Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 1, January 2024
DOI : https://doi.org/10.33395/sinkron.v9i1.13185
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Improving Data Embedding Capacity in LSB Steganography Utilizing LSB2 and Zlib Compression

Joshua Calvin Kurniawan[1)*],Adhitya Nugraha[2)],Ariel Immanuel Prayogo [3)], The, Fandy Novanto[4)]
[1)2)3)4)]Universitas Dian Nuswantoro, Semarang, Jawa Tengah, Indonesia
[1)] kurniawanjoshua05@gmail.com, [2)] adhitya@dsn.dinus.ac.id, [3)] arielemmanuel4@gmail.com, [4)]fandy2832@gmail.com

Submitted : Nov 21, 2023 | **Accepted** : Nov 24, 2023 | **Published** : Jan 1, 2024

Abstract: In an increasingly advanced era, the exchange of information through digital tools has become a common practice. With easy access and advancing facilities, securely and covertly exchanging data has become a challenging task. Therefore, the technique of steganography can be used as a solution for data hiding and protection, enabling safer data exchanges. Steganography is a method to conceal data within a transmission object, which can be an image, video, audio, and more. In this research, steganography will be performed using images as the transmission object. This study is done to offer a modification of the Least Significant Bit (LSB) steganography technique by utilizing the LSB-2 method, along with the utilization of the Zlib compression algorithm. The modification and use of the Zlib compression algorithm aim to increase the message capacity that can be embedded in the transmission object while preserving the image quality. The results of the experiments will be presented in tabular form by comparing the original image with the steganography-processed image using metrics such as Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) as measures of image quality. The experiments conducted results in an increase of capacity of approximately 36.54%, an increase in PSNR value of approximately 4.72%, accompanied by a decrease in MSE value in average of 49.19%, and SSIM values constantly at 0,99999 thus proving the proposed method successfully increased the embedded massage capacity while preserving even enhance the quality of the stego image produced by the embedding process.

Keywords: Compression; Least Significant Bit; LSB-2; Steganography; Zlib

INTRODUCTION

The swift advancement in modern communication technology has streamlined and accelerated data transmission making it convenience for everyone. However, this convenience has also increased the vulnerability of transmitted data to unauthorized copying, modification, or destruction by illegal or unauthorized third party. Consequently, safeguarding data secrecy, whether at rest or during transmission, becomes a crucial concern. Hence, data hiding is essential to network security as it play a pivotal roles in preserving the confidentiality of the data.

Steganography is a technique that focuses on concealing information within seemingly common medium to protect it from unauthorized access or detection. One commonly employed method in steganography is the Least Significant Bit (LSB) technique. LSB involves substituting the least significant bit of pixel values in an image or other digital media with hidden information. This subtle

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

alteration remains imperceptible to the human eye, allowing for the transmission of data using the cover medium.

The conventional LSB method, when employed on 512x512 RGB images, frequently encounters capacity limitations, maxing out at 12.28 kilobytes of data. Due to the restricted space available in the least significant bits of each pixel, the conventional LSB technique struggles to accommodate a significant volume of hidden information without visibly compromising the quality of the final stego image. Consequently, attempting to embed hidden data into more than the last bit can result in a degradation of visual imperceptibility, potentially arousing suspicion from those attempting to steal or manipulate the data. However, this issue can be mitigated by incorporating a compression algorithm into the LSB method as it enables the transformation or reduction of the data into a smaller and more compact form, addressing concerns related to visual imperceptibility while maintaining the security of the concealed information.

The combination of the Least Significant Bit (LSB) method with compression algorithms has become a prevalent practice on image steganography(AbdelWahab et al., 2019a). This widely adopted strategy involves embedding information in the least significant bits of data while applying compression techniques beforehand to optimize storage space. For example, previous study used Discrete Cosine Transform (DCT) compression algorithm to shrink and hide secret messages successfully(AbdelWahab et al., 2019b; Jayapandiyan et al., 2020a). The smaller the data, the better the quality of the stego image will be, as there will always be somekind of trade off between the embedding capacity and the visual quality of the stego image. This kind of approach to LSB had been proven multiple time to improve the quality of the stegoimage ,thus it had a wide application starting from data concealment to digital forensics.

Various compression algorithms, including Huffman coding, Lempel-Ziv, Run-Length Encoding (RLE), Discrete Cosine Transform (DCT), and zlib, are frequently evaluated for their efficiency in reducing file sizes(Jayapandiyan et al., 2020b). Huffman coding excels in achieving high compression ratios through variable-length codes, while Lempel-Ziv focuses on dictionary-based compression, offering excellent performance in certain scenarios. RLE simplifies compression by encoding consecutive identical elements efficiently. DCT, commonly used in image compression, transforms data into frequency components. However, when it comes to overall size reduction and versatility, zlib, Combines with its Deflate algorithm, often outshines others, making it a compelling choice across diverse data types.

This paper introduces an innovative approach to steganography by employing a 2-bit Least Significant Bit (LSB) method for data embedding. This technique aims to enhance the information-carrying capacity of the stego image while mitigating potential degradation in image quality. To counteract any impairing effects on visual fidelity, the paper proposes the use of the zlib compression algorithm. By integrating zlib with the 2-bit LSB method, the paper seeks to strike a balance between increased data concealment and the preservation of the overall quality of the stego image.

LITERATURE REVIEW

Various compression method can be implemented to the data (Aldabagh, 2019), the use of compression for data hiding in image steganography is evident. Additionally, other research has also utilized compression techniques in multilevel image steganography. Others also use the Discrete Cosine Transform (DCT) compression algorithm to compress the image to hide and extract the secret messages(AbdelWahab et al., 2019a; Jadav, 2013). They both manage to produce a high image quality and a higher level of security compared to using LSB alone.

Deflate itself is a data compression algorithm used in the ZIP archive format, while Zlib is a software library that implements the Deflate algorithm for compressing and decompressing data in various contexts. The performance of this algorithm had been proven on a few research papers (Jayapandiyan et al., 2020b; Tayyeh & Al-Jumaili, 2022). The method demonstrated using Deflate manage to produce a higher performance of steganography over the baseline study, indicating the efficacy of using data compression. Zlib allow this proposed method to utilized on many form of data in the future, even if the experiment would be conducted using text hidden data.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 1, January 2024
DOI : https://doi.org/10.33395/sinkron.v9i1.13185
e-ISSN : 2541-2019
p-ISSN : 2541-044X

To further highlight the potential of the cover image, the LSB can be modified to utilize more than just the last bit. As demonstrated in the experiment conducted by Serdar Solak and Umut(Solak et al., 2018), where they discussed the use of the least significant two-bit substitution algorithm for image steganography. This study revealed that this method provides a balanced trade-off between capacity and quality, but it is still recommended to be combined with other types of algorithms. Hence, the proposed method combines a 2-bit LSB with the Zlib compression algorithm.

METHOD

The objective of this paper is to offer a new modification to the commonly known LSB method by conducting experiments. The experiments will be conducted by using the LSB-2 method combined with the utilization of Zlib compression algorithm. The experiments are differentiated into two stages, first is the embedding process and the second is the extracting process. The flow of each stage is as following:

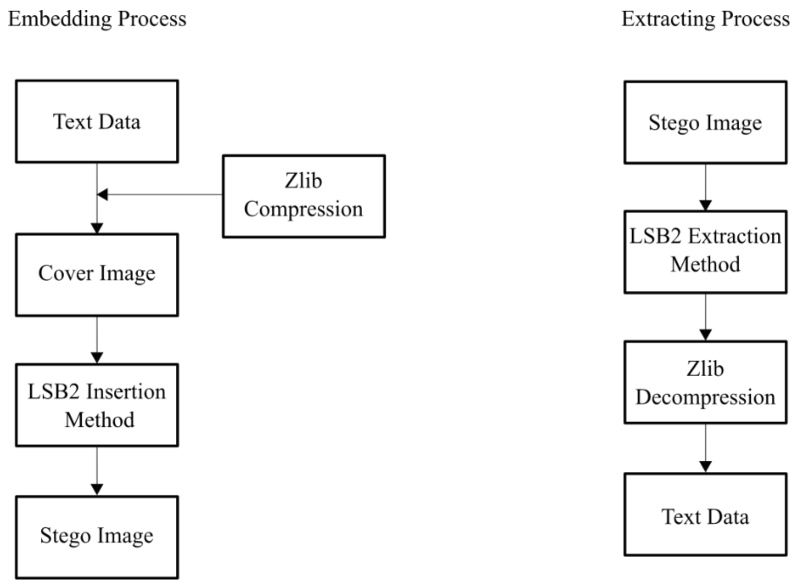

Figure 1. Embedding and extracting process flow

Datasets

The data used to be embedded into the cover image is in form of text data which will be randomly generated with varying data size in bytes, starting from 1000 bytes to 5000 bytes. The text data will be compressed with the Zlib algorithm to decrease the size thus decreasing the amount of pixel used in the

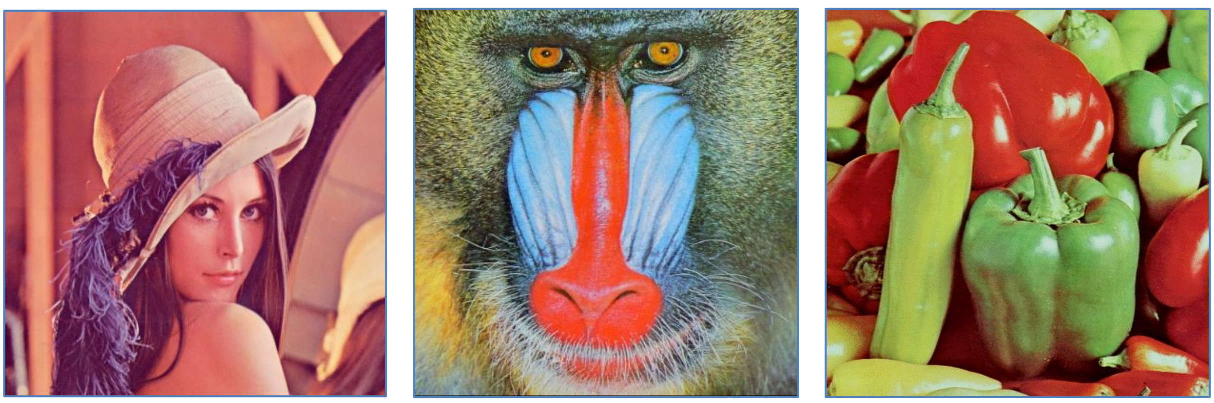

Figure 2. “Lena”, “Baboon”, “Pepper” PNG with 512 x 512

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

cover image in the embedding process. 3 images will be used as the cover image in the experiment, each with the size of 512 x 512, namely "Lena", "Baboon" and "Pepper"

Zlib Compression

The text data to be embedded into the cover image will firstly compressed using the Zlib algorithm. The text data will be processed by the Zlib algorithm, the result of the compression will be encoded using the base64 language to turn the unreadable compression result into readable hexadecimal form, before converting the hexadecimal form into binary form so it can be embedded into the cover image. The compression flow is as follow:

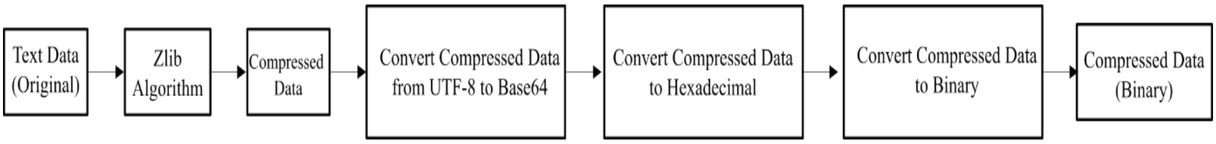

Figure 3. Zlib compression algorithm flow

The Zlib compression is used to compressed the text data to be embedded into the cover image, as mentioned before the data used have varying size in bytes starting from 1000 bytes to 5000 bytes.

Embedding Process

The embedding process will be using the LSB-2 method, unlike the commonly used LSB which uses the last binary value in each RGB of each pixel in the cover image, LSB-2 uses the two last binary value in each RGB. The text data compressed by the Zlib algorithm in binary form will be one by one embedded into the cover image using the LSB-2 method. For example, an image's pixel have an RGB binary bits as the following : R = **10011001**, G = **11001001**, B = **10110011**, and the binary of the text data to be embedded is (**101000**), after the embedding process, the RGB binary value will change into R = **1001110**, G = **11001010**, B = **10110000**.

Extracting Process

To obtain the original text data, firstly extract the two last binary value of the RGB in each pixel in the cover image. The extraction will results in a binary string form, the binary string extracted need to be converted into hexadecimal form then decode the hexadecimal form using the utf64 language to change it into the compressed result of the Zlib algorithm before converting it into the original text data embedded into the cover image.

Stego-image

The embedding process will result with a stego-image which contains the embedded text data. The stego-image will be measured by using Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) to determined its quality. PSNR with the value of 30 dB to 50db is considered having a very good quality image, while MSE value closer to 0 dB is considered having less changes from the original images used in the embedding process, and SSIM value closer to 1 or 100% is considered identical or very similar to the original image (this paper use 1 as an identical indicator). The MSE, PSNR and SSIM formula used is as below.

$$MSE = (1/n)\ \Sigma\ (Y_i - \bar{Y}_i)^2$$

$$PSNR = 10\log_{10}((MAX_i^2) / MSE)$$

$$SSIM = (2 * \mu x * \mu y + c_1) * (2 * \sigma xy + c_2) / (\mu x^2 + \mu y^2 + c_1) * (\sigma x^2 + \sigma y^2 + c_2)$$

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 1, January 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v9i1.13185 p-ISSN : 2541-044X

RESULT

After conducting the experiments twice one using the common LSB method and second using the proposed method using the three images “Lena”, “Baboon”, and “Pepper” each embedded with varying text data size ranging from 1000 bytes up to 5000 bytes. The experiments results is presented in table form, the table is obtained by comparing the common LSB method with LSB-2 with Zlib compression. A few table will be presented, table 1 and 2 will present how Zlib compression compressed the data and increased the image capacity to hold more message, table 3,4 and 5 will compare the regular LSB method and LSB-2 with Zlib compression method using PSNR, MSE and SSIM value as the stego-image quality measurement. The result of the experiments is as follow:

Table 1
Compression Result using Zlib Compression

Size Before Compression (Byte)	Size After Compression (Byte)	Compressed
1000	538	46,18%
2000	982	50,90%
3000	1407	53,08%
4000	1820	54,48%
5000	2130	57,40%

Table 1 present the size of the varying text data in bytes from 1000 bytes to 5000 bytes after being compressed by the Zlib algorithm compression and present the compression in percentage form.

Table 2
Pixel Usage for Each Methods

Data Size (byte)	Regular LSB Method	Regular LSB-2 Method	LSB-2 With Zlib Compression
1000	2667	1334	718
2000	5334	2667	1310
3000	8000	4000	1876
4000	10667	5334	2427
5000	13334	6667	2840

Table 2 present the difference of pixel usage in different method used in the embedding process, from the table it can be seen that the usage of the regular LSB-2 can decrease the pixel usage in the cover image, but to further decrease the pixel usage in the cover image, the usage of LSB-2 with Zlib Compression can furthermore reduce the pixel used in the cover image compared to the other two method, thus, making more space because there are more unused pixel in the cover image, making room to more data to be embedded into the cover image.

Table 3
Comparison Between Common LSB Method with LSB-2 Using Zlib Compression With “Lena” Image

REGULAR LSB WITHOUT COMPRESSION						LSB-2 WITH ZLIB COMPRESSION				
Data	**1000**	**2000**	**3000**	**4000**	**5000**	**1000**	**2000**	**3000**	**4000**	**5000**
PSNR(dB)	67,4637	64,3785	62,5985	61,3693	59,6919	69,8422	67,2376	65,6883	64,5424	63,0707
MSE(dB)	0,0117	0,0237	0,0357	0,0474	0,0698	0,0067	0,0123	0,0175	0,0228	0,0321
SSIM	0,999979	0,999931	0,999853	0,999778	0,999634	0,999992	0,999978	0,999958	0,999938	0,999880

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 1, January 2024
DOI : https://doi.org/10.33395/sinkron.v9i1.13185
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 3 compares the image quality of the commonly used LSB method with the proposed LSB-2 with Zlib compression method with "Lena" as cover image, which results in a significantly improved image quality. Even with using 5000 bytes of data, the LSB-2 with Zlib compression method can reaches the PSNR value of 63,0707 dB while the commonly used LSB only have 59,6919 dB PSNR value.

Table 4
Comparison Between Common LSB Method with LSB-2 Using Zlib Compression With "Baboon" Image

REGULAR LSB WITHOUT COMPRESSION						**LSB-2 WITH ZLIB COMPRESSION**				
Data	**1000**	**2000**	**3000**	**4000**	**5000**	**1000**	**2000**	**3000**	**4000**	**5000**
PSNR(dB)	67,4031	64,4298	62,6535	61,4029	59,6966	69,7187	67,2819	65,5863	64,4082	62,9940
MSE(dB)	0,0118	0,0234	0,0353	0,0471	0,0697	0,0069	0,0122	0,0180	0,0236	0,0326
SSIM	0,999999	0,999998	0,999996	0,999993	0,999988	0,999999	0,999999	0,999999	0,999998	0,999997

Table 4 use the "Baboon" image as cover image. The table proves that the usage of LSB-2 with Zlib compression have an impressive result in enhancing the image quality compared to the commonly used LSB. It can be seen in the SSIM value presented in the table that the embedding process using the proposed method using 1000 bytes of data up to 5000 bytes of data have an impressive 0,99999 SSIM value constantly which means that the stego-image from the proposed method is very identical to the original cover image.

Table 5
Comparison Between Common LSB Method with LSB-2 Using Zlib Compression With "Pepper" Image

REGULAR LSB WITHOUT COMPRESSION						**LSB-2 WITH ZLIB COMPRESSION**				
Data	**1000**	**2000**	**3000**	**4000**	**5000**	**1000**	**2000**	**3000**	**4000**	**5000**
PSNR(dB)	67,4927	64,4176	62,6722	61,4136	59,6965	69,7925	67,1376	65,6548	64,5771	63,1562
MSE(dB)	0,0116	0,0235	0,0351	0,0470	0,0697	0,0068	0,0126	0,0177	0,0227	0,0314
SSIM	0,999994	0,999980	0,999956	0,999930	0,999882	0,999997	0,999994	0,999988	0,999981	0,999962

Table 5 present the usage of image "Pepper" as cover image, after the experiment conducted the proposed method successfully increase the image quality of the stego-image. With the constantly increase PSNR value the decrease of MSE value and SSIM value very close to 1.

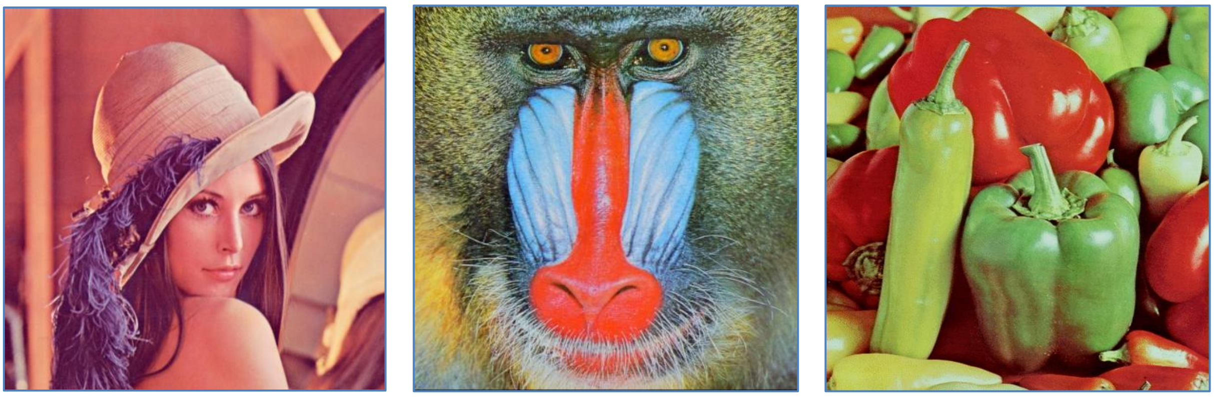

Figure 4. Original image before embedding process using 5000 bytes of text data

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 1, January 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v9i1.13185 p-ISSN : 2541-044X

The image above is the original image "Lena", "Baboon", and "Pepper" used for the embedding process, each having 512 x 512 in size with PNG file format. The PNG file format is chosen because PNG format is resistant to compression thus preserving the image quality.

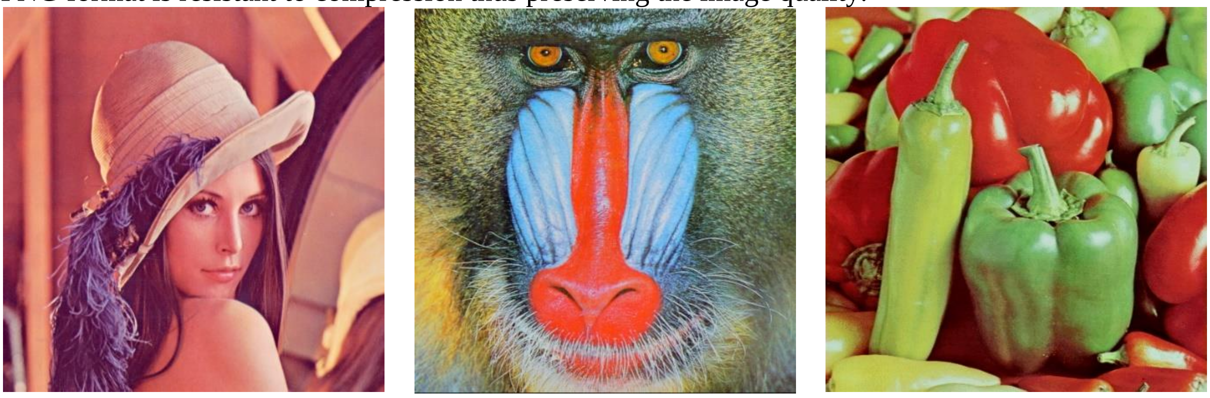

Figure 5. Image after the embedding process using 5000 bytes of text data

The image above is the image after the embedding process is done. The image from the original image and after the embedding process looks very identical, and is undetectable with human eyes. Thus confirming the success of the proposed method used to embed the text data and confirming the SSIM value in table 3, 4, and 5 which indicates the original image and the stego-image is indeed identical.

DISCUSSIONS

The usage of LSB-2 method combined with Zlib compression algorithm has been proven to increase the capacity of cover images to accommodate or receive more embeddable data. The utilization of Zlib compression convert the text data into smaller size, thus, making the pixel needed to embed the text data significantly decrease making much more space in the cover image.

From the table presented above, none of the three images used in the embedding process using the common LSB method reaches the PSNR value of 60 dB while even using 5000 bytes of data to be embedded using the LSB-2 with Zlib compression algorithm the lowest PSNR value of the stego-image is 62,9940 dB using the "Baboon" image while "Lena" image lowest PSNR value is 63,0707 dB and "Pepper" image lowest PSNR value is 63,1562 dB. Considering 30 - 50 dB of PSNR value is the threshold of a good quality image, the proposed method using LSB-2 and Zlib compression lowest PSNR value is 62,9940 dB using 5000 bytes of data is considered impressive.

Seeing from MSE perspective the proposed method have a very impressive score. Using the three image "Lena", "Baboon", and "Pepper", the proposed method constantly having a 0,0 dB score which is really close to absolute 0 dB. This proves the usage of Zlib compression algorithm have a significant effect on improving the quality of the stego-image even with the usage of LSB-2 method which theoretically should damage the stego-image more because of the usage of more bit in the RGB of each image's pixel.

The proposed method also proven to have a very impressive SSIM score. For example using the "Baboon" image, the proposed method can constantly score 0.9999 SSIM value even with the usage of 5000 bytes of data. Using the other two images, the proposed method can also score an impressive SSIM value keeping the score closer to 1 thus making the proposed image produced a stego-image identical to the original image.

CONCLUSION

Based on the experiment results, it can be concluded that this paper's goal to increase the capacity of the cover image to embed more data while preserving and even increasing the quality of the stego-image produced by the embedding process compared to the commonly used LSB method is accomplished. The proposed method which combined the LSB-2 with the Zlib compression algorithm, significantly reduce the pixel used to embed the text data varying from 1000 byte to 5000 byte. The proposed method also proven to significantly improve the quality of the stego-image compared to the

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 1, January 2024
DOI : https://doi.org/10.33395/sinkron.v9i1.13185
e-ISSN : 2541-2019
p-ISSN : 2541-044X

regular LSB method with PSNR, MSE and SSIM as measurement metrics. The experiments conducted results in an increase of capacity of approximately 36.54%, an increase in PSNR value of approximately 4.72%, accompanied by a decrease in MSE value in average of 49.19%, and SSIM values constantly at 0,99999. In further study, the usage of another embedding method combined with other compression algorithm may be used to furthermore increased the cover image's capacity or increasing the stego-image quality furthermore.

REFERENCES

AbdelWahab, O. F., Hussein, A. I., Hamed, H. F. A., Kelash, H. M., Khalaf, A. A. M., & Ali, H. M. (2019). Hiding data in images using steganography techniques with compression algorithms. *Telkomnika (Telecommunication Computing Electronics and Control)*, *17*(3), 1168–1175. https://doi.org/10.12928/TELKOMNIKA.V17I3.12230

Aldabagh, G. M. T. K. (2019). Proposed Hybrid Algorithm for Image Steganography in Text Security Using Fish Algorithm. *Journal of Southwest Jiaotong University*, *54*(6). https://doi.org/10.35741/issn.0258-2724.54.6.28

Ehsan Ali, U. A. Md., Ali, E., Sohrawordi, Md., & Sultan, Md. N. (2021). A LSB Based Image Steganography Using Random Pixel and Bit Selection for High Payload. *International Journal of Mathematical Sciences and Computing*, *7*(3), 24–31. https://doi.org/10.5815/ijmsc.2021.03.03

Hussain, M., Wahab, A. W. A., Idris, Y. I. Bin, Ho, A. T. S., & Jung, K. H. (2018). Image steganography in spatial domain: A survey. *Signal Processing: Image Communication*, *65*, 46–66. https://doi.org/10.1016/j.image.2018.03.012

Jadav, Y. (2013). *Comparison of LSB and Subband DCT Technique for Image Watermarking*.

Jayapandiyan, J. R., Kavitha, C., & Sakthivel, K. (2020). Optimal Secret Text Compression Technique for Steganographic Encoding by Dynamic Ranking Algorithm. *Journal of Physics: Conference Series*, *1427*(1). https://doi.org/10.1088/1742-6596/1427/1/012005

Md, U. A., Ali, E., Mohammad, H., Sohrawordi, M., Uddin, P., Ali, U. A. M. E., & Palash Uddin, M. (2019). *A Robust and Secured Image Steganography using LSB and Random Bit Substitution*. *2*, 39–44. www.ajer.org

Nashat, D., & Mamdouh, L. (2019). An efficient steganographic technique for hiding data. *Journal of the Egyptian Mathematical Society*, *27*(1). https://doi.org/10.1186/s42787-019-0061-6

Nilizadeh, A., Nilizadeh, S., Mazurczyk, W., Zou, C., & Leavens, G. T. (2022). Adaptive Matrix Pattern Steganography on RGB Images. *Journal of Cyber Security and Mobility*, *11*(1), 1–28. https://doi.org/10.13052/jcsm2245-1439.1111

Rustad, S., Setiadi, D. R. I. M., Syukur, A., & Andono, P. N. (2022). Inverted LSB image steganography using adaptive pattern to improve imperceptibility. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 3559–3568. https://doi.org/10.1016/j.jksuci.2020.12.017

Solak, S., Altinisik, U., & Altinişik, U. (2018). The Least Significant Two-bit Substitution Algorithm for Image Steganography. *Article in International Journal of Computer*. http://ijcjournal.org/

Tayyeh, H. K., & Al-Jumaili, A. S. A. (2022). A combination of least significant bit and deflate compression for image steganography. *International Journal of Electrical and Computer Engineering*, *12*(1), 358–364. https://doi.org/10.11591/ijece.v12i1.pp358-364

Walia, G. S., Makhija, S., Singh, K., & Sharma, K. (2018). Robust stego-key directed LSB substitution scheme based upon cuckoo search and chaotic map. *Optik*, *170*, 106–124. https://doi.org/10.1016/j.ijleo.2018.04.135

Wang, Y., Tang, M., & Wang, Z. (2020). High-capacity adaptive steganography based on LSB and Hamming code. *Optik*, *213*. https://doi.org/10.1016/j.ijleo.2020.164685

Zakaria, A. A., Hussain, M., Wahab, A. W. A., Idris, M. Y. I., Abdullah, N. A., & Jung, K. H. (2018). High-capacity image steganography with minimum modified bits based on data mapping and LSB substitution. *Applied Sciences (Switzerland)*, *10*(11). https://doi.org/10.3390/app8112199

*name of corresponding author