International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 6
ISSN: 2088-5334

Steganography on Digital Color Image Using Modulo Function and Pseudo-Random Number Generator

Septia Rani [a,*], Arrie Kurniawardhani [a], Yosa Angela Widya Rendani [a]

[a] *Department of Informatics, Universitas Islam Indonesia, Yogyakarta, Indonesia*
Corresponding author: [*]septia.rani@uii.ac.id

***Abstract*—In this era, many people exchange data digitally either through the internet or other communication channels. However, data sent digitally can be seen by unauthorized people. Therefore, data security is essential. One of the data security techniques is steganography. This study employs steganography based on Pixel Value Modification (PVM) method using the modulo function. This technique will be implemented to insert a text message in digital images in RGB color space. Besides, we use the Pseudo-Random Number Generator with a secret key to enhance the inserted messages' security. To measure the performance of the proposed method, testing is carried out by comparing the stego image to cover image based on three criteria, namely imperceptibility, fidelity, and recovery. PVM method using modulo function successfully hides text that the length is less than 200 characters into 255×255-pixel color images. Imperceptibility testing is done by distributing a questionnaire to six people randomly. The results showed that all respondents answered that each test image's stego image and cover image had no difference. At the same time, the results of the fidelity test show that the MSE value is close to zero, and the PSNR value is above 40 dB. Furthermore, recovery testing will check whether the extracted message is the same as the secret message inserted in the cover image. The results showed that all messages inserted were extracted correctly for recovery criteria if given the correct secret key.**

***Keywords*— Digital image; Modulo; pixel value modification; pseudo-random number generator; steganography.**

Manuscript received 26 Jul. 2020; revised 1 Mar. 2021; accepted 28 Apr. 2021. Date of publication 31 Dec. 2021.

I. INTRODUCTION

The digital era does provide many benefits. One of them is the transfer of information and data that become faster and easier. As a result, digital data is widely spread on communication channels. This allows people who are not entitled to access it can access it either intentionally or not. Looking at the benefits of the digital era, it is important to maintain the security of information that is being exchanged.

There are various methods for securing information; one of them is information hiding. Cryptography is one of the information hiding techniques. Cryptography will convert plain text (readable messages) into ciphertext (unreadable messages) using symmetrical or asymmetrical keys [1]. So, only the person who has the key can see the message. Cryptography also deals with mechanisms for ensuring integrity, techniques for exchanging secret keys, protocols for authenticating users, and more [2]. However, the cryptographic technique raises suspicion because it generates random messages in a normal view. To overcome this problem, the information hiding technique known as steganography is introduced.

Steganography is a technique of hiding secret messages inserted on a media where the results will look normal. There is no suspicion that there is a secret message inserted in the media [3]. The media used to insert the messages can be any type of multimedia file such as text [4], [5], audio [6], [7], and image [8]–[10]. Using text as the media is relatively difficult as compared to the other target media because of the lack of available redundant information in a text file [5], while embedding information into sound files is also generally considered more difficult than images because the human ear is extremely sensitive to perturbations in sound.

Digital images are the most popular media used for steganography. There are two ways of data insertion in a digital image: insertion on spatial domain and insertion on frequency/transform domain. In the frequency/transform domain, the procedure for hiding information in an image is more complex. Transformation is carried out first on a digital image. Then the message can be inserted [11]. Whereas in the spatial domain, messages are directly inserted in pixels. Although the hidden data can be lost with image manipulation,

insertion carried out in the spatial domain will be easier and faster.

The popular methods for inserting data in the spatial domain include the Least Significant Bit (LSB), Pixel Value Differencing (PVD), and Pixel Value Modification (PVM). The LSB method is simple because the secret messages are always inserted in the last bit, but the secret message inserted in the cover image will be easily known. The embedding capacity produced by the PVD method is quite good, but the stego image generated in the color image will be easily detected. On the other hand, the PVD method can produce a good stego image, but the capacity of the inserted secret message cannot be maximized. Another method is PVM. The advantages of PVM are that the secret message is inserted into three different components so that it will be difficult for steganalysis to find the secret messages and know how many bits have been inserted. Besides, hiding messages by PVM is carried out in all RGB color panels so that the capacity of the inserted secret message can be maximized.

Wang proposed applying the PVM method for the first time by adding the modulo function [12]. The modulo function is used to decompose secret messages based on the modulo value used. Wang inserted secret messages in color images. All color layers will be used as an insertion place to enlarge the secret messages' insertion capacity.

In this paper, we design a new steganography method based on Pixel Value Modification (PVM) for digital color images. The proposed method is different from the PVM method [12] because the PVM method is combined with Xorshift Random Number Generator to choose the pixel positions where the secret messages will be inserted. This mechanism will add an additional layer to enhance the security of the messages. This technique is much needed, especially in the medical field, to store patient data on a color medical image. The text will be inserted in the spatial domain using PVM combined with the modulo 3 functions. The main contribution of this paper is to investigate the performance of the proposed method.

II. Materials and Method

A. Steganography

The word steganography comes from Greek, "steganos" which means closed/hidden and "graphein" which means writing [13]. Steganography is a technique for hiding secret messages inserted on a media, such as text, audio, image, and video, where the results will look normal. The media used for hiding messages is called Cover media. The media that has been inserted by secret messages is called Stego media. The process of inserting a secret message into a cover media is called encoding, while the message retrieval process from Stego media is called decoding. Both processes require keys to restrict who can access the secret messages.

B. Pixel Value Modification (PVM)

Pixel Value Modification (PVM) modify intensity value in image pixel. One digit secret message can be inserted into one pixel of the cover image. PVM divides the cover image into three color panels, namely R (red), G (green), and B (blue) [12]. Thus, one color image will have 3 panels or 3 matrices of $M \times N$ pixels according to the length (M) and width (N) of the original image. Each pixel in each panel will be represented by 8 bits. A secret message that will be hidden is inserted into each pixel by sequencing. Insertion of secret messages is done on all color panels. The insertion of the first digit is in the matrix R, the second digit is in the matrix G, and the third digit is in the matrix B. The purpose of inserting pixel digits in the different color panels is to improve security, capacity, and improve image quality on the cover image.

C. Modulo Function

Modulo or mod is an integer operation that the result is the remainder of the division of a number to another number. For example, $9 \, mod \, 2 = 1$, because 9 divided by 2 produces 1 as the remainder. Modulo adds a variety of calculations that are quite complex, making it suitable to use in the encoding and decryption process. Modulo function is used as part of the encoding process to represents hidden messages in n base numbers. The result value of the encoding process will depend on the modulo value used. For example, if modulo 3 is used, secret messages will only be coded with values 0, 1, and 2. Through that encoding process, it is expected that the capacity to insert the secret messages can be greater while maintaining cover media quality.

D. Red Green Blue (RGB) Color Model

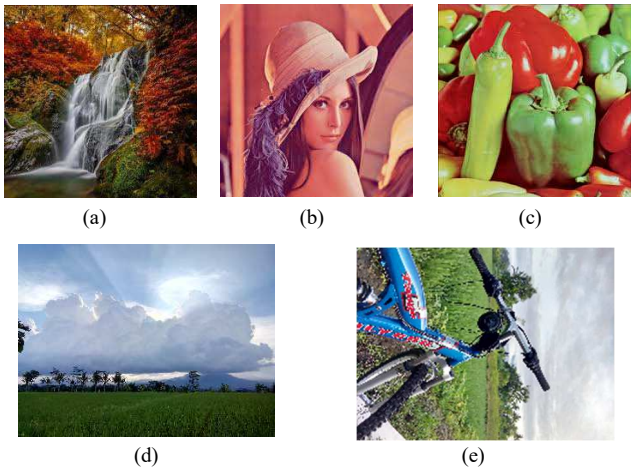
Red Green Blue (RGB) Color Model is a color space consisting of three primary light layers of color R (red), G (green), and B (blue). The RGB coloring system has been adopted in digital appearances such as monitors, cellphones, and televisions. One of the features of this color model is easy to copy or move to another device because many devices apply this color space. Color in a one-pixel image can be represented in 3-bytes or 24 bits. The combination of the three-color components can produce 16 million color variations (2^{24} = 16.777.216).

E. Pseudorandom Number Generator (PRNG)

Pseudo-Random Number Generator (PRNG) is the mechanism used by real-world secure systems to generate values/numbers assumed to be random [14], [15]. PNRG will generate a sequence of numbers that is not truly random but follows a certain algorithm rule. PRNG has a certain cycle length. If the cycle length has fulfilled a certain period, the random number cycle will return to the beginning. Thus, the occurrence of the next random number can be predicted with statistical calculations. Xorshift Random Number Generator (Xorshift RNG) is one of the PRNG algorithms. It passed of the randomness test very well. Xorshift RNG is extremely fast with period $2k - 1$ for $k = 32, 64, 96, 128, 160, 192$ [16][17]. In addition to speed, another advantage is Xorshift RNG uses a smaller space if implemented into a program.

F. Data

Data used in this study are text and color image. Text is used as secret messages which will be inserted into the digital image. The color image is in JPG file format and used as a Cover image. Five Cover images used in this study are shown in Fig. 1. The size of the image in Fig. 1(a)-(c) is 225×225 pixels. These images are obtained from the internet. While the size of the image in Fig. 1(d) and 1(e) is 3264×2448 pixels. These images are taken using a smartphone camera.



(a) (b) (c) (d) (e)

Fig. 1 Cover image. (a) Waterfall, (b) Lena, (c) Peppers, (d) Sky, (e) Bicycle

G. Methodology

This study has two processes: the embedding (encoding) and extracting (decoding) process. The embedding process is the process of inserting the secret message into the Cover image. The substitution process does the embedding process. The extracting process is the process of retrieving the secret message from the Stego image. This study used the symmetry key to encode and decode the message.

The flowchart of the embedding process is shown in Fig. 2. In this process, the inputs given by the user are a Cover image, secret message, and Stego key (number). The secret message entered by the user is a text in TXT file format. Each character of text in the secret message is translated into the ASCII decimal code. Each decimal code is then represented in base-3 numbers. Each decimal code will be represented by 5 bits with the values of each bit 0, 1, or 2. All 5-bits messages are then concatenated to become 1×(5**length of the message*) vector.

Before the secret message is inserted, the user's cover image is separated into 3 panels or matrix of M×N pixels, Red Green Blue. Each matrix is reshaped into a 1×(M*N) vector so that there are three 1×(M*N) vectors. These vectors are concatenated in the order of red vector, green vector, blue vector. The pixel positions where secret messages (in the form of the base-3 number) will be inserted are generated using PRNG based on the Stego key and the length of the message. The intensity value of the cover image in each selected pixel position will be changed.

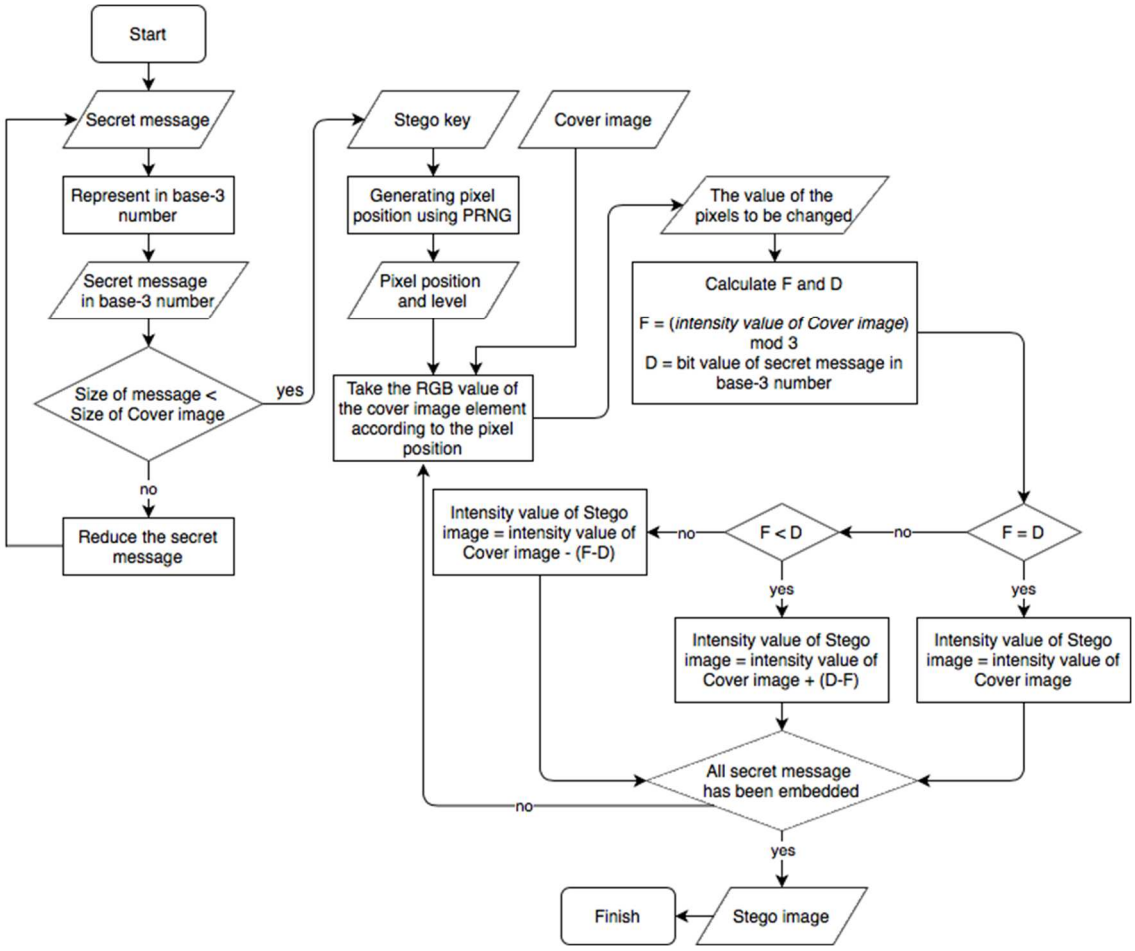


Fig. 2 Embedding process

The intensity value is changed by following several rules chosen based on the secret message bit (base-3 number) that will be inserted. First, the remainder of the intensity value divided by 3 will be calculated (intensity value mod 3). Next, the remainder of the quotient will be called F, and the bit value of a secret message in the form of the base-3 number will be called D. Compare the values of F and D.

If F is equal to D, the intensity value in Stego image is equal to the intensity value in the Cover image at the corresponding pixel position. If F is smaller than D, the intensity value in the Stego image is the intensity value in the cover image at the corresponding pixel position added with the difference value of D and F. If F is greater than D, the intensity value in the Stego image is the intensity value in the cover image at the corresponding pixel position subtracted with the difference between D and F. The result of the embedding process is a Stego image in BMP file format.

The flowchart of the extracting process is shown in Fig. 3. In this process, the inputs given by the user are a Stego image in BMP file format, Stego key (number), and the length of the message. Stego image inputted by the user is separated into 3 panels or matrix of M×N pixels, Red Green Blue. Each matrix is reshaped into a 1×(M*N) vector so that there are three 1×(M*N) vectors. These vectors are concatenated in the order of red vector, green vector, blue vector.

Secret messages in the form of the base-3 number will be retrieved by extracting intensity value at certain pixels. The pixel positions that contain secret messages are found using PRNG based on the Stego key and the length of the message. The intensity value of a certain pixel is divided by 3. The remainder (intensity value mod 3) obtained from that calculation is a bit (base-3 number) of secret message.

After all bits of the secret message is obtained, each 5-bits in the sequence are grouped. Each 5-bits grouped is converted to decimal number. Then, each decimal number is translated to character based on the ASCII decimal code. The result of the extracting process is secret message text.

H. Evaluation

In steganography, three aspects must be considered: imperceptibility, fidelity, and recovery [18], [19]. Imperceptibility ensures that the human senses cannot perceive the existence of hidden messages in container media. Fidelity is the quality of container media that should not be much different after insertion. Recovery is hidden messages in container media should be able to be extracted correctly anytime.

The imperceptibility aspect is measured by comparing the Cover image and Stego image. Imperceptibility testing is done by distributing a questionnaire to six people randomly. In the questionnaire, respondents are asked whether there are differences between two images A and B, in which image A is an image before and after the secret message is inserted, respectively.

To measure the distortion that occurs between the Cover image and Stego image or fidelity, the formula is used. The PSNR value is useful for assessing pixel similarities between the Stego image and Cover image expressed in decibels (dB). The greater the value of PSNR, the more difficult the Stego image can be distinguished from the cover image. Stego image quality is strongly influenced by the size of the hidden message and bit-level used. Before calculating the PSNR value, the Mean Squared Error (MSE) value must be calculated first. MSE can be calculated using (1), while PSNR can be calculated using (2) [20].

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|S(i,j) - C(i,j)\|^2 \quad (1)$$

MSE = the value of the mean square error of the image
m = image length (in pixels)
n = image width (in pixels)
(i, j) = coordinates of each pixel
S = stego image
C = cover image

$$PNSR = 10.\log_{10}\left(\frac{Cmax^2}{MSE}\right) \quad (2)$$

PSNR = image PSNR value (in dB)
Cmax = the highest intensity value in the image
MSE = The value of the mean square error of the image

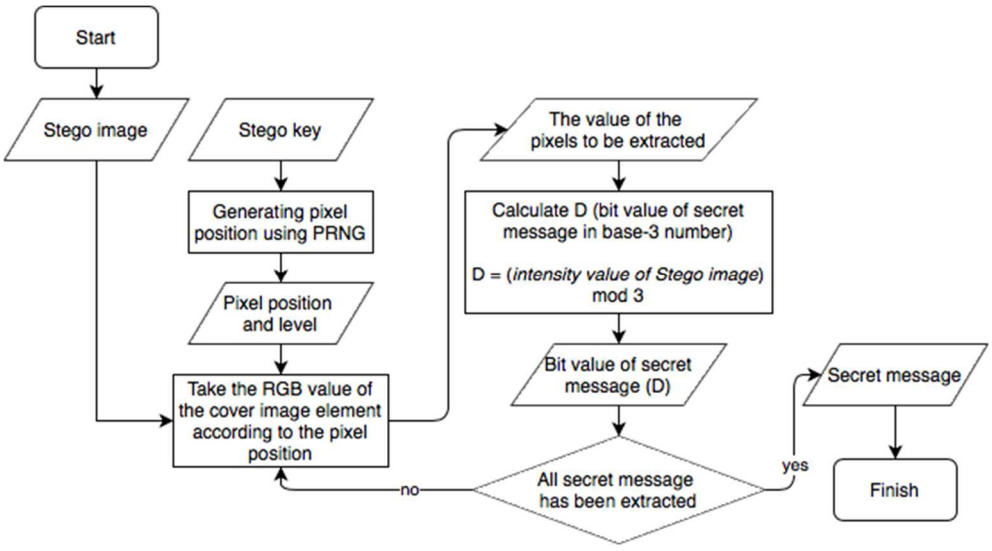


Fig. 3 Extracting process

The relationship between PSNR and MSE is inversely proportional. The smaller the MSE value, the smaller the error value. The greater the value of PSNR, the better the quality of the stego image produced.

Recovery testing is done by observing the extraction results. Recovery testing will check whether the extracted message is the same as the secret message inserted in the cover image.

III. Results and Discussion

The hardware used for implementation and testing is a laptop with Intel Atom @ 2.27 GHz specifications, 4 GB RAM, and a 500 GB HDD. Meanwhile, the software used in the implementation and testing is Windows 10 64 bit, Matlab R2013a, and Matlab R2013a GUI builder. The value is used for a1, a2, and a3 in the Xorshift RNG algorithm is 21, 35, and 4, respectively.

Two processes (embedding and extracting process) have been implemented. Users will be asked to enter some data in the embedding process, namely a secret message, Stego key, and Cover image. The secret message is entered into the system by uploading a .txt file that contains a secret message. The Stego key cannot be zero and consists of a maximum of four digits. When the embedding process has been completed, the system will provide output, namely the Stego image and the length of the secret message. The length of the secret message and Stego key must be remembered in order to be able to do the extracting process. Stego image is in BMP file format. Users will be asked to enter some data in the extraction process, namely the length of the secret message, Stego key, and Stego image. Length of secret message and Stego key is filled in according to the information at the embedding process. When the extraction process has been completed, the system will provide output, namely the secret message. But if the input that is filled in does not match, the extracted message will be incorrect.

Testing is done in two stages. In the first stage, testing is carried out with a length of secret message less than 200 characters. In the second stage, testing is carried out with the length of the secret message more than or equal to 200 characters. In each stage, three aspects: imperceptibility, fidelity, and recovery will be evaluated. Because the Stego image is in BMP format, to be balanced, the cover image will be changed to BMP format also during testing.

To evaluate the imperceptibility aspect, testing is done by distributing a questionnaire to six respondents and comparing the size of the image. The distributed questionnaire asks the respondent whether there are differences between the two images (Stego image and Cover image). All respondents answered that the Stego image and Cover image of each test image had no difference. When comparing the image size before and after the insertion of the secret message, the results show that there are no differences in the image size before and after the insertion. To evaluate the fidelity aspect, testing is done by calculating the PSNR. Test results that calculated PSNR are shown in Table I.

TABLE I
Test Results for MSE and PSNR Values

File	MSE	PSNR
Waterfall	0.00001	98.28775
Lena	0.00006	89.62488
Peppers	0.00442	70.58482
Sky	0.00308	72.15605
Bicycle	0.00193	71.55877
Mean	**0.0019**	**80.442**

The minimum limit of the PSNR value is 40 dB. If the PSNR value is above 40 dB, the Stego image is similar to the Cover image. Likewise, for the MSE value, if the MSE value approaches zero, the Stego image cannot be distinguished to Cover image. Based on Table I, the MSE value is close to zero and the PSNR value is above 40 dB, so it can be said that the result of inserting a secret message is successful. Changes that occur in the Cover image because of the insertion of a secret message cannot be detected by the human senses.

To evaluate the recovery aspect, testing is done by comparing the secret message before it is inserted into the Cover image with a secret message extracted from Stego image. Test results that compare the secret message are shown in Table II.

TABLE II
Test Results for Recovery Aspect

Cover Image	Message Type	Message
Waterfall	Inserted message	Base layer haruslah berwarna terang agar tidak menyerap kalor. Rentang frekuensinya: 40-50
	Extracted message	Base layer haruslah berwarna terang agar tidak menyerap kalor. Rentang frekuensinya: 40-50
Lena	Inserted message	If you want to go fast go alone! If you want to go far Go with me :)
	Extracted message	If you want to go fast go alone! If you want to go far Go with me :)
Peppers	Inserted message	If you want to go fast go alone! If you want to go far Go with me :)
	Extracted message	If you want to go fast go alone! If you want to go far Go with me :)
Sky	Inserted message	Pengukuhan padukuhan Sri Bedugul dihadiri oleh 162 orang berdasi. Mereka membawa beberapa buah tangan yang dipersembahkan oleh Sri Baginda. Arahnya adalah #@9
	Extracted message	Pengukuhan padukuhan Sri Bedugul dihadiri oleh 162 orang berdasi. Mereka membawa beberapa buah tangan yang dipersembahkan oleh Sri Baginda. Arahnya adalah #@9
Bicycle	Inserted message	love the way it is
	Extracted message	love the way it is

It shows that five messages inserted were extracted correctly if given the correct secret key. The wrong secret key will result in an incorrect message as it will pick inaccurate pixel positions in the Stego image.

TABLE III
The Average of Execution Time

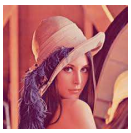
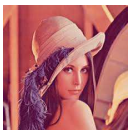
Image	Size	Execution time (s)	
		Embed	Extract
Waterfall	13,1 KB	30.62	58.12
Lena	8,05 KB	22.17	11.45
Peppers	12,1 KB	22.61	16.77
Sky	793 KB	69.64	11.72
Bicycle	1,2 MB	32.70	12.27
	Mean	**35.55**	**22.07**

The performance of the algorithm is tested by calculating the execution time. Table III shows the average execution time. The execution time of the embedding process ranges from 20 to 70 seconds. The average execution time is 35.55

seconds. The execution time depends on the size of the image and the length of the inserted message. Then for the extracting process, the average time needed for execution is 22.07 seconds.

In theory, if the size of the Cover image is 225×225 pixels, the secret message that can be inserted into the Cover image is (225 × 225 × 3)/5 = 30375 characters. But the experiment gave poor results when the length of the secret message inserted was more than or equal to 200 characters. Table IV shows the Stego Image results that have been inserted a secret message with a length of more than or equal to 200 characters. The more messages inserted, the worse the damage that occurs in the Stego image. There is a noticeable change in the Stego image that the human senses can detect. This can evoke suspicion. For this scenario, although the Stego image can be distinguished from the Cover image, we found that the message was still successfully extracted correctly in the recovery test. This may happen because of the use of PRNG, which has a certain cycle length. A further investigation is needed to find why this phenomenon occurred, which currently cannot be covered in this paper.

TABLE IV
MESSAGE INSERTION RESULTS

Cover Image	Length of Char	Stego Image	MSE	PSNR
	200		0.009	67.320
	1.000		0.028	62.583
	6.065		0.048	60.247
	30.375		0.239	53.259

IV. CONCLUSION

This study tries to hide a text message into a color image using the steganography technique. The method used is Pixel Value Modification (PVM) which is combined with the modulo function and Pseudo-Random Number Generator. PVM successfully hides a secret message with a length of fewer than 200 characters in a 255×255 pixel color image. PVM can meet the imperceptibility, fidelity, and recovery criteria. The results of the imperceptibility test indicate that the Stego image cannot be distinguished from the Cover image. While the results of the fidelity test show that MSE value is close to zero and the PSNR value is above 40 dB. Furthermore, the recovery results are indicated by the inserted secret message that can be extracted correctly if given the correct secret key. However, when testing is done by inserting a secret message that the length is more than or equal to 200 characters in a 255×255 pixel color image, this method's imperceptibility component cannot be fulfilled.

REFERENCES

[1] O. R. Shahin, A. Ben Aissa, Y. Fouad, H. Al-Mahdi, and M. Alsmarah, "A New Method of Data Encryption based on One to One Functions," vol. 10, no. 3, pp. 1169–1175, 2020.

[2] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. 2014.

[3] A. S. Ansari, M. S. Mohammadi, and M. T. Parvez, "A multiple-format steganography algorithm for color images," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2991130.

[4] J. R. Jayapandiyan, C. Kavitha, and K. Sakthivel, "Enhanced Least Significant Bit Replacement Algorithm in Spatial Domain of Steganography Using Character Sequence Optimization," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3009234.

[5] K. Wang and Q. Gao, "A Coverless Plain Text Steganography Based on Character Features," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2929123.

[6] A. A. Alsabhany, F. Ridzuan, and A. H. Azni, "The Adaptive Multi-Level Phase Coding Method in Audio Steganography," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2940640.

[7] X. Yi, K. Yang, X. Zhao, Y. Wang, and H. Yu, "Ahcm: Adaptive huffman code mapping for audio steganography based on psychoacoustic model," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 8, 2019, doi: 10.1109/TIFS.2019.2895200.

[8] X. Duan *et al.*, "High-Capacity Image Steganography Based on Improved FC-DenseNet," *IEEE Access*, vol. 8, 2020, doi: 10.1109/access.2020.3024193.

[9] X. Zhang, F. Peng, and M. Long, "Robust Coverless Image Steganography Based on DCT and LDA Topic Classification," *IEEE Trans. Multimed.*, vol. 20, no. 12, 2018, doi: 10.1109/TMM.2018.2838334.

[10] O. F. AbdelWahab, A. I. Hussein, H. F. A. Hamed, H. M. Kelash, A. A. M. Khalaf, and H. M. Ali, "Hiding data in images using steganography techniques with compression algorithms," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 3, pp. 1168–1175, 2019, doi: 10.12928/TELKOMNIKA.V17I3.12230.

[11] Z. Yahya, M. Hassan, S. Younis, and M. Shafique, "Probabilistic Analysis of Targeted Attacks Using Transform-Domain Adversarial Examples," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2974525.

[12] C. M. Wang, N. I. Wu, C. S. Tsai, and M. S. Hwang, "A high quality steganographic method with pixel-value differencing and modulus function," *J. Syst. Softw.*, 2008, doi: 10.1016/j.jss.2007.01.049.

[13] M. Asikuzzaman and M. R. Pickering, "An Overview of Digital Video Watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*. 2018, doi: 10.1109/TCSVT.2017.2712162.

[14] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Cryptanalytic attacks on pseudorandom number generators," 1998, doi: 10.1007/3-540-69710-1_12.

[15] M. Aljohani, I. Ahmad, M. Basheri, and M. O. Alassafi, "Performance Analysis of Cryptographic Pseudorandom Number Generators," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2907079.

[16] G. Marsaglia, "Xorshift RNGs," *J. Stat. Softw.*, 2003, doi: 10.18637/jss.v008.i14.

[17] S. Deshmukh, K. Doshi, and Y. Borse, "Securing Images Using Layered Morphing," 2018, doi: 10.1109/ICCUBEA.2018.8697888.

[18] R. Munir, "Pengantar Ilmu Kriptografi," *Penerbit Andi*, 2008, doi: 10.1017/CBO9781107415324.004.

[19] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research," *Neurocomputing*, vol. 335, 2019, doi: 10.1016/j.neucom.2018.06.075.

[20] E. Emad, A. Safey, A. Refaat, Z. Osama, E. Sayed, and E. Mohamed, "A secure image steganography algorithm based on least significant bit and integer wavelet transform," *J. Syst. Eng. Electron.*, vol. 29, no. 3, 2018, doi: 10.21629/JSEE.2018.03.21.