Steganography on Color Images Using Least Significant Bit (LSB) Method

Tutuk Indriyani[1(✉)], S. Nurmuslimah[1], Audita Taufiqurrahman[1], Rinci Kembang Hapsari[1], Citra Nurina Prabiantissa[1], and Aeri Rachmad[2]

[1] Department of Informatics, Faculty of Electrical and Information Technology, Institut Teknologi Adhi Tama Surabaya, Surabaya, Indonesia
{tutuk,nurmuslimah,rincikembang,citranurina}@itats.ac.id
[2] Department of Information Systems, Universitas Trunojoyo Madura, Madura, Indonesia

Abstract. In some fields, high data security is required for data transmission. This raises concerns about misuse of data to irresponsible parties. To protect it, efforts were made to hide factual information on top of other information, namely steganography using the Least Significant Bit (LSB) method. This method has the advantage of a well-compressed image that is difficult to detect with the naked eye and has a fast process. In this study, the LSB method has two processes, namely the endcoding and decoding processes. The proposed method is tested and then evaluated MSE and computation time. In this research, each test uses 20 text messages and 20 images. The Text messages that were tested for insertion in images consisted of short messages, medium messages, and long messages. The text message will be inserted into the image with a small and a large image. The test results for the category of short messages inserted into thumbnails produce an average MSE value of 0.28 db, an average encoding and decoding processing time of 14 ms. The medium message category embedded in large images produces an average MSE value of 0.029 db, the average encoding and decoding time is 54ms. The long message category inserted in large images produces an average MSE value of 0.11 db, the average encoding and decoding time 1700 ms. The results of the three tests were that all text messages were successfully inserted into the images.

Keywords: *S*teganography · Least Significant Bit (LSB) · Text Security

1 Introduction

As technology develops, the need for data is increasing. The data used by users varies, not only text data but also image data. In some fields, high data security is required for data transmission [1]. The fact that data in several fields or in several institutions is confidential. For example copyright data, data in the medical or military fields [2]. These fields require high security in data transmission. According to research respondents, most of the network incidents reported were related to the loss or leakage of business information such as internal records, customer information, employee and intellectual

I. H. Agustin (Ed.): ICONNSMAL 2022, AISR 177, pp. 39–48, 2023.
https://doi.org/10.2991/978-94-6463-174-6_5

data. This raises concerns about misuse of information by irresponsible parties [3]. These components, incorporating the applicable criteria that follow.

To protect information from being misused by unauthorized parties and then trying to hide factual information on top of other information is called steganography. Steganographic methods in previous studies include: the End Of File (EOF) method tests video files using 80% accuracy [4], the Least Significant Bit (LSB) method is applied to video steganography, this method is capable of storing text messages but the text size does not exceed the capacity of the video closing frame produces an MSE of 15.06 db [5]. The LSB method applied to steganography as a result of its application can work well as expected [6]and the LSB method also has the advantage that the compressed image is difficult to detect with the naked eye, so it does not raise public suspicion, has a fast encoding process [7, 8] and easy to implement. However, this watermarking application still needs to be developed to improve message security.

In this study the development of the LSB method was carried out which was applied to color images. In this study the LSB method has two processes, namely the process of Endcoding and Decoding. The improvement of the LSB method is in the first process, namely Endcoding. The improve process is to insert messages into the image media by modifying the bits of each image pixel. The bit modifying step includes the last bit of each color image pixel which is replaced with the hidden message bits. The second process converts the binary values in the image into decimal matrix form, rearranges the decimal values into characters and then extracts the text message from the stegano image. The process of extracting or retrieving messages from the host image is carried out by taking the pixel bits from the resulting image that are located in the last position and then converting them into characters.

The LSB development method in this study was tested and then evaluated with Mean Square Error (MSE) and time calculations. In this study, each test on the LSB development method uses 20 text messages and 20 images. Text messages tested to be embedded in images consist of short messages (maximum 40 bytes), medium messages (maximum 675 bytes) and long messages (maximum 1.75 KB). Text message data input will be inserted into an image with a small image size of 80 x 80 pixels and inserted into a large image measuring 800 x 800 pixels and then compared. Testing will be carried out crosswise between the image and the embedded text message. The process of further arrangement in this research is as follows: The second part presents related theories, the third part describes the steps of the method, the fourth part sets out the results of the method process and the fifth part concludes this paper.

2 Methodology

In this study, the message data input used 3 types of messages each message inserted at a different image size. The first is ten image data, each of which is color image data with a small image size of 80 x 80 pixels and a large image size of 800 x 800 pixels with a.bmp extension and has 3 components in one matrix or commonly called RGB. This image serves as a hiding place for text messages. The second data input is message or text data input of ten data each with the size of a short message (maximum 40 bytes), medium message (maximum 675 bytes) and long message (maximum 1.75 KB) which

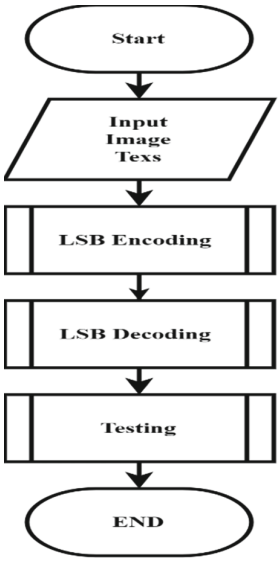

Fig. 1. System design

will be entered into the image data. System design in this study there are four processes. The system design in this study is shown in Fig. 1.

In the first process input image and text data, the second process encodes using the LSB method, the third process decodes using the LSB method, and the last process performs testing.

A. *LSB Encoding*

The encoding process is the process of inserting a text message into an image, where a secret message will be inserted into the image [9]. The stages of the Least Significant Bit (LSB) Encoding process can be shown in Fig. 2. Based on the system flowchart in Fig. 2, we can be explained as follows: Take the pixel value in the RGB image in decimal form and then enter a secret message into the image. The message to be inserted into the image consists of several characters, spaces are still counted because spaces have ASCII values [10]. The improvement in this study lies in the process of taking matrix values according to the length of the message in the form of the number of characters, for example the number of characters is 320 or 107 pixels for each RGB pixel. Converts a decimal image matrix to binary form. Converts text messages to decimal form using ASCII codes. Then from decimal form to binary form. Inserts a text message that has been in binary form into an image that has been converted to binary form as well. This insertion is done by changing the value in the last 1 bit of the LSB.

An illustrative example of bit changes in 10 pixel data after the insertion process is shown in Table 1.

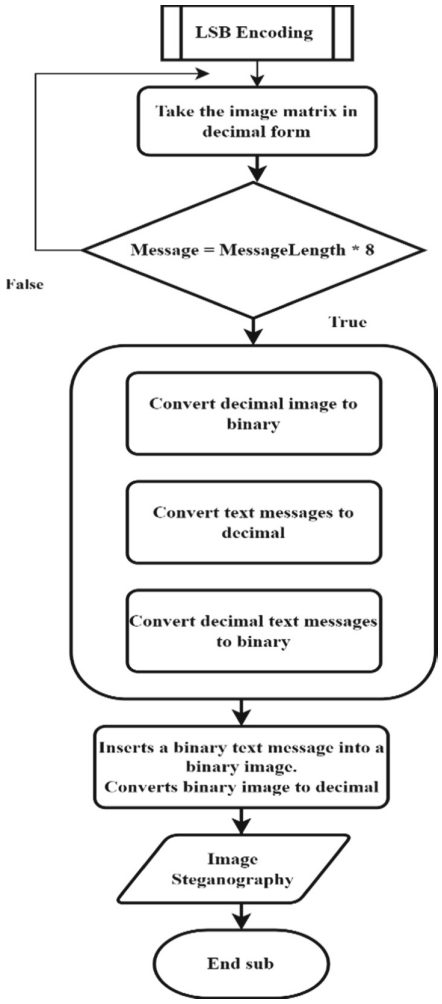

Fig. 2. LSB encoding stages.

Changes in bit values before and after inserting a secret message into the image are shown in bold and underlined for the last 1 bit. Change the image that has been inserted into a decimal which is the output of the stegano image.

B. *LSB Encoding*

Decoding process is a process for extracting hidden messages [11]. The flow of the extraction process on the stegano image by entering the same keywords at the time of the encoding process, so that hidden messages can be reappeared [12]. The stages of the Least Significant Bit (LSB) Decoding process can be shown in Fig. 3.

Table 1. Change of Bits before Insertion and After Insertion of Message

P	Bit value containing text			Bit value doesn't contain text		
	R	*G*	*B*	*R*	*G*	*B*
1	11110111	11001011	10101000	1111011**0**	1100101**1**	1010100**0**
2	11111000	11001100	10101001	1111100**1**	1100110**0**	1010100**0**
3	11111000	11001100	10101001	1111100**1**	1100110**1**	1010100**0**
4	11110111	11001011	10101000	1111011**1**	1100101**0**	1010100**0**
5	11110110	11001010	10100111	1111011**0**	1001001**0**	1010011**0**

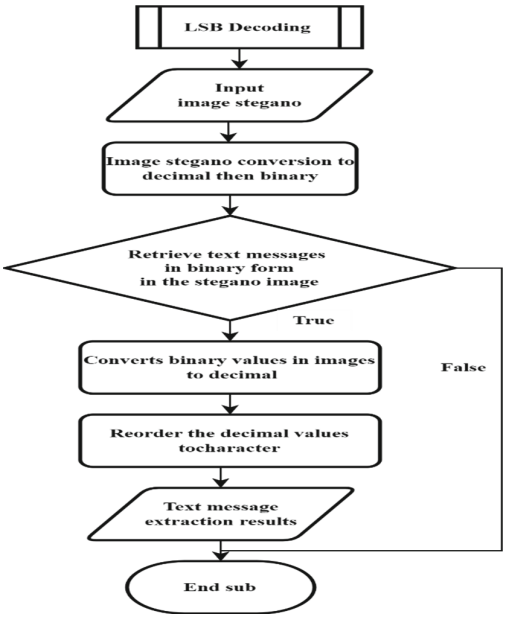

Fig. 3. LSB decoding stages.

Converts a picture that has been inserted from a text message into a decimal form. The first stage is to input the stegano image to extract the message. The second step is to convert the stegano image to a decimal matrix. The next step is to change the stegano image from decimal to binary. The last bit marked with an underscore is the bit that still has the text message in it. Retrieve text messages embedded in binary form on stegano images. An example of changing the bit value after retrieving a text message can be seen in Table 2.

The final stage is to convert the binary values in the image into a decimal matrix then rearrange the decimal values into characters and display the results of the extracted text messages.

Table 2. Change in Bit Value after a Text Message is retrieved

P	Bit value containing text			Bit value doesn't contain text		
	R	*G*	*B*	*R*	*G*	*B*
1	11110110	11001011	10101000	11110111	11001011	10101000
2	11111001	11001100	10101000	11111000	11001100	10101001
3	11111001	11001101	10101000	11111000	11001100	10101001
4	11110111	11001010	10101000	11110111	11001011	10101000
5	11110110	10010000	10100110	11110110	11001010	10100111

C. *Testing*

D. *Means Square Error (MSE): MSE in this study is used to measure the difference between two images, namely the input image and the image that has been inserted with text (steganography). The MSE formula can be shown in Eq. 1* [13, 14].

$$MSE = \frac{1}{M.N} \sum_{x=1}^{M} \sum_{y=1}^{N} \left(S_{xy} - C_{xy}\right)^2 \quad (1)$$

where:

M and N are the height and width values of the image, 1 is the range of dynamic pixel values or the maximum pixel value that can be retrieved, x and y are the coordinates of the points in the image, S is the image with message text, and C is the original image.

E. *Computation time testing*: *Computation time testing is very important to find out the average computation time required by the system in carrying out the process of inserting or extracting messages [15][16]. Computational time testing in this experiment was carried out by calculating how long it took the system to carry out the process of inserting and extracting messages [17][18]. Computation time testing will show that the number of characters in the file will affect the computation time. Another factor that can affect the required computation time is the level of compatibility between the file used and the message text that is inserted, and the number of messages that are successfully inserted* [19, 20].

3 Result and Discussion

The trials in this study included testing small size images inserted with short messages, testing small size images inserted with medium messages, testing small size images with long messages inserted and testing large images with short, medium and long messages inserted. Each of these tests uses ten data.

Table 3. Average Test Results for Message Types Embedded in Small Images

Message Type	MSE Average	Average Encoding and Decoding Time
Short message	0,28 db	14 ms
Medium message	4,67 db	55 ms
Long message	10,97 db	2064 ms

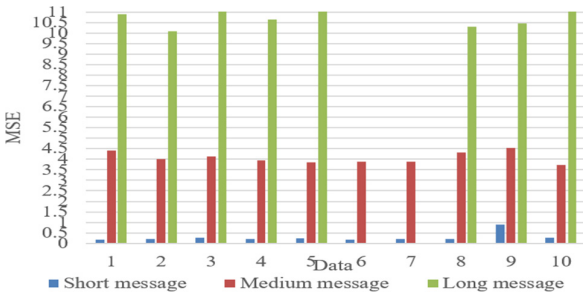

Fig. 4. The test results chart inserts the message type into the small image.

F. *Test Results for Inserted Message Types in Small Images.*

The test in this sub-chapter is a small image test inserted with short, medium, and long message types, each using ten image data and ten message data. The results of testing the MSE value and time can be seen in Table 3.

Based on Table 3, the average value of all these tests results in an MSE value of less than 11 and the testing time for short and medium messages is less than one minute, while long messages take longer because they are influenced by the small image capacity and the resources of the computer device. The three messages embedded in small size images that have an MSE value close to zero are short messages and the shortest average trial time is also the short ones inserted in small images. This means that the LSB method applied to steganography produces small error values and has a short time on small images inserted with short messages. All types of messages embedded in the small-sized image can be decoded (extracted) according to the original input message (all succeeded). A comparison diagram of the three types of messages embedded in an 80x80 image can be shown in Fig. 4.

Based on Fig. 4. The test results of the ten data on each type of message embedded in a small size image, the MSE value of less than one is a small message which is shown in the blue bar while the average MSE value is less than 11db is long messages are shown in the green color chart. This is because the number of pixels in the small size image accommodates all the long message characters, so when it is extracted it takes a

Table 4. Average Test Results for Message Types Embedded in Large Images

Message Type	MSE Average	Average Encoding and Decoding Time
Short message	0.002 db	13 ms
Medium message	0.029 db	54 ms
Long message	0.11 db	1700 ms

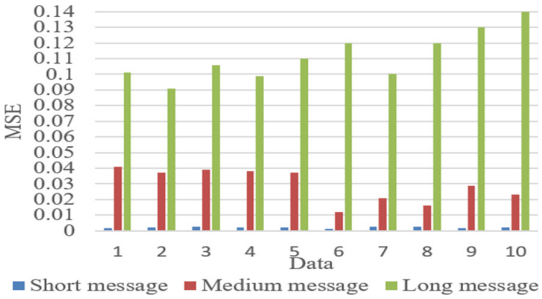

Fig. 5. The test results chart inserts the message type into the big image.

long time and the change in the pixel value of the output image is compared to the input image before text is inserted, the change is large.

G. *Test Results for Inserted Message Types in Big Images*

The test in this sub-chapter is a large image test inserted with short, medium, and long message types, each using ten image data and ten message data. The results of testing the MSE value and time can be seen in Table 4.

Based on Table 4. The average value of all these tests produces an MSE value of less than 0.5 and the test time for short and medium messages is less than one minute, while long messages are less than 30 min. This means that the LSB method applied to steganography produces a small error value if the type of message embedded in the image is large and has a short time. All types of messages embedded in large images can be decoded (extracted) according to the original input message (all success). A comparison diagram of the three types of messages embedded in an 80x80 image can be seen in Fig. 5.

Based on Fig. 5. The results of testing ten data on each type of message embedded in a large image, the average MSE value of less than 0.01db is a small message displayed on a blue bar while the average MSE value of less than 0.1db is message length shown in the green chart. In testing on large images the type of text inserted produces an MSE value of less than 0.15 db, this is because many pixels in the image still contain empty pixels without text characters, so the extraction time is fast and changes to the output image are compared to the input image. The changes are small.

4 Conclusion

Based on the test results in this study, we can conclude that can be concluded that the larger the image size and the smaller the text message inserted, the MSE value is close to zero and the encoding and decoding time is less than 15 ms. Conversely, the smaller the image size and the longer the text message is inserted, the MSE value is more than 10 db and the Encoding Decoding time is greater than 30 min. Steganography with the Least Significant Bit (LSB) method for image size and message size greatly affects the MSE value and Encoding Decoding time.

Acknowledgment. We want to thank *research group and ITATS* for supporting the international ICCGANT seminar in 2022 at Jember.

References

1. S. Rustad, R. Ignatius, M. Rosal, A.Syukur, "Inverted LSB image steganography using adaptive pattern to improve imperceptibility". Journal of King Saud University - Computer and Information Sciences 34(6), 3559-3568 (2022).
2. O. Elharrouss, N. Almaadeed, S.Al-Maadeed, "An image steganography approach based on k-least significant bits (k-LSB)" IEEE 97(3) 81-90 (2020).
3. P. Yani, R. Ignatius, M. Setiadi, "Simple and Secure Image Steganography using LSB and Triple XOR Operation on MSB". IEEE 18(5) 54-70 (2018).
4. B. Amine, A. El, Y. Taoui, " Reversible steganographic method based on interpolation by bivariate linear box-spline on the three directional mesh. Journal of King Saud University", Computer and Information Sciences 32 (7) 850-859, 2020.
5. A. Fauzi, R. Prahasiwi, R. Ignatius, "Image Steganography using Inverted LSB based on 2nd, 3rd and 4th LSB pattern", International Conference on Information and Communications Technology (ICOIACT) 4 (9)178-789, 2019.
6. K. Murat, Y.Yildiray, "Developing LSB Method Using Mask in Colored Images", IEEE 978 (18) 386-399, 2018.
7. H. Wien, M. Chen., T. Chen, "An efficient reversible image authentication method using improved PVO and LSB substitution techniques. Signal Processing", Image Communication 5965(17) 124–138, 2017.
8. K. Qazanfari, R. Safabakhsh, "A new steganography method which preserves histogram: Generalization of LSB++", Information Sciences. 277 (5) 90-101, 2014.
9. A. Todd, H. Ming, "Quantum steganography.3nd edn. Digital Media Steganography Elsevier", Los Angeles, CA, United State , 2020.
10. K.Aditya, S. Gandharba, "An improved method for high hiding capacity based on LSB and PVD", Digital Media Steganography Elsevier. 6nd edn. Department of Computer Science and Engineering, GMRIT, India 2020.
11. R. Isnanto, R. Septiana, A. Fashiha, "Robustness of Steganography Image Method Using Dynamic Management Position of Least Significant Bit (LSB)", International Seminar on Research of Rnformation technology and Intelligent System (ISRITI). 2(1) 55-68 2018
12. K. Rinki, P. Verma, R. Kumar, "A novel matrix multiplication based LSB substitution mechanism for data security and authentication" Journal of King Saud University - Computer and Information Sciences. 34 (8) 5510-5524, 2022.

13. T. Indriyani, M.I. Utoyo, R. Rulaningtyas, “Comparison of Image Edge Detection Methods on Potholes Road Images”,Journal of Physics: Conference Series, 1613(1), 2020.
14. T. Indriyani, M.I. Utoyo, R. Rulaningtyas, “Comparison of Image Smoothing Methods on Potholes Road Images”, Journal of Physics: Conference Series. ICComSET IOP Publishin. vol. 1477, 2019.
15. X. Chai, H. Wu, Z. Gan, Y. Zhang, Y. Chen, and K. Nixon, “An efficient visually meaningful image compression and encryption scheme based on compressive sensing and dynamic LSB embedding”, Nixon Optics and Lasers in Engineering.124(5) 105837, 2020.
16. T. Indriyani, M.I. Utoyo, R. Rulaningtyas, “A New watershed Algorithm for Pothole Image Segmentation”, Studies in Informatics and Control, 30(3) 131-139 , 2021.
17. R. Ignatius, “Improved payload capacity in LSB image steganography uses dilated hybrid edge detection”, Journal of King Saud University - Computer and Information Sciences 34 (2) 104-114, 2022.
18. A Hayder., F Najlae, “LSB based image steganography using McEliece cryptosystem. Journal of Physics: Conference Series, 23(5) 2021.
19. R. Hapsari, Miswanto, R. Rulaningtyas, H. Suprajitno, “Modified Gray-Level Haralick Tex ture Features for Early Detection of Diabetes Mellitus and High Cholesterol with Iris Image“, International Journal of Biomedical Imaging: Hindawi , 2022.
20. R. Hapsari, Miswanto, R. Rulaningtyas, H. Suprajitno, “Comparison of Histogram Based Image Enhancement Methods on Iris Images“, Journal of Physics: Conference Series , 2020.