ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2023 Управление, вычислительная техника и информатика № 65

Tomsk State University Journal of Control and Computer Science

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

INFORMATICS AND PROGRAMMING

Original article
UDK 004.056.5
doi: 10.17223/19988605/65/10

Method of hidden transmission of information based on fractals and its software

Vagif A. Gasimov[1], Jabir I. Mammadov[2], Nargiz F. Mammadzade[3]

[1, 2, 3] Azerbaijan Technical University, Baku, Azerbaijan
[1] vaqif.qasimov@aztu.edu.az
[2] cabir.memmedov@aztu.edu.az
[3] mammadzada.nargizw@gmail.com

Abstract. In this article is considered hiding process of bits on least significant bits that make up the binary codes of confidential information with using fractal based stego-key. In order to increase resistance against stegoanalysis, it is proposed to use only a part of graphic images not all the pixels of the containers, but only a part determined by the appropriate rule. As a key-image file displaying a Mandelbrot and Julian fractal is used to determine positions of pixels for hiding secret information on least significant pixels of container. So after creation of fractal image as key file, border point of it is changed with according represented rule, then position of pixels is determined with obtained new positions to hiding information in least significant bits.

The effectiveness of proposed method is studied with visual and statistical analysis, is verified by implementing it in the C# environment on examples of multiple containers and data files to be hidden.

Keywords: hiding information; fractal; container; stego-key; stegoanalysis.

For citation: Gasimov, V.A., Mammadov, J.I., Mammadzade, N.F. (2023) Method of hidden transmission of information based on fractals and its software. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitelnaja tehnika i informatika – Tomsk State University Journal of Control and Computer Science*. 65. pp. 95–104. doi: 10.17223/19988605/65/10

Научная статья
doi: 10.17223/19988605/65/10

Способ скрытой передачи информации на основе фракталов и его программное обеспечение

Вагиф А. Касумов[1], Джабир И. Мамедов[2], Наргиз Ф. Мамедзаде[3]

[1, 2, 3] Азербайджанский технический университет, Баку, Азербайджан
[1] vaqif.qasimov@aztu.edu.az
[2] cabir.memmedov@aztu.edu.az
[3] mammadzada.nargizw@gmail.com

Аннотация. Рассматривается задача о сокрытии битов, составляющих двоичные коды конфиденциальной информации, в наименее значимых битах графических файлов (контейнеров) с помощью фрактального стегоключа. С целью повышения устойчивости к стегоанализу предлагается использовать не все пиксели графических изображений-контейнеров, а только часть, определяемую соответствующим правилом. Для определения положений пикселей, в которых будет скрыта информация в наименее значимых битах контейнера, используется дополнительный ключевой файл – файл изображения, в котором отображается фрактал Мандельброта или Юлиана. Таким образом, после построения фрактала, описанного в ключевом файле, расположение его граничных точек изменяется в соответствующем порядке, и на основе полученных новых позиций определяются позиции пикселей контейнера, в которых будет скрыта информация в наименее значимых битах. Эффективность предложенного метода исследуется визуальным и статистическим анализом. Проверка работоспособности и эффективности метода осуществляется путем реализации в среде C# на примерах нескольких контейнеров и скрываемых информационных файлов.

Ключевые слова: сокрытие информации; фрактал; контейнер; стегоключ; стегоанализ.

***Для цитирования*:** Касумов В.А., Мамедов Дж.И., Мамедзаде Н.Ф. Способ скрытой передачи информации на основе фракталов и его программное обеспечение // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2023. № 65. С. 95–104. doi: 10.17223/19988605/65/10

Introduction

Recently, to protect the information with standard cryptographic and along with steganographic algorithms, also many methods based on non-traditional and chaotic processes are used [1–3]. One of these methods is information protection based on applying fractal transformations, which at first glance create the impression of chaotic transformations. The analysis of the researches shows that the research works carried out in this sphere cover both the fields of information encryption and protection with steganographic hiding [4–9]. Using Mandelbrot and Julian fractals is more common in steganographic systems [10–16]. In one of the researches, where it is suggested to hide confidential information (image, text, sound, etc.) in Julian fractal, which is an algebraic fractal, the shape and beautiful colours of that fractal, and also its exact dependence on the initial data (this is the parameter C and it is impossible to repeat the fractal without knowing its exact values) are used [13]. In [14], it is suggested to use fractal images, which are easily generated as containers and whose parameters can be easily changed. Here, while building the Julian fractal for the hiding process, it is intended to determine the values of its pixels depending on the content of the hidden information bits. In order to increase the confidentiality level, the information required to be hidden in that research work is encrypted beforehand based on the RSA algorithm.

In [15–16], any image in BMP format is used as a container, and another image - the Mandelbrot fractal - is used in order to determine the position of pixels to hide the information in this image. Here, with the help of the Mandelbrot fractal, the part defined in the container-image is divided into four parts. The bits of the binary codes of the information that is going to be hidden are placed in pairs, passing through each of the four parts in a certain turn in a counter clockwise direction. Writing successive bits of hidden information into separate parts of the container-image makes its detection difficult in the process of stegoanalysis.

One of the possible options to prevent the detection of the information hiding event is using not all the pixels of the container-image, but only a part of them through certain selection ways. During the placement in the BMP format images, the method of using the positions of the border points of the fractals that are described in the graphic files based according to the Mandelbrot and Julian fractals as keys to determine the positions of the pixels where the information is going to be hidden attracts attention in terms of resistance to stegoanalysis. So, the border of the fractal figure with the areas that are not included in this figure is built on the basis of a very complex mechanism, and a slight change in the starting conditions for building the fractal results in a significant change of the set that forms the border. This factor significantly makes the detection of hidden information by means of stegoanalysis complicated.

The purpose of proposed method are creation of new method for hiding data on graphic files and developing software of the method and as well as to study efficiency of the method. The novelty of the work is that the positions of the border points of the fractals, mixed with appropriate methods, described in the graphic files based on the Mandelbrot and Julian sets, were used as keys to determine the positions of the pixels where the information will be hidden.

1. Formulation of the problem

One of the possible options to prevent the detection of information hiding in BMP format files by the LSB method is to use not all the pixels of the container-image, but only a part of them by selecting them in a certain way. By using different methods and algorithms, the positions of the appropriate pixels can be defined, during the replacement of the least significant bits of the pixels of the image files with the bits of the binary code of the hidden information. One of these methods is to use another graphic image file as a key. The fact that the creation algorithms of Mandelbrot and Julian sets have a simple structure, their non-linearity, as well as the fact that fractal figures with complex borders can be easily generated, makes it possible to use these figures as keys. Here, to increase the resistance against stegoanalysis, not all pixels of container-images are used to hide information, but the points appropriate to the new positions received by changing the places of the border points of the mentioned fractals with a certain rule is intended.

2. Mandelbrot and Julian fractals

In 1979, B. Mandelbrot, who studied the sets discovered by the French mathematicians Gaston Julian and Pierre Fatou, discovered such a description on the complex plane, which later became the basis of a whole class of forms called Julian sets [17–18]. A Julian set $x_n \to x_{n-1}^2 + C$ is a set obtained by iterating a quadratic transform sequence. Here, each subsequent value of x is derived from its current value, and C is the control parameter. The values of the numbers that determine the sequence depend on the parameter C and the starting point x_0. In the plane of complex numbers, when we keep C fixed and change x_0, we get the Julian set, and when we change C at a fixed value of $x_0 = 0$, we get the Mandelbrot set (M). Each complex number C may or may not belong to the Mandelbrot set (it is shown in black in fig. 1, a). The point C belongs to M only when the result of the iteration starting with $x_0=0$ does not approach infinity. The set M consists of all C points that are associated with the related Julian set (fig. 1, b), and if the point C is not included in a set M, then the Julian set associated with it is not related (fig. 1, c). When the parameter C leaves M, the Julian set seems to explode and turn to dust. A strong quality change happening at the border of M also affects areas close to the border. The Mandelbrot set reflects in itself the process of transition from regularity to chaos. It is possible to increase the efficiency of the information bit hiding process, based on this chaos, only when determining the positions of pixels in container-images.

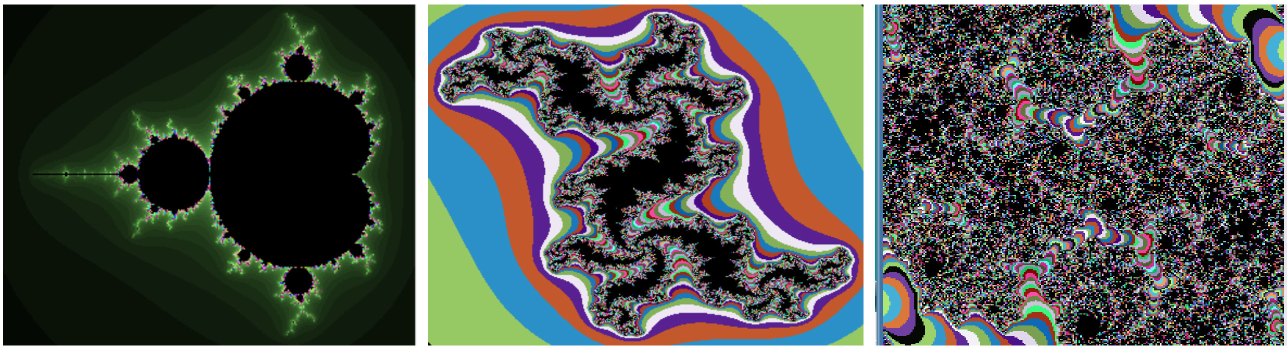


Fig. 1. Julian sets: *a*) Mandelbrot fractal, *b*) related Julian fractal, *c*) unrelated Julian fractal

3. A suggested steganographic hiding method

As mentioned above, the border points of fractal figures have a very complicated structure, and a slight change in the starting conditions causes the positions of the border points to change in a chaotic way. Here, when we say the border point of a fractal, such a point z_{ij} belonging to a fractal is intended that at least one of the adjacent points $z_{i-1,\,j-1}$, $z_{i-1,\,j}$, $z_{i-1,\,j+1}$, $z_{i,\,j-1}$, $z_{i,\,j+1}$, $z_{i+1,\,j-1}$, $z_{i+1,\,j}$, $z_{i+1,\,j+1}$ does not belong to that fractal.

Let's look at the construction of the steganographic method for hiding of information by fractal image obtained with the initial conditions as the initial key (K1) and using the border points of the generated fractals to determine the positions of the pixels to be hidden in another graphic file. It should be noted that Mandelbrot's and Julian's fractals are characterized by the fact that a slight change in the initial conditions causes the positions of the boundary points to change chaotically. Let's use both Mandelbrot and Julian fractals in the proposed method, considering that the construction sequences are also similar.

According to the proposed method, the hidden information is placed instead of the least significant bits of the specified pixels of the BMP format container.

Determining the positions of the pixels of the container where secret information will be placed is based on another image in BMP format - an image with a fractal figure. So, first the points forming the border of the fractal image are defined, and then the positions of these points are changed. According to the new positions of the border points of the fractal figure, the positions of the pixels of the container where the information will be hidden are determined.

Here, changing the places of border points is based on dividing the BMP format image containing the fractal figure into 4 equal rectangular parts (fig. 2) and placing these parts in different combinations. The number of possible combinations is calculated by the expression N=p! (p is the number of parts into which the image is divided). At p=4, the edge points of the fractal image can be located in N=4!=24 different variants.

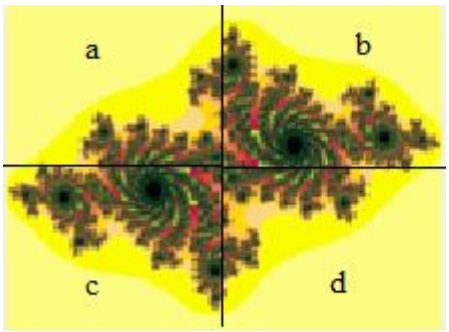


Fig. 2. An example for dividing a fractal image into 4 equal parts

A reflection of the parts of the fractal image by options is given in table. According to the table, in option 1 (abcd) it is intended that there is no displacement in the image parts. This means that the positions of the pixels where the information is hidden in the container overlap with the starting position of the fractal border points. In other options, 2 or more parts are replaced.

Table of displacements of border points in fractal images

Option №	A combination of image parts	Option №	A combination of image parts
1	abcd	13	cbad
2	abdc	14	cbda
3	acbd	15	cabd
4	acdb	16	cadb
5	adbc	17	cdba
6	adcb	18	cdab
7	bacd	19	dbca
8	badc	20	dbac
9	bcad	21	dcba
10	bcda	22	dcab
11	bdac	23	dabc
12	bdca	24	dacb

When the places of the parts are changed from the original version in the image with the size *mxn* (m is the number of rows and n is the number of columns), the new positions of the border points for all variants are calculated as follows:

– in displacement to the left: $i^{'} = i - \frac{n}{2}$; $j' = j$;

– in displacement to the right: $i^{'} = i + \frac{n}{2}$; $j' = j$;

– in downward displacement: $i^{'} = i$; $j' = j + \frac{m}{2}$;

– in upward displacement: $i^{'} = i$; $j' = j - \frac{m}{2}$;

– in displacement to the left and down: $i^{'} = i - \frac{n}{2}$; $j' = j + \frac{m}{2}$;

– in displacement to the right and down: $i^{'} = i + \frac{n}{2}$; $j' = j + \frac{m}{2}$;

– in displacement to the left and upward: $i^{'} = i - \frac{n}{2}$; $j' = j - \frac{m}{2}$;

– in displacement to the right and upward: $i^{'} = i + \frac{n}{2}$; $j' = j - \frac{m}{2}$.

It should be noted that table is kept by the parts that transmit and receive confidential information, and the choice of which option from this table is determined by the secret key (K_y) agreed between the parts beforehands. As one of the possible options, it is suggested to calculate the following expression according to the date on which the information exchange session of that key was implemented:

$$K_y = [\text{date}] \bmod 24.$$

Here, an eight-digit number derived from the concatenation of numbers representing the day, month and year is intended as a numerical expression of the date. For example, the displacement key for May 28, 2023 would be:

$$K_y = 28052023 \bmod 24 = 7.$$

This means that option 7 in table is going to be used to displace the parts of the image that represents the fractal figure, as well as the border points of the fractal together with them.

So, using the covert key K1, agreed between the parts beforehand, the algorithm for steganographic hiding of information in the suggested method will be as follows:

1) *m*×*n* sized BMP format container is selected*;*

2) based on the K1 key, a *m*×*n* sized BMP format fractal image is created;

3) the hidden information is converted into a binary format and its size is determined by bits (the number of bits is denoted by the sign Q);

4) *i*=1, *j*=1, *q*=1 are accepted;

5) it is checked that if the positional point (*i*,*j*) of the fractal image belongs to the fractal: for this it is checked that if the (*i*-1, *j*-1); (*i*-1), *j*; (*i*-1, *j*+1); (*i*, *j*-1); (*i*, *j*+1); (*i*+1, *j*); (*i*+1, *j*-1); (*i*+1, *j*+1) position points adjacent to it in the horizontal, vertical and diagonal directions belong to the fractal.

6) if at least one of these adjacent points does not belong to the fractal, then the point at position (*i*, *j*) is considered a border point, the new position of the border point ($i^{'}$,j') is calculated according to the option selected from table, otherwise, it will go to the 9th paragraph;

7) the least significant bit of the pixel in the position (*i*,*j*) of the container (corresponding to the position of the border element of the fractal image) is replaced by the *q*-th bit of the hidden information;

8) if *q*<*Q*, then *q*=*q*+1, otherwise go to paragraph 11 (the information was completely hidden in the container and the process ended);

9) if *q*<*m*, then *j*=*j*+1 is accepted and it goes to paragraph 5;

10) if *i*<*n*, then *i*=*i*+1, *j*=1 is accepted and goes to paragraph 5, otherwise go to paragraph 11 (the information was not placed in the container and the process was interrupted);

11) THE END.

The process of extracting hidden information at the receiving side is performed in a similar way, and during this process, the quantity K1 is also used as an entry parameter.

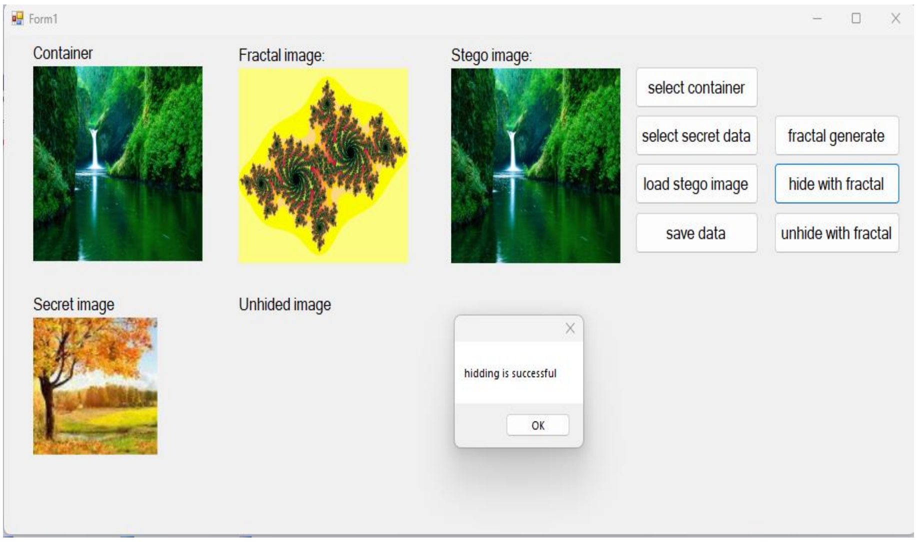
Of course, using only as many container pixels as the number of border points of the fractal figure restricts the amount of hidden information. For example, if we use all the pixels of a 24-bit container in BMP format of size 256 x 256, then theoretically, it is possible to place ≈24.5kB of information there. During the research, through the appropriate program a Mandelbrot fractal sized 256 x 256 was established, and it was defined that the practical number of border points there is up to 5-6 thousand, which means that there is an opportunity of hiding information in the amount of 1.9-2.25 kB. In this sense, practical calculations show that the Julian fractal is superior. Thus, the Julian fractal of that size allows hiding information of 6-8 kB. Of course, the amount of hidden information can be increased by increasing the size of the images. For example, it is possible to define the addresses of pixels for placing information in the amount of 18-20 kB, with a Julian fractal sized 512 x 512.

With the suggested method, the disadvantage of hiding a relatively small amount of information is compensated by a considerable increase in the privacy level of the hidden information.

4. Software realization of the method

To realize the suggested method, the software has been realized using the "Windows Form Application" module intended for designing the C# programming language in the Visual Studio 2019 environment. In the program, according to a certain sequence, a fractal image is first generated according to the starting C_x, C_y values (for this purpose, the "fractal generator" button is provided in the program) and a container is selected (fig. 3), and then the bits of the binary code of the hidden information are consistently hidden in the pixels of the container whose places are defined according to the fractal image.

It should be noted that in the considered example, the image file is taken as the hidden information. In the program window, the container and the image files that are going to be hidden are selected respectively, with the help of the "select container" and "select secret" buttons. Hiding Information in the container is performed by "hide with fractal "button, and saving the result (stegocontainer) is performed by "save data" buttons. When the hiding process is successfully completed, the user is given a "hiding is successful" message.



hidding is successful

Fig. 3. Fractal-based hiding process software window

The process of extracting confidential information from the stegocontainer on the receiving side is carried out in a similar way. For this purpose, the K1 key (C_x and C_y parameters) is included and the "load stego image", "fractal generator", "unhide with fractal" buttons are pressed successively (fig. 4). The appearance of the image file in the "Unhidden image" area informs that the process of extracting information from the stegocontainer is fulfilled successfully.

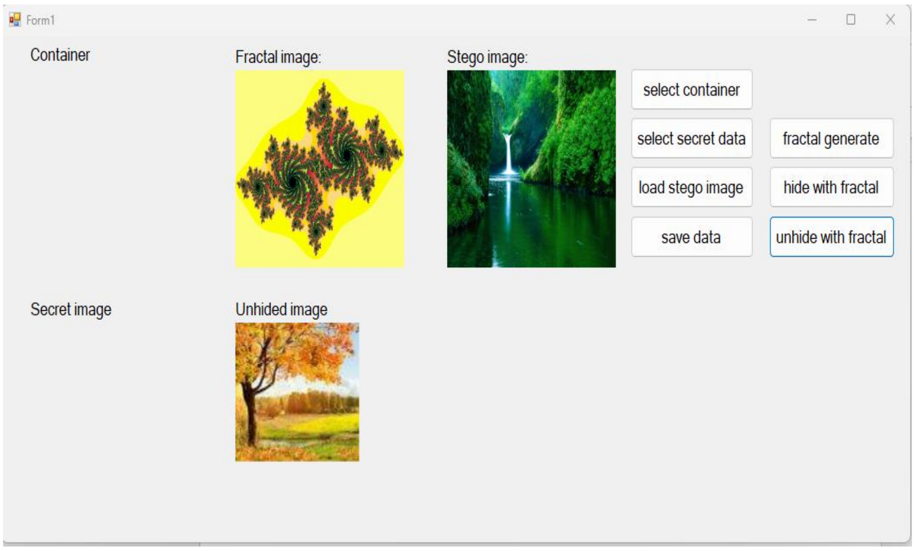


Fig. 4. Extraction of confidential information out of the stegocontainer

5. Efficiency analysis

Visual and statistical methods were used to analyze the effectiveness of the suggested method. Here, it is assumed that the knowledge about the algorithm, key, container, information volume is not known beforehand, it means that, analysis methods useful for any steganographic algorithms have been applied.

5.1. Realizing analysis by visual method

Visual methods, considered the easiest way to analyze graphic files, are based on the ability of the human visual system to detect differences between comparable images. The visual analysis method is realized by simply visual reviewing the captured image.

During the study, more than 30 different stegoimages realized by replacing the least significant bit for each colour category in pixels were visually analyzed. Regardless of the amount of included information, no visual difference was detected between the container and the stegoimages.

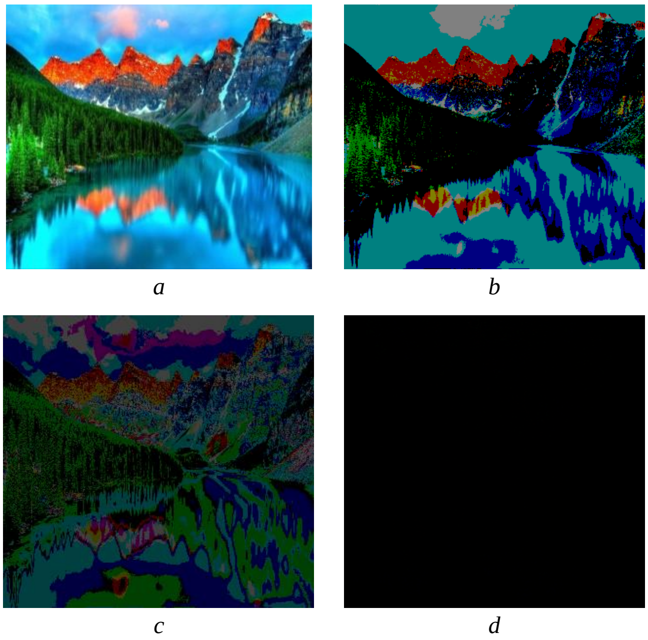


Fig. 5. Stegoimage and its bit cuts: *a* is stegoimage, *b*, *c*, *d* is respectively cuts on bits 7, 6 and 1

It should be noted that, for the analysis of stegoimages, the method of visual analysis of bit cuts is also widely used in practice [19–20]. The essence of this method is that the images are compared with the images got from its bit cuts. Here, the image is considered by means of the program layer by layer, that is, by bits cuts. Since each colour intensity is defined by one byte, a total of 8 cuts must be checked. An image consisting of the least significant layers of all colours gives the first cut, an image consisting of 2 layers gives the second cut, and so on. Images gained with bit cuts are visually compared with the whole image itself.

In the study performed by the considered method, a 256 x 256 sized BMP format stegoimage with 3.6 kB of information hidden inside by replacing the least significant bit was studied by the method of visual analysis of bit cuts (fig. 5). By software realization, cuts were obtained on all bits, including the 1st bit from the right (fig. 5, d), where the information is hidden. In figure 5 the stegoimage and the images formed by cuts consisting of its bits 7, 6 and 1 are given. Images consisting of received cuts and their options gained with different illuminations and screen enlargements were compared with the starting stegoimage, but it was not possible to find out the fact of hiding information there.

5.2. Realization of analysis statistically

Statistical methods are based on checking the "naturalness" of the studied image. Thus, the probability of hiding information during the realization of statistical methods is determined by evaluating the difference of the studied image from that of the "natural" image.

In this research considered, as a statistical method, the method of evaluating transitions between small significant bits in adjacent elements of the image was used. This method is based on the information that there is a correlation contact between the small bits of adjacent elements. As the elements of the studied sequence, the least significant bits of the colour components of the adjacent pixels of the stegoimage are used.

It is known that the dependence between the bits in the appropriate layers of the container elements has a Markov peculiarity [21]. Here, the dependency parameter is defined by layer number. Since the elements of the researched sequence consist of symbols of the binary number system, a histogram is established by analyzing transitions in 4 options (0→0, 0→1, 1→0 and 1→1). For analysis, getting different results in the histograms constructed for the spare container and the stegocontainer is taken as the main issue. Thus, the distribution of the least significant bits in the stegocontainer, as a rule, has the feature of randomness, and therefore the number of transitions is approximately the same, while in the spare container, it differs considerably from each other.

In this research work, the information hiding process is carried out on numerous image samples with the algorithm suggested in Section 3. All received stegoimages were statistically analyzed using the "method of evaluation of transitions between less significant bits in adjacent elements of the image". By no means, the equal distribution even approximately of transitions between the smallest bits in the stegoimages has been observed (fig. 6). In the images, column 1 reflects – 0→0; column 2 – 0→1; column 3 – 1→0; column 4 – reflects – 1→1 transitions.

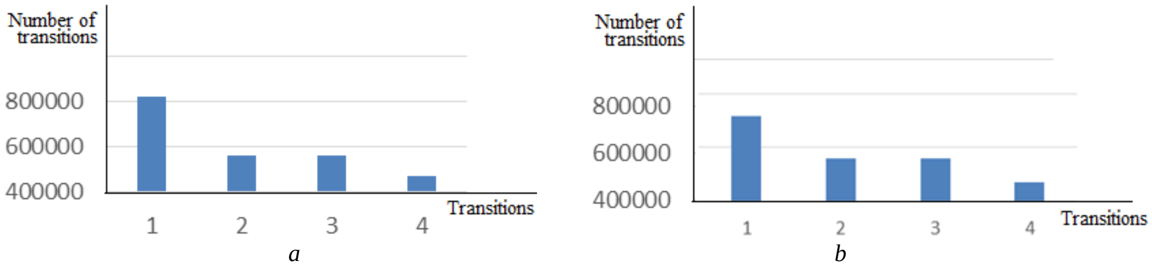


Fig. 6. Histograms of the number of transitions in the succession of least significant bits in the container (*a*) and in the stegocontainer (*b*) in the sample of information hiding according to the suggested algorithm

Conclusion

It has been developed a new fractal-based algorithm for hiding confidential information in the image files. For information hiding, it was suggested to use not all of pixels of container-images, but only pixels selected by a certain rule. Thus, for selecting the pixels in the container, the determination of their positions is based on changing the parts of the fractal figures in the graphic images, which are used as keys, and together with them the places of the border points in a certain manner.

In order to increase the resistance against stegoanalysis, an image file containing a Mandelbrot or Julian fractal was used to determine the positions of the pixels of the containers where the information would be hidden.

It was determined that using the Julian fractal as a key in order to determine the positions of pixels in containers is more appropriate. It has been practically approved that the Julian fractal allows hiding 4-5 times more information than the Mandelbrot fractal.

The suggested algorithm is realized in the C# programming environment. According to the image samples selected to be used as containers, the algorithm was approved to be functional. The effectiveness of the algorithm was tested on numerous samples of containers and information files that are going to be hidden by the methods of "visual analysis" and "evaluation of transitions between less significant bits in adjacent elements of the image" and positive results were gained in all cases.

References

1. Gasimov, V.A. & Mammadov, J.I. (2019) DNA-based image encryption algorithm. *IOP Conf. Series: Materials Science and Engineering.* Art. 734 012162. DOI: 10.1088/1757-899X/734/1/012162
2. Gasimov, V.A., Mammadov, J.I. & Mammadzada, N.F. (2022) Stream encryptıon method based on the chaotıc brownıan motıon model of molecules. *Procedia Computer Science*. 215. pp. 577–588.
3. Gribunin, V.G., Okov, I.N. & Turintsev I.V. (2002) *Tsifrovaya steganografiya* [Digital Steganography]. Moscow: SOLON-Press.
4. Kasapbashi, M.C. (2019) New Chaotic Image Steganography Technique Based on Huffman Compression. *IEEE Access*. 7. pp. 148495–148510. DOI: 10.1109/ACCESS.2019.2946807
5. Milani, M. & Ceyhan, S. (2022) An Efficient Method for Digital Image Encryption Based on Improved Chaotic Map. *Electronic Letters on Science and Engineering*. 18(2). pp. 87–96.
6. Bubere, A., & Ravindra, M. (2021) Steganography Based on Fractal Set. *International Journal of Scientific & Engineering Research*. 12(3). pp. 243–246.
7. Tabbia, B. (n.d.) *Fractal Encryption Algorithm*. [Online] Available from: http://www.codeproject.com/Articles/406389/Fractal-encryption-algorithm
8. Shelukhin, O.I. & Kanaev, S.D. (2017) *Steganografiya. Algoritmy i programmnaya realizatsiya* [Steganography. Algorithms and software implementation]. Moscow: Hotline – Telecom.
9. Abazina, E.S. & Erunov, A.A. (2016) Digital steganography: current state and prospects. *Sistemy upravleniya, svyazi i bezopasnosti – Systems of Control, Communication and Security*. 2. pp. 182–201.
10. Gasimov, V.A., Mammadov, J.I. & Mustafayeva, E.A. (2022) *Steganography: Channels and Technologies of Hidden Transmission of Information*. BakuL Nauka. (In Azerbaijani).
11. Shaw, J., Saha, O. & Chaudhuri, A. (2012) An Approach for Secured Transmission of Data using Fractal based Chaos. *National Conference on Communication Technologies & Its Impact on Next Generation Computing (CTNGC)*. pp. 13–17.
12. Negi, D., Negi, A. & Agarwal, S. (2016) The Complex Key Cryptosystem. *International Journal of Applied Engineering Research*. 11(1). pp. 681–684.
13. Ahmad Sami Nori & Asmaa M. Al-Qassab. (2014) Steganographic technique using fractal image. *International Journal of Information Technology and Business Management*. 23(1). pp. 52–59.
14. Suryakala, E.G., Leelavathy, N. & Sandhya, R.U. (2014) Fractal Image Steganography Using Non Linear Model. *International Journal of Innovative Research in Computer and Communication Engineering*. 2(1). pp. 2644–2649.
15. Desai, H.V. & Desai, A.A. (2014) Image Steganography Using Mandelbrot Fractal. *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)*. 4(2). pp.71–80.
16. Desai, H.V. & Desai, A.A. (2016) Steganography of Messages using Mandelbrot Fractal. *VNSGU Journal of Science and Technology*. 5(1). pp.73–85.
17. Mandelbrot, B.B. (1977) *The Fractal Geometry of Nature*. New York: Freeman.
18. Chernova, E.V. (2015) Chaos and order: a fractal world. *Nature (Mathematics)*. 5. pp. 34–44.
19. Aliev, A.T. (2004) On the application of the LSB steganographic method to graphic files with large areas of monotone fill. *Bulletin of the DSTU*. 4(22). pp. 454–460.
20. Shvidchenko, I.V. (2010) Steganalysis methods for graphic files. *Artificial Intelligence*. 4. pp. 697–705.
21. Barsukov, V.S. & Romantsov, A.P. (2000) Evaluation of the level of secrecy of multimedia steganographic channels for storing and transmitting information. *Special Technique*. 1. pp. 52–59.

***Information about the authors*:**
Gasimov Vagif A. (Professor, Doctor of Technical Sciences, Head of Department of “Computer technologies” of Azerbaijan Technical University, Baku, Azerbaijan). E-mail: vaqif.qasimov@aztu.edu.az
Mammadov Jabir I. (Associate Professor, Candidate of Technical Sciences, Associate Professor of Department of “Computer technologies” of Azerbaijan Technical University, Baku, Azerbaijan). E-mail: cabir.memmedov@aztu.edu.az
Mammadzade Nargiz F. (Lecturer of Department of “Computer technologies” of Azerbaijan Technical University, Baku, Azerbaijan). E-mail: mammadzada.nargizw@gmail.com

Contribution of the authors: the authors contributed equally to this article. The authors declare no conflicts of interests.

***Информация об авторах*:**
Касумов Вагиф Алиджавад оглы – профессор, доктор технических наук, заведующий кафедрой «Компьютерные технологии» Азербайджанского технического университета (Баку, Азербайджан). E-mail: vaqif.qasimov@aztu.edu.az
Мамедов Джабир Исмаил оглы – доцент, кандидат технических наук, доцент кафедры «Компьютерные технологии» Азербайджанского технического университета (Баку, Азербайджан). E-mail: cabir.memmedov@aztu.edu.az
Мамедзаде Наргиз Фируз кызы – преподаватель кафедры «Компьютерные технологии» Азербайджанского технического университета (Баку, Азербайджан). E-mail: mammadzada.nargizw@gmail.com

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации. Авторы заявляют об отсутствии конфликта интересов.

Received 21.07.2023; accepted for publication 08.12.2023

Поступила в редакцию 21.07.2023; принята к публикации 08.12.2023