Ann. Acad. Rom. Sci.
Ser. Math. Appl.
ISSN 2066-6594 Vol. 12, No. 1-2/2020

YASH: YET ANOTHER STEGO HIDING*

Hristo Paraskevov† Aleksandar Stefanov‡ Borislav Stoyanov§

Dedicated to Dr. Vasile Drăgan on the occasion of his 70th anniversary

Abstract

We present a novel pseudorandom insertion least significant bit (LSB) based hiding scheme using Circle map byte output. The proposed algorithm is analysed by means of computer simulation. We evaluated the designed LSB method with NIST and ENT statistical packages, peak signal-to-noise ratio, and histogram analysis. The results data show good performance of the novel stego hiding.
MSC: 68P25, 11K45, 94A60

keywords: chaotic maps, pseudorandom byte generator, least significant bit, steganography.

1 Introduction

Modern information technology is an integral part of our daily lives. Embedding hidden messages into images is an easy way for secure communication between two people.

*Accepted for publication on April 21, 2020

†h.paraskevov@shu.bg University of Shumen, Bulgaria; Paper written with financial support of the project RD-08-96/2019

‡a.stefanov@shu.bg University of Shumen, Bulgaria

§borislav.stoyanov@shu.bg University of Shumen, Bulgaria; Paper written is partially supported of the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)", financed by the Ministry of Education and Science.

There are many applications for techniques that embed information within digital images. Digital image steganographic techniques can also provide forward and backward compatibility by embedding information in an image in an imperceptible manner. If a system has the ability to decode the embedded information, new enhanced capabilities could be provided. If a system did not have the capability to decode the information, the image would be displayed without degradation, leaving the viewer unaware that the hidden data exist.

In this paper, we present novel pseudorandom insertion least significant bit (LSB) based hiding scheme. We are using Circle map [2]. We slightly modified pseudorandom number generation proposed in [1]. The proposed algorithm is analysed by means of computer simulation. We evaluated the designed LSB method with NIST and ENT statistical packages, peak signal-to-noise ratio, and histogram analysis. The results data show good performance of the novel stego hiding.

2 Novel stego hiding scheme based on Circle map

2.1 Circle map

The Circle function [2] maps points on the circle back onto a circle. It is a nonlinear iterated map given by

$$\theta_{n+1} = (\theta_n + \Omega - \frac{K}{2\pi}\sin(2\pi\theta_n)), \tag{1}$$

where Ω is a constant that is the fixed angular progression of the sinusoidal oscillator, K is the coupling strength, and θ_{n+1} is computed mod 1.

2.2 YASH: Yet Another Stego Hiding

Here we present YASH, yet another stego hiding, a new least significant bit algorithm by using the Circle map based post-processed output values. We consider input and resulting images of $n \times n$ size.

The YASH consists of the next steps:

1. Convert input text to binary stream. The initial values θ_0, Ω, and K, of a Circle map from Eq. (1) are determined. The chaotic function is iterated for L times.

2. The Circle map is iterated two times. The two real fractions θ^i and θ^j are post-processed as integer values $i = abs(mod(integer(\theta^i \times 10^9), n)$

and $j = abs(mod(integer$
$(\theta^j \times 10^9), n)$. Repeat this Step until unused pixel position (i, j) is detected.

3. The Circle map is iterated three times and the real fractions are post-processed as bit values $r_{lsb} = abs(mod(integer(\theta^r \times 10^9), 2)$, $g_{lsb} = abs(mod(integer(\theta^g \times 10^9), 2)$, and $b_{lsb} = abs(mod(integer(\theta^b \times 10^9), 2)$.

4. Embed r_{lsb}, g_{lsb}, and b_{lsb} into the last bit of the red, green, and blue values of the position (i, j).

5. Repeat Steps 2–4 until input text is embedded.

3 Computer modelling

The novel YASH is implemented in C++ and Python programming languages and the color images are chosen from USC-SIPI images (sipi.usc.edu/database/). Figures 1 to 3 show the selected 3 images from the database.

Figure 1: Image 4.2.03 Figure 2: Image 4.2.04 Figure 3: Image 4.2.05

4 Security analysis

4.1 Software package testing

Two statistical packages NIST and ENT, peak signal-to-noise ratio, and histogram analysis are used in order to measure randomness of YASH.
The NIST suite [3] is based on 15 statistical tests: frequency, block frequency, cumulative sums, runs, longest run of ones, rank, spectral, non-overlapping templates, overlapping templates, universal, approximate entropy, random excursion, random excursion variant, serial, and linear complexity. The

post-processed bytes from Circle map passed successfully NIST suite, Table 1. The minimum pass rate for each statistical test with the exception of the random excursion variant test is approximately 1966 for a sample size of 2000 binary sequences. The minimum pass rate for the random excursion variant test is approximately 1178 for a sample size of 1201 binary sequences.

Table 1: NIST test results.

NIST test	P-value	Pass rate
frequency	0.797204	1976/2000
block frequency	0.332970	1980/2000
cumulative sums 1	0.064620	1978/2000
cumulative sums 2	0.618385	1976/2000
runs	0.793450	1984/2000
longest run of ones	0.072289	1974/2000
rank	0.390721	1982/2000
spectral	0.546283	1972/2000
non-overlapping templates	0.484767	1980/2000
overlapping templates	0.874764	1972/2000
universal	0.005204	1976/2000
approximate entropy	0.071399	1976/2000
serial 1	0.981151	1973/2000
serial 2	0.492436	1967/2000
random excursion	0.753495	1188/1201
random excursion variant	0.415927	1189/1201
linear complexity	0.938463	1981/2000

The ENT suite is based on 6 tests to pseudorandom sequences. We tested output of 2000000000 bits of the post-processed Circle map values:

- zeros: 1000013353 occurrences with fraction 0.500007; ones: 999986647 with fraction 0.499993.
- entropy is 1.000000 bits per bit.
- optimum compression would reduce the size of this 2000000000 bit file by 0 %.
- χ^2 square distribution for 2000000000 samples is 0.36, and randomly would exceed this value 55.04 percent of the times.
- arithmetic mean value of data bytes is 0.5000 (0.5 = random).

- Monte Carlo value for π is 3.141740978 (error 0.00 percent).
- serial correlation coefficient is -0.000020 (totally uncorrelated = 0.0).

The byte output values passed successfully ENT suite.

4.2 Histogram analysis

The histogram analysis aims to present the overall tone distribution of the colors in the original and stereo files. In stego file are embeded randomly generated by Lorem Ipsum 300 bytes. This analysis shows that the two histograms are very similar, which indicates good performance of the algorithm, see Figure 4 and 5.

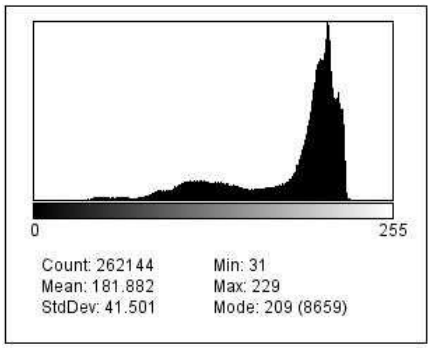

Figure 4: Original file

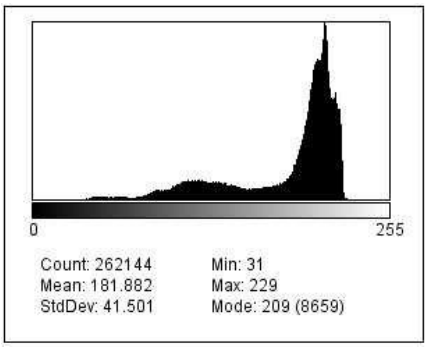

Figure 5: Stego file

4.3 Coordinate distribution

According to step 2 of the proposed algorithm, the generated bytes are converted to pairs of coordinates, used to determine the pixels into which the binary stream will be inserted. An important factor for the correct application of the proposed algorithm is the random like distribution of the pixels used in the image. Figure 6 shows the result with embeded small text and Figure 7 with large text. The figures show that in both cases the coordinate distribution is even.

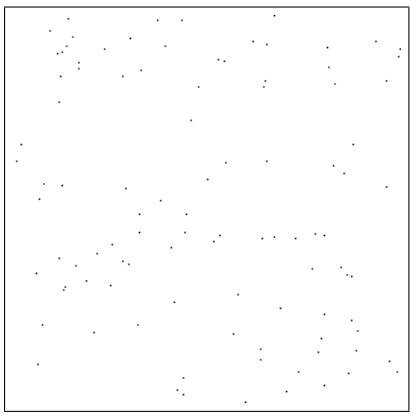

Figure 6: With small text

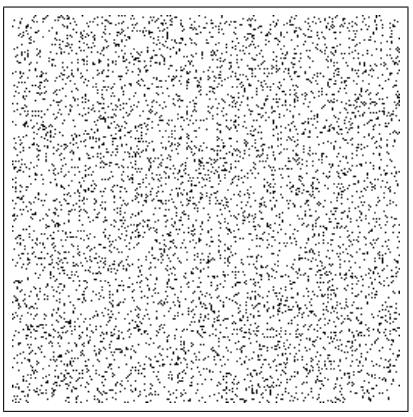

Figure 7: With large text

4.4 Peak signal-to-noise ratio analysis

In order to assess the extent of changing the stego file towards the container the ratio PSNR is calculated.

$$PSNR = 10 \log_{10} \left(\frac{L^2}{MSE} \right) \tag{2}$$

where:
L is the maximum value which is used for color identification
MSE represents the cumulative mean square error between the original and the altered image with dimensions MxN, calculated by the formula:

$$MSE = \frac{1}{MN} \left[\sum_{i=1}^{M} \sum_{i=1}^{N} \left(f(i,j) - f'(i,j) \right)^2 \right] \tag{3}$$

where *f[i, j]*, *f'[i, j]* is the ith-row jth-column pixel in the plain and stego images, respectively.
In Table 2, we provide the computed values for PSNR for the proposed stego algorithm, where PSNR are calculated for images with 100 bytes, 200 bytes, 300 bytes, 400 bytes, and 500 bytes embedded, are presented. From the obtained results, Table 2, it is clear that the PSNR values are very high, above 68 dB, which is an indication that the new LSB chaos-based steganography algorithm has a good level of security.

Table 2: PSNR for images with 100 bytes , 200 bytes, 300 bytes, 400 bytes, and 500 bytes embeded.

Images	100 bytes	200 bytes	300 bytes	400 bytes	500 bytes
4.2.03	85.6886	83.0220	81.4644	80.1238	79.2722
4.2.04	85.4637	82.8060	81.1324	80.0373	78.8029
4.2.05	85.7520	83.1254	81.3163	79.8720	79.0710

In Table 3, we have compare the PSNR of our embedding scheme with similar references [5], [6], [7], and [8].

Table 3: Compare the PSNR with other algorithms

Images	Ref.[5]	Ref.[6]	Ref.[7]	Ref.[8]	Proposed
4.2.03	51.11	44.26	44.54	44.16	79.27
4.2.04	51.12	41.62	44.53	44.19	78.80
4.2.05	51.14	44.24	44.42	-	79.07

Compared to other LSB steganography schemes, we can see that proposed scheme has higher PSNR values.

5 Conclusions

We have presented a novel steganographic spread spectrum image processing techniques that uses chaos based pseudorandom insertion of least significant bit. This process provides a method for concealing a digital signal within a cover image without increasing the size of the image. Additionally, the original image is not needed to extract the hidden message, and a level of security is provided by the necessity that both sender and receiver possess the same keys. An eavesdropper will be unable to decipher the hidden information without possession of the appropriate keys even though the system methodology may be known. Furthermore, the embedded signal power is insignificant compared to that of the cover image, providing low probability of detection and leaving an observer unaware that the hidden data exist.

6 Future work

Two directions for future work could include: the use of multiple container and implementation of parallel algorithms. Using a multiple container will

reduce the change in each of the carrier files, and applying parallel algorithms will reduce processing time.

References

[1] B. Stoyanov. Using circle map in pseudorandom bit generation. In *AIP Conference Proceedings.* 1629:460-463, 2014.

[2] G. Essl. Circle Maps as a Simple Oscillators for Complex Behavior: I. Basics. In *International Computer Music Conference.* 356-359, 2006.

[3] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Application. NIST Special Publication 800-22, Revision 1a (Revised: April 2010), Lawrence E. Bassham III, 2010.

[4] J. Walker. ENT: A Pseudorandom Number Sequence Test Program, http://www.fourmilab.ch/random/, 2004.

[5] Amirtharajan, R., Rayappan, J.B.B. (15 June 2012), An intelligent chaotic embedding approach to enhance stego-image quality, Information Sciences, Volume 193, Pages 115-124.

[6] Aziz, M., Tayarani-N, M.H., and Afsar, M. (2015), A cycling chaosbased cryptic-free algorithm for image steganography, Nonlinear Dynamics, 80 (3), 1271-1290.

[7] Rajendran, S., Doraipandian, M. (2017), Chaotic Map Based Random Image Steganography Using LSB Technique, International Journal of Network Security, Vol.19, No.4, 593-598.

[8] Ranjith Kumar, R., Jayasudha, S., Pradeep, S. (2016), Efficient and secure data hiding in encrypted images, A new approach using chaos Information Security Journal, 25 (4-6), pp. 235-246.