Ann. Acad. Rom. Sci.
Ser. Math. Appl.
ISSN 2066-6594 Vol. 18, No. 2/2026

CHAOS BASED INDISCERNIBLE IMAGE STEGANOGRAPHY SCHEME*

Borislav Stoyanov† Tsvetelina Ivanova‡ Dimitar Dobrev§
Plamen Ribarski¶

Communicated by V. Drăgan

DOI 10.56082/annalsarscimath.2026.2.131

Abstract

In the current digital era, information security is a top priority and outdated algorithms must be updated or replaced. This article presents a novel method that combines encryption and steganography to conceal secret text messages in color graphics. Using a chaotic pseudorandom generator, the position and arrangement of the picture pixels used for information embedding are chosen at pseudorandom. Another layer of protection is to encrypt the secret message before embedding it. This issue will deter attackers from looking for signs of steganography. The suggested indiscernible stegoalgorithm is being evaluated. We perform randomness tests, histograms, peak signal-to-noise ratio analysis, and other tests using conventional statistical and empirical methods. The novel results are presented and analyzed in the current article.

Keywords: chaotic functions, pseudorandom bytes, least significant bit, steganography.

MSC: 68P25, 11K45, 94A60.

*Accepted for publication on January 14, 2026

†borislav.stoyanov@shu.bg, Department of Computer Informatics, Faculty of Mathematics and Informatics, University of Shumen, Shumen, Bulgaria

‡ts.r.ivanova@shu.bg, Department of Computer Informatics, Faculty of Mathematics and Informatics, University of Shumen, Shumen, Bulgaria

§d.d.dobrev@shu.bg, Department of Computer Informatics, Faculty of Mathematics and Informatics, University of Shumen, Shumen, Bulgaria

¶p.ribarski@shu.bg, Department of Computer Informatics, Faculty of Mathematics and Informatics, University of Shumen, Shumen, Bulgaria

1 Introduction

Information security is crucial in today's digital environment. The necessity for efficient data protection techniques is become ever more important as Internet usage rises. The study and practice of hiding and analyzing information is called steganology. Steganography and steganalysis are its two main subfields. The process of concealing a hidden message inside another data file is known as steganography [6]. Data hiding serves as one of the best methods for cover up and protecting sensitive information [14]. In this work, we focus on steganography in bitmap files. Hiding user information in an image file is known as image steganography.

The key contributions of this work are as follows:

- We propose a novel algorithm for generating pseudorandom byte arrays based on a double Ikeda function, demonstrating favourable statistical characteristics.
- This pseudorandom generation algorithm is integrated into a new steganographic scheme.
- Comprehensive evaluation of the proposed approach indicates close to zero Mean Square Error (MSE) values, high peak signal-to-noise ratio (PSNR), strong structural similarity, and very similar histograms.

2 Proposed scheme

2.1 Double Ikeda function-based pseudorandom byte generation

The following equations define the Ikeda map [10]:

$$x_{n+1} = 6 + 0.9\left(x_n \cos\xi_n - y_n \sin\xi_n\right), \tag{1}$$

$$y_{n+1} = 0.9\left(x_n \sin\xi_n + y_n \cos\xi_n\right), \tag{2}$$

where

$$\xi_n = 3.1 - \frac{6}{1 + x_n^2 + y_n^2}. \tag{3}$$

The parameter $u \in (0, 1]$ controls the dissipation of the system.

The double Ikeda function is plotted in Figure 1.

The following steps form the basis for the byte generation:

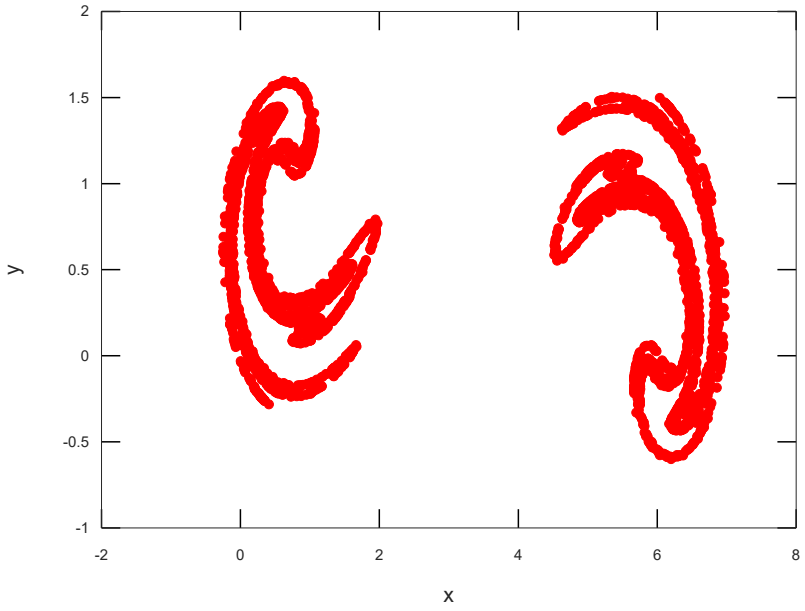

Figure 1: Double Ikeda function

1. The initial values $x_{1,0}, y_{1,0}, x_{2,0}, y_{2,0}, x_{3,0}, y_{3,0}$ for three double Ikeda functions are determined.

2. The values of $x_{1,k}, y_{1,k}, x_{2,k}, y_{2,k}, x_{3,k}, y_{3,k}$ are computed, where k is a fixed constant.

3. For each $i > k$, six non-negative integer values $a, b, c, d, e, f \in [0, 256)$ are computed as follows:

$$a = \left|\lfloor 10^{10} \cdot x_{1,i} \rfloor\right| \bmod 256, \qquad b = \left|\lfloor 10^{10} \cdot y_{1,i} \rfloor\right| \bmod 256, \tag{4}$$

$$c = \left|\lfloor 10^{10} \cdot x_{2,i} \rfloor\right| \bmod 256, \qquad d = \left|\lfloor 10^{10} \cdot y_{2,i} \rfloor\right| \bmod 256, \tag{5}$$

$$e = \left|\lfloor 10^{10} \cdot x_{3,i} \rfloor\right| \bmod 256, \qquad f = \left|\lfloor 10^{10} \cdot y_{3,i} \rfloor\right| \bmod 256. \tag{6}$$

4. A single output byte $byte_i$ is then computed as follows:

$$byte_i = \begin{cases} c \oplus d, & \text{if } (a \oplus b) \bmod 2 = 0, \\ e \oplus f, & \text{otherwise.} \end{cases}$$

5. Steps 3 and 4 are repeated until the desired length of the pseudorandom byte sequence is reached.

2.2 Proposed steganography scheme

The proposed steganography algorithm consists of the following steps:

1. The initial seed for the double Ikeda function-based pseudorandom byte generator is determined.
2. A special end-of-text symbol is appended to the end of the plain message.
3. The plain message is converted into an integer byte array T of length N, where T_i is the ASCII code of the i-th character of the plain text.
4. Assuming the maximum possible value of N is $MAXN = n \cdot m \cdot c$, $MAXN$ pseudorandom bytes b_i are generated using the proposed generator. These bytes will be used to encrypt the message.
5. The array T is encrypted. The elements of the encrypted array are computed as $T'_i = T_i \oplus b_i$.
6. Let $A_{n,m,c}$ be a 3-dimensional array representing a stego image, where n and m are the dimensions of the image, and c is the number of color channels.
7. The array T' is converted into a bit array B' of length $len = 8 \cdot N$. Each encrypted bit B'_i, for $i = 1, 2, \ldots, len$, is embedded into the image using the following procedure:
   - A total of $\lceil \log_2 n \rceil + \lceil \log_2 m \rceil + \lceil \log_2 c \rceil$ bits are generated using the proposed pseudorandom byte generator.
   - These bits are split and interpreted as integers to determine the coordinates (x, y) and the channel index z, by computing each as the result modulo n, m, and c, respectively.
   - If the location (x, y, z) has already been used, repeat the step to generate a new set of coordinates.
   - The bit B'_i is embedded into the least significant bit (LSB) of the pixel located at (x, y) in channel z.
   - The location is marked as used to prevent future overwriting.

The decoding scheme uses the following steps:

1. The initial seed for the double Ikeda function-based pseudorandom byte generator and the end-of-text symbol are identified.

2. Using the proposed pseudorandom byte generator, generate $MAXN = n \cdot m \cdot c$ pseudorandom bytes b_i to decrypt the embedded message.
3. Extract the embedded bits and decrypt them using the following procedure:
   - Generate coordinates (x, y, z) according to the method described in step 6 of the embedding algorithm.
   - If the coordinates (x, y, z) are already marked as used, generate a new set of coordinates.
   - Mark (x, y, z) as used and extract the embedded bit from the LSB of the element $A_{x,y,z}$.
   - Decrypt each byte (group of 8 bits) by performing the exclusive OR ($\oplus$) operation between the extracted byte and the corresponding pseudorandom byte b_i. Append the resulting value to an integer list $extracted\_numbers$.
   - Repeat the extraction and decryption process until the resulting value matches the ASCII code of the special end-of-text symbol, which signals the end of the message.
4. The list $extracted\_numbers$ is converted into the original message by mapping each number to its corresponding ASCII character. The special end-of-text symbol is then removed from the reconstructed message.

The dimensions of the initial seed values, the linear complexity, and the results of the statistical tests suggest that the pseudorandom algorithm is capable of ensuring a favourable pseudorandom-like quality and an acceptable level of security [12], [9], [5], [8], [4].

3 Security analysis

As an illustration, Figure 2 displays the 4.2.01 Woman image in Figure 2(a), along with its stego counterparts in Figures 2(b) to 2(f). Visual inspection shows no apparent differences between the original and the stego images; they look identical to the naked eye, with no visible traces of embedded information.

The randomness of the proposed scheme is evaluated using two statistical tools, NIST and ENT.

(a) (b) (c)

(d) (e) (f)

Figure 2: Illustrated comparison of the 4.2.01 Woman input image and the stego images: (**a**) original image, (**b**) stego image with 100 symbols, (**c**) stego image with 200 symbols, (**d**) stego image with 300 symbols, (**e**) stego image with 400 symbols, and (**f**) stego image with 500 symbols.

Table 1: ENT test results.

ENT test	Bit stream	Byte stream
Entropy	1.000000 bits/bit	7.999998 bits/byte
Optimum compression	decrease the size with 0%	decrease the size with 0%
Chi	2.53 increase with 11.17%	233.17 increase with 83.30%
π value	3.141079566	3.141079566
Correlation coefficient	0.000040	0.000018

The ENT package [13] comprises six different statistical tests. We evaluated the output generated by the proposed pseudorandom byte scheme based on the double Ikeda function using a data array of 100,000,000 bytes. The results in Table 1 met all the criteria of the ENT tests.

A total of 15 statistical tests form the basis of the NIST program [2]. For a sample of 1000 binary arrays, the minimum acceptable pass rate for each statistical test — except the random excursion variant test is around 980. In the case of the random excursion variant test, the minimum pass rate is approximately 603 out of 617 binary arrays. The output numbers are presented in Table 2. The NIST test is successfully passed, with all p-values uniformly distributed across the 10 subintervals and the pass rate falling within the acceptable range.

PSNR quantifies the ratio between the maximum achievable power of a signal and the power of the corrupting noise that degrades its representation. It is formally expressed as:

$$PSNR = 10 \log_{10} \frac{(2^c - 1)^2}{MSE} (dB), \tag{7}$$

here, c denotes the pixel bit depth, and MSE refers to the mean squared error between the input image and the stego image. The MSE is mathematically declared as follows:

$$MSE = \frac{1}{nn} \sum_{i=1}^{n} \sum_{j=1}^{n} (q[i,j] - q'[i,j])^2, \tag{8}$$

where $q[i,j]$, $q'[i,j]$ is the ith-row jth-column pixel in the input and stego images, respectively. Tables 3 and 4 present the calculated MSE and PSNR values for the proposed steganographic scheme, with data embedded in varying payload sizes - 100, 200, 300, 400 and 500 characters (equivalent to

Table 2: NIST test results.

NIST test	P-value	Pass rate
frequency	0.235589	989/1000
block frequency	0.083526	990/1000
cumulative sums	0.881625	990/1000
runs	0.920383	989/1000
longest run	0.763677	988/1000
rank	0.884671	994/1000
fft	0.348869	989/1000
non-overlapping template	0.585863	990/1000
overlapping template	0.043087	981/1000
universal	0.429923	989/1000
approximate entropy	0.821937	991/1000
random excursion	0.430269	611/617
random excursion variant	0.584246	611/617
serial	0.453050	993/1000
linear complexity	0.807412	987/1000

800, 1600, 2400, 3200, and 4000 bits, respectively). The results indicate consistently low, close to 0.0 MSE values, and repeatedly high PSNR values, all exceeding 67 dB, demonstrating that the proposed chaos-based LSB steganography scheme maintains strong imperceptibility and a high level of security.

The Structural Similarity Index (SSIM) is a metric used to evaluate the visual similarity between the input and the stego images [7]. The SSIM between two image patches x and y is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{9}$$

where: μ_x is the average (mean) of image x, μ_y is the average (mean) of image y, σ_x^2 is the variance of image x, σ_y^2 is the variance of image y, σ_{xy} is the covariance between images x and y, $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ are constants to stabilize the division with weak denominators, L is the dynamic range of the pixel values (typically 255 for 8-bit grayscale images), and $K_1 \ll 1$, $K_2 \ll 1$ are small constants (e.g., $K_1 = 0.01$, $K_2 = 0.03$). The SSIM score ranges from -1 to 1, where a value of 1 denotes perfect structural similarity. In contrast to PSNR, which measures pixel-wise intensity differences, SSIM assesses structural variations between two images, offering a metric that aligns more closely with human visual perception. In Table 5, we provide

Table 3: MSE for images with 100, 200, 300, 400, and 500 characters embedded.

Images	100 chars	200 chars	300 chars	400 chars	500 chars
4.1.01	0.0019	0.0042	0.0061	0.0080	0.0104
4.1.02	0.0022	0.0040	0.0061	0.0080	0.0103
4.1.03	0.0020	0.0040	0.0062	0.0083	0.0102
4.1.04	0.0021	0.0042	0.0061	0.0082	0.0100
4.1.05	0.0020	0.0041	0.0060	0.0082	0.0104
4.1.06	0.0021	0.0041	0.0060	0.0082	0.0103
4.1.07	0.0020	0.0041	0.0060	0.0082	0.0103
4.1.08	0.0020	0.0040	0.0061	0.0081	0.0103

Table 4: PSNR for images with 100, 200, 300, 400, and 500 characters embedded.

Images	100 chars	200 chars	300 chars	400 chars	500 chars
4.1.01	75.2348	71.9499	70.2533	69.0995	67.9641
4.1.02	74.7321	72.0905	70.2895	69.0775	68.0069
4.1.03	75.0680	72.1292	70.2282	68.9610	68.0262
4.1.04	74.9178	71.8760	70.2569	68.9663	68.1332
4.1.05	75.0246	71.9659	70.3664	68.9851	67.9535
4.1.06	74.9602	72.0143	70.3370	68.9690	67.9940
4.1.07	75.0680	71.9927	70.3296	68.9743	67.9876
4.1.08	75.0354	72.0631	70.2931	69.0283	68.0198

Table 5: SSIM for images with 100, 200, 300, 400, and 500 characters embedded.

Images	100 chars	200 chars	300 chars	400 chars	500 chars
4.1.01	1.0000	1.0000	1.0000	0.9999	0.9999
4.1.02	1.0000	1.0000	0.9999	0.9999	0.9999
4.1.03	1.0000	1.0000	0.9999	0.9999	0.9999
4.1.04	1.0000	1.0000	1.0000	0.9999	0.9999
4.1.05	1.0000	1.0000	0.9999	0.9999	0.9999
4.1.06	1.0000	1.0000	1.0000	1.0000	0.9999
4.1.07	1.0000	0.9999	0.9999	0.9999	0.9999
4.1.08	1.0000	1.0000	0.9999	0.9999	0.9999

Table 6: Comparison of our steganography scheme with other techniques.

Scheme	MSE	PSNR	SSIM
Proposed	0.0019	75.2348	1.000000
Ref. [7]	0.0016	76.1253	0.999985
Ref. [1]	0.3300	52.9530	-
Ref. [3]	0.2705	53.4623	-
Ref. [11]	0.0191	113.5238	0.999979

the calculated values for SSIM for the presented steganography scheme. The presented results show that the SSIM scores are close to 1.0. These findings suggest that the novel scheme ensures high image quality and strong structural similarity.

The image histograms graphically represent the gray-level distribution within digital images. In this test, the histograms of the input and stego images are compared. Furthermore, histogram analysis was performed using ImageJ (https://imagej.net/ij/) to assess the gray value distribution in the original and stego versions of image 4.2.01 Woman, as shown in Figure 3. The analysis reveals that the stego image histograms closely match those of the original, with no discernible anomalies or evidence of embedded data.

A selection of computed values for the proposed scheme and various existing algorithms is outlined in Table 6. The findings indicate that the proposed scheme produces results that are on par with or superior to those of closely related techniques.

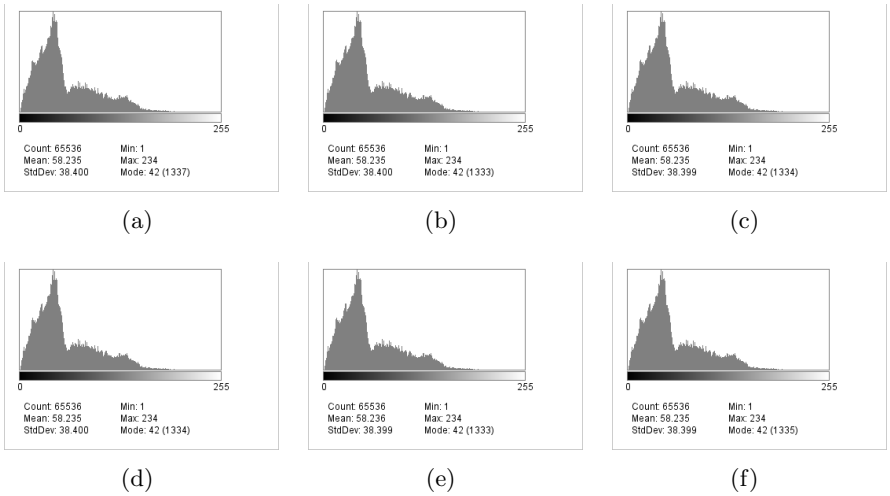

Figure 3: Histogram analysis of the 4.2.01 Woman input image and the stego images: (**a**) original image, (**b**) stego image with 100 symbols, (**c**) stego image with 200 symbols, (**d**) stego image with 300 symbols, (**e**) stego image with 400 symbols, and (**f**) stego image with 500 symbols.

4 Conclusions

We proposed a novel indiscernible steganographic scheme based on spread spectrum image processing, utilizing chaos-driven pseudorandom insertion of least significant bits. This technique enables the embedding of a digital message within an input image without altering its dimensions. Importantly, retrieval of the hidden data does not require the original image, and security is ensured through shared secret keys between sender and receiver. Even if the embedding method is known, unauthorized parties cannot extract the message without the correct keys. Moreover, the embedded signal's power is minimal relative to the cover image, resulting in a low probability of detection and making the presence of hidden data virtually imperceptible.

Acknowledgements. This work is supported by the Scientific research fund of Konstantin Preslavsky University of Shumen under the grant No. RD-08-75/31.01.2025.

References

[1] P.P. Bandekar and G.C. Suguna, LSB Based Text and Image Steganography Using AES Algorithm, In: *3rd International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2018, 782-788.

[2] L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N Heckert and D. Banks, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2010.

[3] E. Emad, A. Safey, A. Refaat, Z. Osama, E. Sayed and E. Mohamed, A secure image steganography algorithm based on least significant bit and integer wavelet transform, *J. Syst. Eng. Electron.* 29 (2018), 639-649.

[4] M. Gupta and R. Chauhan, Hardware Efficient Pseudo-Random Number Generator Using Chen Chaotic System on FPGA, *J. Circuits Syst. Comput.* 31 (2022), 2250043.

[5] A. Hadj Brahim, H. Ali Pacha, M. Naim and A. Ali Pacha, A novel pseudo-random number generator: combining hyperchaotic system and DES algorithm for secure applications, *J. Supercomput.* 81 (2025), 94.

[6] G. Kipper, *Investigator's Guide to Steganography*, Auerbach publications, CRC Press LLC: Boca Raton, FL, USA, 2003.

[7] K. Kordov and S. Zhelezov, Steganography in color images with random order of pixel selection and encrypted text message embedding, *PeerJ. Comp. Sci.* 7 (2021), e380.

[8] R.B. Naik and U. Singh, A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption, *Ann. Data. Sci.* 11 (2024), 25–50.

[9] M. Nazish and M. Banday, Resource-efficient one-dimensional discrete chaotic map-based pseudo-random number generator for IoT applications: a practical analysis, *SN Comput. Sci.* 6 (2025), 779.

[10] C.H. Skiadas and C. Skiadas, *Chaotic Modelling and Simulation: Analysis of Chaotic Models, Attractors and Forms*, Chapman and Hall/CRC, 2008.

[11] B. Stoyanov and B. Stoyanov, BOOST: Medical image steganography using nuclear spin generator, *Entropy* 22 (2020), 501.

[12] A. Tutueva, I.T. Karimov, L. Moysis, E. Nepomuceno, C. Volos and D. Butusov, Improving chaos-based pseudo-random generators in finite-precision arithmetic, *Nonlinear Dyn.* 104 (2021), 727-737.

[13] J. Walker, ENT: A Pseudorandom Number Sequence Test Program, http://www.fourmilab.ch/random/.

[14] S. Zhelezov, B. Uzunova-Dimitrova and H. Paraskevov, An approach for hiding steganography data within web applications, *J. Eng. Appl. Sci.* 12 (2017), 8251–8255.