JINAV: Journal of Information and Visualization *ISSN: 2746-1440 (online)*
Vol. 5 No. 1 (2024)
https://doi.org/10.35877/454RI.jinav2327

Comparative Analysis of Openpuff and Openstego Tools

Linda Heryanti, Wiga Maulana Baihaqi, Ariska Nurul Habibah, Bagus Adhi Kusuma

Amikom Purwokerto University, North Purwokerto, Banyumas City, Central Java, 53127, Indonesia

Abstract

Steganography is the science and art of hiding information in a medium so that its existence is not detected by unauthorized parties. Media that can be used in steganography are text, images, audio, and video. However, the media that is often used is image. Various steganography tools have been developed with their respective strengths and weaknesses, such as Hide in Picture; Openstego; Image Steganography; Invisible Secret 4; S-Tools; Hide 'N' Send; Online Image Steganography, Openpuff, and others. Researchers carried out a comparative analysis of the steganography tools Openpuff and Openstego with test parameters for the quality of the images produced and time efficiency. Test the quality of the resulting image using MSE, PSNR, NCC, SSIM, and time-efficient testing seen from embedding and extraction time. Based on the research results, show that Openstego has better image quality and time efficiency compared to Openpuff. The type of image format used and the size of the embedded message can affect the quality of the resulting image and the time used. The best test results were obtained, namely MSE=0.0009, PSNR=78.5438 dB, NCC=0.999999, SSIM=0.999993 and required embedding time=0.075 second and extraction time=0.084 second.

Keywords: image quality, openpuff, openstego, steganography, time.

Received: 9 January 2024 Revised: 14 May 2024 Accepted: 23 June 2024

1. Introduction

In the current digital era, technology is developing rapidly and the existence of the internet makes information easy to obtain and disseminate. Therefore, information security is needed to maintain the confidentiality, availability, and integrity of information (Nurul et al., 2022). Without security, information can become vulnerable to various risks and threats. Southeast Asia Freedom of Expression Network (SAFEnet) data for 2021, there was an increase in digital attacks in Indonesia by 38% compared to the previous year. The most frequent form of digital attack in Indonesia is hacking with a percentage of 70.46% of the total digital attacks in the previous year (Dihni, 2022).

Steganography is an information security method that is widely used by hiding information in media so that other people cannot see it (Nurhasanah et al., 2023). Steganography comes from the Yunani, steganos, which means "to hide" and graptos, which means "writing" (Permana & Amna, 2022). Media that can be used in the steganography method are text, images, audio and video (Malese, 2021). (Darwis & Pasaribu, 2020) Images are the most widely used media in steganography because images are often used to exchange information. The most important part of applying steganography is the media used as a container and the message or secret information that will be hidden (Rohayah, 2022). (Hidayat et al., 2022) There are three things you must pay attention to when hiding or inserting messages, namely:

1) Fidelity: After adding secret data, the quality of the cover image changes slightly and the stego image is clearly visible. The observer does not realize that the image contains secret data.
2) Robustness: The hidden data must be resistant to any type of cover image manipulation, such as image editing and processing and the data should not be damaged.
3) Recovery: Hidden data must be able to be re-disclosed for further use

Steganography is also called Hidden in Plain Text Sight because the data in steganography is hidden openly and always visible, but it is difficult to detect if there is a secret message in it (Hidayat et al., 2022). In general, the

* Corresponding author.
E-mail address: heryantil74@gmail.com

steganography process is divided into two stages, namely inserting a message using a digital image and a secret message as input, then processing it to produce a steganographic image, while the message extraction stage involves inputting the resulting steganographic image, processing it, and producing a digital image along with a secret message embedded (Mulyono et al., 2023). (Anshori et al., 2019) The steganography method is carried out by inserting confidential information into the media with a certain algorithm so that the media does not visually change or arouse suspicion. Then the message is extracted again by the party authorized to receive it. The purpose of steganography is to keep secret or hide important messages or information in a medium (Siaulhak & Kasma, 2023).

Various steganography tools have been developed with their respective strengths and weaknesses, such as Hide in Picture; Openstego; Image Steganography; Invisible Secret 4; S-Tools; Hide 'N' Send; Online Image Steganography, Openpuff, and others. The implementation and analysis of the process of inserting and extracting messages using steganography tools has been carried out by many previous studies. (Laksiati, 2021) use OpenStego to secure message in image using the AES algorithm to maintain confidentiality. The results of this research are that the size of the secret message and the image pixels used can influence the quality of the resulting image. The larger the size of the secreat message embedded and the smaller the size of the image pixels used, the worse the quality of the resulting image. This is proven by the initial PSNR value in the bmp image file with a pixel value of 1000x1252 with a secret message size of 50, showing a PSNR value = 88.72 dB, then the PSNR value decreases as the message size increases until a message size of 1000 shows PSNR = 83.99. For BMP image files with a pixel value of 100x125, the resulting PSNR value is <70 dB

(Kamil et al., 2018) implementing Openpuff and Oursecreat steganography tools to hide information in image, audio, and video media. The results of this research show that the steganography files produced by steganography tools after going through the sending process via the WhatsApp, Messenger, and BBM applications, the information that is secured cannot be encrypted because the information is damaged.

(Wang et al., 2020)detect hidden information in videos and analyze steganography and watermarking in video files using Openpuff. The results of this research show that Openpuff performance is quite good in identifying differences between original files and files that have been modified, and shows a high level of detection accuracy as proven by the values TP=100%, FP=0%, TN=100%, and FP= 0%.

(Islam et al., 2020) analyzing the performance of steganography tools shows that the Openstego and S-Tool tools have the best performance based on the empirical results tested. The empirical results obtained by Openstego are MSE=0.02; PSNR=66.52; AD=0.002; MD=1.00; MAE=0.026; SSIM=0.98; DSSIM=173.85; SNR=66.16; SC=1.00; NCC=1.00; CQ=178.38; IF=1.00; and PMSE=41.80x10-8. Meanwhile, the empirical results obtained by S-Tool are almost the same as Openstego, namely MSE=0.02; PSNR=66.55; AD=0.002; MD=1.00; MAE=0.02; SSIM=0.98; DSSIM=179.92; SNR=66.19; SC=1.00; NCC=1.00; CQ=178.38; IF=1.00; PMSE=41.70x10-8.

(Oladeji et al., 2020) The most effective steganography tool used with the parameters for comparing size, insertion time, and extraction time is OpenStego with the result that the size of the inserted data becomes larger, so the size of the stego image remains the same for all file formats produced; time for data insertion, if the size increases, a stable insertion time is required.

(Guptaa et al., 2020) conducted research on the performance analysis of the Openpuff steganography tool using various image formats, showing that Openpuff' performance depend on the image format used. The BMP format is the best format compared to JPEG and PNG. By proving that in terms of insertion time and extraction time, it shows the most efficient, namely insertion time = 0.9689 seconds and extraction time = 0.773 seconds.

In this research, researchers will conduct a comparative analysis of the Openpuff and Openstego tools using MSE, PSNR, NCC, and SSIM testing to evaluate and assess the quality of the original image and stego image. (Kasapbaşı, 2019) that SSIM and NCC are able to represent perception-based errors and understanding errors, while PSNR and MSE represent absolute errors. The more the PSNR, NCC, and SSIM values increase and the MSE value decreases, the better the image quality (Singh, 2017). Researchers also analyzed the time required for embedding and extracting messages. The aim of this research is to determine the performance of the Openpuff and Openstego tools as seen from the quality of the images produced.

2. Method

The research method used is experimental to know the comparison of Openpuff and Openstego steganography tools in terms of the quality of the image produced and the efficient time needed. A comparison of image quality produced from each steganography tool used can be done with MSE, PSNR, NCC, and SSIM tests (Rahman et al., 2020). The efficient comparison of the time needed can be done by comparing the time during the embedding and extraction process.

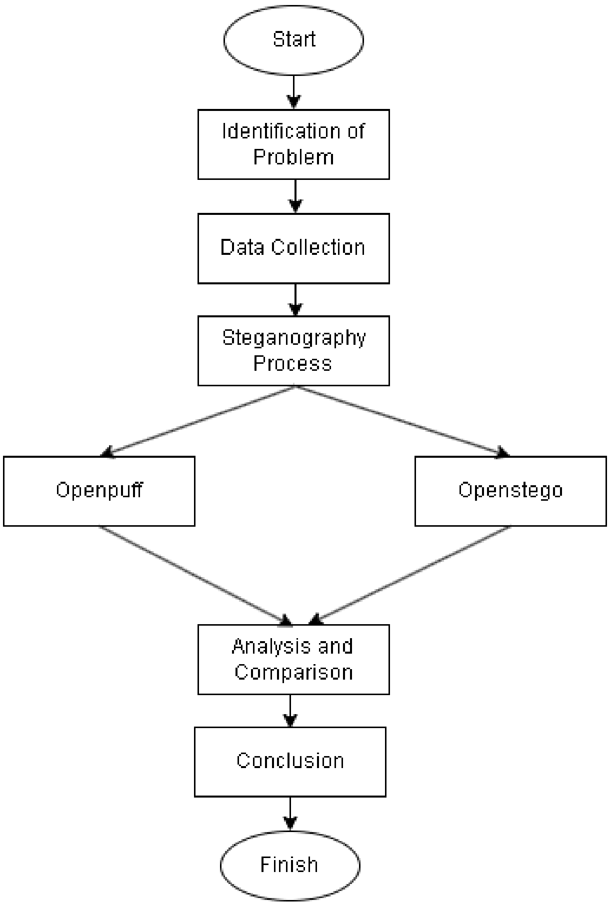

Fig. 1. Research Concept

2.1. Identification of Problem

The initial stage of this research is to identify the problems to be studied. At this stage, the main problem discussed is the lack of a comprehensive understanding of the comparison of the performance of Openpuff and Openstego steganography tools. The aspects that are the focus of research are the quality of the images produced and the speed of time in the process of embedding and extracting messages. With clear problem identification, this study aims to provide a review of research results on each steganography tool, so that it can contribute to the development of steganography technology that is more effective and safe.

2.2. Data Collection

At this stage, researchers collect data through literature studies and take secondary data from search sites in the form of images with various image formats, namely JPG, PNG, and BMP. The literature studies used come from articles, journals, and several books related to Openpuff and Openstego (Kumar et al., 2023). Secondary data is obtained from sources on the search site and the following search site links are used as secondary data retrieval, namely: https://pixabay.com/id/. The embedded secret message data comes from internal sources and varies in size, including but not limited to 1 kb, 22 kb, and 224 kb in text format. All data collected for the study depicted in Fig 2.

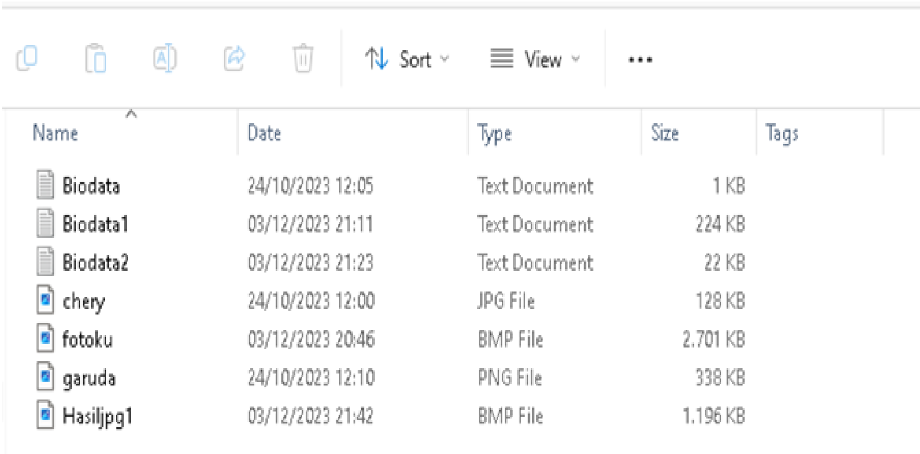

Fig. 2. Research Material

2.3. Steganography Process

The steganography process is carried out on two tools, namely Openpuff and Openstego. According to Sloan & Castro, (2018), Openpuff is one of the steganography tools that is open source and is used to protect secret messages made by people when exchanging messages. The implementation of Openpuff is easy to do, secure, and free in securing data into encrypted files to be sent to other users. Openstego is an open-source and free steganography tool developed using the Java programming language. Openstego tools can be used to insert and extract files and can be used for digital watermarking (Arora, 2022).

2.3.1 Openpuff

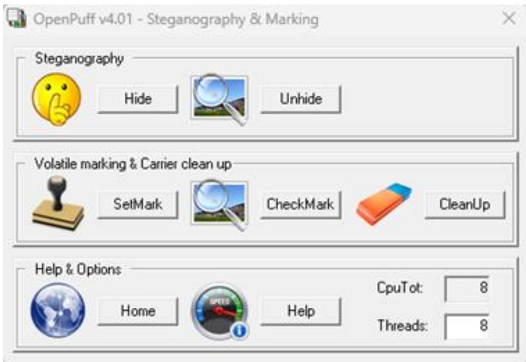

Fig. 3. Openpuff

At this stage, a steganography process is carried out using Openpuff. The steganography process carried out consists of message insertion and message extraction. The steps for inserting a message are carried out in the Openpuff tool, namely:

a. Input password, secret data file, and cover media file

Enter a password as a security key for the stego file that will be generated. There are three password input columns in Openpuff. However, it is not mandatory to fill in all of them, you can fill in anyone. Enter the secret data file that will be inserted and the media cover file that will be the container for inserting the message. The maximum size of the secret data file is 256 MB.

b. Choose percentage embedding and location storage stego file

Select the percentage rate at which message insertion will be performed. Then select the storage location for the stego file that you want to go to. The stego file format that will be produced will be in the same format as the original media cover.

For the message extraction process on Openpuff, this can be done by:

a. Input password and stego file

Enter the same password during the message insertion process. If the password is incorrect then the stego file cannot be extracted. Also, enter the stego file which will be extracted to restore the message stored in the media cover so that it can be reused or used for other purposes.

b. Choose location storage message

Select the storage location for the extracted message.

2.3.2 Openstego

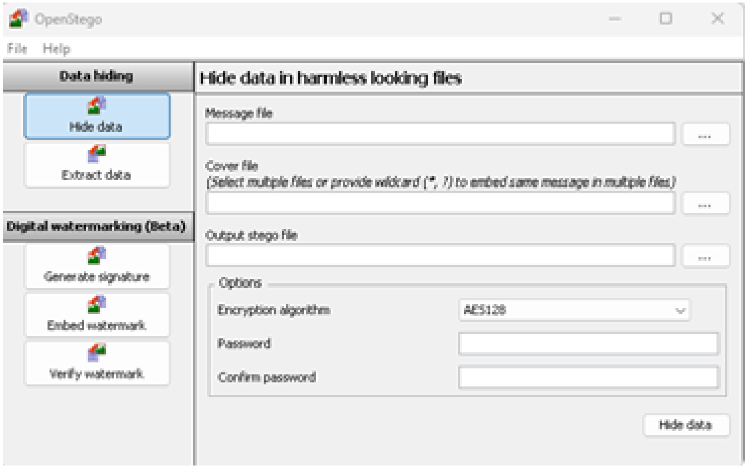

Fig. 4. Openstego

At this stage, a steganography process is carried out using Openstego. The steganography process carried out consists of message insertion and message extraction. The steps for inserting messages are carried out in the Openstego tool, namely:

a. Input massage file and cover file

Enter the secret message file that will be hidden and the media cover file that will be the container for hiding the message. The size of the secret message file to be hidden cannot exceed the size of the cover media file used.

b. Choose location storage and rename stego file

Select the storage location for the generated stego and give the resulting stego file a name. If you just name a file without the desired extension, the resulting stego file format will be BMP, but if you want to give it a stego file extension then it can only be in PNG format.

c. Select security

Select the security used to secure the stego file. There are two types of AES (Advanced Encryption Standard) algorithms, including the AES 128 and AES 256 algorithms. This algorithm is used to encrypt messages hidden in cover media. There is also a password to lock the stego file. However, if you don't enter a password, the message can still be inserted into the media cover. However, the resulting stego file does not have double/full security.

For the message extraction process in Openstego, this can be done by:

a. Input stego file

Enter the secret message file that will be hidden and the media cover file that will be the container for hiding the message.

b. Choose location storage massage file

Select the storage location for the generated stego and give the resulting stego file a name. If you just name a file without the desired extension, the stego file format will produce BMP, but if you want to give it a stego file extension then it can only be in PNG format.

c. Input security

Select the security used to secure the stego file. There are two types of AES (Advanced Encryption Standard) algorithms, including the AES 128 and AES 256 algorithms. This algorithm is used to encrypt messages hidden in cover media. There is also a password to lock the stego file. However, if you do not enter a password, the message can still be inserted into the media cover. However, the resulting stego file does not have double/full security

2.4. Analisis and Comparison

The analysis was carried out by comparing experimental results from both steganography tools. Performance parameters such as the quality of the resulting image and the efficient time required. The resulting image quality can be tested via MSE, PSNR, NCC, and SSIM.

MSE (Mean Square Error) is an assessment that is often used and is the best for a matrix that measures image quality with complete reference metrics and a value close to zero is the best value. An estimator measure that shows an estimate that is much different from the estimate in terms of the variance and degree of distortion of the original secret message. MSE can also represent the quality of the images produced. The smaller the MSE value produced, the better the resulting image quality (Sumijan et al., 2019). The following is the MSE equation:

$$MSE = \frac{1}{xyz} \sum_{i=1}^{y} \sum_{j=1}^{z} (m_{ij} - n_{ij}) \qquad (1)$$

PSNR (Peak Signal to Noise Ratio) is used to determine image quality after embedding a message. The resulting image will be compared with the original image. The greater the PSNR value obtained, the better the resulting image quality and vice versa (Sumijan et al., 2019). The PSNR value can be said to be good, so the value range is between 20-60 dB. The high signal and lots of noise will affect the representation of the reconstructed process and the original data determined using PSNR (Jani Anbarasi et al., 2020). The following is the PSNR equation:

$$PSNR = 10 \, x \, 10 \log \left[255^2 / MSE \right] \qquad (2)$$

NCC (Normalized Cross Correlation) is a measuring parameter in determining image similarity based on the correlation function (Sara et al., 2019). The previously processed image extraction values can also be used by NCC to determine the similarity of two images (Saleh et al., 2020). The following is the equation for calculating NCC:

$$NCC = \frac{\sum_{X=1}^{N} \sum_{Y=1}^{M} [a(x,y).b(x,y)]}{\sqrt{\sum_{X}^{N} \sum_{y}^{M} [a(x,y)]^2} . \sqrt{\sum_{X}^{N} \sum_{y}^{M} [b(x,y)]^2}} \qquad (3)$$

SSIM (Structural Similary Index Measure) is a method based on perception by considering image degradation as changes that occur in the combined structural information. SSIM is used as an assessment parameter for the level of similarity between the stego image and the original image. The SSIM value obtained from the stego image is close to 1, so the stego image is increasingly similar to the original image (Ramadhan & Wirawan, 2021). The following is the SSIM equation:

$$SSIM\ (x,y) = [I(x,y)]^{\propto} . [c(x,y)]^{\beta} . [s(x,y)]^{\gamma} \quad (4)$$

2.5. Conclusion

The conclusion resulting from the analysis include answer to the problem studied, research objective, and suggestion for reader, as well as describing the impact of the research result.

3. Result and Discussion

In this study, researchers used digital images taken from search websites with different formats and sizes, namely jpg format with a size of 128 kb, png format with a size of 338 kb, and bmp with a size of 2,701 kb. Meanwhile, messages that will be embedded in images have TXT format with varying sizes, namely 1 kb, 22 kb, and 224 kb. The following are the results of embedding and extraction tests using Openpuff and Openstego. The data listed is the result of each group for each steganography tool used and can be seen in Table 1 and Table 2.

Table 1. Openpuff Tools Results

Image Format	Message size	MSE	PSNR (dB)	NCC	SSIM	Emmbedding time (second)	Extraction time (second)
png	1 kb	**0.1074**	**57.8227**	**0.9999898**	**0.998726**	40.77	41.23
jpg	1 kb	-	-	-	-	-	-
bmp	1 kb	0.1314	56.9464	0.9999632	0.999048	**5.22**	**5.50**
png	22 kb	**0.1075**	**57.8172**	**0.9999897**	**0.998723**	47.28	48.57
jpg	22 kb	-	-	-	-	-	-
bmp	22 kb	0.1314	56.9456	0.9999630	0.999043	**5.63**	**5.80**
png	224 kb	-	-	-	-	-	-
jpg	224 kb	-	-	-	-	-	-
bmp	224 kb	-	-	-	-	-	-

*Note: The steganography process in Openpuff cannot be carried out, if when entering the message file to be embedded and the image file which is the media cover it produces select/total with a secret message file size larger than the image file, then the embedding process cannot be carried out.

Table 1 shows the results of the performance of the steganography tools on Openpuff based on the resulting image quality test, where the image format used and the size of the embedded message affect the quality of the resulting image. The results show that the PNG image format is better than the BMP and JPG format. The smaller the size of the embedded message, the better the resulting image quality, with values obtained MSE=0.1074, PSNR=57.8227 dB, NCC=0.9999898, and SSIM=0.998726. Meanwhile, looking at the time, the BMP format is more efficient, namely by obtaining the best embedding time = 5.22 second and extraction time = 5.50 second.

Table 2. Openstego Tools Results

Image Format	Message size	MSE	PSNR (dB)	NCC	SSIM	Emmbedding time (second)	Extraction time (second)
png	1 kb	0.0011	77.6451	0.999999	0.999987	0.084	0.090
jpg	1 kb	0.0022	74.7992	0.999999	0.999975	0.085	0.095
bmp	1 kb	**0.0009**	**78.5438**	**0.999999**	**0.999993**	**0.075**	**0.084**
png	22 kb	0.0339	62.8240	0.999997	0.999633	0.091	0.097
jpg	22 kb	0.0665	59.9035	0.999995	0.999281	0.092	1.002
bmp	22 kb	**0.0295**	**63.4401**	**0.999991**	**0.999780**	**0.090**	**0.096**
png	224 kb	0.3794	52.3399	0.999976	0.997384	2.034	3.010
jpg	224 kb	2.1757	44.7547	0.999928	0.992186	3.011	3.022
bmp	224 kb	**0.3299**	**52.9465**	**0.999915**	**0.997782**	**2.010**	**2.093**

Table 2 shows the results of the performance of the steganography tools on Openstego based on the resulting image quality test, where the image format used and the size of the embedded message affect the quality of the resulting image. The results show that the BMP image format is better than the PNG and JPG format. The smaller the size of the embedded message, the better the quality of the resulting image, with values obtained MSE=0.0009, PSNR=78.5438 dB, NCC=0.999999, SSIM=0.999993 and requiring embedding time=0.075 second and extraction time=0.084 second.

4. Conclusion

Based on the results and discussion of the comparative analysis of the performance of the Openpuff and Openstego steganography tools by conducting image quality tests and analyzing the time required for the embedding and extraction processes on both tools, it can be concluded:

a. The best performance of the steganography tool is Openstego, proven by the time used being more efficient than Openpuff, namely embedding time = 0.075 second and extraction time = 0.084 second. Apart from that, it can be seen from the image quality results with the best values MSE=0.0009, PSNR=78.5438 dB, NCC=0.999999, and SSIM=0.999993.

b. The image format and size of the embedded message affect the image quality results and the time required during the embedding and extraction process.
c. The best image format on Openpuff is produced from PNG, while the best image format on Openstego is produced from BMP. The larger the size of the embedded message, the worse the resulting image quality.

References

Anshori, Y., Dodu, A. Y. E., & Purwaningsih, M. (2019). Aplikasi Steganografi pada Media Citra Digital Menggunakan Metode Least Significant Bit (LSB). *SATIN: Sains Dan Teknologi Informasi*, *5*(1), 1–10. https://core.ac.uk/download/pdf/333817100

Arora, N. (2022). Types and Tools of Steganography. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *10*(6), 2049–2053. https://doi.org/10.22214/ijraset.2022.44279

Darwis, D., & Pasaribu, A. F. O. (2020). Komparasi Metode DWT Dan SVD Untuk Mengukur Kualitas Citra Steganografi. *Jurnal Ilmiah NERO*, *5*(2), 100–108. https://doi.org/10.21107/nero.v5i2.175

Dihni, V. A. (2022). *Peretasan, Bentuk Serangan Digital Paling Banyak Terjadi di Indonesia pada 2021*. https://databoks.katadata.co.id/datapublish/2022/04/07/peretasan-bentuk-serangan-digital-paling-banyak-terjadi-di-indonesia-pada-2021

Guptaa, L. K., Singhb, A., Yadavc, V. K., & Srivastava, A. (2020). Performance Analysis of Open Puff Steganography Tool Using Various Image Formats. *International Conference on Innovative Advancement in Engineering and Technology (IAET-2020)*, 1–7. https://doi.org/10.2139/ssrn.3550941

Hidayat, M. A., Sihombing, M., & Sihombing, A. (2022). Teknik Algoritma Elgamal Dan Steganografi First of File (FOF) Untuk Penyisipan Pesan Dalam Citra. *Jurnal Teknik Informatika Kaputama (JTIK)*, *6*(1). https://doi.org/10.21063/jtif.2020.v8.1.25- 31

Islam, M. A., Riad, Md. A.-A. K., & Pias, T. S. (2020). Performance Analysis of Steganography Tools. *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, 428–433. https://doi.org/10.1109/ICAICT51780.2020.9333473

Jani Anbarasi, L., Prassanna, J., MD, A. Q., Christy Jackson, J., Manikandan, R., Rahim, R., & Suseendran, G. (2020). Visual secret sharing: A review. *Journal of Critical Reviews*, *7*(9), 1212–1216.

Kamil, P. H., Masruroh, S. U., Hakiem, N., Simangunsong, F., & Bidari, A. S. (2018). Robustness Analysis of a Steganography File Against a Media Sharing Process in Instant Messaging Applications. *ICONQUHAS 2018*, 1–13. https://doi.org/10.4108/eai.2-10-2018.2295574

Kasapbaşı, M. C. (2019). A New Chaotic Image Steganography Technique Based on Huffman Compression of Turkish Texts and Fractal Encryption with Post-Quantum Security. *IEEE Access*, *7*, 1–16. https://doi.org/10.1109/ACCESS.2019.2946807

Kumar, A., Jamnadas, H., Sharma, V., Muyeen, S. M., & Ali, A. B. M. S. (2023). A Review of Image Steganography Tools. *International Journal for Computers & Their Applications*, *30*(1), 75–87.

Laksiati, D. (2021). Implementasi Steganografi Image Processing Dan Enkripsi AES Menggunakan Openstego. *Jurnal Akbar Juara Yayasan Akrab Pekanbaru*, *6*(1), 30–40. https://doi.org/10.58487/akrabjuara.v6i1.1391

Malese, L. P. (2021). Penyembunyian Pesan Rahasia Pada Citra Digital dengan Teknik Steganografi Menggunakan Metode Least Significant Bit (LSB). *Jurnal Ilmiah Wahana Pendidikan*, *7*(5), 343–354. https://doi.org/10.5281/zenodo.5563416

Mulyono, I. U. W., Kusumawati, Y., & Ningrum, N. K. (2023). Analisa Visual Citra Hasil Kombinasi Steganografi dan Kriptografi Berbasis Least Significant Bit Dalam Cipher. *Jurnal Masyarakat Informatik*, *14*(1), 16–18. https://doi.org/10.14710/jmasif.14.1.51484

Nurhasanah, Rafi, M. Y., & Rosyani, P. (2023). Keamanan Data Menggunakan Teknik Steganografi Dengan Metode End of File (EOF). *Jaringan Sistem Informasi Robotik (JSR)*, *7*(2), 232–240. https://doi.org/10.58486/jsr.v7i2.300

Nurul, S., Anggrainy, S., & Aprelyani, S. (2022). Faktor-Faktor Yang Mempengaruhi Keamanan Sistem Informasi: Keamanan Informasi, Teknologi Informasi Dan Network (Literature Review SIM). *JEMSI: Jurnal Ekonomi Manajemen Sistem Informasi*, *3*(5), 564–573. https://doi.org/10.31933/jemsi.v3i5

Oladeji, F. A., Awe, O., Aro, T. O., Afolorunso, A. A., Ibor, A., & Uwadia, C. O. (2020). Comparative Analysis of Images Based on Least Significant Bit (LSB) Steganography. *FUW Trends in Science & Technology Journal*, *5*(3), 740–744. http://ftstjournal.com/uploads/docs/53%20Article%2019.pdf

Permana, A. A., & Amna, H. (2022). Implementasi Steganografi File Citra Digital Menggunakan Metode Least Significant Bit. *JT: Jurnal Teknik*, *11*(1), 62–72. https://doi.org/10.31000/jt.v11i1.6161

Rahman, S., Masood, F., Khan, W. U., Ullah, N., Khan, F. Q., Tsaramirsis, G., Jan, S., & Ashraf, M. (2020). A Novel Approach of Image Steganography for Secure Communication Based on LSB Substitution Technique. *Computers Materials & Continua*, *64*(1), 31–61. https://doi.org/10.32604/cmc.2020.09186

Ramadhan, K. R., & Wirawan. (2021). Teknik Penyembunyian Data yang Reversible pada Citra JPEG Terenkripsi. *JURNAL TEKNIK ITS*, *10*(2), 277–283. https://doi.org/10.12962/j23373539.v10i2.68560

Rohayah, S. (2022). Implementasi Teknik Steganography Pada File Gambar Dan Audio Dengan Menggunakan Metode LSB. *OKTAL: Jurnal Ilmu Komputer Dan Science*, *2*(2), 496–503. https://journal.mediapublikasi.id/index.php/oktal/article/view/1073/948

Saleh, A., Harahap, M., & Indra, E. (2020). Kombinasi Jaringan Learning Vector Quantization Dan Normalized Cross Correlation Pada Pengenalan Wajah. *JUSIKOM PRIMA (Junal Sistem Informasi Ilmu Komputer Prima)*, *3*(2). https://doi.org/10.34012/jusikom.v3i2.851

Sara, U., Akter, M., & Uddin, M. S. (2019). Image Quality Assessment through FSIM, SSIM, MSE and PSNR — A Comparative Study. *Journal of Computer and Communications*, *7*(3), 8–18. https://doi.org/10.4236/jcc.2019.73002

Siaulhak, & Kasma, S. (2023). Sistem Pengiriman FileMenggunakan Steganografi Pengolahan Citra Digital Berbasis Matriks Laboratory. *BANDWIDTH: Journal of Informatics and Computer Engineering*, *1*(2). https://doi.org/10.53769/bandwidth.v1i2.522

Singh, N. (2017). Survey Paper on Steganography. *International Refereed Journal of Engineering and Science (IRJES)*, *6*(1), 68–71. http://www.irjes.com/Papers/vol6-issue1/K616871.pdf

Sloan, T., & Castro, J. H. (2018). Dismantling OpenPuff PDF steganography. *Sciencedirect*, *25*, 90–96. https://doi.org/10.1016/j.diin.2018.03.003

Sumijan, Purnama, P. A. W., & Arlis, S. (2019). PENINGKATAN KUALITAS CITRA CT-SCAN DENGAN PENGGABUNGAN METODE FILTER GAUSSIAN DAN FILTER MEDIAN. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *6*(6), 591–600. https://doi.org/10.25126/jtiik.20196870

Wang, Y., Zhao, X., & Cao, Y. (2020). Detecting the fingerprint of video data hiding tool OpenPuff. *Forensic Science International: Reports*, 1–6. https://doi.org/10.1016/j.fsir.2020.100088