ILKOM Jurnal Ilmiah Vol. 13, No. 1, April 2021, pp. 67-77
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Steganographic techniques using modified least significant bit and modification reshape transposition methods

Guntoro Barovih [a,1]; Fadhila Tangguh Admojo [a, 2, *]; Yoda Hersaputra [a, 3]
[a] *STMIK PalComTech, alan Basuki Rahmat No.05, Palembang 30127, Indonesia*
[1] *guntoro@palcomtech.ac.id;* [2] *fadhila.tangguh@palcomtech.ac.id;* [3] *hersaputrayoda@gmail.com*
** Corresponding author*

Article history: Received April 28, 2021; Revised April 28, 2021; Accepted April 30, 2021; Available online April 30, 2021

Abstract

A message is a form of conveying information. Various ways are used to secure the information conveyed in the form of messages either in encrypted form or in the form of applying a password in the message. Messages can also be encrypted and embedded in other media such as images (steganography). This research aimed to insert a message into the form of an image by combining the Modified Least Significant Bit (MLSB) method in encrypting messages and reshape modification technique to determine at which position the message encryption will be embedded in the image. Tests were carried out to obtain the quality of the encryption process using the parameters of Fidelity, mean square error, peak signal to noise ratio, testing on file type, robustness, and comparison of message contents. The results of the tests showed that the files that can be used are files with the image file type in the lossless compression category, the rotation can be done at 90, 180, 270 without destroying the message in it, and changing the pixel in the image file will destroy the message inside.

Keywords: Steganography; Modified Least Significant Bit; Reshape; Rewrapping; Encryption

Introduction

A message is a medium of information sent to another party. Messages can be categorized into 2, general messages and secret messages. Secret messages can be distributed to individuals or can also be distributed to an institution. Secret messages are equipped with a series of encryption treatments to form a message that is not easily read by other irresponsible parties. In today's digitalization era, messages can be distributed in a computerized manner and are very vulnerable to data theft actions. Then the encryption treatment or message encoding must be applied. Many forms of message encoding can be done. Encoding a message can also use the ABE schema and combined with other schemas using the charm library to provide better encryption and decryption capabilities [1]. Apart from using the ABE schema, message encryption can also use Message-Locked Encryption (MLE) where the encryption and decryption keys utilize the message itself. Sample-Extract-Encrypt (SXE) was used to construct the MLE schema [2].

Encryption can also be built using or based on identity-based encryption (IBE) from novel primitives or also known as chameleon encryption. IBE can also be built from One-Time Signature with Encryption (OTSE). OTSE is useful in constructing keys that depend on messages, either a secure public key or a secure private key [3]. The encryption of a message has undergone many updates, one of which is using encryption based on the multi-receiver identity. But this method does not guarantee the safety of the encrypted data. In this research, an anonymous multi-receiver identity-based encryption scheme was applied by adopting a lagrange interpolating polynomial mechanism in the encryption process where the results of this research cannot display the identity of the sender [4].

In addition to encryption using MLE, IBE and OTSE, the encryption or message encryption mechanism also uses the message insertion technique on digital image media. This technique is referred to as steganography. The Modified Least Significant Bit (MLSB) method is one of the methods used for digital image steganography, which is a modification of the Least Significant Bit (LSB) method [4]–[6]. In recent studies, MLSB has been combined with another method (water marking) to hide the message inside images [7]. Another test, the MLSB method is also used in the insertion of messages on digital image media using the BMP file type as the image object [8]. This research uses the MLSB method, where 8-bit ASCII numbers will be converted into 5-bit numbers and an image is inserted using file types of bmp, jpeg, png and tiff.

http://dx.doi.org/10.33096/ilkom.v13i1.848.67-77

This research focused on the modified LSB method and modification of image reshaping. The modification was done to randomize the position of the message inserted. Generally, the image reshaping process converts 2-dimensional images into 1-dimensional vectors, MxN images into 1xN vectors (arranged lengthwise). In this research, modification of image reshape with spiral pattern transposition technique was done by dividing part of the image into several levels or more, where the message starts to be inserted at the outer level to the inner level of the image. In other words, the message insertion pattern is spirally inserted.

Method

A. Digital Image

An image defined as information in visual form. An image has characteristics that text data does not have, namely the image is rich with information. There is a proverb that says "a picture is worth more than a thousand words". It means that of course a picture can provide more information than the information presented in the form of words [9]. An image is a representation, similarity, or imitation of an object. An image as the output of a data recording system can be optical in the form of photos, analog in the form of video signals such as images on a television monitor, or digital in nature which can be directly stored on a storage medium. Digital images are images that can be processed by a computer. Some of the most common digital image formats are BMP, JPEG, GIF, PNG, and others [10].

B. Steganography

Steganography is the art of hiding messages in digital media. Digital steganography uses digital media as a container, for example, images, audio, text and video[11], [12]. **Figure 1** is a simple illustration of the process of storing messages into digital image media and producing a new image with the message in it.

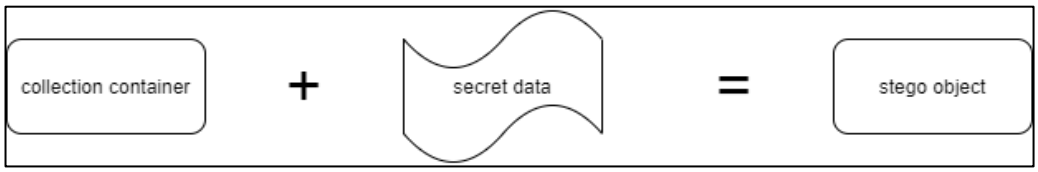

Figure 1. Illustration of Storing Confidential Data into Digital Media[11]

C. Modified Least Significant Bit

Modified Least Significant Bit (MLSB) or modification of the LSB algorithm is used to encode an identity into the original image. MLSB uses some manipulation of the insertion bits before encoding the message [13]. The Modified Least Significant Bit (MLSB) method is a method based on the development of the existing LSB method. In this MLSB method, the numbers used are the 5-bit numbers converted from the 8- bit numbers used in the LSB method. Then the 5-bit number is inserted into the container image using the LSB method [14]. The process involves converting the image value into binary data (8 bits), then 1 bit of the message is inserted from the rightmost row of each image binary value. The insertion of LSB in a 24-bit image can be seen in **Figure 2**.

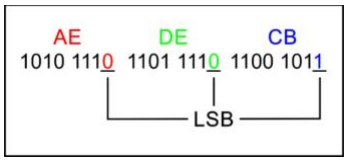

Figure 2. Insertion of LSB at 1 Color Pixel[14]

How the Modified Least Significant Bit (MLSB) algorithm works is as follows:

1. Normalization with the control symbol table.
   Control symbol data that represents message characters to group a data line, as shown in **Table 1**.

Tabel 1. Control Symbol [13]

Hex Representation	Operation
1Bh	Define small letter
1Ch	Define capital letter
1Dh	Define space
1Eh	Define number
1Fh	Define end of text

   a. 61h – 7Ah (Small Letter)
      If the message characters to be inserted represent lowercase letters, first insert the first 5 bits to 1Bh and continue inserting the results of the 8-bit to 5-bit message conversion thereafter.

b. 41h – 5Ah (Capital Letter)
If the message character to be inserted represents a capital letter, first insert the first 5 bits into 1Ch and continue to insert the message conversion results from 8 bits to 5 bits thereafter.
c. 20h (Space)
If the message character to be inserted represents a whitespace character, the whitespace is replaced by the first 5 bits to become 1Dh.
d. 30h – 39h (Number)
If the character of the message to be inserted represents a number, first insert the first 5 bits to 1Eh and continue to insert the message conversion results from 8 bits to 5 bits thereafter.
e. End of Text
If the inserted message has ended, inserted 5 bits at the end becomes 1Fh.

2. The process of changing the ASCII value to 5 bits. For example, the message to be inserted is a text message "STEGO with 05 bits".
   a. The message "STEGO with 05 bits", if converted into ASCII binary requires a memory of 18 x 8 bits = 144 bits.
   b. In this algorithm the message "STEGO with 05 bits" is converted into hexadecimal values: 53h, 54h, 45h, 47h, 4Fh, 20h, 77h, 69h, 74h, 68h, 20h, 30h, 35h, 20h, 62h, 69h, 74h , and 73h.
   c. Normalize the message with the Control System table and XOR it with the smallest tens value in ASCII.
      - Divide the group on the insert message with a white space (20h).
      - The first group, the capital letters sequence is 53h, 54h, 45h, 47h, and 4Fh. Then XOR with the smallest characters in capital letters, namely 41h ('A'), the XOR results are obtained, namely 1Ch, 12h, 15h, 4h, 6h, Eh, and 1Dh.
      - The second group, the lowercase sequence is 77h, 69h, 74h, and 68h. Then XOR with the smallest characters in lowercase letters, namely 61h ('a'), the XOR results are obtained, namely 1Bh, 17h, 8h, 15h, 9h, and 1Dh.
      - The third group, the number sequence is 30h, 35h. Then XOR with the smallest character in numbers, namely 30h ('0'), the XOR results are 1Eh, 0h, 5h, and 1Dh.
      - The fourth group, the lowercase letters are 62h, 69h, 74h, and 73h. Then XOR with the smallest characters in lowercase letters, namely 61h ('a'), the XOR results are obtained, namely 1Bh, 3h, 8h, 15h, and 12h.
   d. Combined the first, second, third, and fourth groups. So that the message to be inserted becomes 1Ch, 12h, 15h, 4h, 6h, Eh, 1Dh, 1Bh, 17h, 8h, 15h, 9h, 1Dh, 1Eh, 0h, 5h, 1Dh, 1Bh, 3h, 8h, 15h, 12h, and 1Fh. Converting this message into ASCII binary requires 23 x 5 bits = 115 bits of memory.
   e. The message will be converted into binary data and pasted using the LSB technique.

3. The process of returning the original message, namely by XORing back the message bytes according to the group of numbers and removing the Control Symbol value.

D. Message Inserting Process Flow

There are several important steps involved in message insertion. The first stage is a reshaping process, then conversion of text messages into 8-bit binary data followed by the process of encoding 8 message bits into 5 bits using the MLSB method until the last stage, namely message insertion using the LSB method. The flow of the message insertion process can be seen in **Figure 3**.

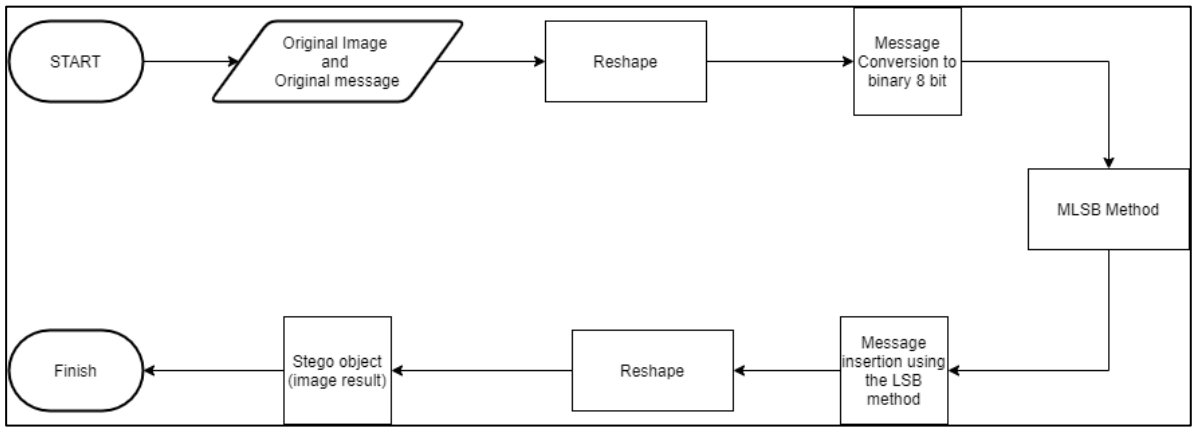

Figure 3. Message Insertion Process Flow with the MLSB Method

E. Reshape Image

The process of reshaping an image aims to transform a 2-dimensional image into a simpler form. Reshaping a two-dimensional image can be seen in **Figure 4**.

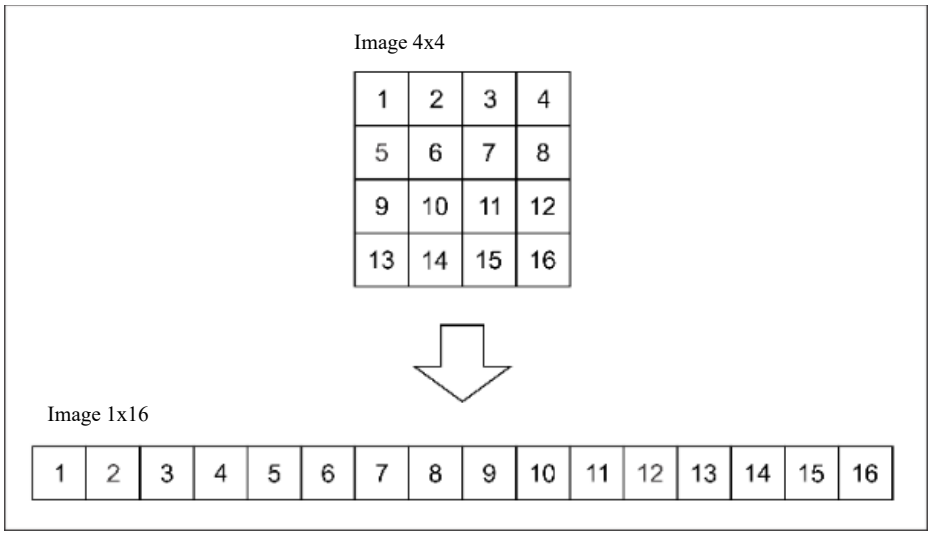

Figure 4. Two-Dimensional Image Reshaping [15]

F. Spiral Pattern Transposition

The use of the spiral reading method is a combination of two reading methods horizontally and vertically, and the use of the spiral reading method will produce a string of image pixels that makes it more complicated to read compared to using the horizontal or vertical reading method alone [16].

The spiral transposition technique was used to determine which part of the text message image to be inserted. The 4x4 image is divided into several levels with the number of levels depending on the dimensions of the image. Then the levels were arranged lengthwise into a 1x16 vector. The message insertion was carried out on the result of the reshape process, namely the 1xN image vector. The image reshaping process can be seen in **Figure 5**.

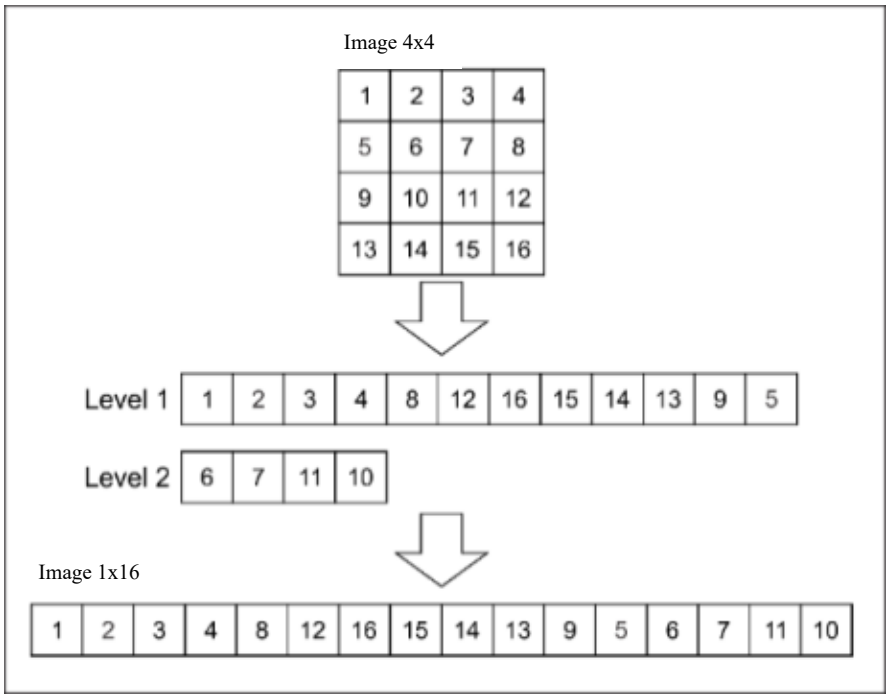

Figure 5. Image Dimensions Before and After Reshaping

After the insertion process was carried out, the re-wrapping process was run. **Figure 6** shows the process of converting the dimensions from 1xN to MxN where the length and width of the original image are known.

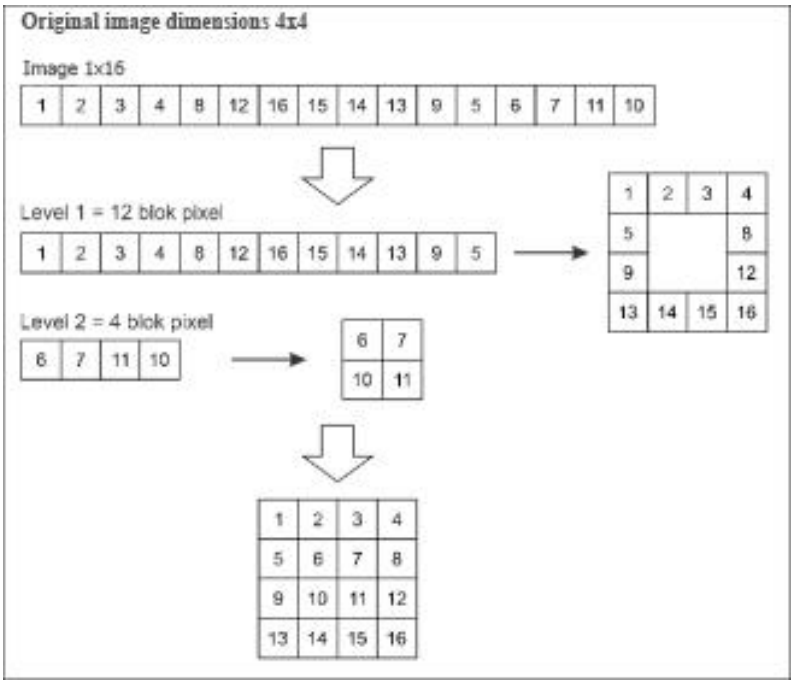

Figure 6. Restoring Image Dimensions

Results and Discussion

A. Types of Research and Data Sources

The type of research used was experimental research, that is, research by recording directly the results of the tests or experiments carried out. The conclusion or the way to collect data was to measure the image quality using the MSE method, PSNR, and test the resilience of the image resulting from external attacks. This research used object data in the form of images. The types of images used were * .bmp, *.jpeg, * .png and * .tiff with a resolution of 512x512 pixels which was obtained from the internet. The maximum capacity of the message depended on the size of the image.

The image sample used consisted of two images with different objects and the same dimensions. The image used can be seen in **Figure 7**. The Lena image sample was used because due to its texture, detail, and shading. Meanwhile, the second image sample was determined randomly which can be found on the internet.

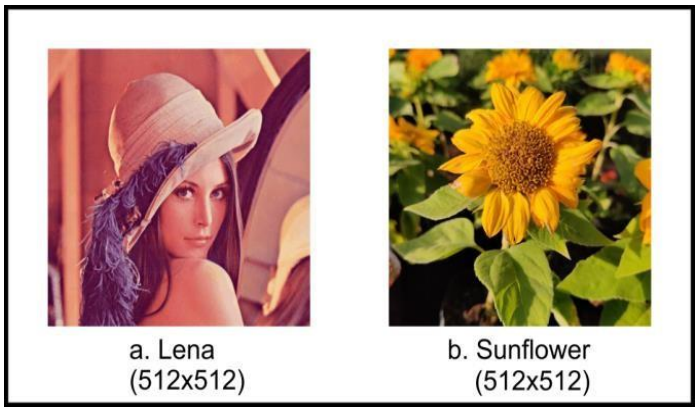

Figure 7. Sample Test Image

B. Testing the Quality Aspects (Fidelity)

1. Mean Square Error (MSE)

MSE is a measure used to assess how well a method performs reconstruction or restoration of an image relative to the original image [17], [18]. The smaller the MSE value shown in equation (1), the image processing results will be closer to the original image.

$$MSE = \frac{1}{MxN}\sum_{x}^{M}\ \sum_{y}^{N}\ [f1(x,y) - f2(x,y)]^2 \quad (1)$$

2. Peak Signal to Noise Ratio (PSNR)

PSNR is a measure of the comparison of the maximum value of the image bit depth measured by the amount of noise that affects the signal shown in equation (2), the noise value is represented by the MSE

value[19]. PNSR was used to improve image quality after inserting a message or modifying the image [20], [21].

$$PSNR = 20Log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \quad (2)$$

Tabel 2. PSNR Value[22]

PSNR Value	Image Quality
60dB	Excellent, no noice
50dB	Good, There is some noise, but image quality is still good
40dB	Reasonable, there are fine grains or snow-like in the image
30dB	Poor picture, there is a lot of noise
20dB	Unusable

C. Robustness Testing

Endurance testing was done to see how reliable the message insertion algorithm was on a digital image, in this case an image [23].

Tabel 3. Image Resilience Level Testing[24]

Basic Operation	Change of Value	Target PNSR Test	Result (dB)
Brightness	-5 to +25	Success/Failed	>= 40dB
Sharpness	-5 to +25	Success/Failed	>= 40dB
Rotation	$90^0/90^0/270^0$	Success/Failed	>= 40dB
Resize	Enlarged 2x	Success/Failed	>= 40dB

The test used 4 24-bit image samples, namely images of bmp, jpeg, png and tiff types with the same dimensions. The image of the attack test results is said to be successful if the message embedded in the resulting image is not damaged after decoding using the MLSB method.

D. Testing Against Comparison of Message Content

Testing was done by comparing the content and size of the message before and after the steganography process. This test can be said to be successful if the original message and the decoded message are exactly the same in terms of both the content and size of the message.

E. Generated Program

A steganography application was produced with the Modified Least Significant Bit (MLSB) method and the spiral pattern reshaping technique in the Python 3.7 programming language.

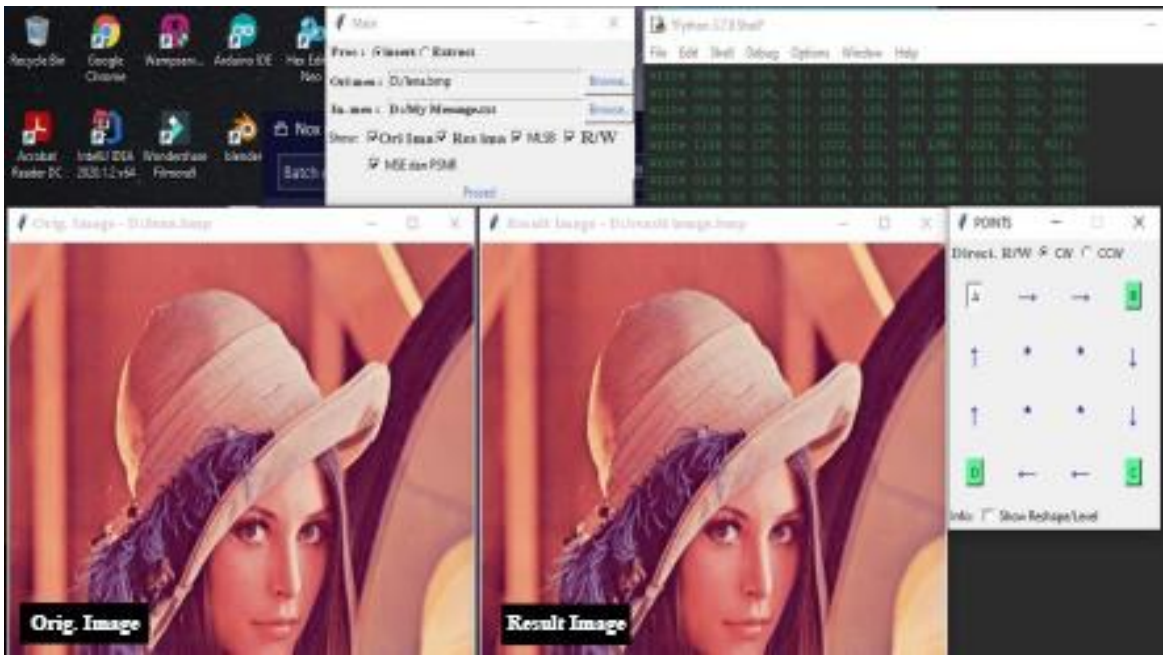

Figure 8. Display of generated program

Figure 8 is a display of software designed to implement the MLSB method and the transposition reshape modification technique in a spiral pattern. The application allows the user to specify an image as a container for text messages in * .txt format and the user can specify the insertion point or point from the starting point, namely A to point D.

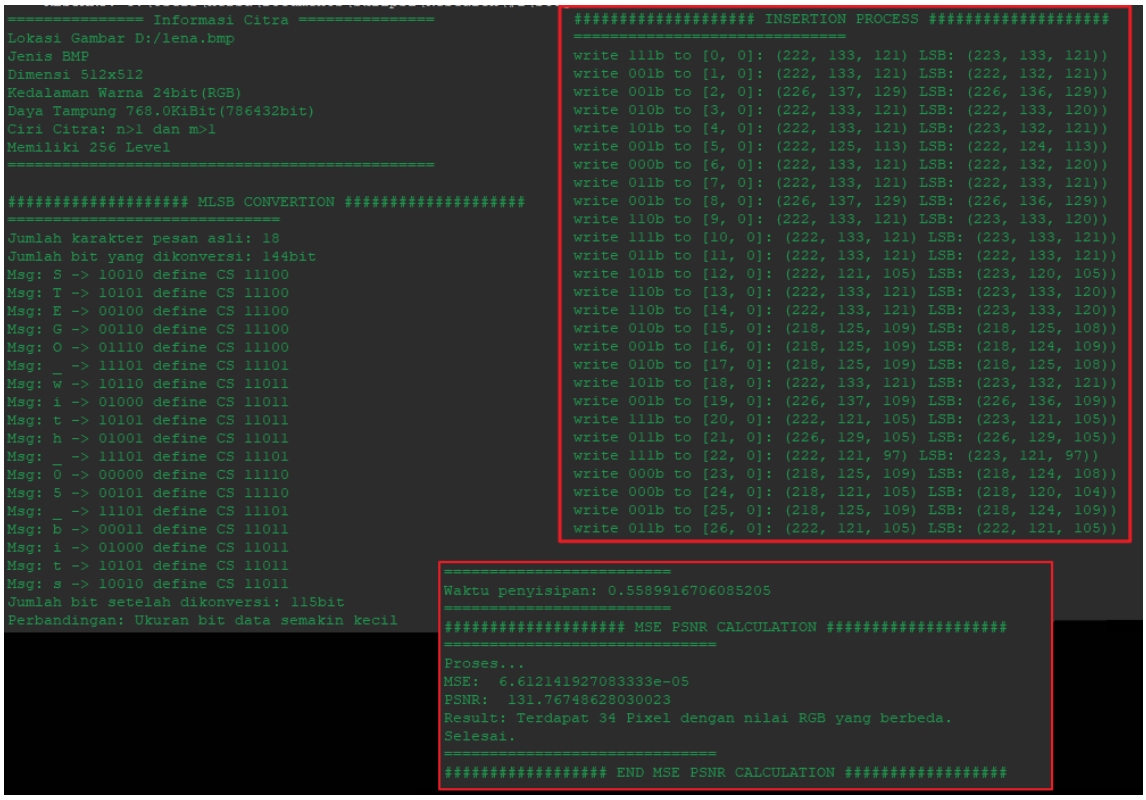

Figure 9. Display of Message Insertion Information

Message insertion and extraction information shown in **Figure 9** will be displayed on the command-line interface with information in the form of MLSB process, read and write MLSB bits on RGB values, insert and extraction processing times and MSE and PSNR calculations.

F. Test Result

The test results on two images with different objects were successfully carried out with a PSNR value of more than 40 dB (image without noise), insertion time of 0.169 - 0.173 seconds and extraction time of 0.169 - 0.175 seconds. The test results on two different images can be seen in **Table 4**.

Tabel 4. Test Results Based on Different Image Objects

Image (512x512)	Message Length	Size File (Bytes)		Running Time (s)		MSE	PSNR (dB)
		Original	Stego	Insertion	Extraction		
Lena.bmp	96	786,486	786,486	0,169	0,171	0,00035	117,19
Sunflower.bmp	96	786,486	786,486	0,173	0,175	0,00035	117,10

Test results were for different file types, including bmp, jpeg, png and tiff. In the jpeg type image the test results failed. This is because the jpeg image is a lossy compression image where data is lost during the compression-decompression process. Specific data damage is on the RGB value where 1 bit of data has been inserted. There will be a change in the LSB value during the compression process and this will affect the MLSB message in it. As a result, the original message and the extracted message are completely incompatible or the data is corrupt. The results of testing different file types can be seen in **Table 5**.

Tabel 5. Test Results Based on Different File Types

Image (512x512)	Message Length	Size File (Bytes)		Running Time (s)		MSE	PSNR (dB)	Test result
		Original	Stego	Insertion	Extraction			
Lena.bmp	96	786,486	786,486	0,432	0,438	0,00035	117,19	Success
Lena.jpeg	96	786,486	786,486	0,175	0,171	0,037	77,45	Failed
Lena.png	96	325,562	326,010	0,169	0,177	0,00035	117,19	Success
Lena.tiff	96	786,572	786,572	0,185	0,179	0,00035	117,19	Success

Image (512x512)	Message Length	Size File (Bytes)		Running Time (s)		MSE	PSNR (dB)	Test result
		Original	Stego	Insertion	Extraction			
Sunflo...bmp	96	786,486	786,486	0,653	0,533	0,00035	117,19	Success
Sunflo...jpeg	96	42,226	42,175	0,188	0,219	0,835	54,88	Failed
Sunflo…png	96	398,624	398,671	0,219	0,196	0,00035	117,19	Success
Sunflo….tiff	96	786,572	786,572	0,222	0,203	0,00035	117,19	Success

The PSNR value obtained was above 40 dB (image without noise) and the test results on images other than the jpeg type were successfully carried out without any damage (corruption) on the message with insertion time of 0.169 - 0.653 seconds and extraction time of 0.171 - 0.533 seconds. The test results for different message lengths were successfully carried out without any damage (corruption) on messages with a PSNR value above 40 dB with an insertion time of 0.199 - 0.251 seconds and an extraction time of 0.174 - 0.688 seconds. The test results based on the message length can be seen in **Table 6**.

Tabel 6. Test Results Based on Message Length

Image (512x512)	Message Length	Running Time (s)		MSE	PSNR (dB)
		Insertion	Extraction		
Lena.bmp	18	0,199	0,198	6,612	131,76
	96	0,196	0,218	0,00035	117,19
	233	0,223	0,174	0,00087	109,33
	1000	0,245	0,688	0,00361	96,96
Sunflo...bmp	18	0,229	0,183	6,993	131,31
	96	0,243	0,197	0,0035	117,19
	233	0,248	0,674	0,00083	109,73
	1000	0,251	0,753	0,03625	96,946

The test results of brightness attack by increasing and decreasing the brightness between -5 to +25 failed. As a result, increasing or decreasing in the brightness can spoil the message inside with a PSNR value of less than 40 dB. The results of testing the image resistance to brightness can be seen in **Table 7**.

Tabel 7. Image Resistance Test Results Against Brightness

Image (512x512)	Brightness		Extraction Time (s)	MSE	PSNR (dB)
	Score	Test results			
Lena Pesan 1000.bmp	-3	Failed	0,202	9,0	29,045
	+5	Failed	0,196	25,0	20,172
Sunflower Pesan 1000.bmp	-3	Failed	0,195	8,676	29,369
	+5	Failed	0,199	24,536	20,337

Tests were also conducted on image resistance to sharpness attacks with a value of 2 (sharp) and 0.05 (blurry). As a result of this attack, it can destroy messages in it with a PSNR value of less than 40 dB. The results of testing image resistance to sharpness can be seen in **Table 8**.

Tabel 8. Image Resistance Test Results Against Sharpness

Image (512x512)	Sharpness		Extraction Time (s)	MSE	PSNR (dB)
	Value (Factor)	Test results			
Lena Pesan 1000.bmp	2	Failed	0,2059	19,435	22,594
	0.05	Failed	0,2149	16,821	23,881
Sunflower Pesan 1000.bmp	2	Failed	0,2139	14,825	25,099
	0.05	Failed	0,1849	15,140	24,770

Testing the image resistance to rotational attacks with degrees of 90, 180 and 270 were successfully carried out without any damage (corruption) to the message with the PSNR value not reaching the target of 40 dB. The PSNR value is below 0 because the pixel value in the image does not change but only changes places due to rotation. However, the spiral reading technique makes it possible to determine the angle where the hidden message is located. Angles of degree other than 90, 180 and 270 cannot be reached and it is difficult for the system to perform and also the image dimensions are different from the dimensions of the resulting image, so MSE and PSNR calculations cannot be performed. The results of testing the image resistance to rotation can be seen in **Table 9**.

Tabel 9. Test Results of Image Resistance to Rotation

Image (512x512)	Rotation		Extraction Time (s)	MSE	PSNR (dB)
	Value (degree)	Test results			
Lena	90/180/270	Success	0,1829	4621	-24,54
	Nilai lain (15)	Failed	0,4008	-	-
Sunflower	90/180/270	Success	0,1664	6630	-25,13
	Nilai lain (45)	Failed	0,5091	-	-

Testing the resilience of the image to the scale by enlarging the scale on the image pixel fails with the message that the extracted results are completely incorrect or the data is corrupt and also the dimensions of the scale image are different from the dimensions of the resulting image, so the MSE and PSNR calculations cannot be done. The results of testing image resistance to resize (scale) can be seen in **Table 10**.

Tabel 10. Image Resistance Test Results Against Resize (Scale)

Image (512x512)	Resize		Extraction Time (s)	MSE	PSNR (dB)
	Value (Factor)	Test results			
Lena.bmp	Enlarged x2	Failed	0,8527	-	-
Sunflower.bmp	Enlarged x2	Failed	0,7787	-	-

Comparison of the message before insertion and the extracted message is identical, so the encryption process and message description are said to be successful. The match between the original message and the extracted message can be seen in **Figure 10**.

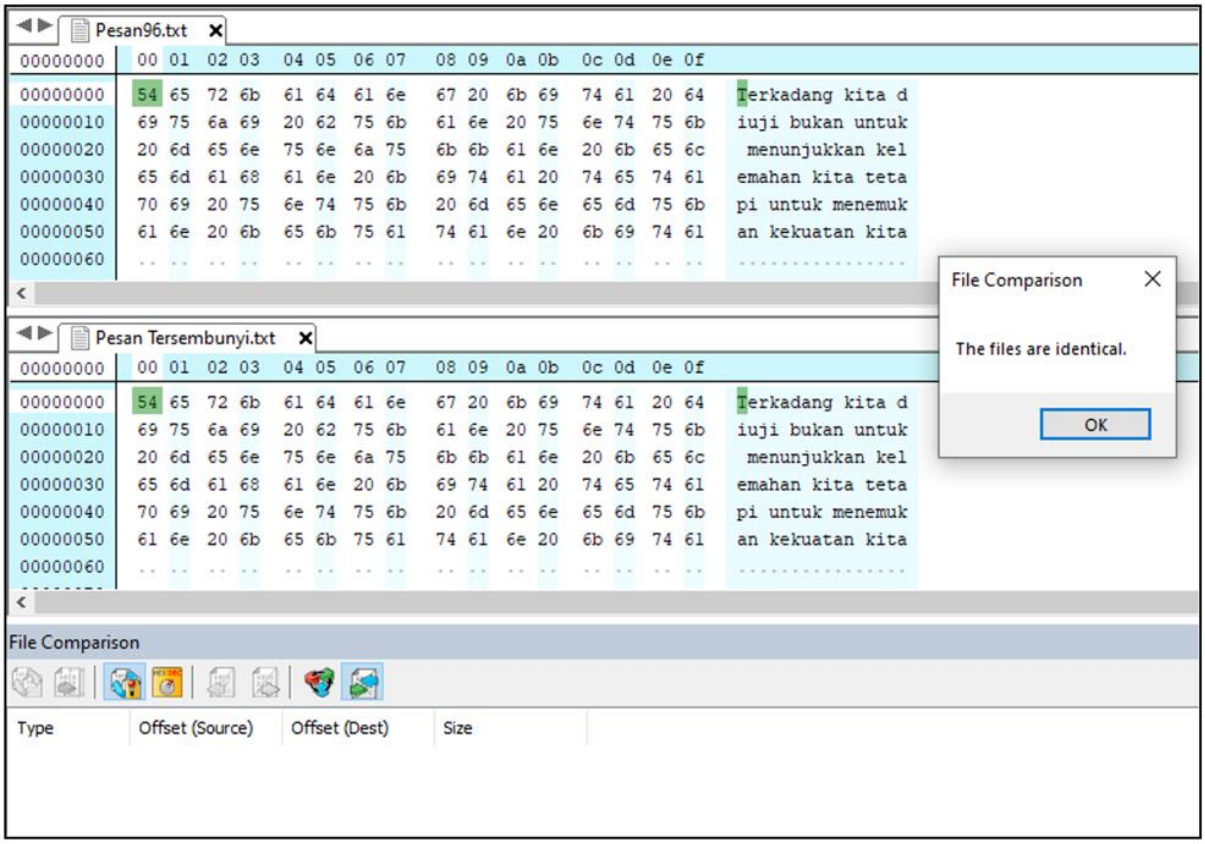

Figure 10. Comparison Results of Original Messages Before and After Extraction on Successful Testing

A comparison of the original message and the extracted message that suffer damage from a failed test. The results of the extraction message are damaged by tagging the messages that are random. The characteristics of the message damage from the extraction can be seen in **Figure 11**.

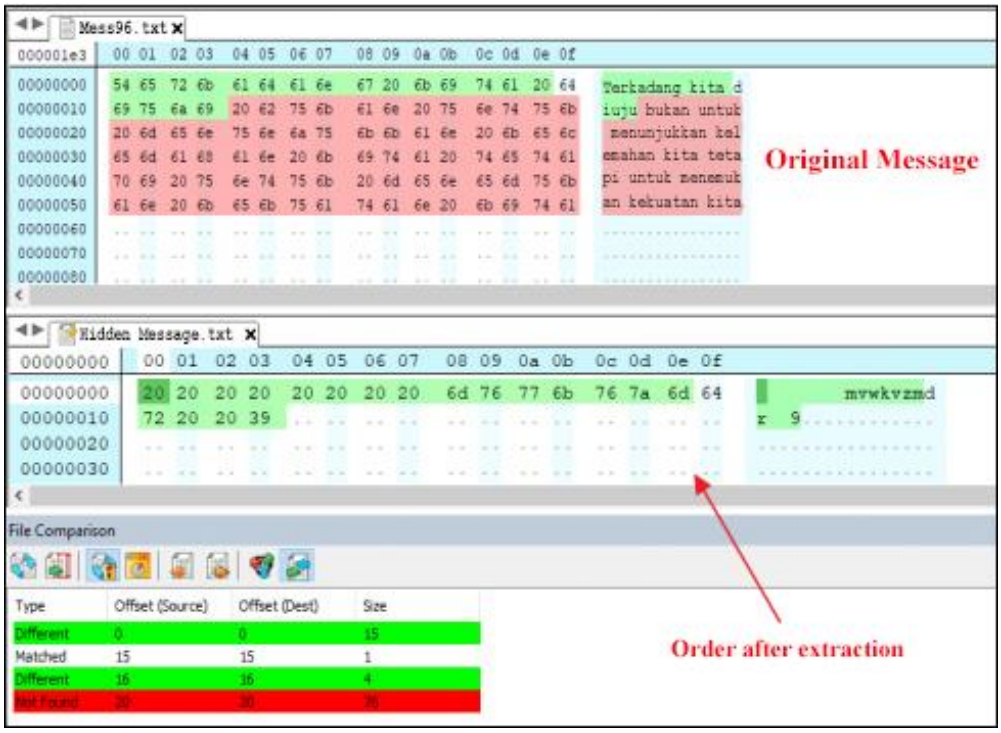

Figure 11. Damage characteristics of the Extraction Result Message on a Failed Test

Conclusion

This research introduced the MLSB method and modification of image reshape with spiral pattern transposition technique. Based on the results obtained from the test, it can be concluded into following points. The starting point of message insertion and extraction can be determined. Points and directions can be keys to steganography. The type of image file that can be used is types of lossless compression images, namely * .bmp, * .png and * .tiff. Images can be rotated with 90, 180, 270 degrees without damaging the message inside. The embedded image is not resistant to changes in the value of the pixels which may result in the message inside being damaged or the data being corrupted. However, the performance of steganography techniques needs to be improved for better message insertion and extraction.

Reference

[1] S. Agrawal and M. Chase, "FAME: Fast Attribute-based Message Encryption," *CCS '17 Proc. 2017 ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 665–682, 2017.

[2] M. Bellare and S. Keelveedhi, "Message-Locked Encryption and Secure Deduplication," *Int. Conf. Theory Appl. Cryptogr. Tech. Adv. Cryptol. – EUROCRYPT 2013*, vol. 32, pp. 296–312, 2013.

[3] N. Döttling, S. Garg, M. Hajiabadi, and D. Masny, "New Constructions of Identity-Based and Key-Dependent Message Secure Encryption Schemes," *IACR Int. Work. Public Key Cryptogr.*, vol. 1, pp. 3–31, 2018.

[4] W. . Luo, F. . Huang, and J. Huang, "Edge Adaptive Image Steganography Based on LSB Matching Revisited," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 2, pp. 201–214, 2010.

[5] C. Irawan, D. R. I. M. ; Setiadi, C. A. ; Sari, and E. H. ; Rachmawanto, "Hiding and securing message on edge areas of image using LSB steganography and OTP encryption," in *1st International Conference on Informatics and Computational Sciences (ICICoS)*, 2017, pp. 1–6.

[6] S. M. M. Karim, M. S. . Rahman, and M. I. . Hossain, "A new approach for LSB based image steganography using secret key," in *14th International Conference on Computer and Information Technology (ICCIT 2011)*, 2011, pp. 286–291.

[7] N. . Voloshina, T. Minaeva, and S. Bezzateev, "MLSB optimal effective weighted container construction for WF5 embedding algorithm," in *10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2018, pp. 1–6.

[8] N. Voloshina, S. Bezzateev, A. Prudanov, M. Vasilev, and A. Gorbunov, "Effectiveness of LSB and MLSB information embedding for BMP images," in *18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT)*, 2016, pp. 378–384.

[9] W. Sipayung, "Perancangan Citra Watermaking Pada Citra Digital Menggunakan Metode Discrete Cosine Transform (DCT)," *Pelita Inform. Budi Darma*, vol. 7, no. 3, 2014.

[10] M. Ricky, F. A. Setyaningsih, and M. Dipenogoro, "Analisis Kompresi Steganography Pada Citra Digital Dengan Menggunakan Metode Least Significant Bit Berbasis Mobile Android," *J. Coding*, vol. 06, no. 03,

2018.
[11] P. N. Andono, T. Sutojo, and Muljono, *Pengolahan Citra Digital*. Yogyakarta: ANDI (Anggota IKAPI), 2017.
[12] S. N. Kishor, G. N. K. Ramaiah, and S. A. K. Jilani, "A review on steganography through multimedia," in *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, 2016, pp. 1–6.
[13] M. A. Zaher, "Modified Least Significant Bit (MLSB)," *Comput. Inf. Sci.*, vol. 4, no. 1, 2011.
[14] A. A. Lubis, N. P. Wong, I. Arfiandi, V. I. Damanik, and A. Maulana, "Steganografi pada Citra dengan Metode MLSB dan Enkripsi Triple Transposition Vigenere Cipher," *JSM STMIK Mikroskil*, vol. 16, no. 2, 2015.
[15] Mulyanto, R. V. Febriyana, and B. W. P. Arief, "Penyisipan Pesan Teks pada Citra Menggunakan Metode LSB dan 2-Wrap Length," *Semin. Nas. APTIKOM*, 2019.
[16] A. Supriyanto and E. Ardhianto, "Penyandian File Gambar dengan Metode Substitusi dan Transposisi," *J. Teknol. Inf.*, vol. 8, no. 2, 2008.
[17] M. P. Singh and H. Singh, "An Efficient Modified LSB technique for Video Steganography," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 4, no. 6, pp. 5253–5259, 2015.
[18] A. B. Nasution, S. Efendi, and S. Suwilo, "Image Steganography in Securing Sound File Using Arithmetic Coding Algorithm, Triple Data Encryption Standard (3DES) and Modified Least Significant Bit (MLSB)," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, pp. 1–6, 2018.
[19] P. Yadav and M. Dutta, "3-Level security based spread spectrum image steganography with enhanced peak signal to noise ratio," in *Fourth International Conference on Image Information Processing (ICIIP)*, 2017, pp. 1–5.
[20] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 287–290, 2011.
[21] S. Gupta, J. Saxena, and S. Singh, "Design of Random Scan Algorithm in Video Steganography for Security Purposes," *IOSR J. Electron. Commun. Eng. Ver. I*, vol. 10, no. 5, pp. 2278–2834, 2015.
[22] E. Y. Hidayat and K. Hastuti, "Analisis Steganografi Metode Least Significant Bit (LSB) dengan Penyisipan Sekuensial dan Acak Secara Kuantitatif dan Visual," *Techno.COM*, vol. 12, no. 3, 2013.
[23] A. A. Abdulla, S. A. Jassim, and H. Sellahewa, "Secure Steganography Technique Based on Bitplane Indexes," in *IEEE International Symposium on Multimedia, Anaheim*, 2013, pp. 287–291.
[24] N. Farid, B. Nurhadiyono, and Y. Rahayu, "Implementasi Metode Steganografi Least Significant Bit Dengan Algoritma Hill Cipher Pada Citra Bitmap," *J. Techno.COM*, vol. 15, no. 2, pp. 109–116, 2016.