Mathematical and Software Engineering, Vol 5, No 1 (2019), 15–22.
Varεpsilon Ltd, http://varepsilon.com

Video steganography with steganalysis

Krasimir Kordov[1] **and Georgi Valchev**[2]

[1]Department of Computer Informatics, Faculty of Mathematics and Informatics, University of Shumen, Bulgaria, e-mail: krasimir.kordov@shu.bg
[2]Department of Computer Informatics, Faculty of Mathematics and Informatics, University of Shumen, Bulgaria, e-mail: razmu@abv.bg

Abstract

This paper presents steganographic algorithm, for hiding secret text in raw video files. The implementing and extraction method are described and empirical tests are performed to evaluate the security of the proposed algorithm. The steganographic analysis includes files comparison, histogram analysis, Peak to signal noise ratio and Chi-square analysis.

Keywords: steganography, steganography in video, steganalysis

1 Introduction

Steganology is an art of science including two general aspects - Steganography and Steganographic analysis (or Steganalysis) [1, 12, 16]. The first aspect - Steganography, means ”covered writing” and its main purpose is ”data hiding”. The are different ways to hide data used in the past, but in recent years concerning the information is mostly digital, Steganography methods are improved to work with digital files containing the hidden data. Usually the files caring the secret message are called stego-containers and the most used files are digital images [8, 10]. New approaches for steganography explore the possibilities for hiding information in different types of files such as video files and this paper is presenting an algorithm for covering data in digital raw video files.

Steganalysis is the other main aspect ot Steganology and has the purpose to determine the possibility for existence of hidden data[13, 15]. The steganalysis is also a part information security [9, 14] and uses different methods for analysing the files with file size comparison, histogram analysis, measuring values such as Peak to signal noise ratio (PSNR), Chi-square etc.

2 Stego algorithm

Embedding digital information in specific type of files requires knowledge of the file structure. For example images are composed of pixels [2], audio files are composed of samples [3] and video files are composed of frames. Our focus are video files and we use their frames for data embedding. Video files are decomposed to frames and the frames are processed as images. The proposed scheme is based on standard Least significant bit (LSB) method[8, 10]. Programming language Python is used for the software realization of the algorithm.

2.1 Embedding algorithm

The embedding process includes the following steps:

- The secret text message (T) for hiding is being typed

- A control symbol is appended at the end of the secret message (in our case the symbol is "$\sim$", because is rarely used in text)

- The text is converter in binary sequence using ASCII table values

- The video file (P) (plain video) is decomposed into frames

- The number of the frames n is divided to the number of the letters (l) and the result (lpf) (letter per frame) is rounded to determine the total number of the letters that will be embedded in every frame

- The number of total symbols (l) of the secret message and the number of letters per frame (lpf) are being embedded in the first frame as a binary sequence with LSB method

- The embedding process of the secret message starts from the second frame and in frame lpf letters are being embedded using LSB method

- The result stego frames are composed again into stego video file - S (stego-container)

2.2 Extraction algorithm

The extraction process includes the following steps:

- Stego video file S is decomposed into frames

- From the fist frame, number of total letters of the secret message (l) and the number of letters per frame (lpf) are extracted and converted into *integer* values

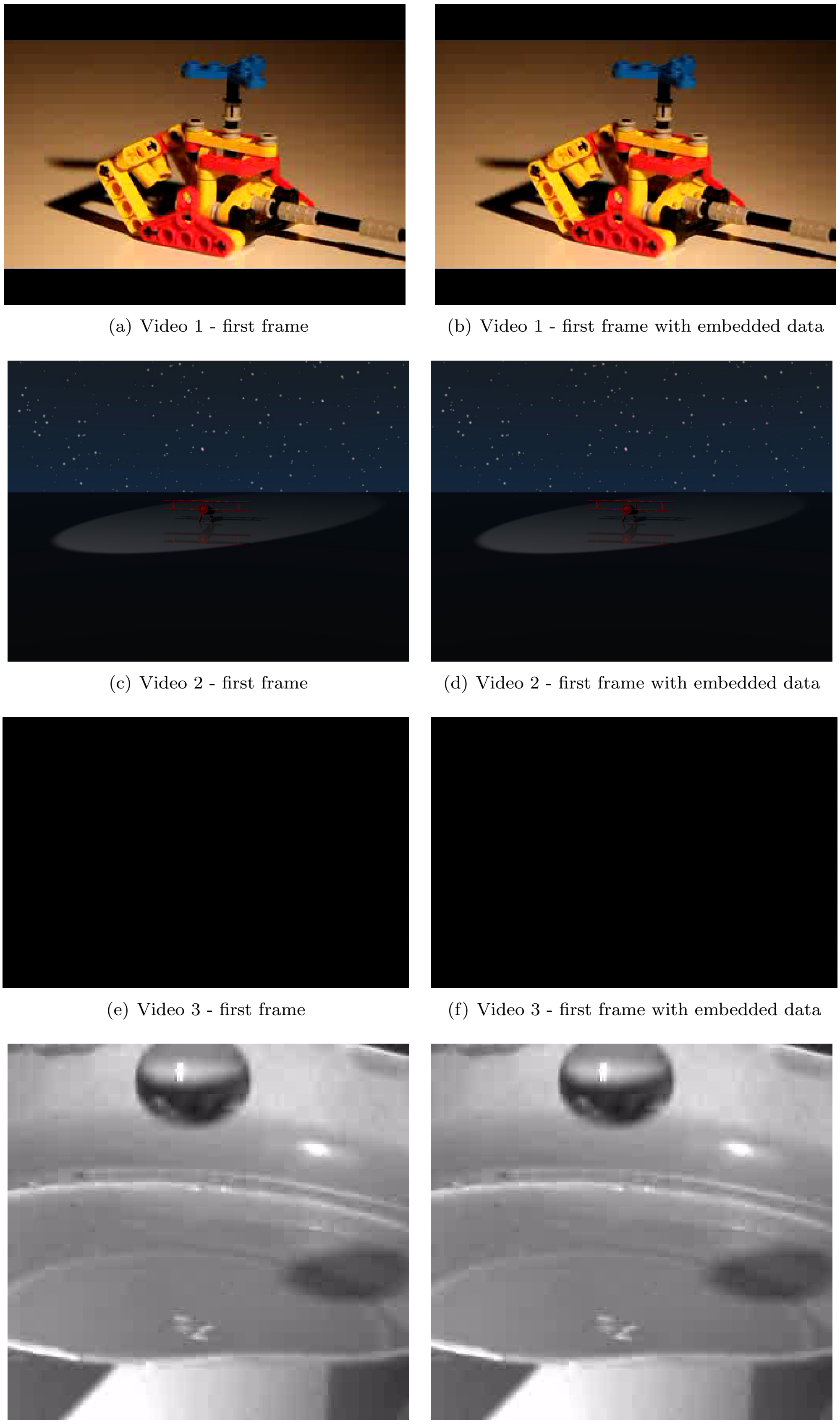
- From every frame (until the control control symbol) (lpf) letters are extracted and appended to restore the secret message (T)

3 Stego analysis

The steganographic analysis that we performed includes file comparison and some stego indicators.

3.1 Visual analysis

The purpose of this test is to determine if there are any visual differences between plain video files and their corresponding files with hidden data. Figure 1 shows the first frames of out test video files with their corresponding stego frames with embedded information.



(a) Video 1 - first frame

(b) Video 1 - first frame with embedded data

(c) Video 2 - first frame

(d) Video 2 - first frame with embedded data

(e) Video 3 - first frame

(f) Video 3 - first frame with embedded data

(g) Video 4 - first frame

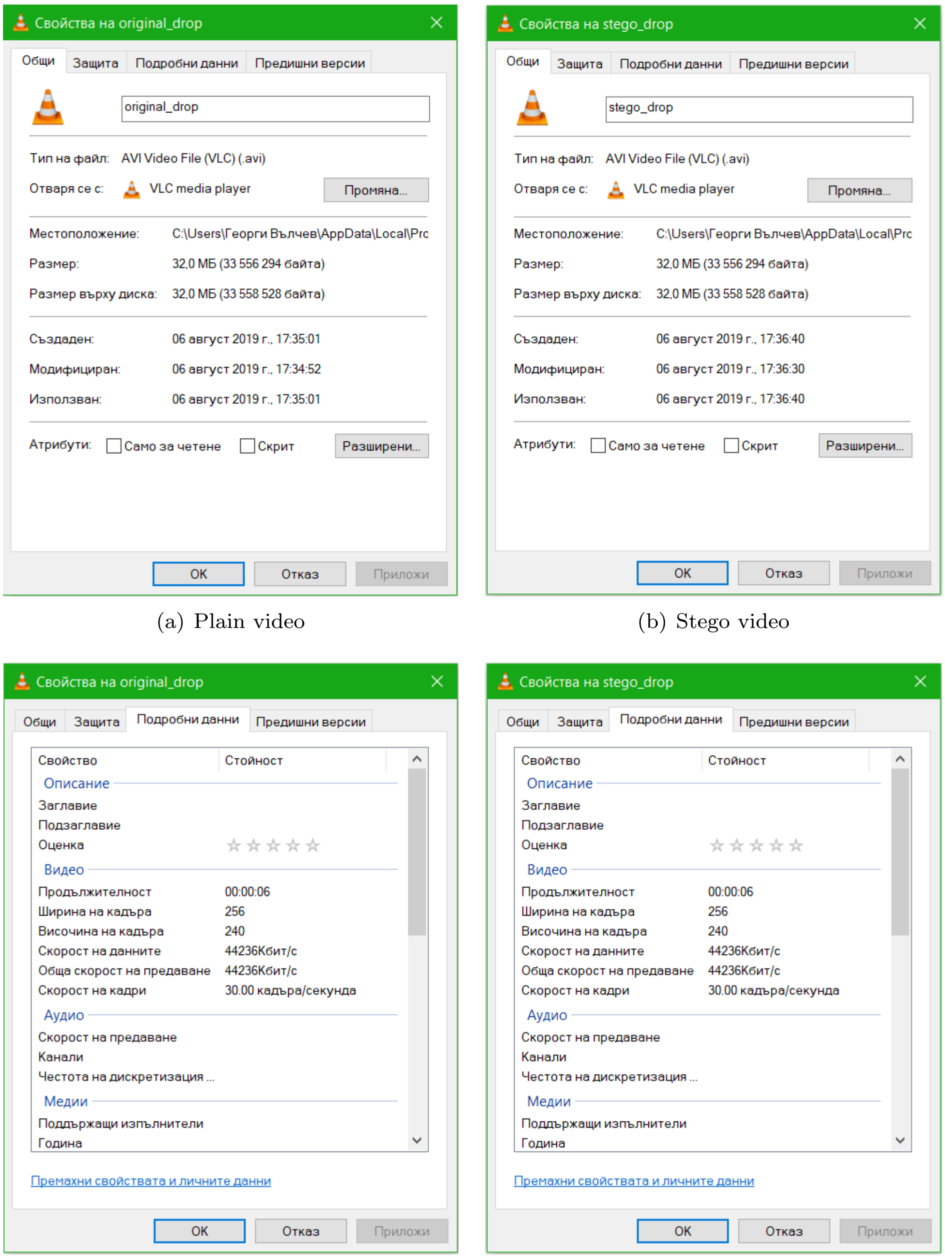
(h) Video 4 - first frame with embedded data

Figure 1: Visual analysis - plain video files with their corresponding stego video files

This tests demonstrate there is no visual difference between video files even if all the pixels in a frames are the same or similar color (see Figure 1(e) and Figure 1(f)).

3.2 File size comparison

One of the most important aspects in steganography is to leave no trace of file modifications. Changing the file size in embedding process can be considered as an indicator for steganography. Figure 2 demonstrates that the our algorithm do not change the file size in hiding process.



original_drop

Тип на файл: AVI Video File (VLC) (.avi)

Отваря се с: VLC media player

Местоположение: C:\Users\Георги Вълчев\AppData\Local\Prc

Размер: 32,0 МБ (33 556 294 байта)

Размер върху диска: 32,0 МБ (33 558 528 байта)

Създаден: 06 август 2019 г., 17:35:01

Модифициран: 06 август 2019 г., 17:34:52

Използван: 06 август 2019 г., 17:35:01

Атрибути: □ Само за четене □ Скрит

(a) Plain video

stego_drop

Тип на файл: AVI Video File (VLC) (.avi)

Отваря се с: VLC media player

Местоположение: C:\Users\Георги Вълчев\AppData\Local\Prc

Размер: 32,0 МБ (33 556 294 байта)

Размер върху диска: 32,0 МБ (33 558 528 байта)

Създаден: 06 август 2019 г., 17:36:40

Модифициран: 06 август 2019 г., 17:36:30

Използван: 06 август 2019 г., 17:36:40

Атрибути: □ Само за четене □ Скрит

(b) Stego video

Свойства на original_drop
Общи
Защита
Подробни данни
Предишни версии
Премахни свойствата и личните данни
OK
Отказ
Приложи

Свойство	Стойност
Описание	
Заглавие	
Подзаглавие	
Оценка	☆☆☆☆☆
Видео	
Продължителност	00:00:06
Ширина на кадъра	256
Височина на кадъра	240
Скорост на данните	44236Кбит/с
Обща скорост на предаване	44236Кбит/с
Скорост на кадри	30.00 кадъра/секунда
Аудио	
Скорост на предаване	
Канали	
Честота на дискретизация ...	
Медии	
Поддържащи изпълнители	
Година	

(c) Plain video - detailed info

Свойство	Стойност
Описание	
Заглавие	
Подзаглавие	
Оценка	☆☆☆☆☆
Видео	
Продължителност	00:00:06
Ширина на кадъра	256
Височина на кадъра	240
Скорост на данните	44236Кбит/с
Обща скорост на предаване	44236Кбит/с
Скорост на кадри	30.00 кадъра/секунда
Аудио	
Скорост на предаване	
Канали	
Честота на дискретизация ...	
Медии	
Поддържащи изпълнители	
Година	

(d) Stego video - detailed info

Figure 2: File size comparison

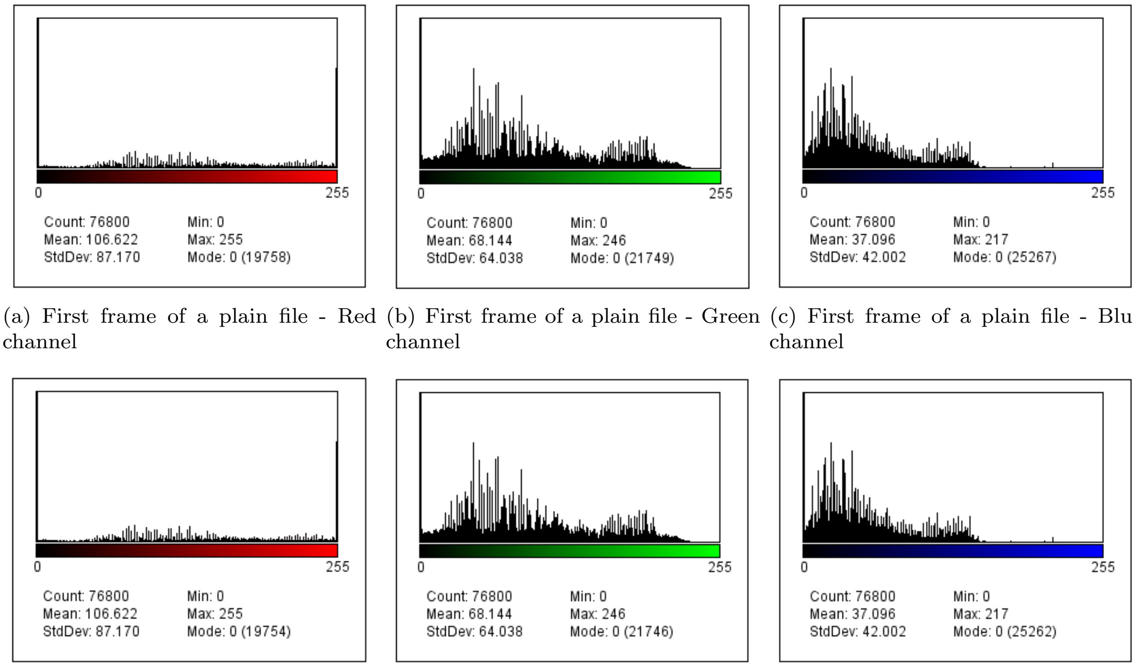
The full file comparison tests we made are presented in Table 1.

File Name	File Size	Stego File Size
Toy.avi	30.5MB	30.5MB
Plane.avi	87.8MB	87.8MB
Canyon.avi	1.10GB	1.10GB
Drop.avi	32.0MB	32.0MB
Stream.avi	137.0MB	137.0MB

Table 1: File size comparison

3.3 Histogram Analysis

The histogram analysis compares the frames (processed as images) of the plain and their corresponding stego frames (frames with embedded data). Image histograms represent the tonal distribution of the colors in the images. This test compares histograms of plain and stego frames.



(a) First frame of a plain file - Red channel

(b) First frame of a plain file - Green channel

(c) First frame of a plain file - Blue channel

(d) First frame of a stego file - Red channel

(e) First frame of a stego file - Green channel

(f) First frame of a stego file - Blue channel

Figure 3: Histogram analysis - color histograms

Figure 3 demonstrates the similarity of the plain and stego frames which is indicator of successfully realized steganography.

3.4 Peak Signal to Noise Ratio

Peak Signal to Noise Ratio (PSNR) measure the power of clean signal against the power of noise. For our tests we calculated the PSNR for the fist three frames of every plain and its corresponding stego frame of the video files. PSNR is calculated as follows:

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE} dB, \tag{1}$$

where MAX is the maximum possible value of pixel color (In our case the maximum value is 255) and MSE is the mean square error between the plain and stego frame. MSE is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2, \qquad (2)$$

where N is the total number of pixels in the frame, x_i and y_i are the corresponding pixel values of the plain and stego frames.

Table 2 and Table 3 contain the results of our tests.

Table 2: Mean Square Error

File Name	Total Frames	Embedded Chars	Chars in Frame	Frame 1 MSE	Frame 2 MSE	Frame 3 MSE
Toy.avi	139	1000	8	0.52083	0.00015	0.00013
Plane.avi	100	2000	21	0.18446	0.93316	0.87891
Canyon.avi	1142	3000	3	0.15432	0.13503	0.11574
Drop.avi	182	4000	23	0.00015	0.00049	0.00049
Stream.avi	157	5000	33	0.34722	0.00012	0.00013
Ref [8]	-	-	100	0.0021	-	-

Table 3 contains the results of our tests.

Table 3: Peak Signal to Noise Ratio

File Name	Total Frames	Embedded Chars	Chars in Frame	Frame 1 PSNR	Frame 2 PSNR	Frame 3 PSNR
Toy.avi	139	1000	8	90.9638	86.3149	86.8420
Plane.avi	100	2000	21	95.4717	88.4312	88.6914
Canyon.avi	1142	3000	3	96.2466	96.8265	97.4959
Drop.avi	182	4000	23	86.1625	81.1961	81.1486
Stream.avi	157	5000	33	92.7247	87.0208	86.6703
Ref [8]	-	-	100	74.9390	-	-
Ref [10]	-	-	-	73.98	-	-

Higher PSNR indicates more clear signal and values over 60 dB are considered as acceptable for steganographic analysis.

3.5 Chi-Square analysis

In order to check if our steganographic algorithm can resist to Chi-square attacks we used Chi-square steganography test software. The red curve indicates Chi-square values of the tested frames and the green values represent the average value of the LSBs. If the green values are above the red curve the test is passed successfully. Figure 4 demonstrates the results of our tests. On the left are diagrams of the first frames and on the right are their corresponding stego frames.

The Chi-square tests shows that there is no trace of steganography in the video files, indicated that the proposed algorithm can withstands against Chi-square attacks.

4 Future work

For additional security the stego algorithm can be combined with cryptographic pseudo-random generator [4, 5, 6, 7, 13] for encrypting the secret message before embedding or for random selection of the frames of pixel positions for applying the LSB method.

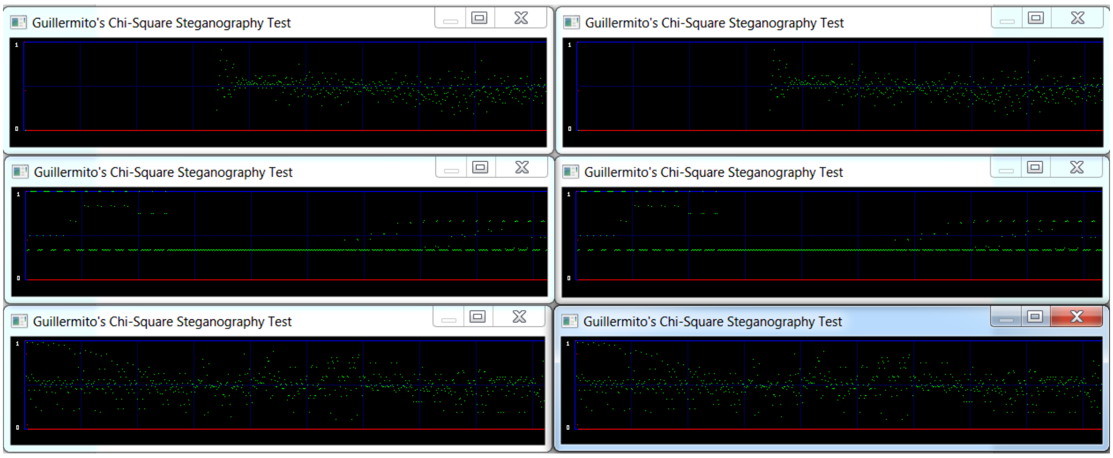

Guillermito's Chi-Square Steganography Test
Guillermito's Chi-Square Steganography Test
Guillermito's Chi-Square Steganography Test
Guillermito's Chi-Square Steganography Test
Guillermito's Chi-Square Steganography Test
Guillermito's Chi-Square Steganography Test
1
0

Figure 4: Chi-Square analysis

5 Conclusion

We designed a video steganographic algorithm for data hiding in video files. The algorithm is software realized in Python for Steganographic analysis. Stegoanalysis shows there are no visual signs of steganography and there are no file size changes in data embedding process. Steganographic analysis also indicates high values of PSNR and resistance against Chi-square attacks.

Acknowledgement

This work is partially supported by the Scientific research fund of Konstantin Preslavski University of Shumen under the grant No. RD-08-71/29.01.2019.

References

[1] Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T. (2007). Digital watermarking and steganography. Morgan kaufmann.

[2] Hristova, R. P. (2017). An example of using computer animation in astronomy lessons (in secondary school).

[3] Kordov, K. (2019). A Novel Audio Encryption Algorithm with Permutation-Substitution Architecture. Electronics, 8(5), 530.

[4] Kordov, K. M. (2014, November). Modified Chebyshev map based pseudo-random bit generator. In AIP Conference Proceedings (Vol. 1629, No. 1, pp. 432-436). AIP.

[5] Kordov, K. (2015). Signature Attractor Based Pseudorandom Generation Algorithm. Advanced Studies in Theoretical Physics, 9(6), 287-293.

[6] Stoyanov, B., Kordov, K. (2013, June). Pseudorandom bit generator with parallel implementation. In International Conference on Large-Scale Scientific Computing (pp. 557-564). Springer, Berlin, Heidelberg.

[7] Kordov, K. (2015). Modified pseudo-random bit generation scheme based on two circle maps and XOR function. Applied Mathematical Sciences, 9(3), 129-135.

[8] Kordov, K., Stoyanov, B. (2017). Least Significant Bit Steganography using Hitzl-Zele Chaotic Map. International Journal of Electronics and Telecommunications, 63(4), 417-422.

[9] Nachev, A., Zhelezov, S. (2013). Assessing the efficiency of information protection systems in the computer systems and networks. Collection" Information technology and security", 2(1), 79-85.

[10] Stoyanov, B. P., Zhelezov, S. K., Kordov, K. M. (2016). Least significant bit image steganography algorithm based on chaotic rotation equations. Comptes rendus de l'Academie bulgare des Sciences, 69(7), 845-850.

[11] Stoyanov, B., Szczypiorski, K., Kordov, K. (2017). Yet another pseudorandom number generator. International Journal of Electronics and Telecommunications, 63(2), 195-199.

[12] Stanev, S., Szczypiorski, K. (2016). Steganography Training: a Case Study from University of Shumen in Bulgaria. International Journal of Electronics and Telecommunications, 62(3), 315-318.

[13] Stanev, S., Zhelezov, S., Yakimov, I. (2012). An aproach for parallel steganalisys based on data compression. In Proceedings of Jubilee International Congress (Vol. 40, pp. 360-367).

[14] Todorova, M., Kapralov, S., Dyankova, V. (2018). Application of sorting algorithms for convex hull determination. Mathematical and Software Engineering, 4(2), 24-27.

[15] Zhelezov, S. (2016). Modified Algorithm for Steganalysis. Mathematical and Software Engineering, 1(2), 31-36.

[16] Zhelezov, S., Paraskevov, H. (2015). Possibilities for steganographic parallel processing with a cluster system. Contemporary Engineering Sciences, 8(20).