Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 2, April 2023
DOI : https://doi.org/10.33395/sinkron.v8i2.12266
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Analyzing Image Malware with OSINTs after Steganography using Symmetric Key Algorithm

Anni Karimatul Fauziyyah[1)*], Ronald Adrian[2)], Sahirul Alam[3)]
[1,2,3)]Program Studi Sarjana Terapan Teknologi Rekayasa Internet, Departemen Teknik Elektro dan Informatika, Sekolah Vokasi Universitas Gadjah Mada
[1)]anni.karimatul.f@ugm.ac.id , [2)]ronald.adr@ugm.ac.id, [3)]sahirul.alam@ugm.ac.id

Submitted : Mar 17, 2023 | **Accepted** : Mar 29, 2023 | **Published** : Apr 1, 2023

Abstract: Steganography is the practice of hiding a message or information within another file, such as an image (Singh & Singla, 2022). OSINT (Open Source Intelligence) involves using publicly available information for intelligence gathering purposes. In this research, the asymmetric key algorithm will be applied to the steganography method, using 10 images with different sizes and dimensions. Images tested for steganography are in tiff, gif, png, jpg, and bmp format. A combination of steganography and OSINT could involve analyzing and decoding images found on publicly available platforms, such as social media, to uncover hidden messages. On the other hand, steganography within OSINT can also be used to protect sensitive information from prying eyes. Overall, the combination of Symmetric Key Algorithm steganography and OSINT can be a powerful tool for both intelligence gathering and secure communication. Here in this work, malware is developed, and using that malware the victim's machine is exploited. Later, an analysis is done via freely available OSINTs to find out which is the best OSINT that gives the best results. OSINTs have been very helpful in identifying whether the URLs and files are malicious or not. But how binding an image with the malware makes it difficult for OSINTs to identify they are malicious or not is being analyzed in this work. The analysis shows that the best OSINT is VirusTotal which has a greater number of engines that could detect the malware whereas others don't have a variety of engines to detect the malware. Also, when it comes to malware afore binding it with an image is easier to detect whereas for an OSINT it was difficult to identify and detect the malware after binding with an image.

Keywords: ***OSINTs;asymmetric key; Steganography;Malware;Image***

INTRODUCTION

Malware is a type of software designed to harm or exploit any computer system, network, or individual device. It is a broad term that includes various types of malicious software, such as viruses, worms, Trojans, adware, spyware, ransomware, and more. Malware can damage files or steal sensitive information, slow down or crash computers, and disrupt the normal operation of networks. It spreads through download links, email attachments, social engineering, and other methods (Murali, Ravi, & Agarwal, 2020). The malware Trojan can execute various damaging actions on the victim's computer, such as stealing sensitive data, deleting files, modifying system settings, and installing additional malicious software (Witte, 2020). Image malware Trojans are often spread through phishing emails or fake websites that trick users into downloading or clicking on them. It is recommended to use reliable antivirus software and keep the system up-to-date to prevent such attacks (Liu, Li, Liu, Xiaoling Gao

*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

College of Computer and Data Science, & Liu, 2021) . Steganography is the practice of hiding a message or information within another file, such as an image (Singh & Singla, 2022). OSINT (Open Source Intelligence) involves using publicly available information for intelligence gathering purposes (Bryushinin, Dushkin, & Melshiyan, 2022). In this research, the asymmetric key algorithm will be applied to the steganography method. A combination of steganography and OSINT could involve analyzing and decoding images found on publicly available platforms, such as social media, to uncover hidden messages. For example, a terrorist organization might use steganography to hide instructions or plans within an innocent-looking image posted on a public forum (Bogdanoski, Risteski, & Pejoski, 2013) . On windows malware research (Demetrio, Biggio, Lagorio, Roli, & Armando, 2021), investigate trade-off on two popular static Windows malware detectors, and show that our black-box attacks can bypass them with only a few queries and small payloads, even when they only return the predicted labels, also evaluate whether our attacks transfer to other commercial antivirus solutions, and surprisingly find that they can evade, on average, more than 12 commercial antivirus engines. (Zhang, Zhao, He, & Zhang, 2022) researching about steganography to achieve that in lossy channels, robust steganography has been proposed. In this letter, the ability against JPEG recompression of robust steganography is further improved by introducing a robustness cost function. For OSINT experiment (Gong, Cho, & Lee, 2018) choose four a kind OSINT: Threat Crowd, Virus Total, Open Threat exchange (OTX), and Cymon. Their conducted threat report crawling and collected other related resources from that report repeatedly. At first, popular network resources such as the domain of search engines or government website, IP address of common DNS server, or MD5 hash value of notorious malware are used to collect related threat reports. From these reports, their collected the related network resources and repeat crawling using these resources. From these open threat reports of OSINT, a number of network resources such as IP, domain, MD5 hash value are randomly collected. OSINT analysts could use software tools to detect and extract hidden data from these images, providing valuable intelligence to law enforcement agencies. On the other hand, steganography within OSINT can also be used to protect sensitive information from prying eyes. By hiding messages in images that are publicly available, parties can communicate securely without alerting potential threats. Overall, the combination of steganography and OSINT can be a powerful tool for both intelligence gathering and secure communication. Here in this work, malware is developed, and using that malware the victim's machine is exploited. Later, an analysis is done via freely available OSINTs (Kowta, Bhowmick, Kaur, & Jeyanthi, 2021) to find out which is the best OSINT that gives the best results. OSINTs have been very helpful in identifying whether the URLs and files are malicious or not. If before running those files on the system are checked then it can help many from losing important information. But how binding an image with the malware makes it difficult for OSINTs to identify they are malicious or not is being analyzed in this work.

LITERATURE REVIEW

Symmetric Key Algorithm

Algorithm for steganography was implemented on the research (Andi Marwan Elhanafi, 2022) he AES cryptography application is used as an additional layer on RGB Overlapping Block Based PVD to provide additional security. (Almazaydeh & Sheshadri, 2018) proposed method for Steganography based on a secret key between the sender and the receiver, this method have called it a dynamic symmetric key. Secret key steganography is similar to a symmetric encryption, where the sender chooses a medium and ensures the secret message in it using a secret key. It is supposed that the key used in the process of embedding is known to the receiver, so the receiver can reverse the process and retrieve the message. In absence of the key, an intercepting party cannot retrieve the embedded message. Applying the Least Significant Bit (LSB) algorithm in images means that the least significant bits for some or all the data in the image are replaced with one bit of the secret message. In LSB techniques, each pixel can hide three bits of the secret message. The LSB algorithm is one of the most common techniques to hide secret messages in an image. The main problem with LSB is knowing whether such an image has a secret message inside, making it easy to retrieve it by collecting the least significant bit from the stego-image (K. Sathish Shet, 2016). This research implemented LSB-2

*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 2, April 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i2.12266 p-ISSN : 2541-044X

algorithm combine symmetric encryption makes use of a secret key steganography to improve the security level, can be described as follows:

1. converts the image pixels to binary values using zigzag scanning with size equal to (M×N×8) where M is the number of rows in the original image, N is the number of columns and 8 is the number of bits per pixel,
2. gets the two least significant bit of each pixel value according to the position, where the (LSB) position equal to 0 and the bit before (LSB) position equal 1,
3. in a parallel process, the secret message is converted to a row of binary values with size equal (1×K) where K is the number of bits in the secret message.
4. each bit of the secret message is compared with the two bits of the (LSB), if the bit of the secret message doesn't match the first and the second position of the (LSB), we will the position 0 of the (LSB) to the value of that bit of the secret message and the key will be 0.
5. the size of the data (Secret Message) that can imbed into the image by using this method can be calculated by using the following formula:

$$S2 = (M \times N) - 27 \qquad (1)$$

where S2 is the size of the secret message, M is the number of rows in the image, N is the number of columns in the image, and the number 27 is: the first 7 bits from 1 to 7 are reserved to the Steganography type may be [1, 2, 3, …, 127], for example, when the Steganography type equals 1 means that Steganography process is LSB, when the Steganography type 2 means LSB-2.

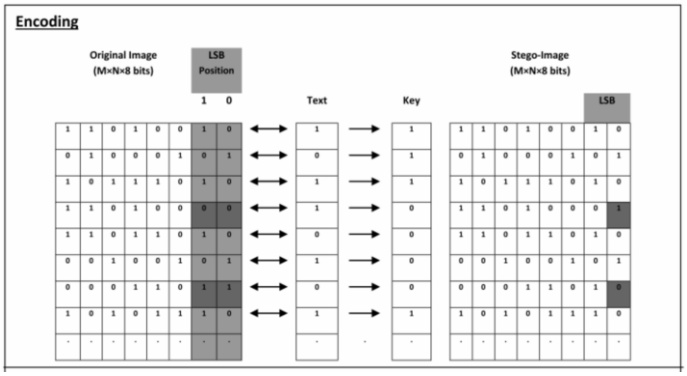


Fig. 1 Encoding technique on dynamic symmetric key method

METHOD

In this research, a brief overview is given on how malware is created and how it is used to exploit the victim's machine. Later the same malware is analyzed on different freely available OSINTs.

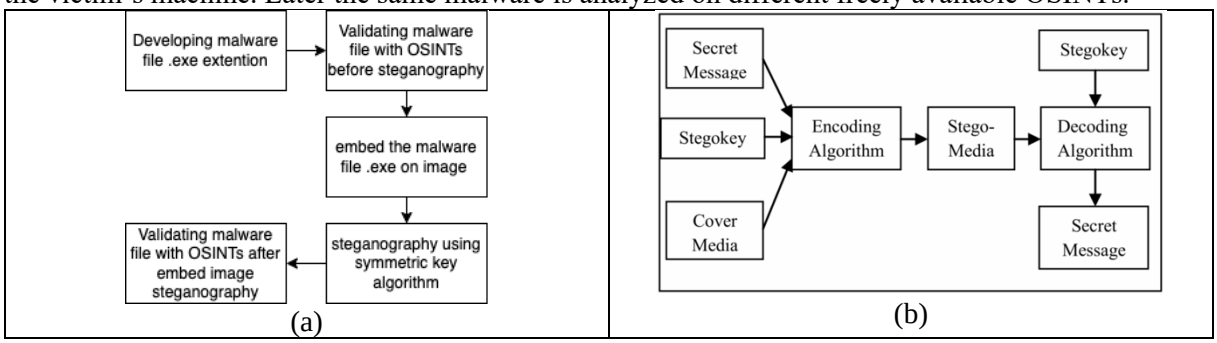


Fig. 2 (a) Research Flowchart (b) Steganography technique

In this section, a brief overview is given inspect a steganography technique step-by-step and implemented to hide .exe malware files in an image file. We also have to encrypt these files to provide more security (Nurhayati, 2022). The algorithm step in this research can be described as follows:

*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 2, April 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i2.12266 p-ISSN : 2541-044X

Algorithm for hiding

1. Select the file(s) to be hidden and the file which we will use as a host to hide our secret files
2. Encrypt secret files
3. Create an archive file(with an .zip extension) and zip the encrypted files
4. Using shell commands, concatenate the image with this zip
5. Obtain the image which contains encrypted hidden files

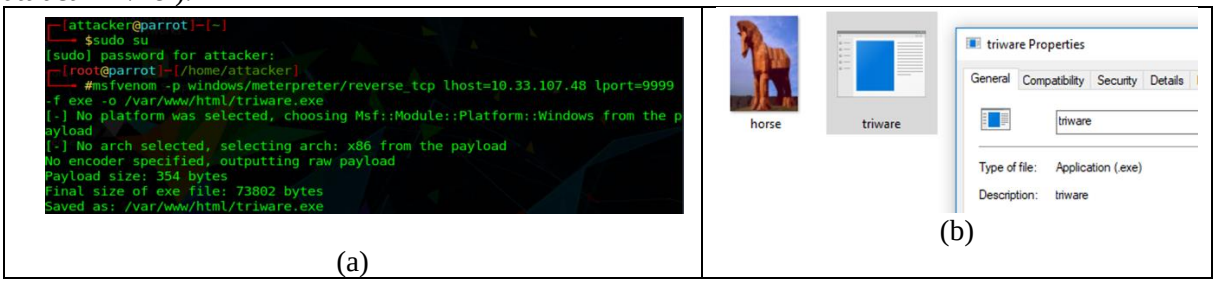
Algorithm for revealing

1. Rename the image file with the appropriate archive extension(.zip etc.)
2. Extract the secret files from zip
3. Decrypt files

RESULT

Developing the Malware

Malware is created in Parrot OS, keeping security and privacy in mind this OS is built for security purpose. Using the command as shown in Fig 3., the malware triware.exe is created. For Steganography it is embedded with the horse image (or any other picture can also be taken of the victim's interest to attract him/her).



(a) (b)

Fig. 3 (a) Develop malware triware.exe (b) Embed malware with the horse image on windows

Steganography using symmetric secret algorithm

After malware is created, the next step implementing steganography. File triware.exe is embedded with the horse image. We implemented os libray for accessing shell, and zip file library for zip operations of Python. Then, since we have to traverse directories to find appropriate files to hide and check the specific conditions for these files. The genKey function takes one argument which is for the key file name, and writes the generated key to the file created with this name while the getKey function simply reads and returns the key. We also need functions for encryption and decryption. The encryptFile function encrypts the files with the given key using Fernet. Then, it writes the file to the current folder. The decryptFile function decrypts the files using the same key. In main function, we specify the name of the key file, name of the output image file, folder to extract and the host image filename. The key file and output(host) image is generated. Then we comparing with the original image, the increase on the size of file, since our host image contains the other file, it's size is increased.

Table 1
Encode File .exe messages

image	size of malware	Afore binding the Malware		After binding the Malware	
		image size	image dimention	output size	image dimention
Horse.jpg	76 KB	44,9 KB	485 x 497	87,1 KB	
Mountain.jpg	80 KB	50 KB	500 x 678	107 KB	
Home.jpg	76 KB	67 KB	800 x 975	128 KB	

*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 2, April 2023
DOI : https://doi.org/10.33395/sinkron.v8i2.12266
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 2
Secret key and md5 code

Image	Secret key	MD5 image	MD5 Output
Horse.jpg	sk2TCAC37d5l_UPWyPNzOqNlRN6Lpw-82TjlhLF6hUs=	fce8552170cced3dd545566309124097	dde1f3c75e10a7b970d538f427291f3a
Mountain.jpg	3epaQ4l917z8yolQpgP3RBEMD38j3oselZvYUcMn7t8=jPGuqUFuwZMzcGBORPjM7m0Afzkstuozy2MlvtbYpbU=	ec2d539554778615d303c5a942cd93b2	23c739701788f94e92fc4c6adf585bd7
Home.jpg	jPGuqUFuwZMzcGBORPjM7m0Afzkstuozy2MlvtbYpbU=	61dade587d1ce6de8f7819a80bcb8d37	e52e3b4c429c2fdbf834931df2b2b06f

After steganography is performed using the asymmetric secret key method, each .exe file embedded in the image file gets a generate key. Comparison of the size of the original image with the image that has been embedded with the malware file has a difference and the key hash of the md5 of each original image compared to the image embedded by the malware has a different key md5. But original image and output embed image there is no difference in size and visual dimensions. This is proven by analyzing the histogram values in the original image and the image that has been steganography

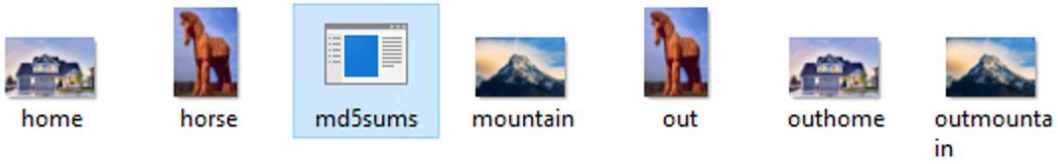


Fig. 4 Original image and output embed image

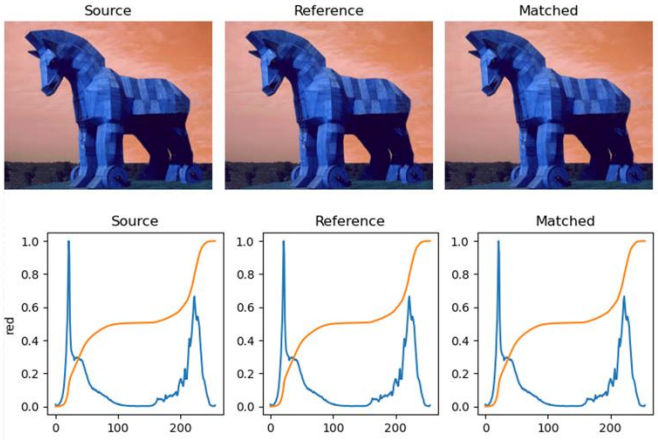


Fig. 5 Analyzed of histogram original image and output embed image

Analysis using OSINT: VirusTotal

As demonstrated in Fig 6 (a) and (b) the results of the triware.exe and horse.jpeg are shown. And it can be seen that there is a huge difference between the results afore & after binding malware with an image. As afore embedding malware with an image 55 out of 68 engines were able to scan the file using VirusTotal as demonstrated in Fig .6 (a) , whereas after embedding malware with an image only 0 out of 58 engines were able to scan the file as shown in Fig 6 (b).

*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 2, April 2023
DOI : https://doi.org/10.33395/sinkron.v8i2.12266
e-ISSN : 2541-2019
p-ISSN : 2541-044X

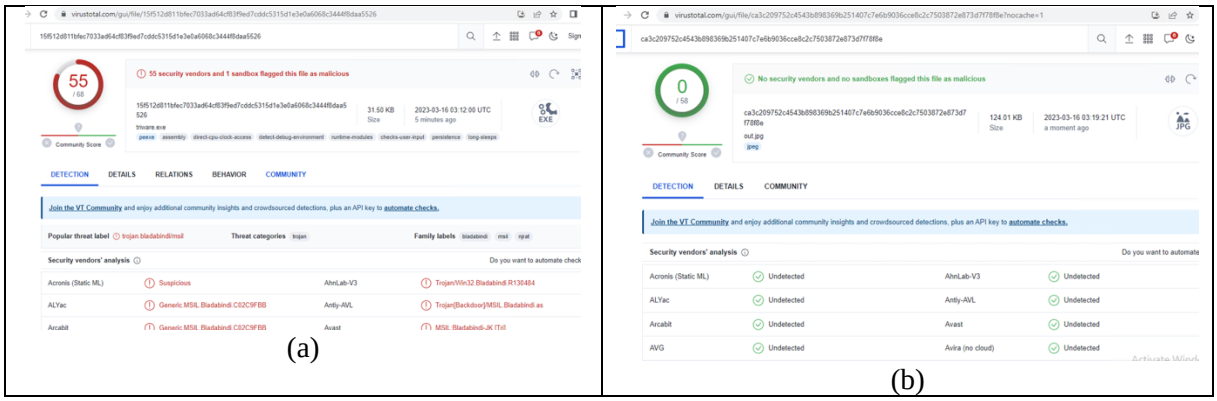


Fig. 6 (a) VirusTotal results afore binding the malware with an image (b) VirusTotal results after binding the malware with an image

In this section, similar to VirusTotal other OSINTs are taken to identify how many engines can detect the malware as one should never rely on one source's results. As the analysis shown in Table 3, after steganography, the malware is not detectable as afore steganography it is. This means steganography is very effective and strong as there is a huge difference in the results. Along with the scanning results, the VirusTotal gives a lot of other information also asMD5, SHA-1, SHA-256, Vhash values, etc. Similarly, other OSINTs also provide the same or more information about the files

Table 3
OSINTs Analysis

OSINTs	Afore binding the Malware	After binding the Malware
VirusTotal	55/68	0/68
OPSWAT	29/40	6/40
Jotti	27/49	9/49
Bitbaan	13/15	0/15
MaLab	11/21	4/20
PolySwarm	11/15	6/11

DISCUSSIONS

The results show how the engines were able to scan afore & after steganography. Similarly, the anti-malware works. They go into the file and check whether the file is malicious or not but if the anti-malware doesn't have any information about the malware's signature then the anti-malware won't be able to detect the malicious content and will tell the user that the file is non-malicious and safe to use and after that the machine gets infected. So, it's very important to have good defenders or anti-malware. Only freely available OSINTs are considered which doesn't have many known engines to detect the malware. In the future, paid tools also can be used for enhancing the results and to provide more information on the malware

CONCLUSION

Malware is a rapidly growing and never-ending concern, but effective and defensive measures can be taken to overcome the damage that they can cause if properly not taken care of. This paper gives a thorough overview of the phishing technique, which is widely used to manipulate victims, and how the malware can exploit the system to gain access and information. In this work, malware is created, and then the same malware is being analyzed on different freely available OSINTs. A light has been thrown on how binding a malware makes it difficult for OSINTs to identify and detect them. The analysis shows

*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

that the best OSINT is VirusTotal which has a greater number of engines that could detect the malware whereas others don't have a variety of engines to detect the malware. Also, when it comes to malware afore binding it with an image is easier to detect whereas for an OSINT it was difficult to identify and detect the malware after binding with an image.

REFERENCES

MMurali, R., Ravi, A., & Agarwal, H. (2020). A Malware Variant Resistant To Traditional Analysis Techniques. 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). Vellore, India.

Liu, Y., Li, J., Liu, B., Xiaoling Gao College of Computer and Data Science, F. U., & Liu, X. (2021). Malware Identification Method Based on Image Analysis. Wuyishan, Fujian, China: IEEE.

Witte, T. N. (2020). Phantom Malware: Conceal Malicious Actions From Malware Detection Techniques by Imitating User Activity. IEEE Access, 8, 164428 - 164452.

Singh, J., & Singla, M. (2022). A Novel Method of high-Capacity Steganography Technique in Double Precision Images. Shillong, India: 2021 International Conference on Computational Performance Evaluation (ComPE).

Bryushinin, A. O., Dushkin, A. V., & Melshiyan, M. A. (2022). Automation of the Information Collection Process by Osint Methods for Penetration Testing During Information Security Audit. Saint Petersburg, Russian Federation: IEEE.

Kowta, A. S., Bhowmick, K., Kaur, J. R., & Jeyanthi, N. (2021). Analysis and Overview of Information Gathering & Tools for Pentesting. India: IEEE.

Demetrio, L., Biggio, B., Lagorio, G., Roli, F., & Armando, A. (2021). Functionality-Preserving Black-Box Optimization of Adversarial Windows Malware. Published in: IEEE Transactions on Information Forensics and Security, 3469 - 3478.

Zhang, J., Zhao, X., He, X., & Zhang, H. (2022). Improving the Robustness of JPEG Steganography With Robustness Cost. IEEE Signal Processing Letters, 164 - 168.

Bogdanoski, M., Risteski, A., & Pejoski, S. (2013). Steganalysis — A way forward against cyber terrorism. 2012 20th Telecommunications Forum (TELFOR). Belgrade, Serbia: IEEE.

Gong, S., Cho, J., & Lee, C. (2018). A Reliability Comparison Method for OSINT Validity Analysis. IEEE Transactions on Industrial Informatics , 5428 - 5435.

Andi Marwan Elhanafi, T. R.-N. (2022). Cryptography Application on RGB Overlapping Block Based PVD Using AES. Sinkron : Jurnal dan Penelitian Teknik Informatika, 7.

Almazaydeh, W. I., & Sheshadri, H. S. (2018). Image Steganography Using a Dynamic Symmetric Key. 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). India: IEEE.

Nurhayati, S. S. (2022). Steganography for Inserting Message on Digital Image Using Least Significant Bit and AES Cryptographic Algorithm. Informatics Engineering Department, Science and Technology Faculty Syarif Hidayatullah State Islamic University (UIN) Jakarta, Indonesia.

*name of corresponding author

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.