Increasing the Steganographic Resistance of the LSB Data Hide Algorithm

A. Y. Buchaev[1], A. G. Mustafaev[2], V.S. Galyaev[3], A. M. Bagandov[4]
Department of Information Technologies and Management
Dagestan State University of National Economy
Makhachkala, Russian Federation

***Abstract*—The robustness of the security algorithm is one of the most important properties that determines how difficult it is to break it. Increasing the robustness of the algorithm directly affects the degree of secrecy when it is used for confidential transmission. The paper analyzes the steganographic algorithm Least Significant Bit, represents a method of counteracting the algorithm of the "visual attack" and statistical methods used against stego-containers generated using the LSB algorithm. To prove the increase in resistance, the study used the PSNR index, Chi-square test. The proposed technique involves the use of a uniform distribution and compression method. The paper presents the results of computer experiments demonstrating the effectiveness of the proposed technique.**

Keywords*—Steganography; steganalysis; visual attack; least significant bit

I. Introduction

Steganography is the science of methods of transferring information or storing it, in which the fact of transfer or storage is hidden [1]. Currently, such directions as digital steganography (hiding information in digital objects) and network steganography (hiding information using the features of network protocols) are developing. Modern steganographic systems use steganography and cryptography algorithms together in order to not only encrypt and protect a message [2-4], but also to transmit it secretly.

Some steganographic algorithms have become widespread, such as algorithms for applying digital watermarks, which used to embed an image into an image and provide protection against illegal copying or dissemination of information [5]. The most famous and simple steganography algorithms have obvious drawbacks, for example, replacing the color of the hidden message with the background color, such a substitution is easy to notice and reveal the hidden data [6-7]. The book cipher (or its modifications, for example, the book cipher of Aeneas), in which each character of the secret message is replaced by a pointer (for example, the number of a row, column or table), has several significant disadvantages: transmission of small volumes of a secret message; storage and transmission of the key, which can be used to collect a "scattered" message from the so-called stego-container, weak degree of security [8]. Methods of hiding secret information in special fields of attributes of files of various formats or in service fields of network packets are also popular, but these methods have a significant limitation on the amount of information transmitted per unit of time [9-10]. Along with these methods, steganography includes various algorithms are based on distortion (introduction of changes into the structure of a digital object), statistical methods of concealment [11] and structural methods.

Specialized attacks are carried out on steganographic algorithms, the main purpose of which is to reveal the presence of an embedded secret message. The statistical method has become widespread, which makes it possible to determine the characteristic distortions both in the file structure and in the semantic information of a digital object [12]. Attacks are usually directed against specific vulnerabilities of a particular steganographic algorithm [13]. When hiding information in the service part of files or transmitted packets, an attacker can compare with reference values or empty containers to identify potential corruption [14-15]. In modern steganalysis, separate areas of investigation to identify hidden messages have been formed.

Reliable masking algorithms are being developing to counter attacks, for example, one of them hides secret information during a "handshake" when using the TCP data transfer protocol in a response packet [16], but this method is describing theoretically, is implementing only as part of the study and has small bandwidth.

Within the framework of this work, a technique makes it possible to reduce the efficiency of steganalysis methods of the "visual attack" type. The Least Significant Bit (LSB) method [17-18] was chosen as the most illustrative example of a steganographic algorithm. LSB is an efficient algorithm used to embed information into container files [19-20]. A hypothesis was put forward, according to which the algorithm can become more resistant to the mentioned types of attacks if the modified bytes are uniformly distributed in the container image. However, when using a uniform distribution of a large number of bytes, a large number of zones with a high density of modified bytes will inevitably appear [21-22]. Lossless compression methods are using to reduce the number of modified bytes.

II. Steganographic Algorithm least Significant Bit

LSB (Least Significant Bit) is a method of embedding a secret message into an image. The algorithm includes the following steps:

- Conversion of secret message into a binary code, followed by splitting into separate bits or into blocks of two bits.

- Replacing the last bit or two bits in the bytes of the container image with the corresponding number of bits of the transmitted secret message.

For example, the binary form of the secret message looks like this: 101101. We form groups of two bits: 10, 11, 01. We get three groups, the number of modified image bytes is equal to the number of groups, and therefore, the last two bits in the three bytes will be replaced in the container image. Suppose the bytes of the container image look like this: ... 10111010 11010001 10000011 ..., the last two bits in the given bytes will be replaced with bits from the previously received groups. The transformation will look like this: the set 10111010 11010001 10000011 goes to 10111010 11010011 10000001. This transformation leads to changes that a person does not perceive during normal visual observation.

Such conversions can be carried out with graphic file formats that do not compress data (BMP, PNG), otherwise the hidden information will be lost. The algorithm described above is also applicable to the WAV format [23], similar conversions can be performed with video clips and sound files.

For containers, either a unique digital object is often used, or images and sound files that are distributed over the network in a variety of ways. If you select objects that are in the public domain in the most common variant, then an attacker will be able to compare the source file with the container and reveal secret information.

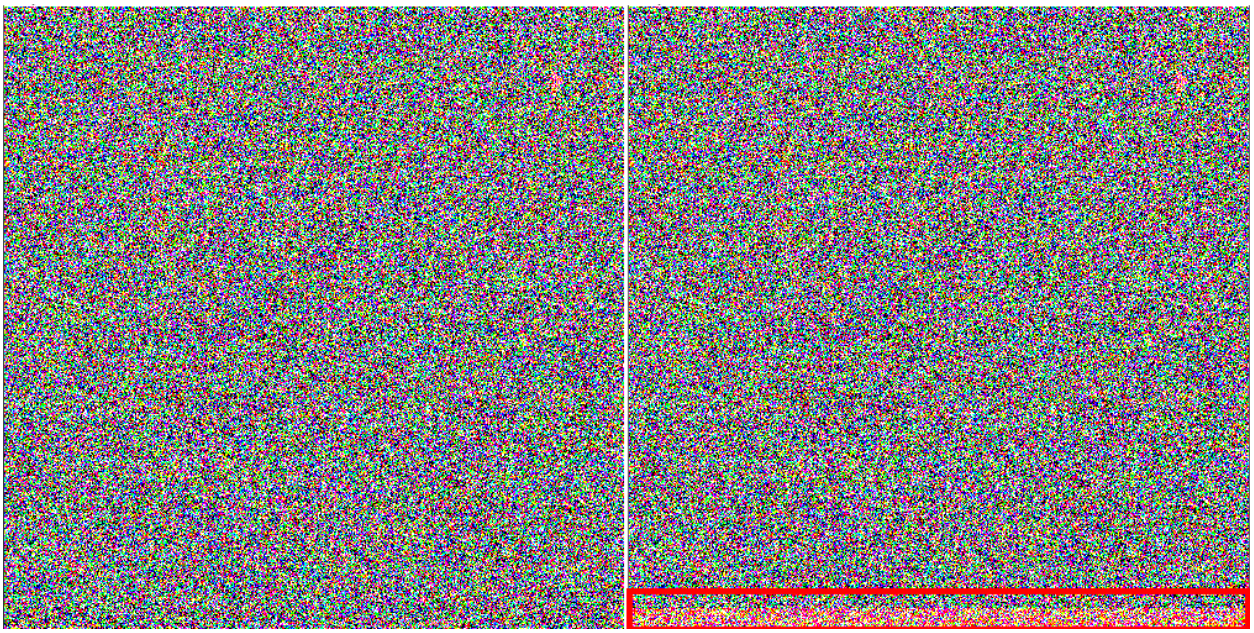
III. Methods of Detecting the Fact of Transfer of Information

There are a number of attacks on image containers that reveal the presence of a hidden message. One of the common attack methods is "visual attack" [24-25]. The essence of the attack is the formation of new images from the low-order bits of the original image with amplification of values (maximization), an example of an image without hiding is shown in Fig. 1(a). At the same time, areas with a high data density usually appear on the images that are generated on the basis of stego-container files with an existing hidden message, as can be seen in Fig. 1(b). The embedded message is located in the areas of visual distortion.

When using a small container image (242 976 bytes), the result of a visual attack is more apparent. For example, when the container is filled by 30% (Fig. 2(b)), one can see a characteristic difference from the result of a visual attack on the original image (Fig. 2(a)).

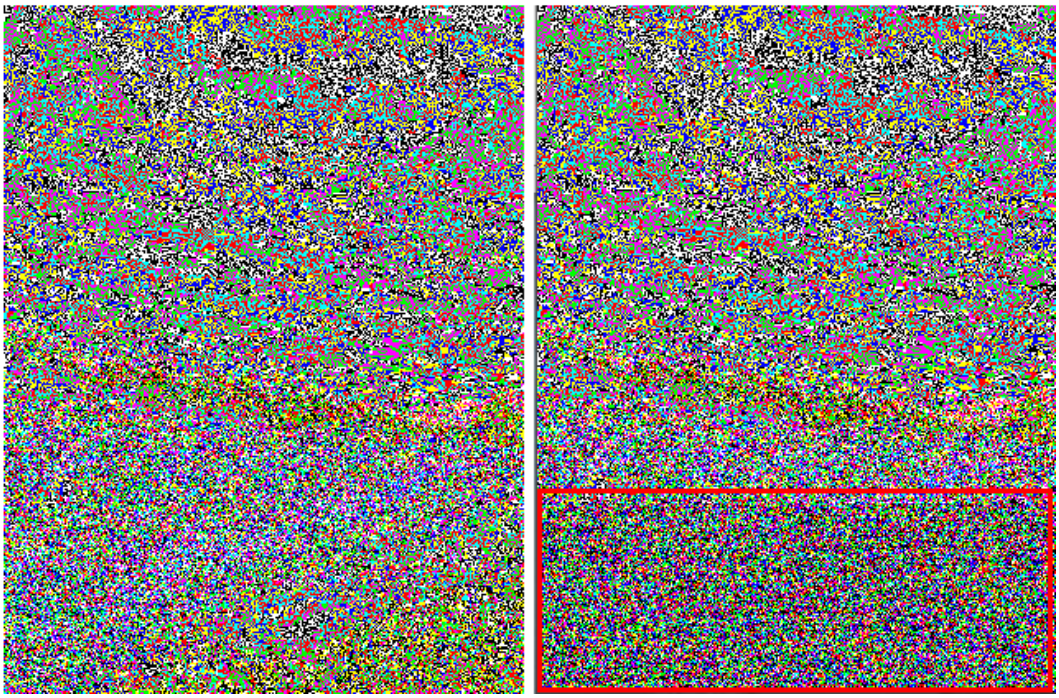
To identify hidden information usually use method that named the Chi-square statistical test [26]. In steganalysis, the use of the criterion, which based on the fact that two neighboring colors (colors are adjacent if they are different only in the least significant bits) differ significantly in the number of points relative to the untransformed image [27]. Fig. 3(a) shows a chi-square rendering of an original container image. The clear difference between the rendering results of the original image and the image processed with classical LSB is shown in Fig. 3(b).

Because both steganalytical methods are based on identifying dependencies throughout the digital object, the research was aimed at finding modifications to the algorithm that would allow disguising the introduction of distortions when placing a steganographic message under random distortions or noise. To do this, it was necessary to achieve two parameters of the algorithm: to increase the coverage of the involved parts of the container, but at the same time to reduce the total number of modified bits. To increase the coverage of the involved areas of the container, a uniform distribution of modified bytes in the container image was used (Fig. 3(c)). This solution only partially improves the overall picture when performing visual attacks, as well as Chi-square analysis. To reduce the number of modified bytes, it was decided to use lossless compression methods.



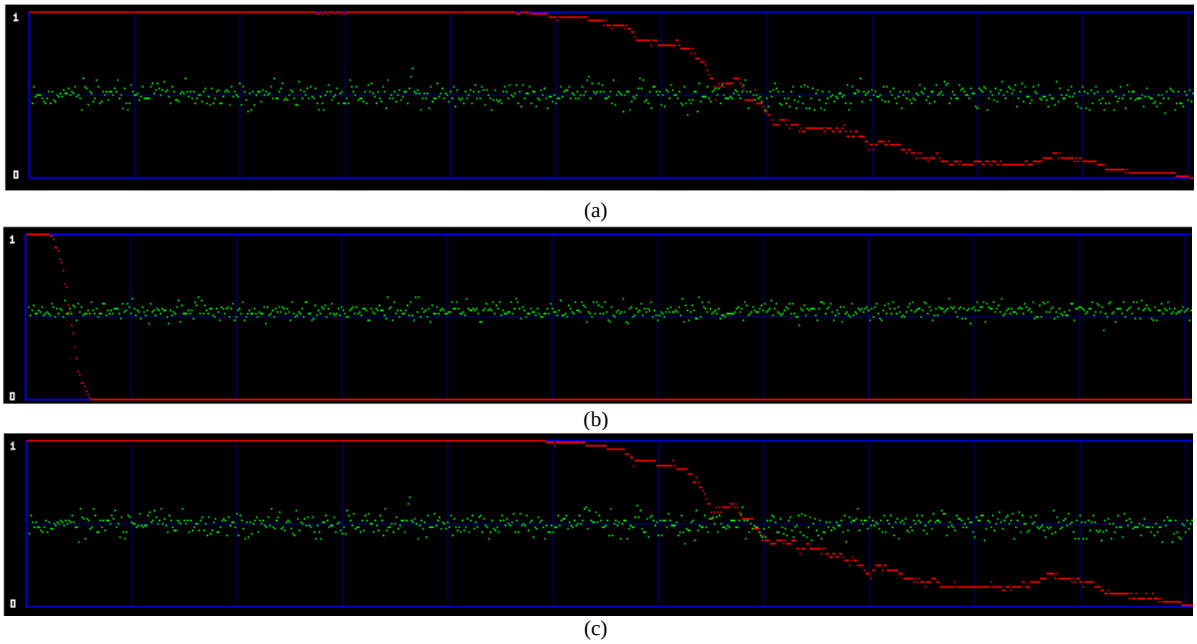
(a) (b)

Fig. 1. The Result of a Visual Attack on Images: (a) – Original Image; (b) – Image Filled with Standard LSB Method.



(a) (b)

Fig. 2. The Result of a Visual Attack on Small Image: (a) – Original Image; (b) – Image Filled with Standard LSB Method.



(a)

(b)

(c)

Fig. 3. Chi-Square Visualization: (a) – The Result of the Analysis of the Original Image; (b) – The Result of the Analysis of the Standard LSB; (c) - is the Result of LSB Analysis using a Substitution Table.

Studies have shown that for small messages, the size of the container has practically no effect on the result of the attacks described above. In other words, the attacks themselves are ineffective in finding low-capacity messages. The study of the results of the modified steganographic algorithm was tested on examples of hiding medium and large messages in the corresponding containers.

IV. Analysis of Compression Algorithms

To solve the problem of compression, several methods, which perform lossless data compression [28], have been considered. All the considered compression algorithms were implemented in the C ++ programming language and tested on several samples of medium and large texts.

Run-length encoding (RLE) [29]: An easy-to-implement algorithm with the best, average and worst compression ratios equal to 1/32, 1/2, 2/1. Test #1 results: text 608 characters long and 1115 bytes in size was converted to 1135 bytes. Test #2 results: a text of 3040 characters long and 5575 bytes in size was converted to 5669 bytes. Test #3 results: text 9120 characters long and 16725 bytes in size was converted to 17028 bytes. The increase in volume is due to the construction of key-value pairs. With a small number of repetitive sequences, the size of the original file will grow. This algorithm shows satisfactory results with texts containing repetitive sequences.

Compression of information based on binary coding trees (Huffman compression) [30]. Test #1 results: text 608 characters long and 1115 bytes in size was converted to 1321 bytes. Test #2 results: a text of 3040 characters long and 5575 bytes in size was converted to 3499 bytes. Test #3 results: text 9120 characters long and 16725 bytes in size were converted to 8944 bytes. With an increase in the volume of text, the compression ratio increases, but the work of the algorithm with medium-sized texts does not give positive results. This specificity is due to the implementation of the algorithm based on binary trees.

Sliding window compression (LZ77) [31]. Test #1 results: text 608 characters long and 1115 bytes in size were converted to 704 bytes. Test #2 results: text 3040 characters long and 5575 bytes in size were converted to 792 bytes. Test #3 results: text 9120 characters long and 16725 bytes in size were converted to 1009 bytes.

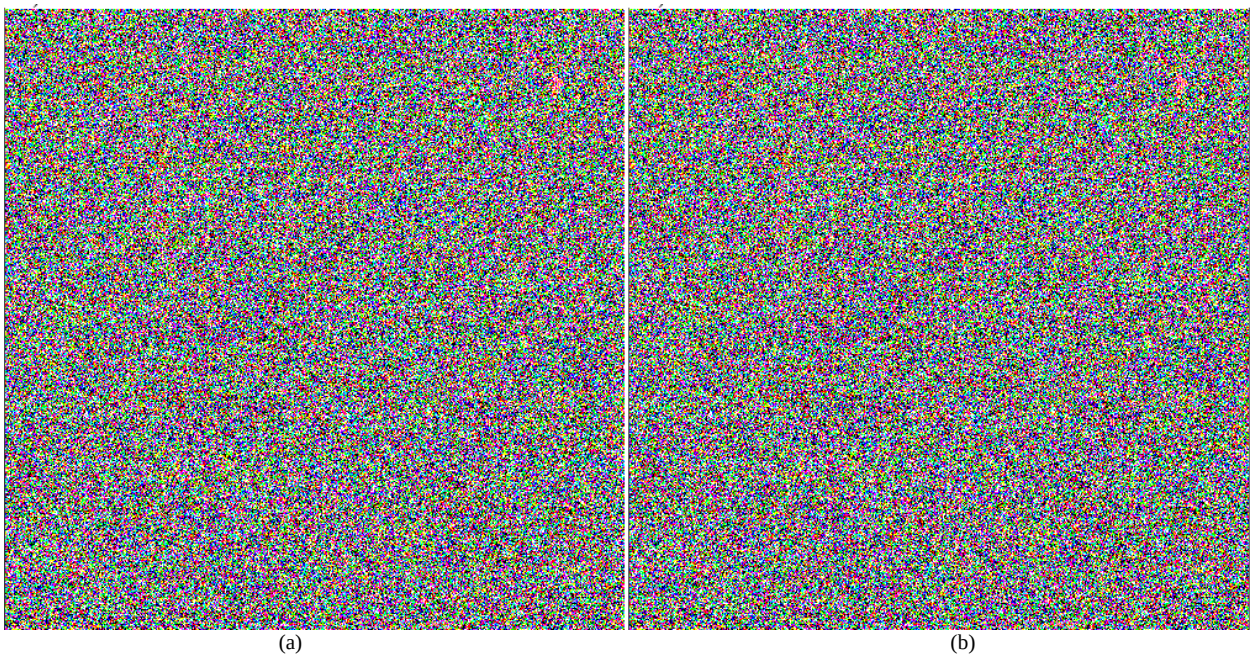
According to the results of the analysis (Table I) and testing of the methods, the LZ77 method is the most optimal for solving the problem of compressing medium and large texts.

TABLE I. COMPARISON OF THE RESULTS OF THE COMPRESSION ALGORITHMS

Compression algorithms	Text No. 1 608 characters, 1115 bytes	Text No. 2 3040 characters, 5575 bytes	Text No. 3 9120 characters, 16725 bytes
RLE	1135 bytes	5669 bytes	17028 bytes
Huffman compression	1321 bytes	3499 bytes	8944 bytes
LZ77	704 bytes	792 bytes	1009 bytes

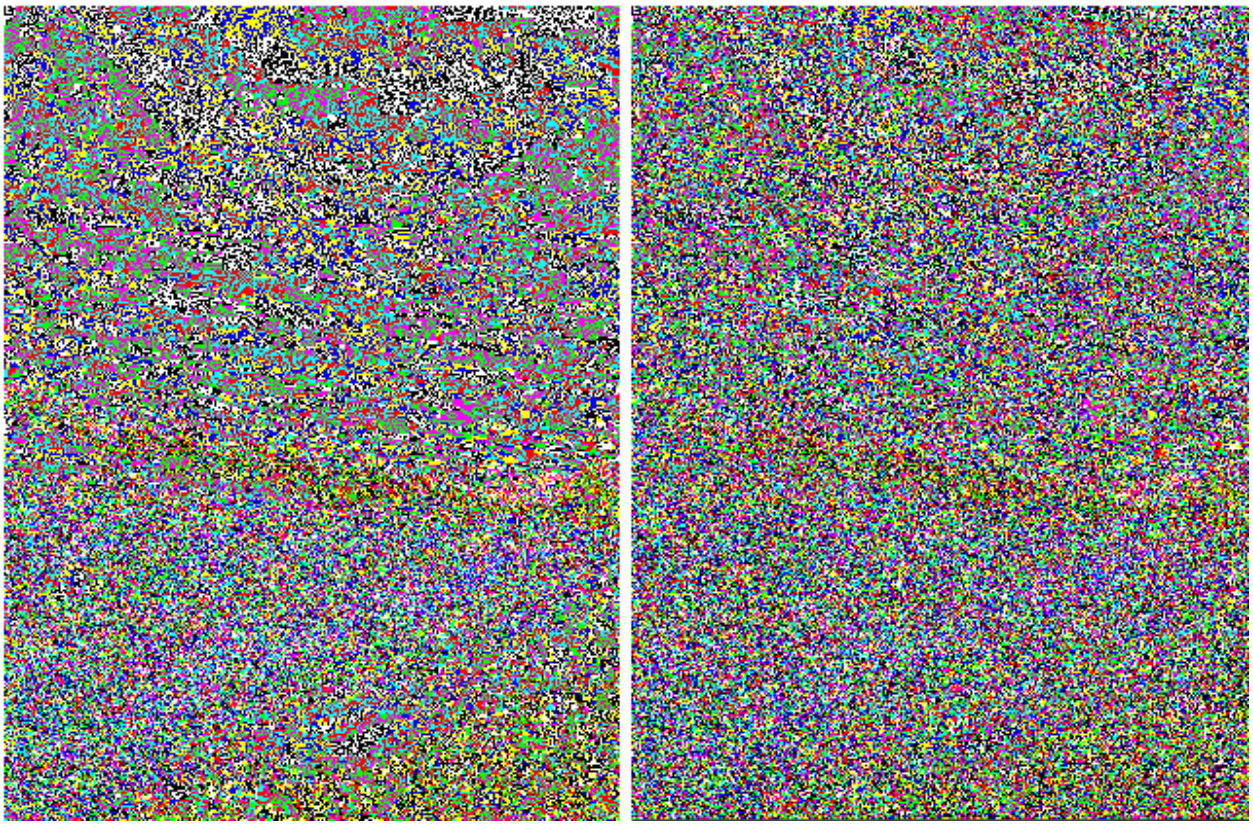
V. KEY-GENERATED REPLACEMENT TABLE

The use of a uniform distribution of compressed data does not sufficiently improve the robustness of the steganographic algorithm, since any steganographic algorithm has the property of symmetry, in other words, it is possible to extract hidden information performing the reverse actions of hiding. To solve this problem, an analogue of the secret key was introduced. A byte replacement table in the container image is formed based on this key. The number of elements in the table is equal to the number of groups of two bits formed from the binary representation of the compressed bytes of the secret message. The replacement table forms a uniform distribution of the modified bytes in the container image (Fig. 4(b), 5(b)), due to which the embedded compressed message is similar to the noise that occurs in the image when it is projected in different channels. The important thing is that the receiver and the sender generate the table independently of each other, using only a shared secret key that both parties know. In this case, the replacement table for both sides is generated the same. This property is very important because transferring and storing such a large table is difficult, and the key is easy to use.



(a) (b)

Fig. 4. Visual Attack on Images: (a) – Original Image; (b) – Image Filled with Improved LSB Method.



(a) (b)

Fig. 5. Visual Attack on Small Images: (a) – Original Image; (b) – Image Filled with Improved LSB Method.

VI. Results of a Computer Experiment

A program that implements an improved version of the LSB hiding method was written in the Python programming language as part of the study. The results of testing this program are shown below. For comparison, the result of executing the original LSB algorithm will be shown. The comparison will be carried out on the same compressed data (the original text is 150 474 characters long, the text size is 150 948 bytes; the compressed text is 58 364 bytes).

VII. The Effectiveness of the Developed Method

For an objective assessment of the improvement of the algorithm, it was decided to use the PSNR (Peak Signal to Noise Ratio) index [32]. The index value indicates the similarity between the two images, and therefore, the higher the value, the more the similarity. When calculating the index, you need to calculate the mean square error (MSE) between the images.

$$MSE = \frac{1}{\mathrm{N}}\sum_{i=1}^{N}(X_i - Y_i)^2$$

X and Y are equal to the values of the original image and the container image, N is the number of pixels in the image.

$$PSNR = 20\log_{10}\frac{MAX}{MSE}$$

MAX is the maximum value that the pixel color can take, equal to 255.

During the study, the index between the original image (Fig. 1(a)) and the image processed with the standard LSB (Fig. 1b) was calculated, which is 57.72449648521831. The index between the original image and the image processed by the improved LSB algorithm (Fig. 4(b)) is 61.39096278660813, which indicates a greater similarity with the original image. There is a slight improvement when calculating the index between the smaller images. So the index between the original image (Fig. 2(a)) and the image processed with the standard LSB (Fig. 2(b)) is - 49.47831069914614. The index between the original image and the image processed by the improved LSB algorithm (Fig. 5(b)) is 50.5708622069228. A brief comparison of the results of the computer experiment is presented in the Table II.

TABLE II. Comparison of PSNR Indices

	Standard LSB algorithm	**Improved LSB algorithm**
Large volume of hidden message, medium container volume	49.47831069914614	50. 5708622069228
Large volume of hidden message, large container volume	57.72449648521831	61.39096278660813

VIII. Conclusion

The study compared two aspects: PSNR index and security. Based on the results of calculations and comparison of visualizations of various types of attacks on images processed by the standard LSB method and images processed by the improved LSB method, we can conclude that the use of a substitution table with a uniform distribution gives a structural result similar to the original container image. Distortion areas are minimized, and fragments with a high data density have disappeared, from which it is possible to calculate the fact of information transfer, an analogue of a secret key has been added, without which an attacker will not be able to extract useful information from the container image. The PSNR index when using the improved algorithm increases by ~ 5.9% in the best case and by ~ 2.16% in the worst case. In addition, visualization of the result of the improved LSB algorithm by the Chi-square criterion also indicates an increase in steganographic resistance. Computer experiments have shown that the improved LSB algorithm is more resistant to visual attacks and the use of statistical analysis methods.

The improved algorithm can be applied to process a large number of images, for example, the covert transmission of information in social networks and services for storing and transmitting images.

In the future it will be analyzed the applicability of various distributions to increase the PSNR index and adapt the algorithm to other data formats, including video file formats.

References

[1] N.F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen", Computer, vol. 31, no. 2, pp. 26-34, 1998.

[2] P.P. Aung and T.M. Naing, "A novel secure combination technique of steganography and cryptography", International Journal of Information Technology Modeling and Computing (IJITMC), vol. 2, no. 1, pp. 55-62, 2014.

[3] B. L. Sirisha, S. S. Kumar and B. C. Mohan, "Steganography based information security with high embedding capacity," 2015 National Conference on Recent Advances in Electronics & Computer Engineering (RAECE), Roorkee, India, 2015, pp. 17-21, doi: 10.1109/RAECE.2015.7510218.

[4] A. Shamir, "How to share a secret", Communication ACM, vol. 22, no. 11, pp. 614-613, 1997.

[5] J. Bloom, I. Cox, J. Fridrich, T. Kalker and M. Miller, Digital watermarking and steganography, San Francisco, CA, USA:Morgan Kaufmann Publishers, Inc., 2007.

[6] P. Johri, A. Mishra, S. Das and A. Kumar, "Survey on steganography methods (text, image, audio, video, protocol and network steganography)," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2016, pp. 2906-2909.

[7] N. Subramanian, O. Elharrouss, S. Al-Maadeed and A. Bouridane, "Image Steganography: A Review of the Recent Advances," in IEEE Access, vol. 9, pp. 23409-23423, 2021, doi: 10.1109/ACCESS.2021.3053998.

[8] Changda Wang; Shiguang Ju (2008). "Book Cipher with Infinite Key Space". 2008 International Symposium on Information Science and Engineering. p. 456. doi:10.1109/ISISE.2008.273. ISBN 978-0-7695-3494-7. S2CID 15768123.

[9] A. Kuznetsov, K. Shekhanin, A. Kolhatin, I. Mikheev and I. Belozertsev, "Hiding data in the structure of the FAT family file system," 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kiev, 2018, pp. 337-342, doi: 10.1109/DESSERT.2018.8409155.

[10] K. Shekhanin, A. Kolhatin, K. Kuznetsova and S. Kavun, "Steganographic hiding information in a file system structure," 2018 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), Odessa, Ukraine, 2018, pp. 1-6, doi: 10.1109/UkrMiCo43733.2018.9047551.

[11] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding", IBM Syst. J., vol. 35, pp. 313-336, 1996.

[12] C. Zhi-li, H. Liu-sheng, Y. Zhen-shan, L. Ling-jun, and Y. Wei, "A Statistical Algorithm for Linguistic Steganography Detection Based on Distribution of Words," in Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, pp. 558-563, 2008.

[13] Fridrich, M. Goljan and R. Du, "Detecting LSB steganography in color and gray-scale images", IEEE Multimedia, vol. 8, no. 4, pp. 22-28, 2001.

[14] T. Sharp, "An implementation of key-based digital signal steganography", Proc. of the 4th Information Hiding Workshop, pp. 13-26, 2001.

[15] F. Petitcolas, R. Anderson and M. G. Kuhn, "Information hiding – a survey", Proceedings of the IEEE, vol. 87, no. 7, pp. 1062-1078, 1999.

[16] Mazurczyk, Wojciech & Smolarczyk, Milosz & Szczypiorski, Krzysztof. (2011). Retransmission steganography and its detection. Soft Comput.. 15. 505-515. 10.1007/s00500-009-0530-1.

[17] Deepesh Rawat Vijaya Bhandari "Steganography Technique for Hiding Text Information in Color Image using Improved LSB Method" International Journal of Computer Applications Vol. 67 No. 1 April 2013 pp. 22-25.

[18] Shailender Gupta Ankur Goyal Bharat Bhushan " Information Hiding Using Least Significant Bit Steganography and Cryptography I.J. Modern Education and Computer Science 2012 Vol. 6 pp. 27-34.

[19] K. Thangadurai and G. Sudha Devi, "An analysis of LSB based image steganography techniques," 2014 International Conference on Computer Communication and Informatics, Coimbatore, India, 2014, pp. 1-4, doi: 10.1109/ICCCI.2014.6921751.

[20] V. Verma, Poonam and R. Chawla, "An enhanced Least Significant Bit steganography method using midpoint circle approach," 2014 International Conference on Communication and Signal Processing, Melmaruvathur, India, 2014, pp. 105-108, doi: 10.1109/ICCSP.2014.6949808.

[21] N. M. Abdali and Z. M. Hussain, "Reference-free Detection of LSB Steganography Using Histogram Analysis," 2020 30th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, Australia, 2020, pp. 1-7, doi: 10.1109/ITNAC50341.2020.9315037.

[22] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2012, pp. 234-239, doi: 10.1109/WIFS.2012.6412655.

[23] Q. Liu, A. H. Sung and M. Qiao, "Detecting information-hiding in WAV audios," 2008 19th International Conference on Pattern Recognition, Tampa, FL, USA, 2008, pp. 1-4, doi: 10.1109/ICPR.2008.4761650.

[24] R. Chandramouli and Nasir Memon, "Analysis of LSB Based Image Steganography Techniques", IEEE, no. 1019–1022, February 2001.

[25] A. Arora, M. P. Singh, P. Thakral and N. Jarwal, "Image steganography using enhanced LSB substitution technique," 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 2016, pp. 386-389, doi: 10.1109/PDGC.2016.7913225.

[26] M. Göl and A. Abur, "A modified Chi-Squares test for improved bad data detection," 2015 IEEE Eindhoven PowerTech, Eindhoven, Netherlands, 2015, pp. 1-5, doi: 10.1109/PTC.2015.7232283.

[27] Stanley, C.A., "Pairs of Values and the Chi-squared Attack", Department of Mathematics, Iowa State University (2005).

[28] R. J. van der Vleuten, "Low-complexity lossless and fine-granularity scalable near-lossless compression of color images," Proceedings DCC 2002. Data Compression Conference, Snowbird, UT, USA, 2002, pp. 477-, doi: 10.1109/DCC.2002.1000020.

[29] D. S. Bhadane and S. Y. Kanawade, "Comparative study of RLE & K-RLE compression and decompression in WSN," 2016 3rd International

Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2016, pp. 1-5, doi: 10.1109/ICACCS.2016.7586319.

[30] B. Ergude, L. Weisheng, F. Dongrui and M. Xiaoyu, "A Study and Implementation of the Huffman Algorithm Based on Condensed Huffman Table," 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 2008, pp. 42-45, doi: 10.1109/CSSE.2008.1432.

[31] C. Fraser, "An instruction for direct interpretation of LZ77-compressed programs", Technical report MSR- TR-2002-90, 2002.

[32] K. Joshi, R. Yadav and S. Allwadhi, "PSNR and MSE based investigation of LSB," 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), New Delhi, India, 2016, pp. 280-285, doi: 10.1109/ICCTICT.2016.7514593.