ADVANCES IN CYBER-PHYSICAL SYSTEMS

VOL. 7, NUM. 1, 2022

SOFTWARE IMPLEMENTATION OF MODIFIED LSB ALGORITHM WITH SHAMIR`S SECRET SHARING

Maksym Pavlov, Iryna Yurchak

Lviv Polytechnic National University, 12, Bandera Str., Lviv, 79013, Ukraine.
Authors' e-mail: maksym.pavlov.mkiks.2021@lpnu.ua; iryna.y.yurchak@lpnu.ua

https://doi.org/10.23939/acps2022.__.___
Submitted on 30.09.2022

Abstract: Today, it is often necessary to transmit a confidential message of a small volume, while the use of complex cryptographic systems is difficult for some reasons. One of these reasons is the impossibility of using reliable products, which, as a rule, are commercial and unavailable to the average computer user. In modern information society, many services are provided with the help of computer networks and information technologies. Information presented in digital form must be reliably protected from many threats: unauthorized access and use, destruction, forgery, leakage, violation of license agreements, disclaimer of authorship, etc. Information protection is extremely important in both commercial and government spheres. The issues of developing effective methods of protecting digital information, in particular methods of computer steganography and steganalysis, are relevant and important for the state and society. To achieve the goal, it is necessary to propose a method of increasing stego-resistance, determine the effectiveness of the created solution and analyze the obtained results. The object of research is the process of protecting information embedded in a graphic e-container. The subject of research is methods and algorithms of computer steganography and steganalysis for images. The research methods used in this work are based on steganographic algorithms.

Index Terms: steganography, least significant bit, Shamir's secret distribution, digital image steganalysis.

I. INTRODUCTION

Steganography is the science of transmitting information in a hidden form by keeping the very fact of transmission secret [1]. The main task is to ensure that a person does not suspect that some message is hidden inside the transmitted information. The information transmitted does not appear to be of any external value, but it contains hidden valuable information. Thus, steganography allows the transmission of secret information through open channels, hiding the fact of its transmission.

Currently, steganographic systems are widely used to solve such tasks as:

- Protection of information from third-party access
- Hiding information from monitoring systems
- Hiding software
- Protection of intellectual property

The goal of classical steganography is to hide secret data [2] in other open data sets or data streams in a way that does not reveal that they have any hidden component, thereby distinguishing these messages from others. It could be said that steganography is the art and science of ways of transmitting (storing) messages that hide the fact of the existence of a hidden communication channel (hidden data).

One of the most common methods is Least Significant Bit Insertion (LSB) [3], a common, simple approach to embedding information into a container. The last bit of each pixel is replaced by a secret message bit. When using a 24-bit image, each of the red, green, and blue color components [4] can be used, each represented by a byte [2].

A slight modification of this steganographic technique allows two or more low-order bits per byte to be used to create a message. This includes the amount of received information in object containers, but the secrecy is greatly reduced [5], which facilitates the process of steganography recognition. Other variations of this method include statistical image changes [6]. Some intelligent steganalysis software checks for regions that consist of a single solid color [7]. To increase stealth, you should avoid recording changes in these pixels.

Alternatively, secret sharing schemes based on secrets are shared among the group of participants, and without the knowledge of all participants involved in the encryption, the image cannot be retrieved [8]. Additionally, there is an advantage of this secret sharing scheme. Without the knowledge of complete subsets of secrets, it is not possible to obtain the information regarding secrets. Image encryption is proposed based on Shamir's secret sharing [9] scheme and this method is applied to black and white, gray and color images.

There are methods of statistical and structural analysis that allow revealing hidden information. These methods are based on the idea that stego-images can be detected by anomalous changes [10] in the parameters of the cover image. However, these methods differ in their approaches to image parameter estimation - using statistical or structural analysis methods.

This paper presents an overview of a developed modification of the LSB algorithm combined with Shamir's secret distribution to improve the robustness of embedded messages to attacks.

II. PROBLEM STATEMENT

Currently, with the wide use of digital multimedia formats and the problems of digital resource management, research in the field of steganography is becoming more and more relevant. In the conditions of the developed network communication infrastructure of users of the global computer network, it is also relevant to solve the problem of hidden information, with the development of which it is possible to transfer electronic documents quickly and economically to different corners of the earth, a large amount of transferred material is often accompanied by illegal reproduction and distribution. As a result, this forces us to look for ways to hide copyright information in a variety of text, image, audio, video, and other file types.

Today, there are quite a few software products that are used for steganography and the implementation of techniques for entering sensitive data into various types of files.

The classic task of steganography is to organize the transmission of secret messages so that the message's content and the transmission are hidden from all but the parties involved. To solve such a problem, a special message called a stego-container is used, in which the secret message necessary for transmission is embedded. At the same time, the developers of steganographic methods must organize the transparency of confidential data transmitted: changing a certain number of bits of information in the container should not lead to a specific loss of its quality (there should be no built-in visualization artifacts). The most used containers are files containing digital photos, text, music, and video. So, for example, when using a graphic file as a container for an outside observer, the message transfer process will be treated as a normal exchange of digital graphic files. At the same time, it is important to remember the importance of observing one condition: no one should have access to both the source file chosen as the container and the file containing the hidden messages, because in this case, a simple comparison of the files will immediately reveal the existence of the message. As it has been mentioned above, almost any file format can serve as a container in computer steganography, but the most common type of media is BMP image files. This is because the best file formats for steganography purposes are those that use lossless compression (this type of compression is often used for images in BMP, TIFF, PNG, TGA, etc. formats). In addition, the high quality of the images and the simplicity of the format are positive aspects in favor of choosing the BMP format.

III. PURPOSE OF THE WORK

Today, on the Internet, you can find a lot of free or shareware steganography software. Algorithms, which are the basis of such programs, embed a confidential message in so-called containers (images, audio and video). The use of such programs makes it possible to transmit any closed information simultaneously with open (visible) non-confidential information over open communication channels, imperceptibly for third parties. The invisibility of such data transfer can be used to implement criminal intentions. Steganalysis allows to prevent unauthorized transmission of information using steganography methods. The main task of stego-analysis is to establish the fact of the existence of hidden information in the container. In general, detecting the hidden transmission of data hidden by one of many existing steganography methods in various container formats is a rather complex process. For example, the use of the well-known method of stego-analysis based on the chi-square test allows to obtain good results if information interpolation was conducted by the method of sequential replacement of the least significant bits of the elements of the image container or by the method of interpolation with filling, however, this method does not work when pseudo-random selection of the youngest bits occurs (distributed scattering). As you may know, the reliability of the message transmission in the container drops rapidly with the increase in the size of the message that is embedded in the container, which is quite difficult to prevent. This means that if an attacker intercepts an overflowing container, it will be quite easy for him to extract the message from the container.

The purpose of this work is to improve the reliability of hiding information, using the LSB method, in a stego-container by using the Shamir secret sharing.

IV. LSB METHOD ALGORITHM

LSB - Least Significant Bits, the essence of this method is to hide information by changing the last bits of the image, which encode the color, to the bits of the hidden message [2]. The difference between empty and filled containers should not be perceptible to human perception. The principle of hiding information is shown in Fig. 1.

The stage of converting text into a byte sequence is as follows. As it has been described earlier, in the BMP format [1], an image is stored as a matrix of color hue values for each point of the stored image. If each of the components of the RGB space (they are also called color channels) is stored in one byte, it can take on values from 0 to 255 inclusive, which corresponds to a 24-bit color depth. The peculiarity of human vision is that it poorly distinguishes minor variations in color [3]. For 24-bit color, a change in each of the three channels of one least significant bit (that is, the rightmost bit) leads to a change of less than 1% of the intensity of a given point, which allows you to change them imperceptibly to the eye at your discretion. Let's calculate the bandwidth of the method. If we ignore the service information at the beginning of the file, which is usually insignificant with the size of the image, then we have the opportunity to secretly transmit a message with a size of 1/8 of the size of the container ("smear" by the last bits in each byte of the pixel color matrix) or a size of 1 / 4 containers (respectively when using the last 2 bits in bytes) (see Fig. 2).

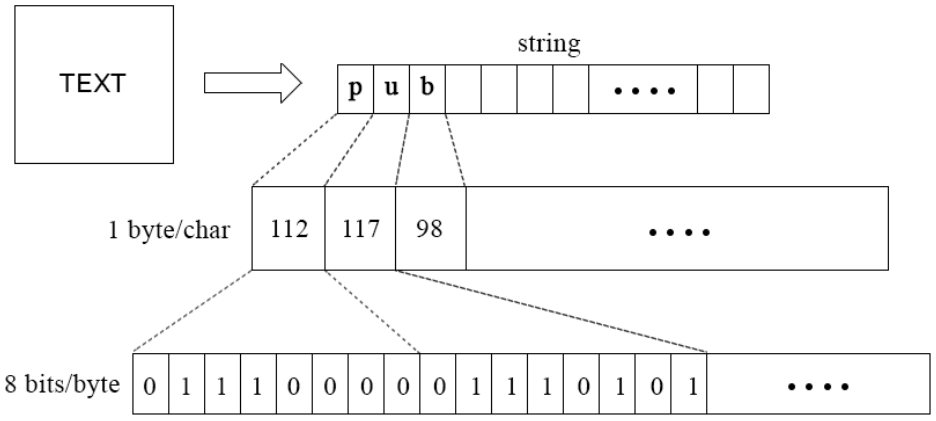

Fig.1. Data representation in the computer.

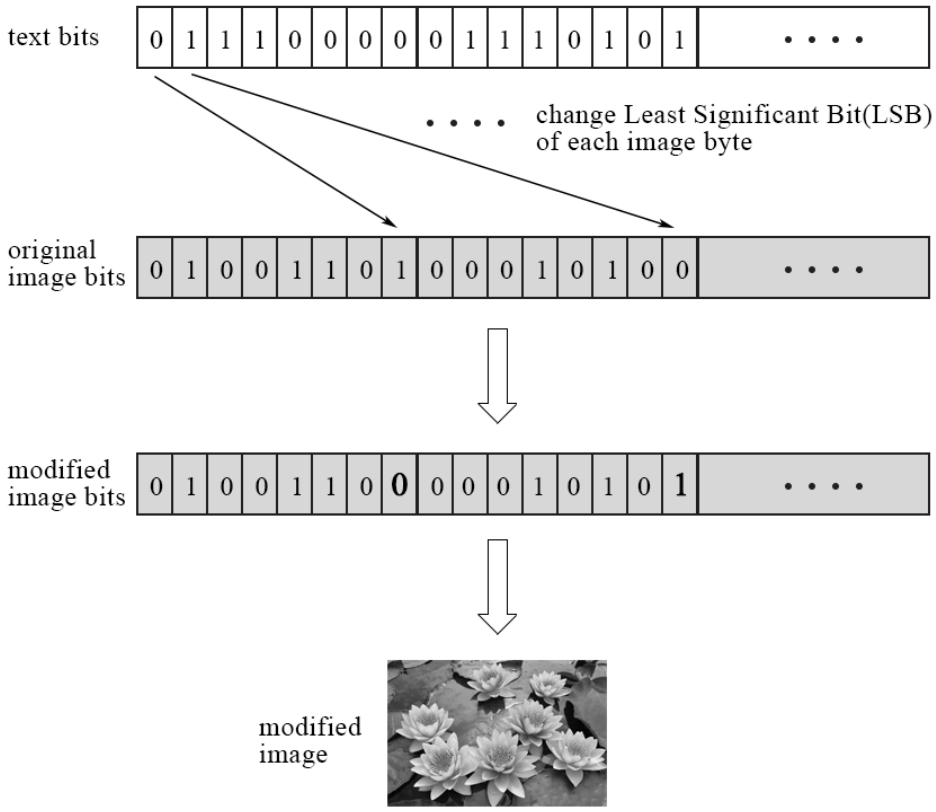

Fig.2. Hiding information in the image.

The working principle of the steganographic method is as follows. Let there be a 24-bit grayscale image. A pixel is coded by 3 bytes, and they contain the values of the RGB channels [7]. By changing the least significant bit, we change the value of the byte by one. Such gradations are imperceptible to humans and may not be display at all when using low-quality output devices. The following example (see Fig. 3 and Fig. 4.) shows how a message (01000001) can be hidden in the first eight bytes corresponding to three pixels in a 24-bit image [4].

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

Fig. 3. – Pixels without embedded message.

(00100110 11101001 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

Fig.4. – Pixels with an embedded message.

In the example, only the three bits that were changed are underlined. Applying the LSB steganographic method requires, on average, that only half the bits of the container image are changed [2]. A slight modification of this steganographic technique allows two or more low-order bits per byte to be used to embed the message. This increases the volume of hidden information in the object container, but the secrecy is greatly reduced, which facilitates the process of steganography recognition.

LSB methods are unstable to all types of attacks and can be used only when there is no noise in the data transmission channel. Detection of the LSB-coded container is conducted based on anomalous characteristics of the distribution of the values of the range of the lower bits of the digital signal counts.

V. SHAMIR'S SECRET DISTRIBUTION SCHEME

The idea on which Shamir's scheme is based is that interpolation of a polynomial of the degree $k-1$ requires k points [8]. If fewer points are known, then interpolation will be impossible. Let us denote: p is a large prime number (larger than any secret M that is supposed to be divided in this scheme). The $M \in Z_p : n$ - number of particles of the secret; k is the minimum size of the allowed group. The work of the algorithm can be divided into 3 stages [9]:
Preparatory stage: the dealer randomly selects coefficients $S_1,...,S_{k-1} \in Z_p$ and makes a secret polynomial:

$$S(x) = S_{k-1}X^{k-1} + S_{k-2}X^{k-2} + \ldots + S_1X + M \bmod p,$$

where M is the separating secret, and the other coefficients are arbitrary elements of the field (the dealer keeps the polynomial coefficients secret). It is obvious that: $S(0) = M$. Next, the dealer chooses n different non-secret non-zero elements, $r_1, r_2, r_3, ..., r_n$ each of which corresponds to one participant in the scheme.

The stage of secret distribution is as follows. The dealer calculates the value of the following polynomial:

The stage of secret distribution is as follows. The dealer calculates the value of the following polynomial:

$$c_1 = S(r_1), c_2 = S(r_2), \ldots, c_n = S(r_n).$$

The share of each user A_i is a pair of numbers $(r_i, c_i), i = 1, 2, \ldots, n$. Shares are distributed to participants of the scheme.

To restore the secret, you need to use Lagrange's interpolation formula [8]: if you need to construct a polynomial $S(x)$ of degree $k-1$, which takes the value $x_1, x_2, \ldots, x_k$ corresponding to $y_1, y_2, \cdots, y_k$ then this polynomial will be:

$$S(x) = \sum_{j=0}^{k-1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

k as the polynomial in the secret distribution, the scheme should be chosen so that $S(0) = M$, then the Lagrange formula follows:

$$M = \sum_{i=0}^{k-1} c_i S_i,$$

where

$$S = \prod_{j \neq i} \frac{r_j}{r_j - r_i}.$$

From the above, it becomes clear that for larger threshold values, the calculation becomes slower.

VI. DESCRIPTION OF THE DEVELOPED ALGORITHM

Preparation of input data: the container in which the distribution of the secret will be interspersed; the key for extracting the message (distribution) from the container; the message that needs to be transmitted reliably; the number of distribution sides; the number of distribution parties required to recover the original message from the distributions.

Perform a distribution using Shamir's secret distribution scheme to the initial message with the specified distribution parameters: the number of sides to split and the number of sides to restore the split.

You need to specify a key that controls the interpolation of information into the container using a pseudorandom number generator. The generator in turn affects the choice of interpolation and reverse extraction bits.

All distributions are interpolate in a copy of the initial container, which are transferre to the participants.

Obtaining stego-containers with distributions in the number specified in the parameter corresponding to the number of participants of the distributions, due to any sampling of them in the number specified in the parameter corresponding to the number of participants required to recover the message.

VII. SOFTWARE APPLICATION DEVELOPMENT

For images of BMP format, the method of replacing the least significant bit is implemented. This method is very unstable to any distortions of the container, as well as to its compression, which destroys all hidden information. In addition, the introduction of information in this way is easily detected by steganographic attacks. For a greater degree of protection of the information hidden in the image by the substitution method, cryptographic methods can be used that will encrypt the message before its embedding, and, in addition, the Shamir secret distribution scheme is applied, while imposing a key. In this way, the information will be protected by four levels of security, which will ensure the concealment of the fact of the transmission of the message, and the impossibility of deciphering it if it is detected. Along with cryptography, it is possible to develop and apply algorithms that will ensure the selection of a pixel for introducing a message bit into it according to some specific rule. This approach, in comparison with the sequential selection of pixels for the introduction of information, will ensure the protection of the message from decryption even when extraneous information is detected in the container since the bits of the message will be removed from the image in a random order determined by the algorithm. Such a modification of the method of replacing the least significant bit is applied in this work by using a function that generates a random value of the ordinal pixel numbers. After loading the

image, the useful capacity of the container is calculated. The text entered by the user is presented in binary form according to the UTF-8 encoding, in which the characters of different alphabets are encoded by a different number of bytes. For example, the letters of the Ukrainian alphabet are encoded with two bytes, that is, sixteen bits, and the letters of the Latin alphabet, Arabic numerals, and punctuation marks are encoded with one byte.

Next, the image, which is the container, is split into R, G, and B components. The generation of the pixel order for the change occurs according to a function that generates pseudo-random numbers to be used as the sequence numbers of the components in a vector containing all components of all pixels. The function generates pseudo-random numbers concerning an initial value specified by the program. The sequence of message bits is introduced in a pseudo-random order. In each component, one least significant bit is changed, that is, the one that is the smallest. When zero is entered into the zero least significant bit and a unit is entered into the least significant bit equal to one, the value of the component does not change. This is considered when calculating the peak signal-to-noise ratio. The peak signal-to-noise ratio is calculated using the calculated modified image elements. An image with implemented information is created by copying the downloaded image for implementation and making the necessary changes to it. The original image remains original, and the image with embedded information is preserved. Extraction of data from the stego-container takes place in the same stages that are used when entering data, only in the reverse order. The components of all image pixels are recorded in the following order R, G, and B into one vector. The length of the message is calculated. This value is written to the beginning of the message array, so it can be retrieved. Once the length of the message is obtained, the entire message can be extracted.

Fig. 5 shows the interface of the developed program. It consists of three main parts: area for selecting an image, area for displaying an image, area with key selection.

The key may be selected from the specified file, or just entered in the proper field. The area with the main functionality contains three tabs “Embade” - to embed, to select the message to be embedded in the container, and two parameters that are responsible for the number of distributions and the required number of containers for successful message recovery; “Exclude” – for excluding the message from the image; “Restore” – to restore from multiple files.

From the main window, it is possible to insert a message with a preliminary distribution according to the Shamir scheme, to extract a message from the stego container using a key, to restore the initial message, if there is a required number of a secret distributions without which the initial secret cannot be restored. To start the work demonstration, it is necessary to select a file container for the message by clicking the "Select" button. After selecting a container, it will be displayed in a special place

To specify a non-sequential embedding in the container, it is necessary to specify a key for specifying the message embedding sequence algorithm. The key can be specified manually or selected from a text file. This key makes it difficult for an attacker to extract the message from the container if intercepted: an incorrect key provides the wrong order of characters in the embedded message.

The next step is to specify the text of the message. This is done in one of two possible ways. The first method consists of the fact that the user manually types the text of the message from the keyboard, while the second method requires the presence of the message in a text file of .txt format. It is also worth paying attention to the size of the container to roughly understand the maximum possible length of the message that will be processed by the algorithm and embedded in the container.

The next stage is setting the parameters for Shamir's secret distribution scheme. It is necessary to specify the number of distribution participants and the required number of participants to restore the message. This allocation guarantees that the data will not be restored if there are fewer allocations than the number of participants to restore, which were specified before the allocation in the corresponding field. Due to the Shamir scheme, the application adds reliability if an attacker intercepts one of the stego containers. Even if the attacker extracts the contents of the container, he will receive only one of the distributions, which by itself does not provide any information.

By clicking the "Embed" button, the message is embedded in the containers with an additional key overlay, after processing by Shamir's secret distribution scheme. The output of the algorithm produces files that are filled with containers containing the distribution. Files are stored in the number specified in the number of distributions.

To extract embedded distributions from stego-containers using a key, you need to go to the Extract tab, load the stego-container with the embedding, specify the key that was set when embedding, and go to the Extract tab. Fig. 6 shows an example of extracted information from one of the stego-containers. As you can see, there is no information that even slightly resembles the initial message. The appearance of each secret distribution is uninformative for an attacker and guarantees that the original message cannot be recovered.

At the stage of embedding, the number of distributions was specified as four, with the number of distributions for recovery - three. This means that after extracting data from three different containers. ~~It~~ Just three containers will be enough to restore the message, so firstly they must be combined to reconstruct the original message. To do this, you need to go to the "Recovery" tab and paste in the text field some number of partitions removed from the containers.

After re-verification of the given roses, it is necessary to press the button "Restore". As a result, the recognition of the cob informing (secret) behind the roses, vectorizing the Lagrange interpolation polynomial, which is victorious in the scheme of sharing Shamir's secret at the stage of renewal. The result of the operation is shown in Fig. 7.

The restored message completely matches the initial one, which indicates that the application is working correctly.

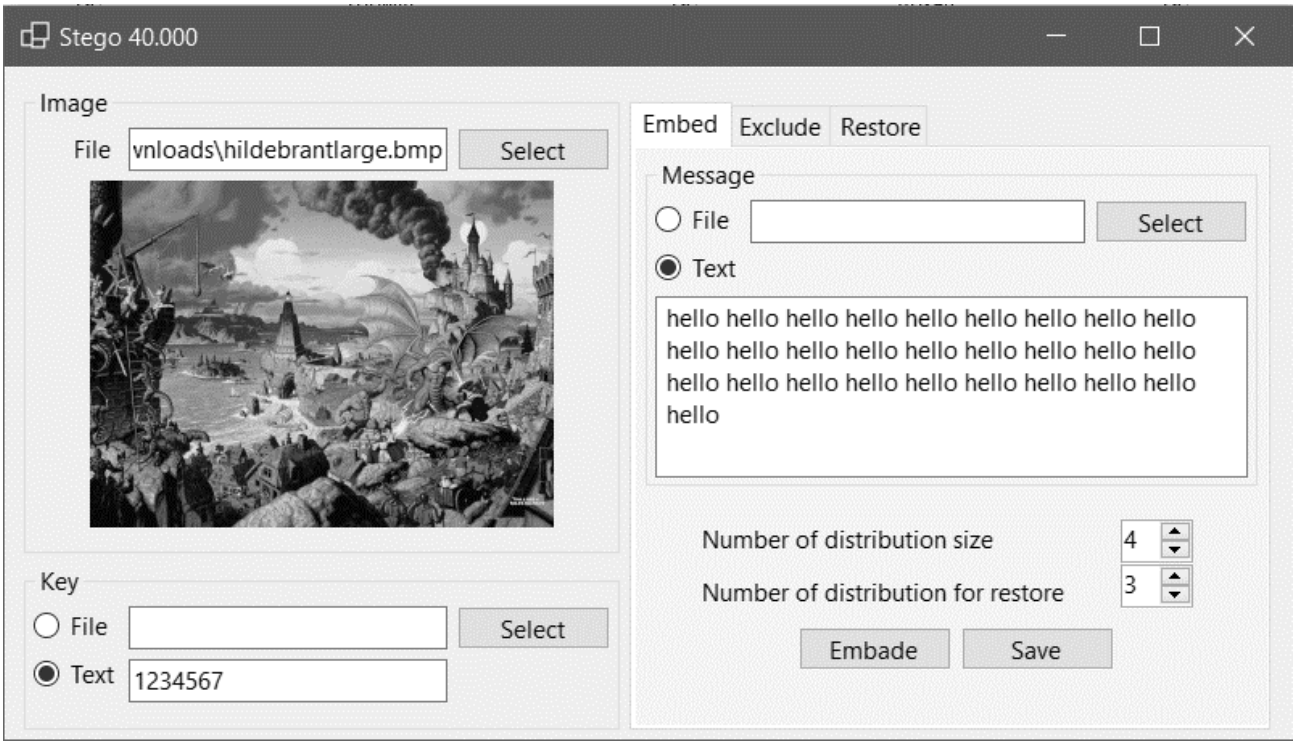

Fig.5. Interface for entering a message.

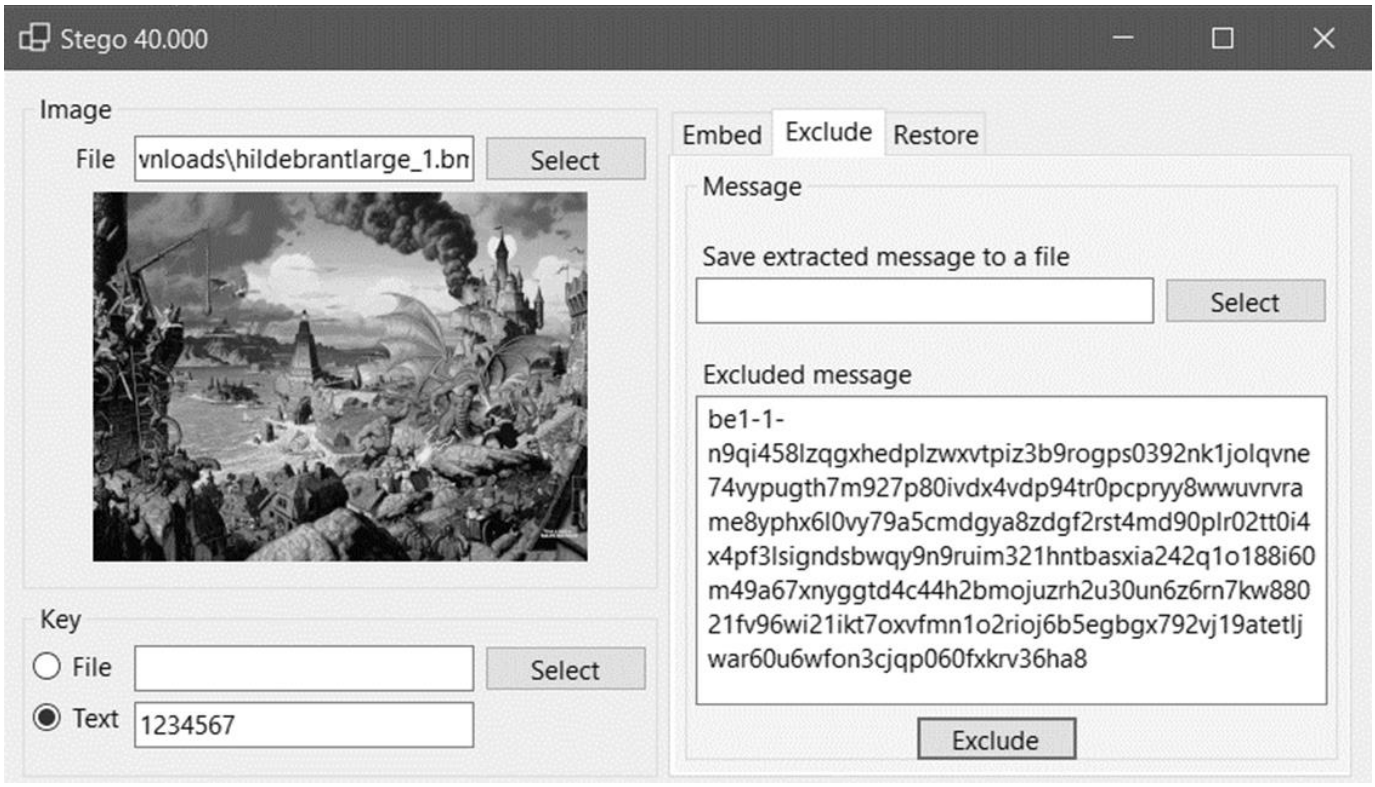

Fig.6. Interface for extracting the message.

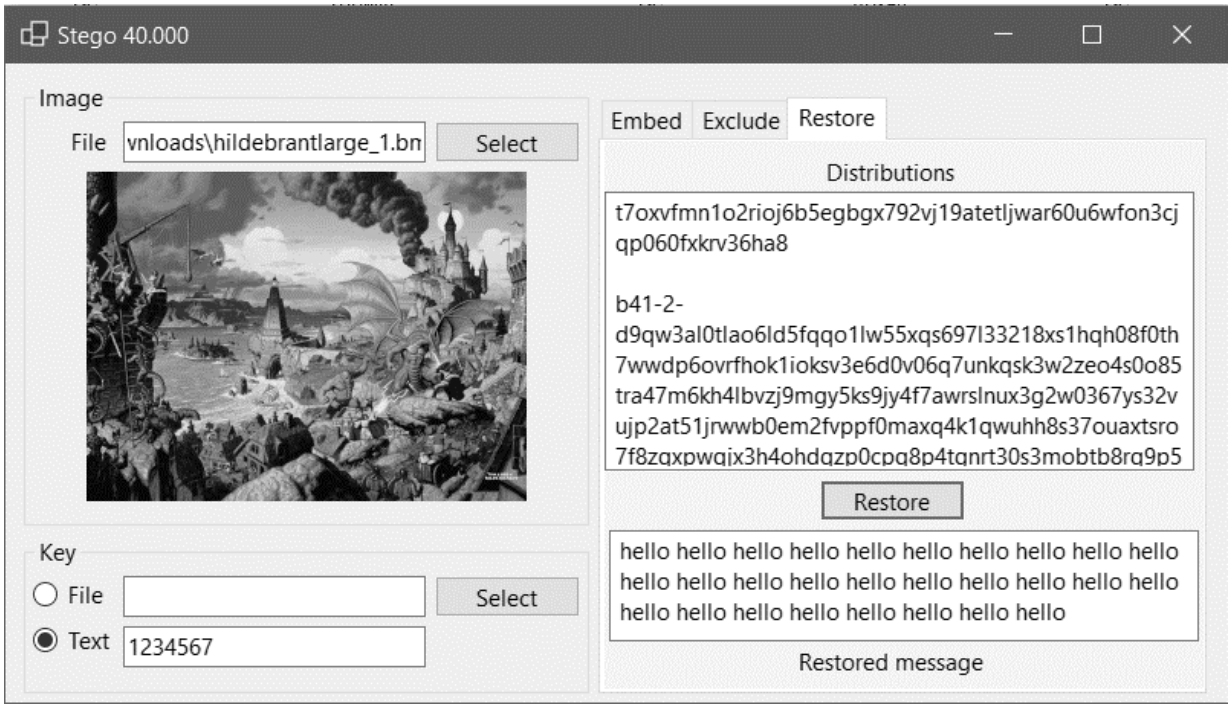

Fig.7. Interface for restoring the message.

VIII. APPLICATION TESTING

To analyze the results of the program, a chi-square attack will be performed, followed by an RS attack. These attacks are statistical methods of stego-analysis, which compare pixel regions, and ideally, detect the presence of a message in the case of a regular serial LSB interpolation. For the experiment, let's take a stego-container in which the first distribution of the secret according to Shamir's scheme is interspersed (see Fig.6)

The size of this distribution segment is 312 bytes, since its length is 312 characters, and the encoding is in the UTF-8 system, hence the size of 312 bytes is obtained. The size of the container is 2.01 kB, which is 2010 bytes. That is, the message is 16% of the size of the container. Chi-square and RS-attack methods should tell us the amount of information hidden in the container submitted for processing by stego-analysis methods. The results of the attacks are shown in Fig. 8 and Fig. 9 images.

As it can be seen from the above figures, the results of the attacks are false, which indicates fairly reliable hiding for the given length of the message embedded in the container. The Chi-Square attack method did not detect any embedded information, and the RS attack only recognized 4.5% of the data from the container's volume, when in fact 16% of the container's size was there.

It can be seen from the above results that the chi-square attack accurately detected the size of the message and the RS attack with a small error. So, according to the analysis of two types of attacks, the proposed algorithm performs the task better.

To compare the effectiveness of the modified and the basic method, 100 chi-square and 100 RS attacks were conducted on containers with different amounts of embedded information. Attacks were conducted 20 times for containers, filled using the base LSB implementation and using the modified LSB method, on 15%, 30%, 45%, 60%, and 75% volume of the container. The size of the container taken for testing is 2010 bytes. The results with the average amount of detected information for each container occupancy level are shown in Table 1 and Table 2.

To compare the effectiveness of the method, let's do the same steps, but the interpolation of the same data will be carried out by the usual sequential method of the least significant bit. The results of a chi-square attack and an RS attack on such a container are shown in Fig. 10 and Fig. 11, respectively.

Table 1.

Chi-square attack results

n	Embedded information (byte)	V_b (byte)	V_m (byte)
1	301	183.1	147.8
2	603	531.6	314.7
3	904	823.2	600.4
4	1206	1137.3	1084.3
5	1507	1470.9	1405.1

Table 2.

RS attack results

n	Embedded information (byte)	V_b (byte)	V_m (byte)
1	301	142.3	16.1
2	603	418.4	227.5
3	904	739.1	509.9
4	1206	1148.1	1006.7
5	1507	1494.5	1465.4

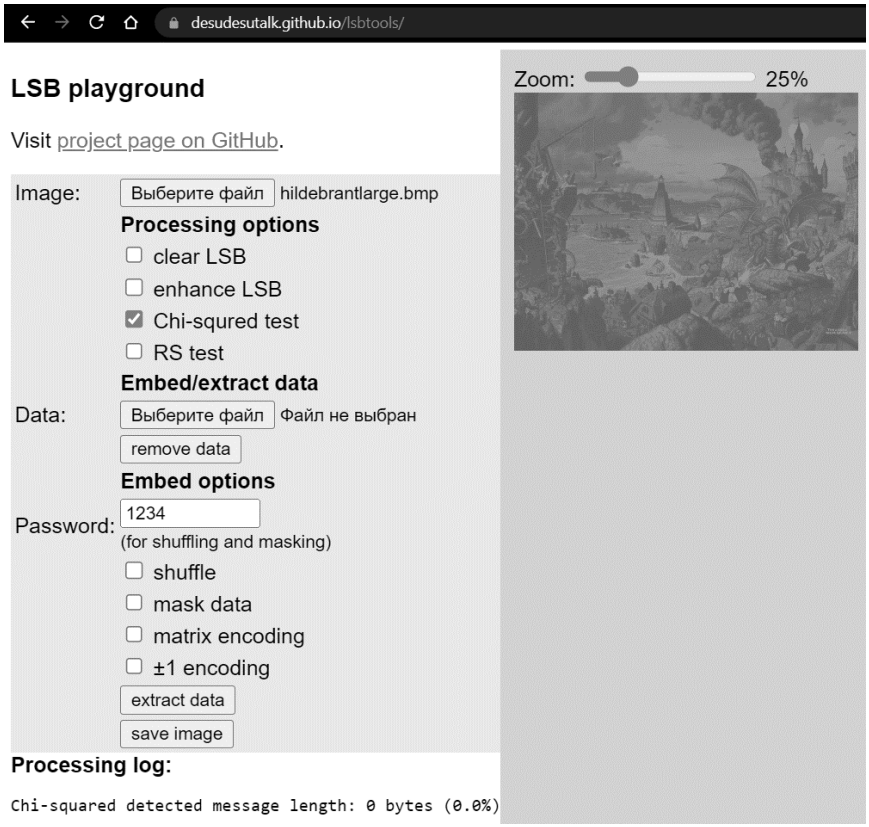

Fig.8. Chi-square attack (modified LSB).

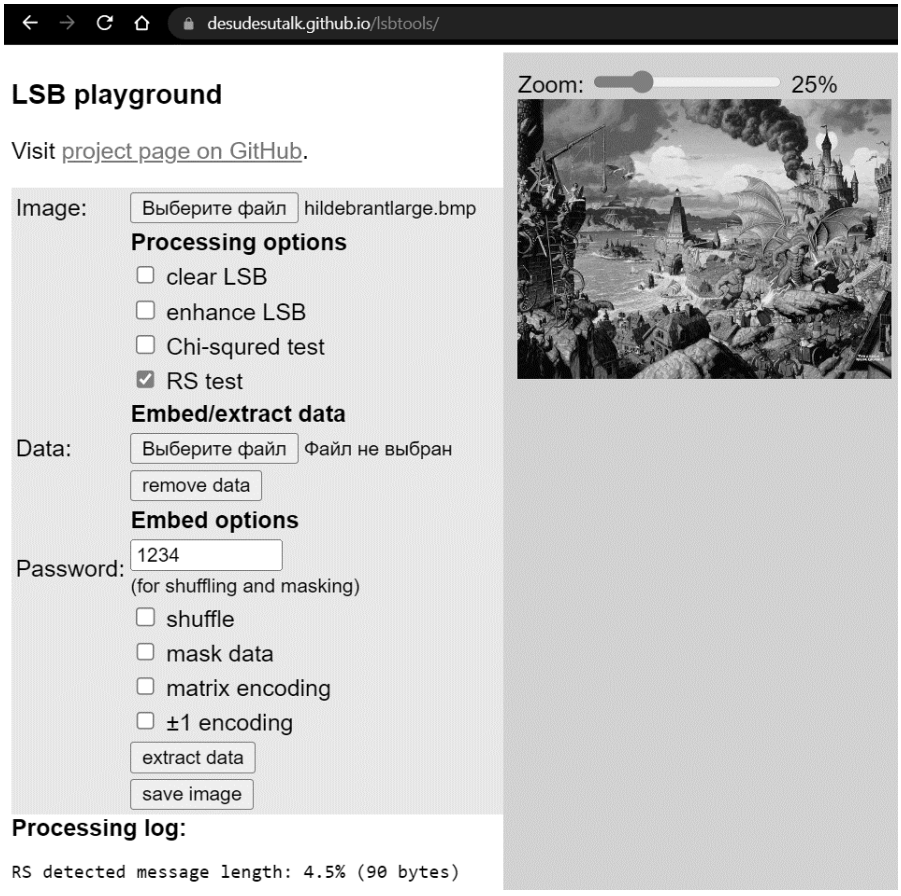

Fig.9. RS attack (modified LSB).

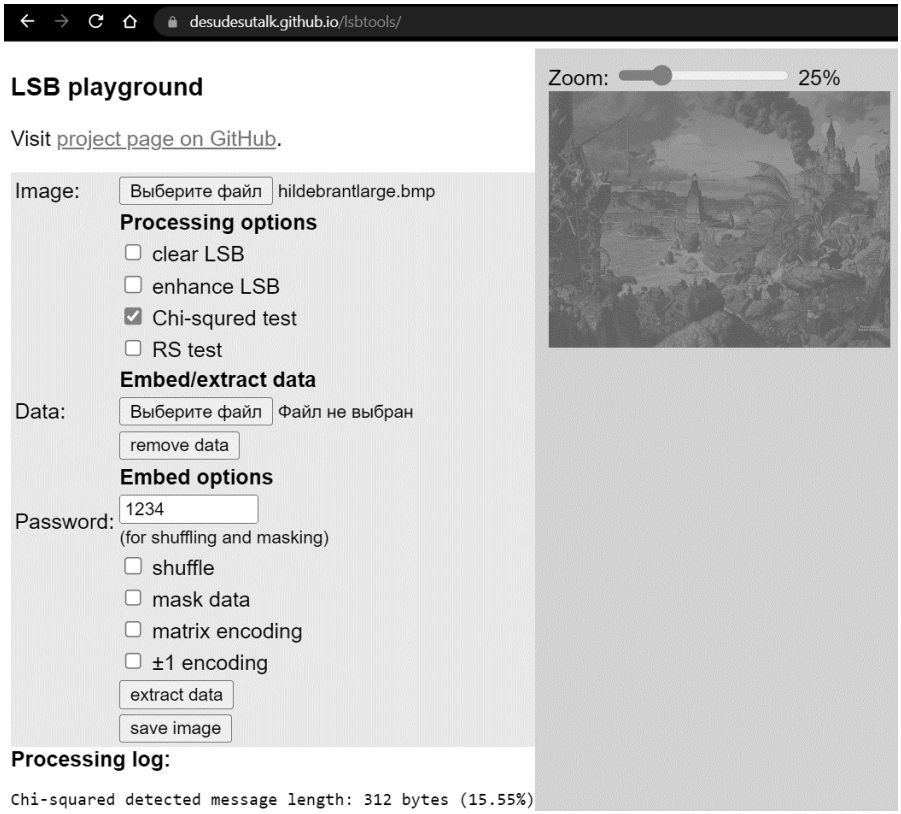

Fig.10. Chi-square attack (simple LSB).

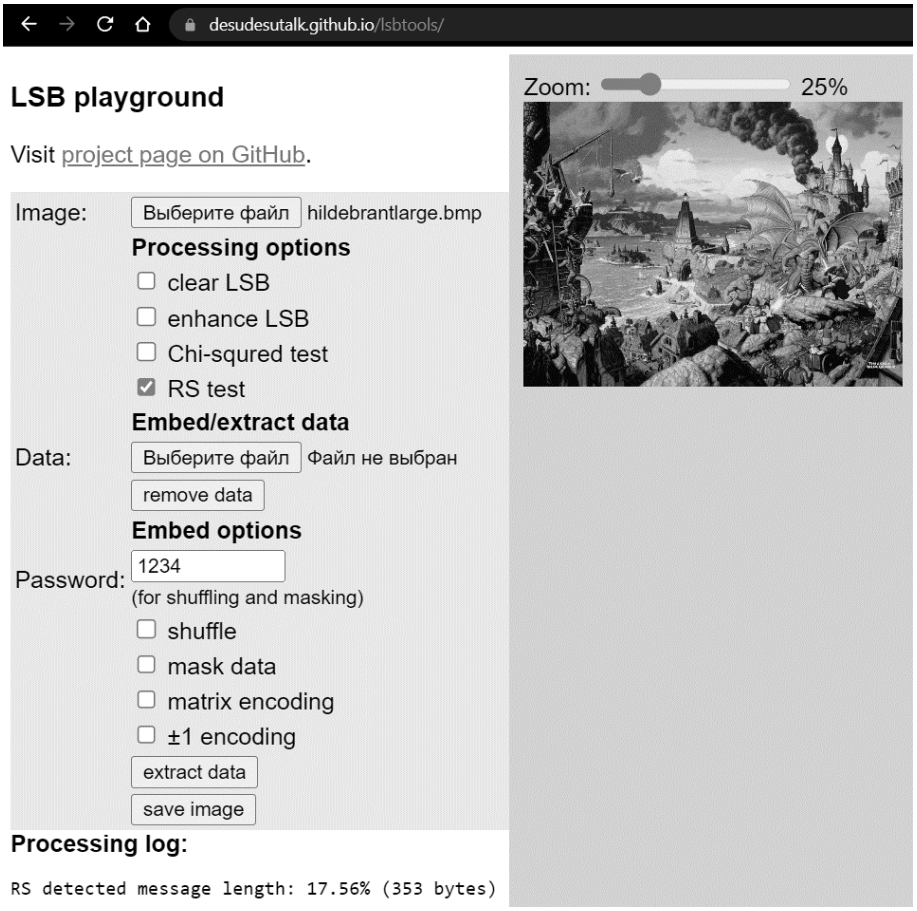

Fig.11. RS attack (simple LSB).

According to results shown in Table 1 and Table 2, the average percentage increase in resistance to attacks can be calculated using the following formula:

$$\frac{\sum_{k=1}^{n}\frac{100-V_{m,k}}{V_{b,k}}}{n},$$

where V_m is amount of detected bytes in a container filled using modified LSB method, its values are presented in Table 1 and Table 2 in fourth columns; V_b is the amount of detected bytes in a container filled using base LSB method, its values are presented in Table 1 and Table 2 in third columns; n is the number of different degrees of filling the container

Based on the formula above and the results of Table 1 and Table 2, the average increase in resistance to chi-square attacks is 19.27% and 35.92% for RS attacks, respectively.

IX. CONCLUSION

As a result of this work, a modification of the LSB algorithm for embedding information into the container was developed.

Since the interpolation in random order by key was not conducted sequentially with a certain step, where a regularity can be traced, but in an unpredictable order, it complicated the task of analysts to identify hidden information in the container and its correct extraction by attackers.

Based on the results obtained after testing, the modified LSB method can resist chi-square attacks by an average of 19.27 % better, and RS attacks by an average of 35.92 % better. At the same time, the difference in comparison with the unmodified algorithm almost disappeared with an increase in the stego container used percentage.

References

[1] Sajid M. M. and Tanvir M. P. et al. (2019). “A Comparative Study of Recent Steganography Techniques for Multiple Image Formats”, Modern Education and Computer Science (MECS), pp. 1–20. DOI: 10.5815/ijcnis.2019.01.02.

[2] Mahdi M., Shafry M. M., Abass F. J., Sabah M. T. et al. (2018). “Performance evaluation measurement of image steganography techniques with analysis of LSB based on variation image formats”, International Journal of Engineering & Technology, pp. 3505-3514. DOI: 10.14419/ijet.v7i4.17294.

[3] Subramanian N., Elharrouss O., Al-Maadeed S. and Bouridane A. et al. (2021). “Image Steganography: A Review of the Recent Advances”. IEEE Explore. pp. 23409-23423. DOI: 10.1109/ACCESS.2021.3053998.

[4] Polinovskyi V.V., Korolyov V.Y., Gerasimenko V.A. and Gorinshtein M.L. et al. (2011). “Information technology for research methods steganography and steganalysis”. Interuniversity collection "Computer-integrated technologies: education, science, production" Lutsk, 2011. Issue No. 5. Available at: https://icyb180.org.ua/wp-content/uploads/2012/04/informatsiyna-tehnologiya-dlya-doslidzhennya-metodiv-steganografiyi-i-stegoanalizu.pdf (Accessed: 28 September 2022).

[5] A. Suresh, K.L. Shunmuganathan, et al. (2012). “Image Texture Classification using Gray LevelCo-Occurrence Matrix Based Statistical Features”. European Journal of Scientific Research. pp. 591-597. Available at: https://www.academia.edu/36645235/Image_Texture_Classification_using_Gray_Level_Co_Occurrence_Matrix_Based_Statistical_Features (Accessed: 28 September 2022).

[6] T. Filler, J. Fridrich. et al. (2011). “Design of adaptive steganographic schemes for digital images”, in Electronic Imaging, Media Watermarking, Security, and Forensics: The International Society for Optical Engineering, San Francisco, CA, 2011, DOI: 10.1117/12.872192.

[7] C.-Y. Lin, S.-F. Chang et al. (1999). “Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process”.International Symposium on Multimedia Information Processing (ISMIP 99). Available at: https://www.ee.columbia.edu/ln/dvmm/publications/99/cylin-modelscan.pdf (Accessed: 28 September 2022).

[8] Vidhya R. and Brindha M. et al. (2018) “Polynomial Substitution based Image Encryption using Shamir Scheme”, Proceedings of International Conference on Computational Intelligence & IoT (ICCIIoT), pp. 311-314. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3355268 (Accessed: 28 September 2022).

[9] Hineman A. and Blaum M. et al. (2021). “An efficient implementation of the Shamir secret sharing scheme”, ArXiv, pp. 1-7. DOI: 10.48550/arXiv.2108.05982.

[10] D. Progonov. et al. (2021) “Detection Of Stego Images With Adaptively Embedded Data By Component Analysis Methods”, Advances in Cyber-Physical Systems, Vol. 6, Number 2, pp. 146-154, DOI: 10.23939/acps2021.02.146.

Maksym Pavlov was born in 1999 in Kirovograd, Ukraine. He received a Bachelor's degree in Computer Engineering at Lviv Polytechnic National University. He was involved in some technological startups. His interests are computer vision, image processing, and the integration of artificial intelligence into various fields.

Irina Yurchak received her B.S. and M.S. degrees at Lviv Polytechnic Institute, Lviv, in 1987. Her research interests include work with Artificial intelligence systems, intelligent computing systems, neural networks, genetic algorithms, recognition systems, prediction problems, computer graphics, computer modeling and animation, and web design.