Flipping the Message Bits to Increase Imperceptibility in the Least Significant Bit Image Steganography

Erna Zuni Astuti1, De Rosal Ignatius Moses Setiadi1, Eko Hari Rachmawanto1, Christy Atika Sari1 and Md Kamruzzaman Sarker2

1Department of Informatics Engineering, Dian Nuswantoro University, 207 Imam Bonjol Street Semarang 50131, Indonesia.
2Department of Computer Science, Wright State University, 3640 Colonel Glenn Hwy, Dayton, Ohio 45435 USA.

Abstract. Imperceptibility is one of the issues that continue to be studied in the science of steganography in images. Imperceptibility means that the message embedded in the cover image cannot be perceived by the human senses, especially the sense of sight. Flipping the message bits is the method proposed in this study, which is inspired by the inverted LSB method, but this method is simpler because it only flips the message bit when the change in the cover image pixel exceeds 50%. Although the proposed method is simpler than the inverted LSB method, based on the test it is found that the imperceptibility quality can increase significantly when compared to the traditional LSB method. This is evidenced by the PSNR value increasing around 9dB, from 51dB to 60dB and the MSE value dropping from around 0.44 to 0.05.

1. Introduction
Steganography is one of the data hiding techniques. In the science of steganography there are several issues that are still being researched and developed, including imperceptibility, payload capacity, and security [1] [2]. Imperceptibility mean messages hidden in the cover cannot be perceived by the human visual perception, the image steganography imperceptibility sometimes called invisibility [3] [4]. Payload capacity is related to the ability of the cover to accommodate the amount of data [5]. The least significant bit (LSB) method is often used to overcome these two issues. LSB is a method included in the spatial domain [6] [7] [8] [9] [10]. This method directly changes the smallest bit of the cover with the message. This method is very simple, by changing the smallest bit by default it has relatively good imperceptibility with a maximum payload of one-eighth the size of the cover [5].

Very good Imperceptibility is produced by LSB because changes in pixel values are only one value difference. But in terms of security LSB is very weak, this is because the technique is very simple. Many studies combine cryptographic techniques to solve the issue of message security issues [11] [12] [13]. Even though LSB has the advantage of solving payload capacity and imperceptibility problems, many studies continue to try to contribute to optimizing the issue of imperceptibility or payload capacity or both. As in the study [2], [3], [14] and [15] using the inverted LSB technique, research [4] and [12] used insertion techniques at the edge area, all of these studies contributed to improving the quality of the stego image imperceptibility. But each method has weaknesses and strengths. The inverted bits method can increase imperceptibility and possibly without reducing payload capacity, but this method requires a calculation table before inverted bits. Some or all table information must also be stored for the extraction process, storage of these tables can be done and sent separately or pinned to a specific pixel area on the cover image. Separate storage and senders require additional bandwidth, while embedding
in certain areas of the cover image can reduce payload and imperceptibility capacity, so this creates new problems. In other studies such as [5] and [8] contribute to increasing payload capacity. The inverted LSB method has been widely used to improve the quality of imperceptibility as evidenced by the increased PSNR value. But in research [2] and [14] the increase in PSNR was not more than 1dB, this indicates that the effect was not significant enough. In the study [15] obtained a higher increase in PSNR, which is around 0.4dB-8.3dB. But the inverted LSB technique is a fairly complicated technique compared to traditional LSB. In the inverted LSB technique a table is needed to calculate the total profit if inverted bits are carried out on the four kinds of combinations of the 3rd and 4th bits, namely 00, 01, 10, and 11. The advantage is the difference in the number of bits compared to the LSB method. Inspired by this technique, this study uses the flipping bit technique in the message with a simpler algorithm that contributes to the improvement of imperceptibility quality.

2. Related Research

In research related to steganography in the image there are three things that continue to be examined to be optimized, namely imperceptibility, payload capacity and security. Many methods and algorithms are proposed to improve one or more of these three things. Here are some studies that contribute to increasing imperceptibility. The Study [14] proposed a message bit complement technique and then inserted the message bit complement with the inverted LSB technique. In the study, experiments were carried out on three types of cover images, namely peppers, lena and baboon with sizes of 512 * 512 pixels, equivalent to 2097152 bits, while messages that were inserted contained three types of sizes, namely 4225 bits, 16384 bits, and 24964 bits. So if calculated the number of message bits pinned at each pixel is 0.016 bits, 0.063 bits and 0.095 bits. The average PSNR results produced in this study are 59.3151dB, 53.4431dB and 51.6067dB and when compared to traditional LSB techniques the maximum PSNR value increase is only around 1dB.

Another study conducted by Akhtar et al. [15], proposed the improvement of an inverted LSB technique. By default, the inverted LSB technique will reverse the message based on four types of patterns on the 6th and 7th bits, namely 00, 01, 10 and 11. The reversal of the message is based on the benefits obtained on each pattern. If there is no gain, the message bit is not reversed. This study adds variations in bit patterns to 6 and 7 with changes in the 0 and 1 bits on the LSB. This means that this technique is increasingly complicated because it requires a more complex variation of calculations. In this study, the trial was carried out on a cover image with a size of 1024 * 1024 pixels with a message size of 256 * 256 pixels or the equivalent of 524288 bits. If calculated for the size of the message bit embedded in each pixel the cover is 0.5 bits. The results of PSNR measurements produced in this study experienced an increase of about 0.4dB to 8.3dB when compared to traditional LSB techniques (without inversion). Other studies use other techniques to increase imperceptibility, namely by pinning messages to the edges of the image. Embedding in the edge area is believed to give fewer visual changes to the image, because the image edge area is the most insensitive part of the change. In the research conducted by Irawan et al. [4], the cover image was carried out by edge detection with the Canny algorithm. The results of edge detection are used as a key to extracting messages. In the testing use a text message with the largest message size is 8192 bits to cover grayscale image is 512 * 512 pixels. The PSNR generated is 69.11dB. Research carried out by Kusuma et al. [12] also applied almost the same technique, namely pinning on the edge area of the image coupled with DES encryption in the message. The size of the RGB cover image with a size of 1024 * 1024 pixels and the message image in the form of an RGB image with a size of 32 * 32 pixels, equivalent to 24576 bits obtained PSNR ranges from 72dB.

3. Flipping the Message Bits

Flipping the message bits method has similarities with the inverted LSB method, because this method is inspired by the method. The similarity is to complement the message that has been converted in binary form. This method is carried out after checking the bits of the cover image, if the bits of the cover image change more than fifty percent then this technique is carried out on the message bits. Below are the steps for the proposed flipping bits method:

1. Read the message that you want to flip bits.
2. If the message is an image of length m and width n, then the matrix after reshaping becomes a matrix with a length of m*n and width 1, then convert the reshaped matrix into a binary form as in Figure 1. If the message is text do transpose operation on the message, change text messages to ASCII values, and then change to the binary form, as in Figure 2.

```
    u   d   i   n   u   s
     117 100 105 110 117 115

    ASCII
```

**Figure 1.** The process to convert image message into binary form.

**Figure 2.** The process to convert the text message to binary form.

3. Reshaped messages become an array of length 1 and width m*n*8 for image messages and for text messages, the length of the text message is multiplied by 8. Based on the example in Figure 1 it produces the binary array as follows: 01101001 00011110 10011100 00110010 11001000 01001101 01011111 01011000, whereas for the example in Figure 2 it produces the binary array as follows: 1110101 1100100 1101001 1101110 1110101 1110011.

4. From the two sample bits above do flip bits in binary messages (BM) with the modulus 2 operation using formula (1).

\[
BM_z = \text{mod}((BM_z + 1), 2)
\]  

(1)

Where z is the binary message index.

5. Next, an array containing the complement value of the message will be obtained. After the complement value of the message is successfully obtained, the message can be inserted in the cover image with the LSB method, or can use other appropriate steganography methods. After being inserted and sent to the receiving side, of course the message will be extracted and converted to its original form. This is called the deflipping process. The proposed deflipping process is as follows:

1. Read binary messages as a result of message extraction.
2. Perform the flip bit process in the message using formula (1), as a note of the process of flipping and deflipping using the same formula.
3. Use the reshape function, to convert the message array into a matrix with a length of m*n and width 8 specifically for image messages, whereas for text messages use the message length size.
4. Convert binary values to decimals, so the matrix width becomes 1.
5. For image messages use the reshape function to change according to the original size, while for text messages change the ASCII value to char and then transpose the message.

4. Proposed Embedding and Extracting Method

4.1. Proposed Embedding Method

The following are the stages of the proposed message embedding process:
1. Read the cover image and message image, make sure the message image size can be pinned entirely to the cover image, if the message image size is too large the process cannot continue.
2. Change the message image into a binary form, then reshape the matrix message to form an array.
3. Make a counter variable to calculate the changes in the pixel bit of the cover image that changes if all message bits are pinned. Where the process of calculating pixel bit changes in a cover image can use double looping.
4. If the change in pixel bit of the cover image exceeds 50%, do flipping the message bit, with formula (1).
5. Embed the flipping bits of the message, on the smallest bits in each cover image sequentially.
6. Get a stego image.

4.2. Proposed Extracting Method
The following are the stages of the proposed message extraction process:
1. Read the stego image.
2. Perform message extraction by reading all the smallest bits in each pixel of the stego image, then save it in an extraction array.
3. Flipping bits in the extraction array using formula (1), so that the original message is obtained in binary form.
4. Do the reshape process to get a matrix with width 8, this is done because later every eight bits will be converted into decimal numbers.
5. After getting a matrix containing decimal numbers, do the reshape process according to the size of the message image.
6. Get the message image.

5. Results
In this study a standard image is used as a cover image that can be downloaded from [16] [17] [18] and for message image grayscale images are used. The software used for testing in this study is Matlab. The color cover image is converted into a grayscale image with the rgb2gray function. All cover images have a size of 512 * 512, while message images are 128 * 256. The message size is selected because all the pixels of the cover image will be inserted with 1 message bit. Where the message capacity used is far greater than the research conducted [2], [14] and [15]. Images shown in Figure 3 are used after being converted into a grayscale image.

![Figure 3. Original cover images used in the experiment](image_url)
Image presented in Figure 4, is used as message image, the left side is the original message image, while the right side is the message image after flipping bits. The result of the flipping bit process in the message image looks like a negative image. It needs to be emphasized that the process of flipping bits is only done if the bit changes in the entire cover image are more than 50%. This means that flipping bits are not necessarily done on all cover images. Then the message embedding process is carried out using the proposed method so that the stego image is obtained as shown in Figure 5.

The results of the stego image shown in Figure 5 look like the original image and there is no difference at all, if seen in plain view. This is indeed an advantage of the LSB technique that excels at imperceptibility and large embedding capacity. This is also coupled with the method of flipping message bits which further enhances the imperceptibility aspect. Imperceptibility quality is measured by PSNR and MSE [19] [20] [21]. The MSE formula is shown in formula (2), while the PSNR formula is shown in formula (3). Table 1 shows the comparison of the PSNR and MSE values generated by the proposed method and the LSB method without flipping the message bits.

$$\text{MSE} = \sum_{m=0}^{255} \sum_{n=0}^{255} \| o(m,n) - s(m,n) \|^2$$  \hspace{1cm} (2)\nonumber$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\sqrt{\text{MSE}}} \right)$$ \hspace{1cm} (3)\nonumber

Where $m, n$ is the sizes of images, $o$ is the original cover image, $s$ is the stego image.
Table 1. Imperceptibility Measurement of Stego Image Results.

| Image Name   | Pixel Counter | Flip? | Flip (proposed) | Non Flip (traditional) |
|--------------|---------------|-------|-----------------|------------------------|
|              |               |       | PSNR (dB)       | MSE                    |
| Aerial       | 130852 (49.92%) | No    | 51.6334         | 0.4464                 |
| Barbara      | 131476 (50.15%) | Yes   | 60.7253         | 0.0550                 |
| Bird         | 122456 (46.71%) | No    | 51.2837         | 0.4839                 |
| Cameraman    | 131152(50.03%)  | Yes   | 60.6968         | 0.0554                 |
| Waggon       | 140175(53.47%)  | Yes   | 60.9760         | 0.0519                 |
| Livingroom   | 133795(51.04%)  | Yes   | 60.8249         | 0.0538                 |
| Lena         | 131321(50.09%)  | Yes   | 60.7410         | 0.0548                 |
| Lichtenstein | 130282(49.76%)  | No    | 51.6220         | 0.4476                 |
| Average      |               |       | 57.3129         | 0.2061                 |

From the results shown in table 1, it appears that out of the eight cover images there are three cover images whose flip process is not done. This is because the change in the cover image bit does not exceed 50% so the process of flipping the message bit is undone. This method depends on the cover image used, however the method proposed in this study can significantly increase the value of MSE and PSNR. The average MSE value of the proposed method 0.2061, this is twice better than the MSE value without the flip method. While the PSNR value can reach 60 dB and an average increase of around 9dB, compared to the traditional LSB method (without the flipping bit technique). It can be concluded that the method of flipping message bits has a very good impact on imperceptibility quality.

Even when compared with the inverted LSB method used in research [2], [14], and [15] the proposed method produces a better improvement. In the three studies the proposed method was also compared with traditional LSB techniques, but the increase in PSNR value was not significant as described in the related research section. Although it does not use the same message so it cannot be compared with previous research. But the message size used in the study is far greater than in previous studies. Where the larger the message size will be equivalent with the greater capacity, and with the greater message capacity embedded will greatly affect the quality of imperceptibility. Algorithm for flipping message bits is also much simpler than inverted LSB, so it's easy to implement.

In the extraction process this method is the same as the traditional LSB method, only if the extracted message cannot be read, the message bits can be flipped. In the trials carried out in this study, all messages can be extracted perfectly, proved PSNR with infinity values in the message image extracted. Figure 6 shows the message image resulting from the extraction process.

6. Conclusions

This study proposes a flipping the message bit method that has a simple algorithm but results in a significant increase in imperceptibility quality. This is evidenced based on PSNR measurements which experienced an average increase of 9dB. But this method is very dependent on the cover image, because flipping bits are done if the change in bits per pexel is more than 50%, so it needs to be selective in choosing the cover image. Because the algorithm of this method is simple and that adds to the flipping step, the message security level is only one step ahead of the traditional LSB method. For the next research, the quality of security can be further developed, perhaps by a combination of cryptographic techniques or other techniques.
7. References

[1] Ahani, S. & Ghaemmaghami, S. 2015. Colour image steganography method based on sparse representation, *IET Image Processing*, vol. 9, no. 6, pp. 496 - 505.

[2] Sharma, V. & Bhardwaj, R. 2013. A lossless Data Hiding method based on inverted LSB technique, in International Conference on Image Information Processing, Wagnhat.

[3] Kusuma, E. J., Sari, C. A., Rachmawanto E. H. & Setiadi, D. R. I. M. 2018. A Combination of Inverted LSB, RSA, and Arnold Transformation to get Secure and Imperceptible Image Steganography, *Journal of ICT Research and Applications*, vol. 12, no. 2, pp. 103-122.

[4] Irawan, C., Setiadi, D. R. I. M., Sari, C. A. & Rachmawanto, E. H. 2017, *Hiding and securing message on edge areas of image using LSB steganography and OTP encryption*, in International Conference on Informatics and Computational Sciences, Semarang.

[5] Setiadi, D. R. I. M., Santoso, H.A. Rachmawanto, E. H. & Sari, C. A.,2018, *An improved message capacity and security using divide and modulus function in spatial domain steganography*, in International Conference on Information and Communications Technology, Yogyakarta.

[6] Jiang, W., Guo, Z., Wang, K. & Huang, Y. 2016. *A Self-contained Steganography Combining LSB Substitution with MSB Matching*, in International Conference on Computer Science and Network Technology, Chanchun.

[7] Shehzad D. & Dag, T. 2017. *A Novel Image Steganography Technique based on Similarity of Bits Pairs*, in Control and System Graduate Research Colloquium, Shah Alam.

[8] Liu, Y., Chang, C.-C. & Chien, T.-Y. 2016. *A Revisit to LSB Substitution Based Data Hiding for Embedding More Information*, in International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung.

[9] Rodrigues A. & Bhise, A. 2017. *Reversible Image Steganography Using Cyclic Codes and Dynamic Cover Pixel Selection*, in International Conference on Wireless Communications, Signal Processing and Networking, Chennai.

[10] Subong, R. A., Fajardo, A. C. & Kim, Y. J. 2018. *LSB Rotation and Inversion Scoring Approach to Image Steganography*, in International Joint Conference on Computer Science and Software Engineering, Nakhonpathom.

[11] Ardiansyah, G., Sari, C. A., Setiadi, D. R. I. M. & Rachmawanto, E. H. 2017, *Hybrid Method using 3-DES, DWT and LSB for Secure Image Steganography Algorithm*, in International conferences on Information Technology, Information Systems and Electrical Engineering, Yogyakarta.

[12] Kusuma, E. J., Indriani, O. R., Sari, C. A., Rachmawanto, E. H & Setiadi, D. R. I. M. 2017, *An Imperceptible LSB image Hiding on Edge Region using DES Encryption*, in International Conference on Innovative and Creative Information Technology, Salatiga.

[13] Setyono, A., Setiadi, D. R. I. M. & Muljono. 2017, *StegoCrypt method using wavelet transform and one-time pad for secret image delivery*, in International Conference on Information Technology, Computer, and Electrical Engineering, Semarang.

[14] Bhardwaj R. & Sharma, V. 2016. *Image Steganography Based on Complemented Message and Inverted Bit LSB Substitution*, in International Conference on Advances in Computing and Communications, Kochi.

[15] Akhtar, N. S. Khan & Johri, P. 2014. *An improved inverted LSB image steganography*, in International Conference on Issues and Challenges in Intelligent Computing Techniques, Ghaziabad.

[16] Wikipedia, "File: Lichtenstein img processing test.png," 17 April 2017. [Online]. Available: https://commons.wikimedia.org/wiki/File:Lichtenstein_img_processing_test.png. [Accessed 2018].

[17] Ming Hsieh Department of Electrical Engineering. "SIPI Image Database," [Online]. Available: http://sipi.usc.edu/database/. [Accessed 2018].

[18] ImageProcessingPlace.com, "Image Databses," [Online]. Available: http://www.imageprocessingplace.com/root_files_V3/image_databases.htm. [Accessed 2018].
[19] Ardy, R. D., Indriani, O. R., Sari, C. A., Setiadi, D. R. I. M. & Rachmawanto, E. H. 2017, *Digital image signature using triple protection cryptosystem (RSA, Vigenere, and MD5)*, in International Conference on Smart Cities, Automation & Intelligent Computing Systems, Yogyakarta.

[20] Winarno, A., Setiadi, D. R. I. M., Arrasyid, A. A., Sari, C. A. & Rachmawanto, E. H. 2017. *Image Watermarking using Low Wavelet subband based on 8×8 sub-block DCT*, in International Seminar on Application for Technology of Information and Communication, Semarang.

[21] Setiadi, D. R. I. M., Sutojo, T., Rachmawanto, E. H. & Sari, C. 2017. *Fast and efficient image watermarking algorithm using discrete tchebichef transform*, in International Conference on Cyber and IT Service Management, Denpasar.