International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 2 (2018) pp. 1354-1358

Image Steganography Based on Rubik's Cube and Convolutional Code for Error Correction

Fitria Rahma Nur Sulistyo[1], Dr. Ir. Bambang Hidayat[2], Suci Aulia S.T., M.T[3]

[1,2]Study Program S1 Telecommunication Engineering, Faculty of Electrical Engineering, Telkom University, Indonesia.
[3]Study Program D3 Telecommunication Engineering, Faculty of Applied Science, Telkom University, Indonesia.

Abstract

Steganography has the main purpose which is to send a message to a recipient while concealing the content of the message to others. In this study, rubik's cube steganography method is used as the stego message positioning technique for text message insertion into a digital image. The rubik's cube scrambling pattern is used as a reference for the text message insertion into a digital image. To reduce errors during disruptions, convolutional code for error correction method is used. Based on the test results of this system, when not disturbed and the secret message is still within the limit of insertion capacity obtained BER value: 0 with the value of PSNR above 75 dB. The number of characters that can be inserted onto the cover with convolutional code is less when compared to insertion without convolutional code. System test using Salt & Pepper Noise, Gaussian Noise, Poisson Noise, Localvar Noise, resizing, cropping, rotating, brightness

Keywords: steganography, digital images, rubik's cube

INTRODUCTION

One way to provide security to the data is by using steganography techniques. Steganography is the science and art of hiding messages on other media so that the existence of such secret messages can not be known unless the steganography insertion key is known [1-4]. While the steganography image is the insertion of the message with the image as the media or often referred to as cover image [5].

Some researches relating to steganography image are using general method in insertion techniques such as LSB, MSB and MLSB. What will be discussed here is a study that examines the security technique in the steganography process, by adding encryption and randomization key at the placement location of the message. The method that will be directly discussed here is the Rubik's cube algorithm based on some reviews, among them are; [6] has proved that the Rubik's cube algorithm has high PSNR value besides short computation time and resistant to interference, [7] has tested the Rubik's cube algorithm by adding visual attack to the system and Chi square analysis and the results show that the message is very difficult to extract, [8] Chi square analysis itself is a technique often used to detect the existence of a secret message in an image or audio, [9] Kaziwa proved that the Rubik's cube algorithm produces high SNR values for the insertion process in steganography audio. Besides being used for insertion technique, the Rubik's cube algorithm is also widely tested for encrypting the message before it is inserted into the host. Related studies [10] who have conducted the study prove the value of PSNR is above 56dB.

In this study, a study with the Rubik's cube method is used to determine the position of secret messages insertion in the digital cover image. As for the process of secret messages insertion on the image is done by using the Modified Least Significant Bit (MLSB) method. To ensure a good level of accuracy of the received data additional steps are used, such as the uses of detection technique and error correction technique with convolution code.

LITERATURE REVIEW

This section describes the insertion of a Rubik's cube based message. In this study, the Rubik's cube that will be used as a reference is cube 4x4x4. The Rubik's cube method is used to determine the position of secret message bits in the cover image. Rubik's cube will be rotated so the position will change, the changed position will later become the reference position of the secret message bits insertion [11]. Equation (1) is an equation of the Rubik's cube algorithm used in this study with the illustration is shown in Figure 1.

$$f(x, y) = i, 16i, 33i, 49i, 65i, 81i, \quad \text{Where} \quad i = 1,2,3, \ldots ,8 \tag{1}$$

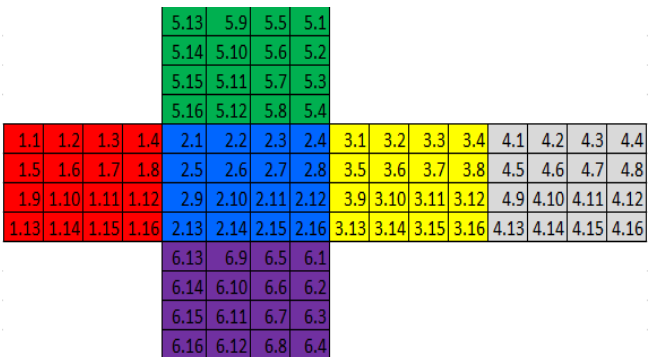

Figure 1: Corresponding Index of a Rubik's Cube

From Figure 1 [12], it appears that the cube has 6 faces with each face divided into 16 sections (4x4) so the total coordinates if converted into pixels are 64 coordinates (coordinate 1.1 to 6.9). The 64 point coordinates are randomized according to keys used by the sender. The new coordinate sequence as a result from randomization indicates the location where the message is stored within the cover image. Randomization can also be repeated in accordance to the iteration's initialization set to increase the time for Brute Force Attack (BFA). The BFA itself is an attack technique to know the keys of a security system [13].

Convolutional Code

Convolutional code consists of four parameters, i.e. k (number of input bit), n (number of output bit), m (sum of register memory) and L (bit code length). Code rate is the division of input bit into output bit, the code rate is a efficiency measurement of a sent code. In general the number of bit k and m ranges from 1 to 8 bits, m consists of 2-10, while for the code rate ranges from 1/8 s.d 7/8. The L value is derived from the following (2) [14]:

$$L = k\,(m-1) \qquad (2)$$

Convolution codes may be represented in various ways as state diagram, tree diagram, and trellis diagram . A convolutional code is generated by passing the information sequence through the shift register. Figure 2 is an example of a convolutional encoder.

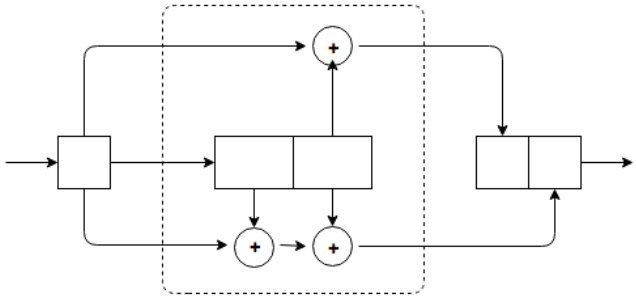

Figure 2: A convolutional encoder [n,k = 2,1]

RESULT AND ANALYSIS

A. Analysis of the effect of cover image size and Convolutional Code use on message insertion capacity.

Tabel I : Message Insertion Maximum Capacity

Image	Maximum Number of Character		
	with CC (R= 1/3)	with CC (R= ½)	without CC
lena.bmp	4439	6659	13319
mickeymouse.bmp	13956	20934	41868
dinding.bmp	14802	22203	44406
gradasi1.bmp	2126	3190	6380
gradasi2.bmp	1514	2271	4543

On Table 1 above, it can be seen that the larger the image size, the capacity of insertion message also increases. The effect of using convolutional code (cc) is the number of character that can be inserted less than the insertion without cc.

B. Analysis of the effect of Salt & Pepper Noise and Convolutional Code use on stego image's PSNR and extracted message's BER

Based on Figure 3 and Figure 4, it can be seen that the effect of the Salt & Pepper Noise is, the greater the variance of the Salt & Pepper Noise the smaller the PSNR value becomes as seen on Table 2 . This happens because the greater the noise variance, the more image pixel value to be changed into red, green, or blue.

Table 2. A Convolutional Code on image PSNR and BER's value with R=1/2.

Image	Variance	Using Convolutional Code (R=1/2)		
		PSNR	BER	Text
lena.bmp	10^-5	55.0618	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-4	44.5527	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-3	35.1103	0.134259	T e
	10^-2	24.9801	0.175926	x%0ae9c;r `/w<b4o(`*w-r! }$w r4z%`l!z)p$owp !2#$5 ? ±
mickey mouse.bmp	10^-5	51.9804	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-4	43.6269	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-3	33.4249	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-2	23.387	0.127315	dh%pqeic+`rb • g.`f • hrm}p+` • nu"xd`%`lqz!`lo/00123456?¸©

dinding.bmp	10^-5	55.5697	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-4	46.6679	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-3	36.3042	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-2	26.0235	0.166667	$x%0ae9c;p"b/w>`6o(`*u-p# • &u"p H%@L1Z ` OoP 1 4 ? • !
gradasi1.bmp	10^-5	53.5624	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-4	43.3353	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-3	33.3211	0.377315	gDfenlohrfcLt`0ekrkq1coe~q&}ik+e • ipi%m{b?q}`$yrzr5nlx
	10^-2	23.6278	0.479167	K p+
gradasi2.bmp	10^-5	54.8722	0	The quick brown fox jumps over the lazy dog 0123456789
	10^-4	43.1607	0.3125	lnqrUie!qthbk!brnvn!foy!kumqr!over the lazy dog 012345
	10^-3	34.1393	0.395833	JØgòéi· Ñ(¢Í¯ C- @2 p x \02rh*j&f'x>n8h+e,y"}&o8*p(q<p
	10^-2	24.0937	0.483796	qP

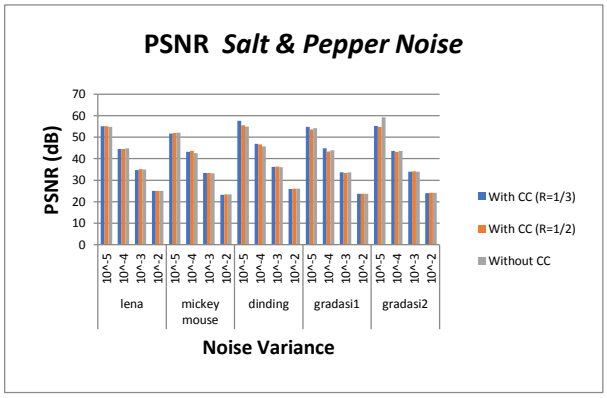

Figure 3: The Effect of Salt & Pepper Noise on image's PSNR value

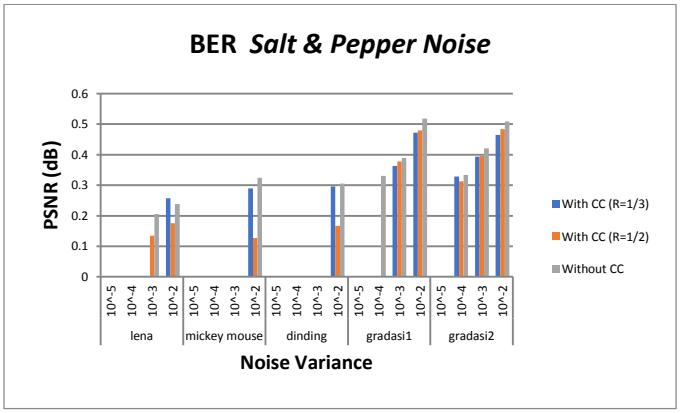

Figure 4: The Effect of Salt & Pepper Noise on Image's BER value

As the noise variance becomes larger, the BER value also becomes larger, as shown in Figure 4 and as the reverse is shown in Figure 3 for the PSNR parameter. On lena.bmp image, the system has resistance to Salt & Pepper Noise until 10^{-4} noise variance. On mickeymouse.bmp and wall.bmp images, the system has resistance to Salt & Pepper Noise until 10^{-3} noise variance. In gradation1.bmp image, the system has resistance to Salt & Pepper Noise until 10^{-4} noise variance. While on the image gradasi2.bmp, the system has resistant to Gaussian Noise until 10^{-5} noise variance. BER value when convolutional code is used is less than when convolutional code is not used. The smaller the rate of the convolutional code results in a smaller BER value. The image that has the highest resistance to Salt & Pepper Noise is the mickeymouse.bmp and the dinding.bmp image. While the image that has the lowest resistance to Salt & Pepper Noise is the gradasi2.bmp image.

C. Analysis of the effect of the Gaussian Noise dan Convolutional Code use on stego image's PSNR and extracted message's BER

The effect of the Gaussian Noise on the PSNR value is the greater the noise variance resulting in smaller PSNR value. This can be seen in Figure 5. As for BER when not using convolutional code it has a greater value than when using convolutional code. The smaller the convolutional code rate then the error correction ability becomes larger, as shown in Figure 6. This system is resistant to Gaussian Noise until 10^{7} noise variance

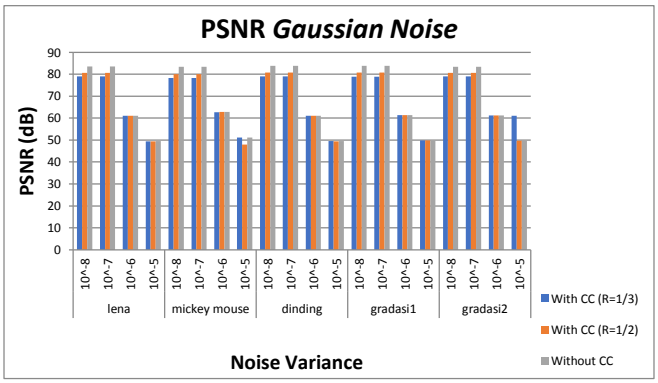

Figure 5: The effect of Gaussian Noise on Image's PSNR Value

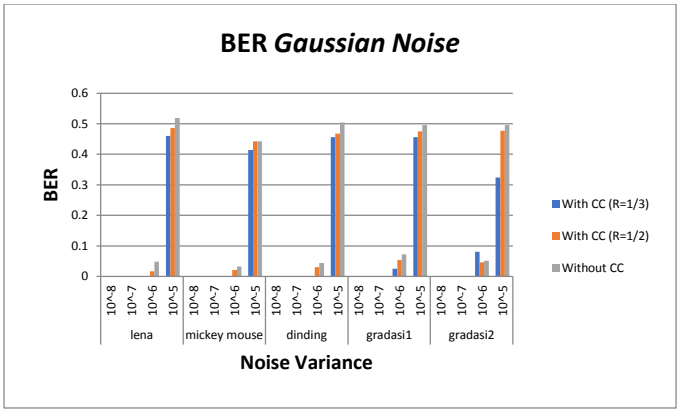

Figure 6: The effect of Gaussian Noise on images's BER value.

D. Analysis of the effect of the Poisson and Localvar Noise using CC on Stego Image's

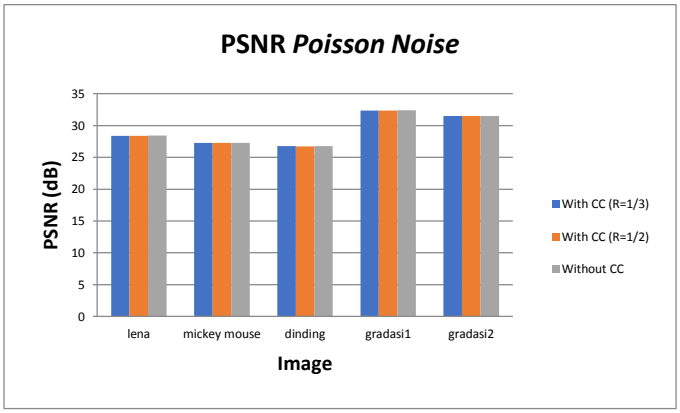

Figure 7: The effect of Poisson Noise on image's PSNR value.

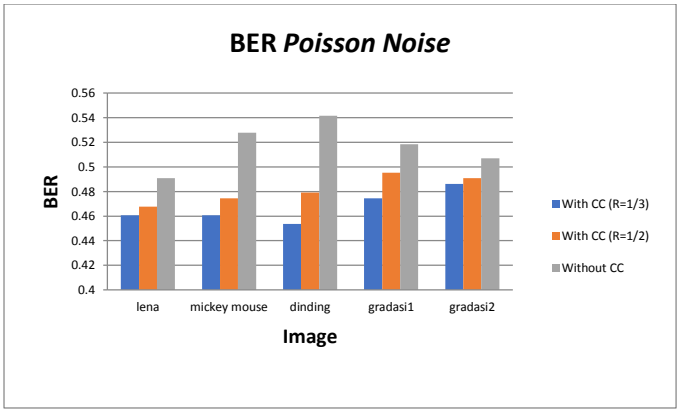

Figure 8. The effect of Poisson Noise on images's BER value

Based on Figure 7, the effect of Poisson Noise on PSNR depends on the image used. The smallest PSNR value is obtained when using dinding.bmp image. The BER value can be seen in Figure 8. The convolutional code correction ability with 1/3 rate is better than the convolutional code with 1/2 rate. The result can be seen in Figure 8.

E. Processing Pictures (Analysis of the effect of the Resize, Crop, Rotate, Brightness and Convolutional Code on stego image's PSNR and extracted message's BER)

Resizing

The effect of resizing on BER depends on the size of the stego image . Whether the insertion is using the convolutional code or not, it does not have a definite pattern and nothing is superior. This is because when the image changes in size, the pixel value also goes through the changes, but the change in pixel value occurs randomly. The value of BER also depends on the image used. This system can not resist resizing attacks.

Cropping

The BER value when the cropping is done is depends on the position of the image cropping. If the cropped image is in the position of the secret message is inserted, then the value of BER becomes smaller. However, when the image is cropped in a position where the secret message is not inserted, the BER value becomes larger.

Rotating

Same as the effect of resizing,The rotating effect on the PSNR value depends on the image used, the smallest PSNR value obtained when the image is rotated by 180° angle. Whereas on other images the value of BER does not have a definite pattern. On mickeymouse.bmp image, the smallest BER value is obtained when using convolutional code with 1/3 rate.

Brightness

The value of PSNR will be greater when given the brightness value close to 0. This happens because when the brightness value close to 0, the changed pixel value is not too large.

CONCLUSION

The conclusion obtained from the design through the testing, shows that the capacity of the secret message is greater when the steganography system without convolutional code error correction is about 49.99%. The BER value of the extracted messages reaches a value of 0 when the inserted message is still within the limit of the inserted message capacity and without a noise addition. The system resistance to the noise tested with the BER value of the extracted message is to a value of 0 when using convolutional code i.e. $\sigma_2=10^{-3}$, 10^{-7} and 10^{-6}, for Salt & Pepper Noise, Gaussian Noise, and Localvar Noise respectively. As for Poisson Noise the BER value is lower i.e. 0.5 due to its very random nature.

Stego image's BER value when attacked by resizing, cropping, rotating, does not have a definite pattern. As for the brightness attack, the BER value is directly proportional. The average PSNR value obtained in this test when it is not given interference is above 75 dB

REFERENCE

[1] A. Kumar and K. Pooja, "Steganography-A Data Hiding Technique," Int. J. Comput. Appl., vol. 9, no. 7, pp. 975–8887, 2010.

[2] A. Toumazis, "Steganography," 2009.

[3] P. Kamble and S. Engineering, "Steganography Techniques: A Review," Int. J. Eng. Res. Technol., vol. 2, no. 10, 2013.

[4] E. Nazora, B. Hidayat, and S. Aulia, "Analisis Steganografi Citra Digital Dengan Metode Sudoku Puzzle (Analysis Digital Image Steganography Using Sudoku Puzzle Method)," Eproceeding Telkom Univ., 2013.

[5] T. Morkel, J. H. P. Eloff, and M. S. Olivier, "An Overview of Image Steganography," Inf. Comput. Secur. Archit. Res. Gr., vol. 83, no. July, pp. 51–107, 2005.

[6] A. Khare and B. Gupta, "Survey on Steganography with Cryptography," Int. J. Emerg. Technol. Adv. Eng., vol. 7, no. 7, pp. 335–339, 2017.

[7] S. Raniprima, B. Hidayat, and N. Andini, "Digital image steganography with encryption based on rubik's cube principle," ICCEREC 2016 - Int. Conf. Control. Electron. Renew. Energy, Commun. 2016, Conf. Proc., pp. 198–201, 2017.

[8] N. Arifah, B. Hidayat, and S. Aulia, "Steganalisis Pada Citra Digital Dengan Format Jpeg Menggunakan Uji Chi-Square," in Eproceeding Telkom University, 2014.

[9] K. Saleh, M. Muhammad, and K. Ahmed, "Using Rubik ' s Cube and a Modified LSB for Audio Steganography," J. Zankoi Sulaimani, vol. 4, pp. 187–194, 2015.

[10] Monica Deswanti Fista, H. Bambang, and A. Suci, "Steganografi Citra Digital Menggunakan Enkripsi Berdasarkan Prinsip Kubus Rubik Dan Kode BCH," in Universitas PGRI Yogyakarta, 2015, no. 10, pp. 425–432.

[11] V. T. and J. B. B. R. Rengarajan Amirtharajan, M.Venkata Abhiram, G.Revathi, J. Bharathsimba Reddy, "No Title," Res. J. Inf. Technol., vol. 5, 2013.

[12] R. Shelke and S. Metkar, "Image scrambling methods for digital image encryption," in 2016 International Conference on Signal and Information Processing (IConSIP), 2016, vol. 5, no. 1, pp. 1–6.

[13] S. Aulia, K. Usman, and U. Wijayanti, "Desain dan implementasi sistem steganografi berbasis SSB-4 dengan pengamanan baker map untuk citra digital," in DISC Maranatha, 2010, vol. 7.

[14] C. Langton, "Tutorial 12 Coding and decoding with Convolutional Codes," in Complex2Real. com Complex Communications …, C. Langton, Ed. 1999, pp. 1–29