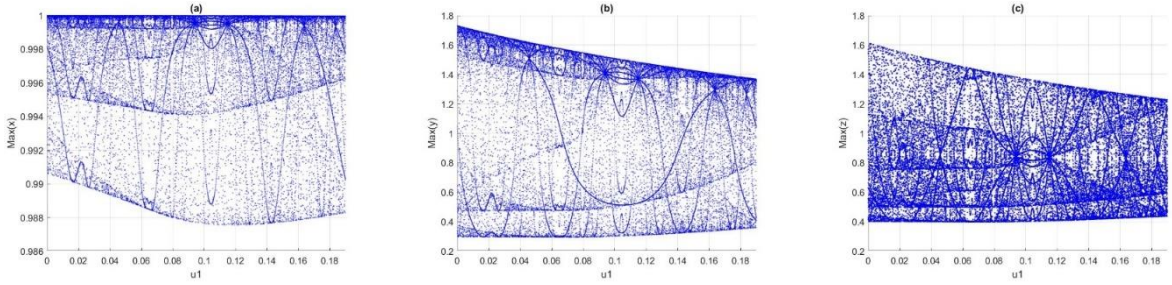


Figure 2. The bifurcation diagrams for the system when $u_1 \in [0,0.19]$ vs. (a) Max (x) (b) Max (y) and (c) Max (z).

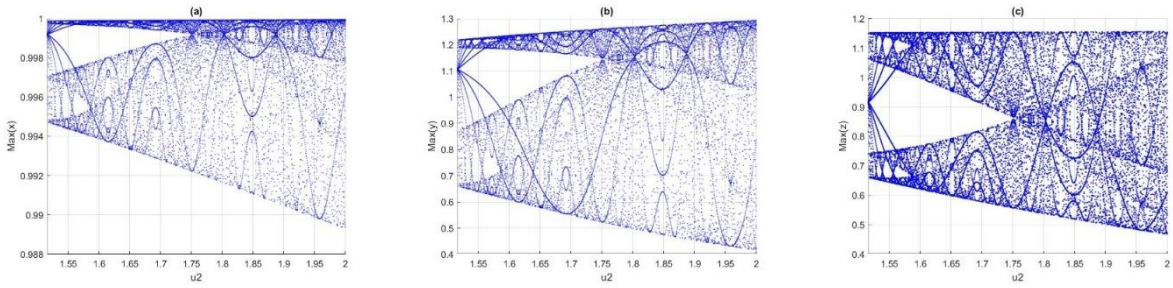


Figure 3. The bifurcation diagrams for the system when $u_2 \in [1.516,2]$ vs. (a) Max (x) (b) Max (y) and (c) Max(z).

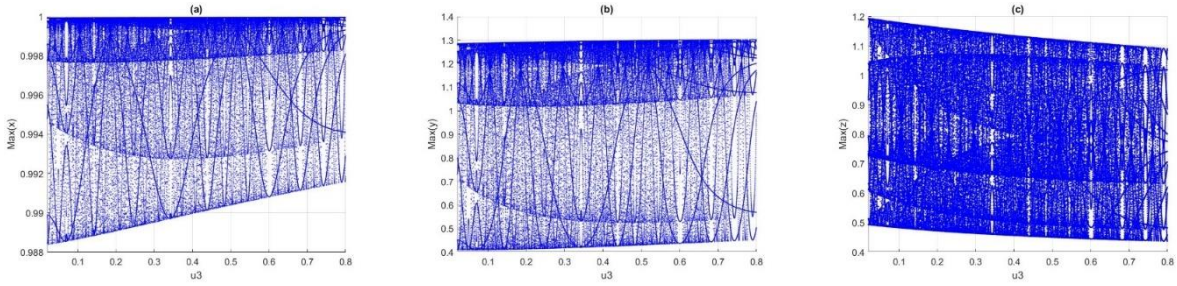


Figure 4. The bifurcation diagrams for the system when $u_3 \in [0.02,0.8]$ vs. (a) Max (x) (b) Max (y) and (c) Max(z).

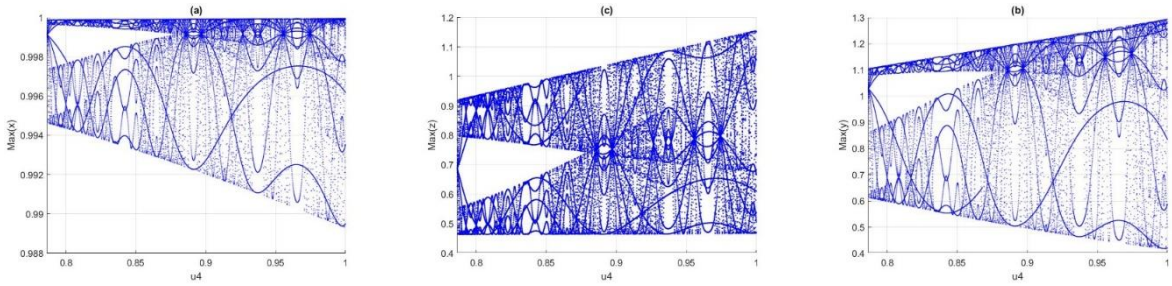


Figure 5. The bifurcation diagrams for the system when $u_4 \in [0.7866,1]$ vs.(a) Max (x) (b) Max (y) and (c) Max(z).

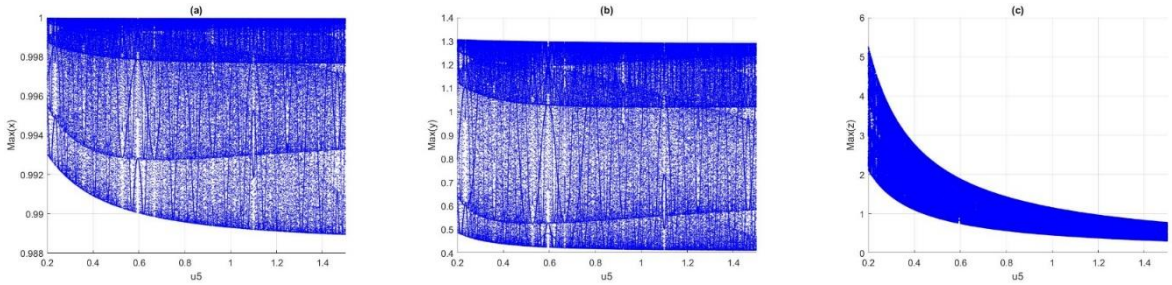


Figure 6. The bifurcation diagrams for the system when $u_5 \in [0.2,1.5]$ vs. (a) Max (x) (b) Max (y) and (c) Max(z).

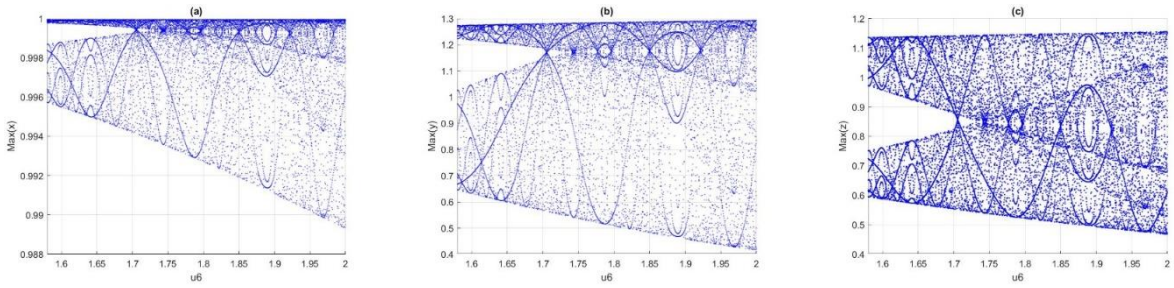


Figure 7. The bifurcation diagrams for the system when $u_6 \in [1.58,2]$ vs. (a) Max (x) (b) Max (y) and (c) Max (z).

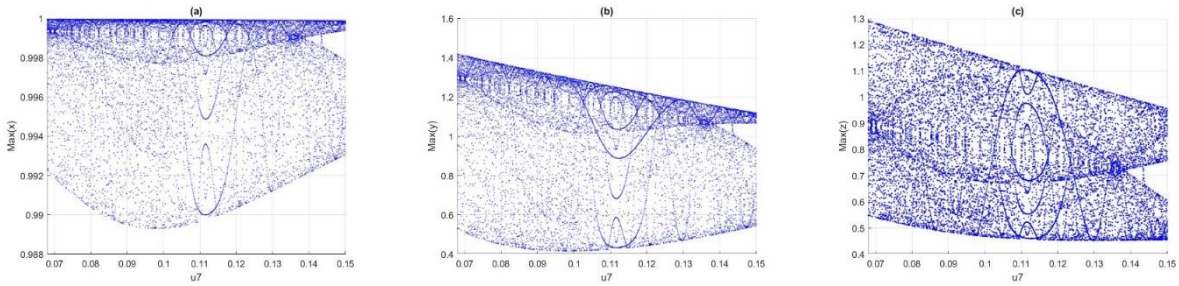


Figure 8. The bifurcation diagrams for the system when $u_7 \in [0.066,0.15]$ vs. (a) Max(x) (b) Max (y) and (c) Max (z).

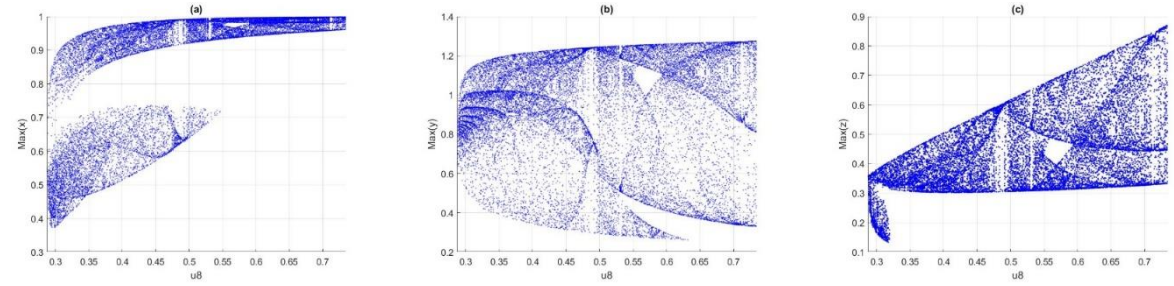


Figure 9. The bifurcation diagrams for the system when $u_8 \in [0.2875,0.734]$ vs.(a) Max (x) (b) Max (y) and (c) Max (z).

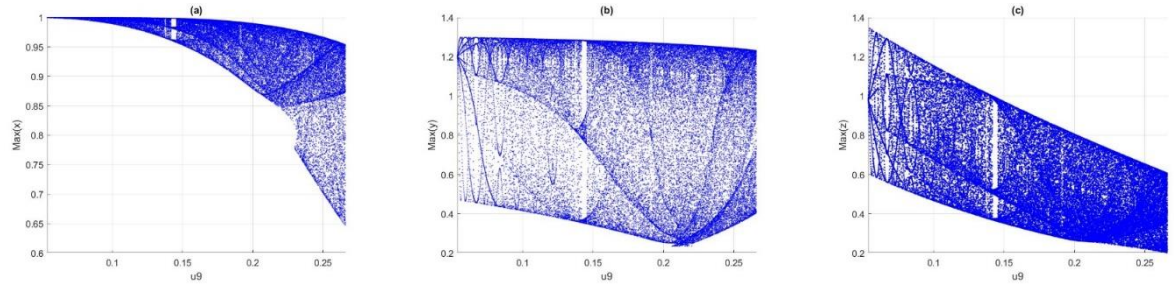


Figure 10. The bifurcation diagrams for the system when $u_9 \in [0.0536,0.266]$ vs. (a) Max (x) (b) Max (y) and (c) Max (z)

The new parameter values are employed to solve the 3D chaotic dynamic system, resulting in the generation of three vectors: X, Y, and Z. To address any discrepancies in these vectors, a normalization process takes place. Each vector (X, Y, and Z) is normalized using the equations

$$XN = (X - X_{min})/(X_{max} - X_{min}),$$
$$YN = (Y - Y_{min})/(Y_{max} - Y_{min}), \quad (9)$$
$$ZN = (Z - Z_{min})/(Z_{max} - Z_{min}).$$

where X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, and Z_{max} represent the minimum and maximum values of each respective vector.

After the process of normalization, the authentic numeric entities of XN, YN, and ZN and are transformed into their corresponding binary values. This conversion is achieved using the equations

$$Xb = floor(XN * 10^4)\ mod\ 2,$$
$$Yb = floor(YN * 10^4)\ mod\ 2, \quad (10)$$
$$Zb = floor(ZN * 10^4)\ mod\ 2.$$

Finally, a sequence op is calculated by applying the equation

$$op = (Xb \oplus Yb) \oplus Zb \quad (11)$$

If the resulting op sequence successfully passes at least 12 NIST tests, it is deemed a random binary key. However, if the sequence does not fulfill the criteria, an alternative keyword is chosen, and the entire procedure is restarted anew.

Algorithm 1 details the methodology for generating an unpredictable binary key, whereas **Figure (11).** presents a graphical representation of the entire key formation procedure.

Algorithm 1: Random Binary Key Generator

Input: Keyword (string).

Output: Random Binary Sequence op.

Step 1: Convert the keyword to its corresponding ASCII code and store it as an integer value ***keyword_ascii***.

Step 2: Convert ***keyword_ascii*** to a binary sequence ***keyword_binary***.

Step 3: Extract the initial 8 bits from ***keyword_binary*** and convert them to a decimal number $\boldsymbol{d_1}$.

Step 4: Calculate the parameter number $\boldsymbol{i}$ using the equation: $\boldsymbol{i = (d_1\ mod\ 9) + 1}$.

Step 5: Extract bits 9 to 12 from ***keyword_binary*** and convert them to a decimal number $\boldsymbol{d_2}$.

Step 6: Calculate the parameter range length $\boldsymbol{h}$ using the equation: $\boldsymbol{h = (UB_i - LB_i)/d_2}$.

Step 7: Extract bits 13 to 16 from ***keyword_binary*** and convert them to a decimal number $\boldsymbol{d_3}$.

Step 8: Calculate the serialization index $\boldsymbol{j}$ using the equation: $\boldsymbol{j = d_3\ mod\ (d_2 + 1)}$.

Step 9: Modify the value of parameter $\boldsymbol{\#i}$ using the equation: $\boldsymbol{U_i = LB_i + j * h}$.

Step 10: Solve the 3D chaotic dynamic system using the modified parameter values, resulting in three vectors: $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$.

Step 11: Normalize each vector ($\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$) using the equations:

$$\boldsymbol{XN = (X - X_{min})/(X_{max} - X_{min})}$$
$$\boldsymbol{YN = (Y - Y_{min})/(Y_{max} - Y_{min})}$$
$$\boldsymbol{ZN = (Z - Z_{min})/(Z_{max} - Z_{min})}$$

Step 12: Convert the normalized values $\boldsymbol{XN}$, $\boldsymbol{YN}$, and $\boldsymbol{ZN}$ to binary values using the equations:

$$\boldsymbol{Xb = floor(XN * 10^4)mod\ 2}$$
$$\boldsymbol{Yb = floor(YN * 10^4)\ mod\ 2}$$
$$\boldsymbol{Zb = floor(ZN * 10^4)\ mod\ 2}$$

Step 13: Calculate the sequence ***op*** using the equation: $\boldsymbol{op = (Xb \oplus Yb) \oplus Zb}$.

Step 14: Perform at least 12 NIST tests on the ***op*** sequence.

- If the sequence passes all tests, return ***op*** as the random binary key.
- If the sequence fails any test, go back to step 1 with a new keyword.

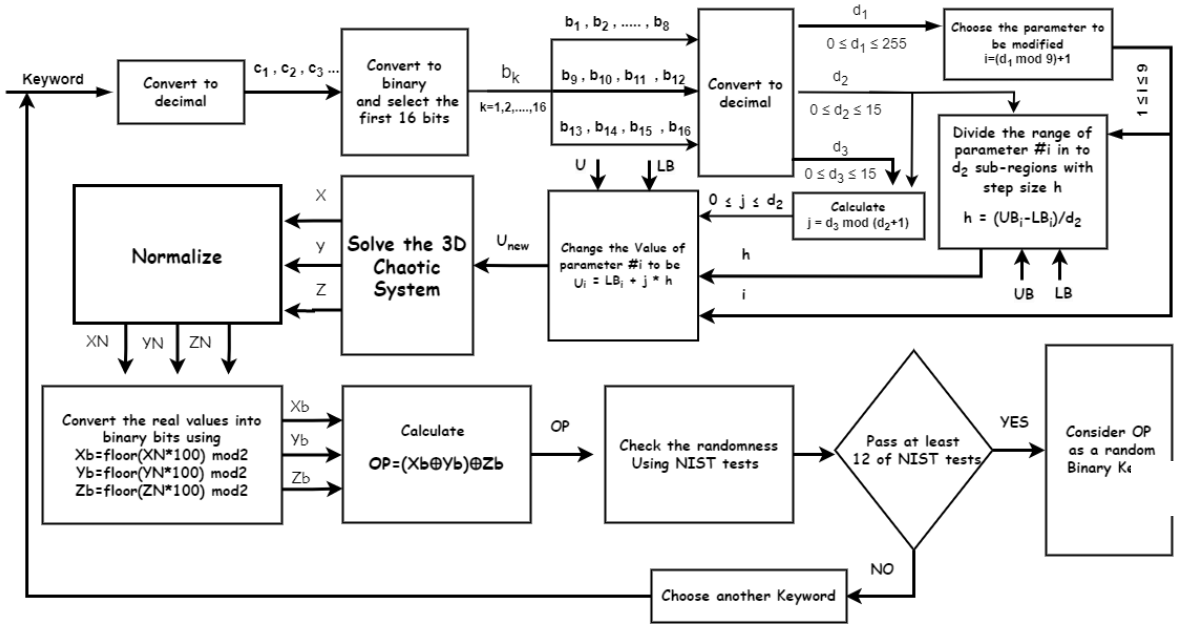


Figure 11. Block diagram of random binary key generator.

4.Experimental Results and Analysis

This section demonstrates the performance of the proposed random binary key generation algorithm using system (1) with the initial point (0.8, 0.7, 0.6) and the parameter values specified in Equation (12):

$$U = [0.25, 5, 0.25, 1, 1, 2, 0.1, 1, 0.1] \quad (12)$$

Considering the keyword "Student," its corresponding ASCII code is

$$[83, 116, 117, 100, 101, 110, 116]$$

The binary representation for the keyword is:

$$'10100111110100111010111001001100101110111011101100'$$

Based on this information, the values of d_1, d_2, and d_3 are determined as 167, 13, and 3, respectively. These values lead to i being equal to 6, h being equal to 0.0323, and j being equal to 3. Consequently, the value of u_6 is defined as 1.6769, resulting in the following values for the nine parameters replace the values in (12) and the new set of parameter's values be:

$$U = [0.25, 5, 0.25, 1, 1, 1.6769, 0.1, 1, 0.1] \quad (13)$$

Figure (12 a) illustrates the strange attractor for the 3D dynamic system, while **Figure (12 b-d)** show the projection in the xy-, xz-, andyz-plane, respectively.

An illustration of the conversion process from real values of the chaotic system outputs to binary values is presented in **Table (1)**.

Table 1. Example of converting the real values to binary values

X	Y	Z	XN	YN	ZN	Xb	Yb	Zb	op
0.3496	1.1743	0.1159	0.3450	0.8185	0.1004	0	1	0	1
0.2474	1.2481	0.1487	0.2423	0.8700	0.1289	1	0	1	0
0.1577	1.2650	0.1898	0.1524	0.8817	0.1645	0	1	1	0
0.1005	1.2040	0.2429	0.0947	0.8392	0.2105	1	0	1	0
0.0714	1.0775	0.3143	0.0655	0.7510	0.2725	1	0	1	0
0.0611	0.9012	0.4133	0.0551	0.6281	0.3583	1	1	1	1
0.0658	0.6809	0.5496	0.0599	0.4746	0.4765	1	0	1	0
0.0920	0.4317	0.7153	0.0861	0.3008	0.6201	1	0	1	0
0.1604	0.2192	0.8464	0.1549	0.1528	0.7337	1	0	1	0
0.3035	0.0988	0.8858	0.2987	0.0688	0.7679	1	0	1	0

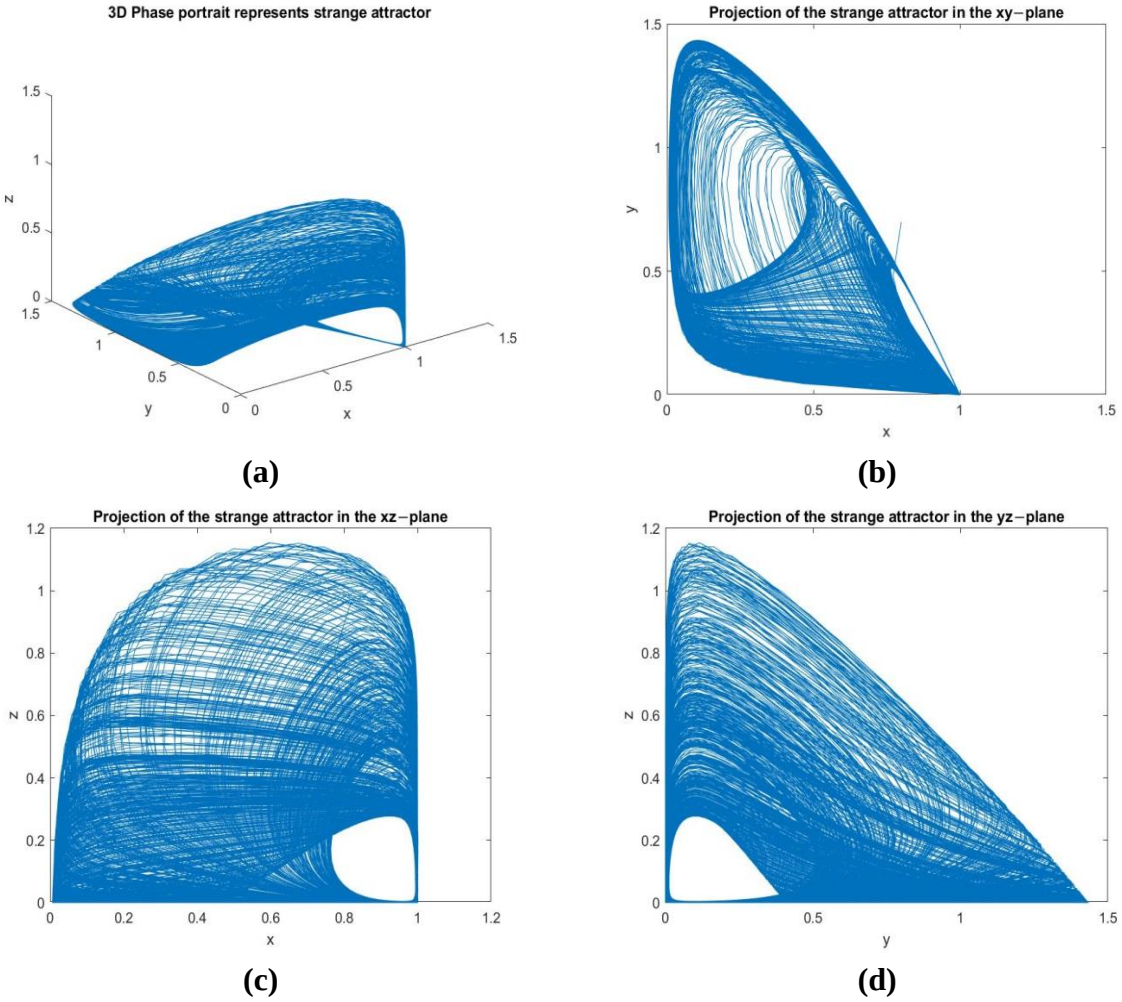


Figure 12. The trajectories of system (1) using parameters values (13).

To verify the statistical characteristics of the generated key, the SP800-22 test package, developed by the National Institute of Standards and Technology (NIST), is employed for detecting random performance (15). In this evaluation, a keystream sequence of 1,000,000 bits generated by the proposed keystream generator is subjected to the tests. The results of these tests are presented in **Table (2)**, which includes the test name, the corresponding P-value, and the test result.

Table 2. Evaluating key randomness

Test	P-value	Result
Frequency (Monobit)	0.9792	Success
Block Frequency	0.9792	Success
Runs	0.1822	Success
Longest Run of Ones	0.8138	Success
Binary Matrix Rank	0.5780	Success
DFT	0.6202	Success
Non Over Lapping Templates	0.4901	Success
Over Lapping Template	0.2873	Success
Universal Statistical	0.0803	Success
Serial Test	0.4194	Success
Approximate Entropy	0.4801	Success
Cumulative Sums (Forward)	0.9144	Success
Random Excursions Test	0.5954	Success
Random Excursions Variant	0.4553	Success
Linear Complexity	0.6541	Success

It is noteworthy that no deviation from a truly random sequence is observed, as indicated by all P-values exceeding the significance value $\alpha = 1\%$. Therefore, the results in the table demonstrate the high quality of randomness achieved by the proposed keystream generator.

Given that the binary sequence generated successfully passed all the randomness tests mentioned earlier, it becomes a viable option for employing as a key in various encryption techniques or steganography, as demonstrated in this study.

In order to evaluate the outcomes of employing this key to conceal data within an image using random techniques, we utilized the two algorithms proposed in our previous research (16). Where, the first algorithm operates within the Spatial domain, while the second algorithm conducts the hiding process in the Frequency domain. Performance evaluation and image quality assurance involve utilizing several commonly used metrics. Key assessments include the signal-to-noise ratio (PSNR), mean squared error (MSE), and normalized cross-correlation (NCC), which are considered crucial in this context (17,18).

Figure (13) showcases a collection of standard images, demonstrating their appearance before and after the inclusion of a hidden binary message. Additionally, the histograms for each image are presented. The final column of the figure showcases the randomization map, where red points represent the embedding of 1 bit per pixel, green points represent 2 bits per pixel, and blue points indicate 3 bits per pixel. Meanwhile, black points inwdicate skipped pixels.

Furthermore, **Table (3)** presents a comprehensive overview of the numerical values associated with the three performance metrics mentioned earlier: PSNR, MSE, and NCC.

Table 3. Performance metric in spatial domain

Cover Image	Image size	Payload (bits)	PSNR	MSE	NCC
Lena	512×512	345216	43.5242	2.8884	0.9994
Barbara	512×512	345216	43.5302	2.8844	0.9994
Baboon	512×512	345216	43.5200	2.8912	0.9992
Peppers	512×512	345216	43.5210	2.8906	0.9997
Goldhill	512×512	345216	43.5364	2.8803	0.9994
Cameraman	512×512	345216	43.5658	2.8609	0.9996
Average	512×512	345215	43.5392	2.8826	0.9996

Figure (14) depicts the representation of embedding in the frequency domain. **Table (4)** presents the numerical values for the three metrics. It is important to highlight that the quantization step employed the modified quantization table proposed by Li and Wang (19).

Table 4. Performance metric in frequency domain

Cover Image	Image size	Payload (bits)	PSNR	MSE	NCC
Lena	512×512	159744	42.3139	3.8167	0.9992
Barbara	512×512	159744	35.3070	19.1594	0.9958
Baboon	512×512	159744	30.3099	60.5462	0.9843
Peppers	512×512	159744	42.2903	3.8375	0.9994
Goldhill	512×512	159744	40.8418	5.3567	0.9989
Cameraman	512×512	159744	40.1989	6.2114	0.9992
Average	512×512	159744	38.5436	16.4880	0.9961

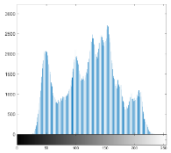
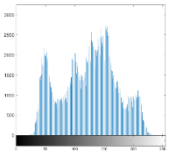
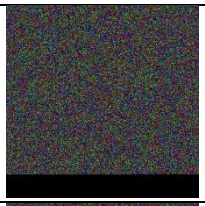
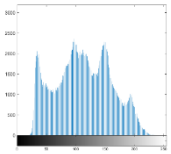
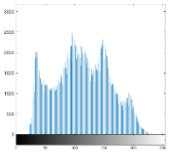
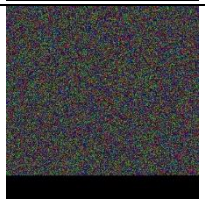
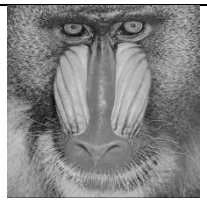
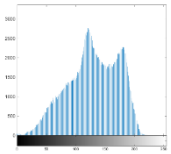
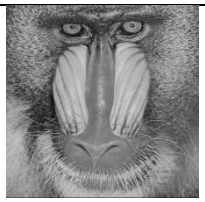
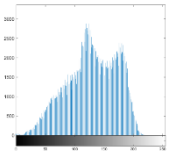
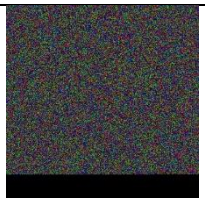
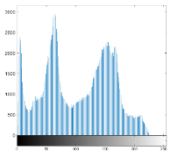
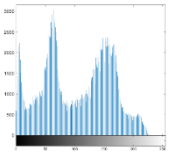
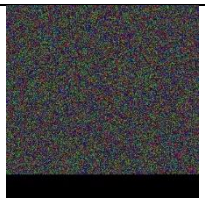
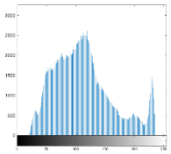
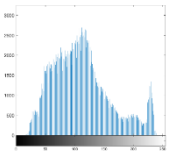
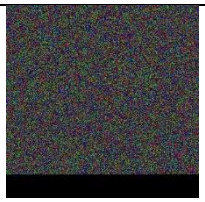
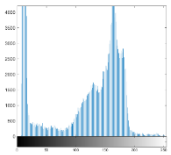
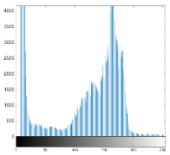
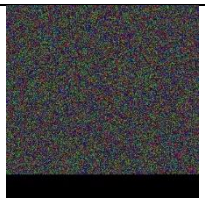
Name	Cover image	Cover image histogram	Stego image	Stego image histogram	Randomization map
Lena					
Barbara					
Baboon					
Peppers					
Goldhill					
Cameraman					

Figure 13. Images pre- and post-embedding of secret message in spatial domain

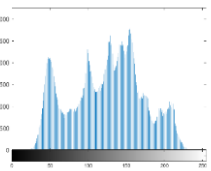
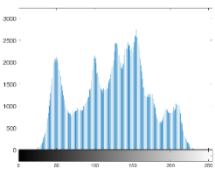
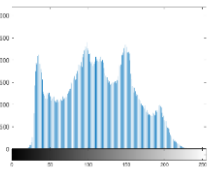
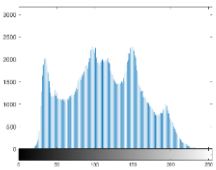
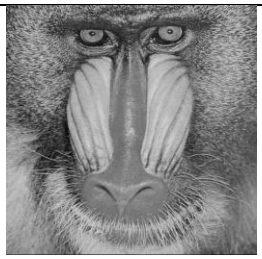
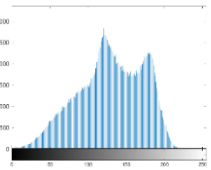
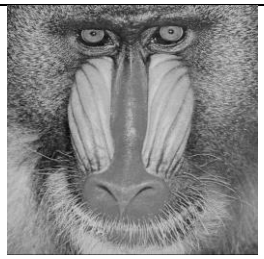
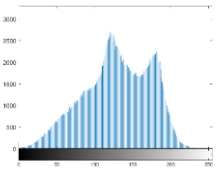
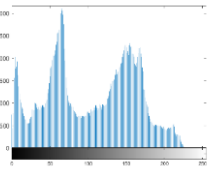
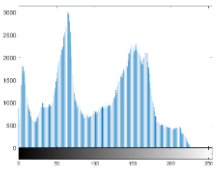
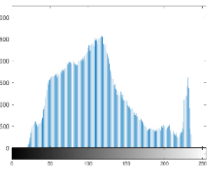
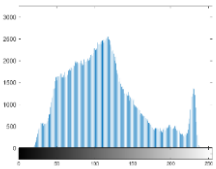
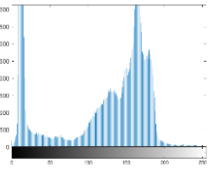
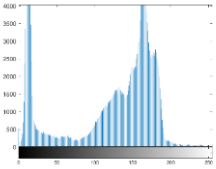
Name	Cover image	Cover image histogram	Stego image	Stego image histogram
Lena				
Barbara				
Baboon				
Peppers				
Goldhill				
Cameraman				

Figure 14. Images pre- and post-embedding of secret message in the frequency domain

In addition, the frequency domain embedding results were compared with two other methods: the well-known Jsteg method and the method proposed by Senthooran and Ranathunga (20). To ensure a fair comparison, the same images and payloads as those used in (20) were employed. The images used for this comparison are shown in **Figure (15)**.

Airplane **Man** **Barbara** **Lena** **Roar**

Figure 15. The collection of images utilized for conducting the comparison

The average results showcased in **Table (5**) demonstrate the superior performance of the proposed method compared to the other two methods. It achieves the lowest error and highest PSNR. Moreover, a visual depiction of the comparison, focusing on MSE and PSNR, is provided in **Figures (16** and **17)**, respectively.

Table 5. Comparative results for (20), and (16)

Cover Image	Payload	MSE			PSNR		
		Jsteg	(20)	(16)	Jsteg	(20)	(16)
Airplane	105536	31.4586	26.1262	2.6343	33.1534	37.2589	43.9242
Man	121714	50.345	34.801	1.6070	31.1112	36.7316	46.0708
Barbara	122952	65.5876	55.8191	18.7003	29.9626	33.0522	35.4123
Lena	95936	19.8841	20.0296	3.0215	35.1457	33.0567	43.3286
Roar	59624	9.8284	12.8471	0.7914	38.206	37.4658	49.1468
Average	101152	35.4207	29.9246	5.3509	35.5158	35.5130	43.5765

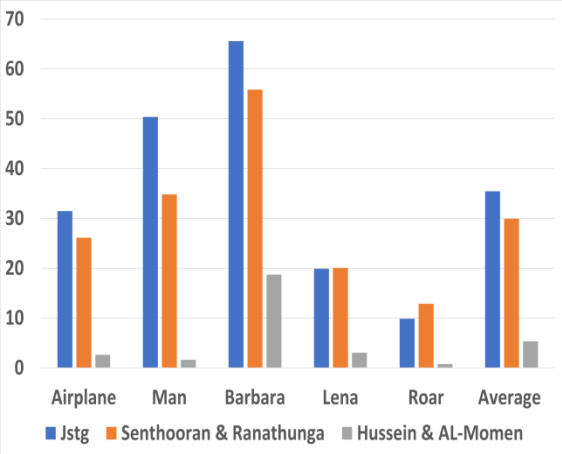


Figure 16. Comparison of MSE values

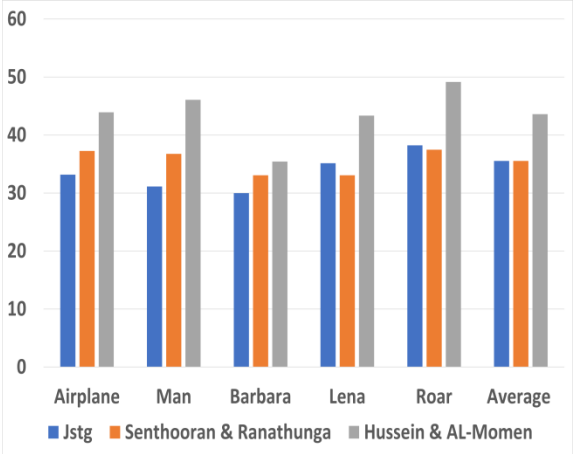


Figure 17. Comparison of PSNR values

5. Conclusion

This paper presents an approach to hide data within images using a 3D chaotic dynamic system as a random key generator. Lyapunov exponents, fractal dimensions, and bifurcation diagrams are used here to check the chaotic behavior of the selected system. This analysis insure suitability of this system to generate random keys. Rigorous randomness tests confirm the effectiveness of the generated binary key. By employing this key to govern the data embedding process in both spatial and frequency domains, the steganography method achieves heightened security. The study's outcomes demonstrate the promising potential of utilizing chaotic dynamical systems as random key generators for data hiding, as the generated keys exhibit a remarkable level of unpredictability, thwarting attackers' efforts to detect hidden messages.

Furthermore, by incorporating confusion during the data hiding process, discrete communication and robust data protection can be attained without compromising the appearance of the cover image. The proposed method offers simplicity, speed, efficiency, and

a high level of incomprehensibility while strengthening security. These findings pave the way for further exploration and application of chaotic dynamic systems in information security and various domains that demand confidential communication and robust data protection.

Acknowledgment

None.

Conflict of Interest

The authors declare that they have no conflicts of interest.

Funding

No funding.

References

1. Almuhammadi S, Al-Shaaby A. A survey on recent approaches combining cryptography and steganography. Comput Sci Inf Technol 2017;:63–74. https://doi.org/10.5121/csit.2017.70306
2. Abdulwahed MN, T MS, Mohd Rahim MS. Image spatial domain steganography: a study of performance evaluation parameters. IEEE Int Conf Syst Eng Technol 2019;:309–314. https://doi.org/10.1109/ICSEngT.2019.8906402
3. Tanna S. Codes, ciphers, steganography & secret messages. Answers 2000 Ltd, England; 2000.
4. Pramanik S, Raja SS. A secured image steganography using genetic algorithm. Adv Math Sci J 2020;9(7):4533–4541. https://doi.org/10.37418/amsj.9.7.22
5. Öztürk I, Kılıç R. A novel method for producing pseudo random numbers from differential equation-based chaotic systems. Nonlinear Dyn 2015;80(1–2):1147–1157. https://doi.org/10.1007/s11071-015-1932-5
6. Majeed MA, Sulaiman R, Shukur Z, Hasan MK. A review on text steganography techniques. Mathematics 2021;9(21):2829. https://doi.org/10.3390/math9212829
7. Melad SJ. A new technique based on chaotic steganography and encryption text in DCT domain for color image. J Eng Sci Technol 2013;8(5):508–520.
8. Elkamchouchi H, Salama WM, Abouelseoud Y. Data hiding in a digital cover image using chaotic maps and LSB technique. Int Conf Comput Eng Syst 2017;:198–203. https://doi.org/10.1109/ICCES.2017.8275302
9. K M. Analysis of HSI color model on chaos theory in spatial domain. Int J Anal Exp Modal Anal 2019;11:3162–3165.
10. Kasapbaşı MC. A new chaotic image steganography technique based on Huffman compression of Turkish texts and fractal encryption with post-quantum security. IEEE Access 2019;7:148495–510. https://doi.org/10.1109/ACCESS.2019.2946807
11. Hameed SM, Ali ZH, Al-Khafaji GK, Ahmed S. Chaos-based color image steganography method using 3D Cat Map. Iraqi J Sci 2021;62(9):3220–3227. https://doi.org/10.24996/ijs.2021.62.9.34
12. Maghool FH, Naji RK. Chaos in the three-species Sokol-Howell food chain system with fear. Commun Math Biol Neurosci 2022;2022:Article ID 14. https://doi.org/10.28919/cmbn/7056
13. Silva-Juarez A, Rodriguez-Gomez G, de la Fraga LG, Guillen-Fernandez O, Tlelo-Cuautle E. Optimizing the Kaplan–Yorke dimension of chaotic oscillators applying DE and PSO. Technol 2019;7(2):38. https://doi.org/10.3390/technologies7020038
14. Leonov GA, Kuznetsov NV, Korzhemanova NA, Kusakin DV. Lyapunov dimension formula for the global attractor of the Lorenz system. Commun Nonlinear Sci Numer Simul 2016;41:84–103. https://doi.org/10.1016/j.cnsns.2016.04.032
15. Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S. SP 800-22 Rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Natl Inst Stand Technol 2010. https://doi.org/10.6028/NIST.SP.800-22r1a

16. Hussein MA, Al-Momen S. Linear feedback shift registers-based randomization for image steganography. Iraqi J Sci 2023;64(8):4416–4428. https://doi.org/10.24996/ijs.2023.64.8.34
17. Alyousuf FQA, Din R, Qasim AJ. Analysis review on spatial and transform domain technique in digital steganography. Bull Electr Eng Inform 2020;9(2):573–581. https://doi.org/10.11591/eei.v9i2.2068
18. AbdelWahab OF, Hussein AI, Hamed HF, Kelash HM, Khalaf AA, Ali HM. Hiding data in images using steganography techniques with compression algorithms. TELKOMNIKA 2019;17(3):1168–1175. https://doi.org/10.12928/TELKOMNIKA.v17i3.12230
19. Li X, Wang J. A steganographic method based upon JPEG and particle swarm optimization algorithm. Inf Sci 2007;177(15):3099–3109. https://doi.org/10.1016/j.ins.2007.02.008
20. Senthooran V, Ranathunga L. DCT coefficient dependent quantization table modification steganographic algorithm. Int Conf Netw Soft Comput 2014;:192–196. http://doi.org/10.1109/CNSC.2014.6906644.