SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 2, April 2024
DOI : https://doi.org/10.33395/v8i2.13573
e-ISSN : 2541-2019
p-ISSN : 2541-044X

LSB-2 Steganography with Brotli Compression and base64 Encoding for Improving Data Embedding Capacity

Muhammad Yiko Satriyawibawa[1)*], Pulung Nurtantio Andono[2)], Lim Way Soong[3)], Ng Poh Kiat[4)]
[1,2)]Universitas Dian Nuswantoro, Indonesia [3,4)]Multimedia University, Malaysia
[1)]yikonyk@gmail.com, [2)]pulung@dsn.dinus.ac.id

Submitted :Dec 21, 2023 | **Accepted** : Jan 14, 2024 | **Published** : Apr 1, 2024

Abstract: Steganography functions as a technique for embedding messages or data in various forms of media, such as images, audio, video, or text, with the aim of avoiding detection by unauthorized parties. Steganography techniques can be used as a solution to hide and protect data. In this research, steganography will be carried out using images as the transmission object. This research was conducted to offer a modification of the Least Significant Bit (LSB) steganography technique using the LSB-2 method with Brotli compression and base64 encoding. Modification and use of Brotli compression and base64 coding aims to increase the message capacity that can be embedded in a transmission object while maintaining the quality of the transmission object. Experiments using small data and big data. The experimental results will be presented in tabular form by comparing the original image with the steganographically processed image using metrics such as Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) as a comparison. The experiments carried out resulted in an increase in image capacity by reducing capacity usage with an average of 47.13% for small data and an average of 71.34%. The big data experiment resulted in an increase in the PSNR value of around 3.49%, accompanied by a decrease in the average MSE value of 33.85%, and a constant SSIM value of 0.9999, thus proving that the proposed method was successful in increasing image capacity and improving stego-image quality. when embedding big data.

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

INTRODUCTION

The rapid development of communication technology in the modern era has made the process of sending data easier and faster. However, this convenience also increases the risk to the data that will be sent. From a security perspective, data is vulnerable to illegal actions such as theft, modification, or destruction by unauthorized third parties. Therefore, maintaining data confidentiality, both when stored and when transmitted, is very important. Therefore, data hiding has an important role in ensuring the security of data.

Steganography is a technique that focuses on hiding information in media generally to protect information from unauthorized access or detection. steganography is achieving attraction by people caused by the security issues over the internet. Steganography has retreated a digital strategy of hiding a file in some form of multimedia, such as an image, an audio/video files (Saxena et al., 2018). The aim of steganography is hiding the embedded information in the cover image. One of the common methods in steganography is the Least Significant Bit (LSB) method. LSB involves replacing the smallest pixel values in an image or other digital media with hidden information. These changes are subtle so they are not visible to the human eye, this makes it possible to maintain the security of information by using images or cover media.

The maximum conventional LSB method can only store around 12.28 kilobytes of data on a 512x512 RGB image. The limited space available in the least significant bit of each pixel makes it difficult for the LSB technique to hide large amounts of information without reducing the quality of the stego-image. By combining it with the data compression algorithm in the LSB method, it is possible to reduce the size of data and the use of pixels in images, so that images can contain more information without worrying about reducing the visual quality and security of the data.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 2, April 2024
DOI : https://doi.org/10.33395/v8i2.13573
e-ISSN : 2541-2019
p-ISSN : 2541-044X

In previous research, the LSB method using Discrete Cosine Transform (DCT) compression to reduce and hide secret messages was successful. Using LSB and DCT effectively reduces the overall number of bits/bytes in a file so it can be transmitted faster over slower Internet connections, or take up less space on a disk (AbdelWahab et al., 2019). DCT is used to transform the image into the frequency domain. According to the results obtained in this paper, it is clear that we can hide the intended data while minimizing its size, enabling us to transfer the data more securely with less overall burden in capacity in comparison to other messages. Various compression algorithms have also been tested, including Huffman coding, Lempel-Ziv, Run-Length Encoding (RLE), Discrete Cosine Transform (DCT), and zlib (Jayapandiyan et al., 2020).

Brotli is an open source data compression library. Brotli is developed keeping in view the, Huffman coding, modern variant of algorithm LZ77 and 2nd order context modeling. Here context modeling permits multiple Huffman trees for the similar alphabet using similar block (Zubair et al., 2018). There are more advantages with the Brotli algorithm, some of them are high compression ratio, fast compression and decompression, and can be used for any type of files.

This paper introduces an innovative approach to steganography using the 2-bit Least Significant Bit (LSB) method for data embedding. This technique aims to increase the information carrying capacity of the stego image while minimizing the potential for image quality degradation. To overcome any effects that compromise visual fidelity, this paper proposes the use of brotli compression and base64 encoding. By integrating brotli and base64 encoding with a 2-bit LSB method, this paper attempts to achieve a balance between increasing data hiding capacity and overall stego-image quality.

METHOD

Color images have parts for each pixel, the color images used in this research are images that have 3 parts for each pixel, namely R, G and B. The LSB-2 method will replace 2 values from R, G and B for inserting data.

Proposed Method

Our methodology represents a novel approach to significantly enhance both the data-hiding capacity and fidelity of conventional LSB (Least Significant Bit) steganography. This section provides a detailed account of the methods and procedures employed in our research. 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 Brotli compression algorithm and base64 encoding. 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:

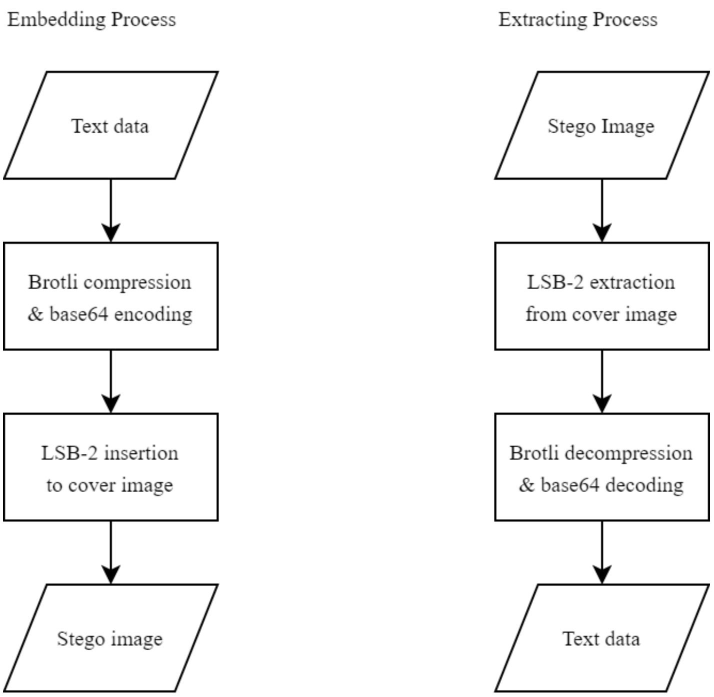

Figure 1 Embedding and extracting process flow

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 2, April 2024
DOI : https://doi.org/10.33395/v8i2.13573
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Later, the number of pixels used in the cover image and the quality of the stego-image will be compared.

Datasets

The data used to be embedded into the cover image is in form of text data which will be generated with varying data size in bytes, starting from 1,000 to 5,000 bytes for small data and starting from 10,000 to 20,000 bytes for big data. The text data generated from https://www.blindtextgenerator.com. The text data will be compressed with the Brotli algorithm to decrease the size thus decreasing the amount of pixel used in the 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 “baboon”, “lena” and “pepper”.

Figure 2 “baboon”, “lena”, and “pepper” PNG with 512x512

Brotli Compression & base64 Encoding

The text data that will be inserted into the image will previously be compressed using the Brotli compression algorithm. Then the results will be encoded using base64 so that the results of brotli compression can be read and become text as usual so that later it will be converted to binary type so that it can be inserted into the image. This process uses the library provided in the Python programming language. The compression process can be seen in the following chart.

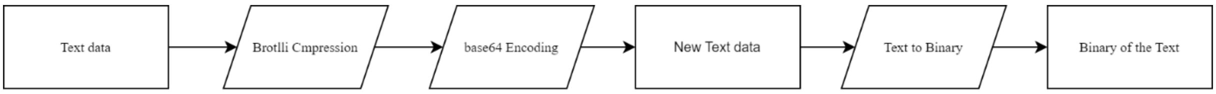

Figure 3 Brotli compression and base64 encoding flow

Embedding Process

Unlike the widely used LSB approach, which uses the last binary value in each RGB of each pixel in the cover picture, the embedding process will be using the LSB-2 method which employs the two last binary values in each RGB for the embedding process. Using the LSB-2 technique, the text data that has been binary-compressed using the Brotli compression and base64 encoding algorithm will be inserted into the cover image. An image's pixel, for instance, has RGB binary bits that look like this: **R = 10111001**, **G = 11001011**, and **B = 10110010**. The binary of the text data to be embedded is **101001**. After the embedding procedure, the RGB binary value will change to **R = 10111010**, **G = 11001010**, **B = 10110001**.

Extracting Process

In order to get the original text data, first extract the final two binary values for each RGB pixel in the cover image. The extraction process will produce a binary string form that must be transformed into hexadecimal form. This hexadecimal form must then be decoded using the base64 to become the compressed output of the Brotli compression algorithm, which can then be transformed back into the original text data embedded in the cover image with Brotli decompression technique.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 2, April 2024
DOI : https://doi.org/10.33395/v8i2.13573
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Stego-image

The embedding process will produce a stego-image containing the embedded text data. The quality of the Stego image can be seen using Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM).

Evaluation Metrics

This research uses Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) as methods for measuring research results. Stego-image will be measured using Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) to determine its quality. PSNR with a value of 30 dB to 50 dB is considered to have excellent image quality, while MSE values close to 0 dB are considered to have smaller changes from the original image used in the embedding process, and SSIM values are close to 1 or 100%. considered identical or very similar to the original. The MSE, PSNR and SSIM formulas used are as below.

$$MSE = (1/n)\ \Sigma\ (Y_i - \bar{Y}_i)^2$$
$$PSNR = 10\log_{10}((MAX_i^2) / MSE)$$
$$SSIM = (2 * \mu x * \mu y + c_1) * (2 * \sigma xy + c_2) / (\mu x^2 + \mu y^2 + c_1) * (\sigma x^2 + \sigma y^2 + c_2)$$

RESULT

After conducting the experiments twice one using the conventional LSB method and second using the proposed method using the three images “Baboon”, “Lena”, and “Pepper” each embedded with varying text data size ranging from 1.000 bytes up to 5.000 bytes for small data and 10.000 bytes to 20.000 bytes for big data. The experiments results is presented in table form, the table is obtained by comparing the conventional LSB method with LSB-2 method with Brotli compression and base64 encoding. A few table will be presented, table 1 will present result of Brotli compression and base64 encoding compressed the data text. Table 2 will present the pixel usage of data text. The fewer pixels used, the less data capacity used, so the image can accommodate more message. Table 3,4 and 5 will compare the regular LSB method and LSB-2 method with Brotli compression and base64 encoding using PSNR, MSE and SSIM value as the stego-image quality measurement. The result of the experiments is as follow:

Table 1
Text Data Compression Result with Brotli Compression and base64 Encoding.

Default Size (Byte)	Size after Compression (Byte)	Compression Precentage
1,000	664	33.60%
2,000	1,144	42.80%
3,000	1,492	50.27%
4,000	1,848	53.80%
5,000	2,240	55.20%
10,000	3,820	61.80%
20,000	3,824	80.88%

From table 1, the results of the data size and percentage of Brotli compression and base64 encoding carried out with a data range between 1.000 to 2.0000 can be seen.

Table 2
Pixel Usage from each Methods

Data Size (Byte)	Conventional LSB Method	LSB-2 Method	LSB-2 Method with Brotli Compression and base64 Encoding
1,000	2,667	1,334	886
2,000	5,334	2,667	1,526
3,000	8,000	4,000	1,990

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 2, April 2024
DOI : https://doi.org/10.33395/v8i2.13573
e-ISSN : 2541-2019
p-ISSN : 2541-044X

4,000	10,667	5,334	2,464
5,000	13,334	6,667	2,986
10,000	26,667	13,334	5,094
20,000	533,334	26,667	5,099

Table 2 presents the differences in pixel usage in the various methods used in the embedding process. From this table it can be seen that using the LSB-2 method with brotli compression and base64 encoding can reduce more pixel usage in the cover image than the other two methods, even though the LSB-method 2 regular can already reduce the use of pixels on the cover image, the addition of brotli compression and base64 encoding can further reduce the use of pixels used on the cover image, resulting in more space that can be used for embedding messages in the cover image.

Table 3
Comparison Between Conventional LSB Method with LSB-2 Method with Brotli Compression and base64 Encoding on "baboon" Image

Conventional LSB				**LSB-2 with Brotli Compression and base64 Encoding**			
Data (Byte)	**PSNR**	**MSE**	**SSIM**	**Data (Byte)**	**PSNR**	**MSE**	**SSIM**
1,000	70.991471	0.005175	0.999999	1,000	69.174721	0.007863	0.9999997
2,000	67.998814	0.010309	0.999998	2,000	66.890853	0.013304	0.9999994
3,000	66.257050	0.015395	0.999996	3,000	65.554756	0.018097	0.9999989
4,000	65.008111	0.020524	0.999994	4,000	64.765080	0.021706	0.9999983
5,000	64.049841	0.025592	0.999993	5,000	63.915560	0.026395	0.9999978
10,000	61.034688	0.051240	0.999973	10,000	61.594072	0.045048	0.9999943
20,000	58.035503	0.102219	0.999947	20,000	61.593704	0.045052	0.9999942

Table 3 compares the image quality of the commonly used LSB method with the proposed LSB-2 method with Brotli compression and base64 encoding on "baboon" as cover image, which results in a significantly improved image quality when using 10,000 and 20,000 bytes of data. From table, in 20,000 bytes of data the PSNR increased to 61.593704 from 58.035503 and MSE decreased to 0.045052 from 0.102219. But in 5,000 bytes of data PSNR decreased to 63.915560 from 64.049841 and MSE increased to 0.026395 from 0.025592. So when using between 1,000 to 5,000 the quality of conventional LSB method is more better. The SSIM always increased on proposed methods.

Table 4
Comparison Between Conventional LSB Method with LSB-2 Method with Brotli Compression and base64 Encoding on "lena" Image

Conventional LSB				**LSB-2 with Brotli Compression and base64 Encoding**			
Data (Byte)	**PSNR**	**MSE**	**SSIM**	**Data (Byte)**	**PSNR**	**MSE**	**SSIM**
1,000	70.950054	0.005225	0.999981	1,000	69.462130	0.007360	0.999991
2,000	67.985442	0.010340	0.999938	2,000	66.996307	0.012985	0.999979
3,000	66.240938	0.015452	0.999888	3,000	65.735134	0.017361	0.999965
4,000	64.993606	0.020593	0.999841	4,000	64.873795	0.021169	0.999951
5,000	64.032612	0.025693	0.999796	5,000	64.051136	0.025584	0.999928
10,000	61.042779	0.051145	0.999576	10,000	61.661277	0.044356	0.999912

*name of corresponding author

20,000	58.042153	0.102062	0.999244	20,000	61.654932	0.044421	0.999902

Table 4 compares the image quality of the commonly used LSB method with the proposed LSB-2 method with Brotli compression and base64 encoding on "lena" as cover image, which results in a significantly improved image quality when using 5,000, 10,000 and 20,000 bytes of data. From table, in 20,000 bytes of data the PSNR increased to 61.654932 from 58.042153 and MSE decreased to 0.044421 from 0.102062. But in 4,000 bytes of data PSNR decreased to 64.873795 from 64.993606 and MSE increased to 0.021169 from 0.020593. So when using between 1,000 to 4,000 the quality of conventional LSB method is more better. The SSIM always increased on proposed methods.

Table 5
Comparison Between Conventional LSB Method with LSB-2 Method with Brotli Compression and base64 Encoding on "pepper" Image

Conventional LSB				**LSB-2 with Brotli Compression and base64 Encoding**			
Data (Byte)	**PSNR**	**MSE**	**SSIM**	**Data (Byte)**	**PSNR**	**MSE**	**SSIM**
1,000	71.053806	0.005102	0.99999	1,000	68.970325	0.008242	0.999998
2,000	68.052175	0.010183	0.99997	2,000	66.716339	0.013850	0.999993
3,000	66.305386	0.015224	0.99994	3,000	65.426076	0.018641	0.999987
4,000	65.067985	0.020243	0.99992	4,000	64.576389	0.022669	0.999980
5,000	64.113531	0.025219	0.99989	5,000	63.793843	0.027145	0.999968
10,000	61.086185	0.050636	0.99975	10,000	61.575844	0.045237	0.999908
20,000	58.094519	0.100839	0.99943	20,000	61.528856	0.045729	0.999909

Table 5 compares the image quality of the commonly used LSB method with the proposed LSB-2 method with Brotli compression and base64 encoding on "pepper" as cover image, which results in a significantly improved image quality when using 10,000 and 20,000 bytes of data. From table, in 20,000 bytes of data the PSNR increased to 61.528856 from 58.094519 and MSE decreased to 0.045729 from 0.100839. But in 5,000 bytes of data PSNR decreased to 63.793843 from 64.113531 and MSE increased to 0.027145 from 0.025219. So when using between 1,000 to 5,000 the quality of conventional LSB method is more better. The SSIM always increased on proposed methods.

Figure 4 Original "baboon", "lena", and "pepper" images

Figure 4 is the original image before the experiment was carried out. This image measures 512x512 with RGB channels. From the image it looks still clean, clear and without noise. This image is a comparison with the resulting image after steganography which is shown in figure 5.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 2, April 2024
DOI : https://doi.org/10.33395/v8i2.13573
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Figure 5 “baboon”, “lena”, and “pepper” images after embedded with 20,000 bytes of data using proposed method

Figure 5 shows the image that has been embedded with 20,000 bytes of data. Images that have embedded data look very similar to the original image and are not identifiable when viewed directly by the human eye. This is because the proposed method is successful in compressing the number of bytes of data before embedding it into the cover image so that the number of pixels used is also less. The result is that the SSIM values shown in tables 3, 4 and 5 are almost close to 1, this results in the stego-image looking almost the same as the original image.

DISCUSSIONS

The usage of LSB-2 method with Brotli compression and base64 encoding has been proven to increase the capacity of cover images to accommodate or receive more embeddable data. The utilization of Brotli compression and base64 encoding 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. The brotli method is very effective when compressing large data sizes. The larger the data size, the smaller the compression results of the Brotli method. Judging from the table, when Brotli compresses data measuring 10,000 bytes and 20,000 bytes, the result is only a 4 bytes difference in data size results and a 5 pixel difference in pixel usage.

From the table presented, when 10,000 bytes of data are inserted using the conventional LSB method or using the proposed method, the PSNR is not less than 60 dB, which means it is good and without noise. However, when 20,000 bytes of data are inserted, in the conventional LSB method the PSNR value is between 50 dB and 60 dB, which is included in the good but noisy criteria, whereas in the proposed method the PSNR value is still above 60 dB. Considering that PSNR values between 30 dB and 50 dB are common values for image assessment, the proposed method provides impressive results while maintaining a PSNR value above 60 dB even though it has inserted 20,000 bytes of data, meaning a good improvement by the proposed method. PSNR value test results after being inserted into the image. The PSNR value of "baboon" is 61.593704 dB, "lena" is 61.654932 dB, and "pepper" is 61.528856 dB.

In terms of MSE values in testing, the conventional LSB method and the proposed method both have values that almost touch 0 dB, namely remaining at a value of 0.0 dB when inserting 10,000 bytes of data. However, when inserting data of 20,000 bytes, the MSE value of the conventional LSB method is 0.1 dB for the three images, while the MSE value of the proposed method is still at 0.0 dB, this is a good improvement.

In terms of the SSIM value in testing, the proposed method can have a constant SSIM value of 0.9999 when inserting 20,000 bytes of data. This result is very impressive where the image after inserting 20,000 bytes of data still looks similar to the original because the SSIM value is almost 1.

CONCLUSION

Based on the experimental results, it can be concluded that the aim of this paper is to increase the capacity of the cover image to embed more data while minimizing damage and even improving the quality of the stego-image produced by the embedding process compared to the conventional LSB method. The proposed method which combines LSB-2 with Brotli compression algorithm and base64 encoding, significantly reduces the pixels used to embed text data varying from 1,000 bytes to 5,000 bytes for small data and 10,000 bytes to 20,000 bytes for big data. The experiments carried out succeeded in increasing image capacity by reducing image capacity usage by an average of 47.13% on small data and an average of 71.34% on big data. In Brotli compression, the larger the data to be compressed, the greater the compression percentage produced. The proposed method is also proven to improve the quality of stego-images when experiments using big data compared to the conventional LSB method with PSNR, MSE and SSIM as measurement metrics with the result of increasing the PSNR value by around

*name of corresponding author

3.49%, accompanied by a decrease in the average MSE value of 33.85%, and a constant SSIM value of 0.9999. In further studies, the use of other embedding methods combined with other compression algorithms could be used to further increase the cover image capacity or further improve the stego-image quality.

ACKNOWLEDGMENT

The authors wish to express their sincere gratitude to Pulung Nurtantio Andono for his invaluable guidance and support as the academic supervisor during this research. Additionally, we extend our appreciation to the Faculty of Computer Science at Universitas Dian Nuswantoro, Indonesia, and the Faculty of Engineering and Technology at Multimedia University, Malaysia, for their contributions to this study. Your mentorship and resources have been instrumental in the successful completion of this research.

REFERENCES

Saxena A. K., Sinha S., & Shukla P.(2018). Design and Development of Image Security Technique by Using Cryptography and Steganography: A Combine Approach. *International Journal of Image, Graphics and Signal Processing*, *10*(4), 13-21. https://doi.org/10.5815/ijigsp.2018.04.02

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

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

Zubair A. S., & Soomro T. R (2018). Compression Algorithms: Brotli, Gzip and Zopfli Perspective. *Indian Journal of Science and Technology*, *11*(45), 1-4. https://doi.org/10.17485/ijst/2018/v11i45/117921

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

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

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

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

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

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

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

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

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

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/

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

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

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

*name of corresponding author