Analysis of LSB Algorithm Modification with Bit Inverse and Insertion based on Length of Message

Fahrul Ikhsan Lubis[1], Saib Suwilo[2] and Poltak Sihombing[2]
[1]*Magister Informatics, Universitas Sumatera Utara, Medan, 20155, Indonesia,*
[2]*Department of Informatic, Universitas Sumatera Utara, 20155, Indonesia*

Keywords: Steganography, LSB Algorithm Modification, Bit Inverse, Length of Message, LCG Algorithm, Text Message, Image Massage.

Abstract: Steganographic demolition using the LSB Algorithm is increasingly being carried out by researchers for modifications as needed. Previously, the LSB algorithm has been modified by doing a bit inverse where the message is inserted into the container image and then modified by looking at the 2nd and 3rd LSB bits. If the two bits are different then the inverse of the entered message bit (bit 1 LSB) is performed. However, if the two values are the same then the entered message will not be inverted. This modification is considered good but still lacks in the pixel position of the image that is inserted with the message (modification is done by inserting sequential images in an image). To cover this weakness, in this study a modification of the LSB with insertion based on the length of the message was carried out. Message insertion was carried out on random pixels of an image using the LCG algorithm and the number of message bits that were inserted varies according to the message length. Moreover, the modifications made should produce a better stegoimage than before, as evidenced by calculating the PSNR of a stegoimage. From the modification test results, it was found better results than before both in text message archiving (66.29 dB > 61.8 dB) or in image messages (54.20 dB > 50.01 dB).

1 INTRODUCTION

The insertion of messages into images using steganography techniques has experienced rapid development. The researchers competed to make their own modifications according to their knowledge and experience of the insertion. This occurs because the basic theory of message insertion is considered to have been overused and tried to modify the method.

Previously, the algorithm (LSB) has been modified by inverse bit where the message will be inserted into the container image first. Furthermore, the 2nd and 3rd bit LSB is scanned. If the two bits have different binaries then the inserted message binary will be converted to the inverse binary that was inserted. If the 2nd and 3rd bit LSB are the same, then the inserted message does not change (Bharwaj and Sharma, 2016).

Previous research has produced a good message insertion, but the modification only inserts one message bit in each pixel and the insertion position is done on consecutive pixels in an image. This modification has gaps to be recognized because if a more detailed check is carried out, the reduction in taste quality in certain parts in sequence will be detected. To cover the weakness of the previous researchers, the second modification inserts messages on random pixels in an image and the number of message bits inserted is adjusted to the message bit length. If the length of the message in modulo 2 is 0, then 2 message bits will be inserted at each image layer at positions 1 and 2 LSB. Meanwhile, if the result is 1, the insertion of 1 message bit on each image layer at position 1 LSB is carried out. Thus the insertion will be difficult to detect because the insertion of random messages using steganographic techniques will increase the security of the message (Sitompul *et al*, 2018) and the number of bits inserted can change according to the message length.

Steganoimage of the two LSB modifications will be tested by calculating the Peak Signal to Noise Ratio (PSNR), Mean Squared Error (MSE), and the time required to insert the message.

Lubis, F., Suwilo, S. and Sihombing, P.
Analysis of LSB Algorithm Modification with Bit Inverse and Insertion based on Length of Message.
DOI: 10.5220/0010333500003051
In *Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (CESIT 2020)*, pages 522-529
ISBN: 978-989-758-501-2

2 RELATED STUDIES

Steganography was a process to embed data to cover object. The cover object might be image, text, audio, or video. Cover object were a container to hidden data or secret messages and they were the main material of steganographic systems, where some of their characteristics were altered or manipulated to conceal a confidential message. However, these modifications or manipulation that occurred during the process of concealing should rest unnoticeable for anyone not participating in the process of communication. The ability of files used as a cover for embedding confidential data depends on the availability of redundant or unimportant areas inside these files, so the cover file size might be larger than the message size to include (Shtayt *et al*, 2020).

Image steganography was the most widely used, compared with the other types of steganography. This popularity was because the images have a large amount of redundant information. Image steganographic techniques were evaluated by three principles: (i) Capacity, The amount of information was hidden inside the cover file; (ii) Imperceptibility, The invisibility of the hidden data was in the cover file without destroying image quality; (iii)Security, How could a stego file resist the different steganalysis detection attacks. (Al-Aidroos and Bahamish, 2019)

The terms in steganography were: (-) Embedded message or Hidden text: secret message to be inserted; (-). Cover-object: the object of the message insertion (embedded message); (-) Stegoimage or Stego-object: objects that had been inserted secret messages (embedded message); (-) Embedding or Encoding: the process of inserting a message into the Cover object; (-) Extraction or Decoding: Stegoimage extraction process to issue an original message (embedded message) (Gunawan and Sumarno, 2018).

Steganography techniques were divided into Spatial (time) domain and Transform (frequency) domain technique: Spatial Domain Technique, uses the pixel values of images directly for encoding the secret message. This class of technique was Least Significant Bit (LSB) replacement technique in which firstly binary representation of the image's pixel value was calculated then bits were used to hide the secret messages. Initially, for a 24-bit image, each of the red, green and blue colour components of bit could be used, as each was represented by a byte. In other words, one could store 3 bits in each pixel. This technique hide the secret message bits into the 3 LSB bits of the cover image.

Transform Domain Technique, These transform domain techniques mainly included Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT). This technique used DCT and Blowfish algorithm in which the LSB of each DC coefficient replace with each bit of secret message. This algorithm was proposed to increase the security of hidden message. (Singh *et al,* 2018).

2.1 Least Significant Bit (LSB) Algorithm

LSB algorithm was modifying the last bits in the cover object in every byte of color in a pixel with replacing each last bit with the secret bits of information (Sari and Siahaan, 2018). In the using of LSB algorithm, firstly the message and the cover image must be converted to binary, after it completed, one bit message would be added to each layer in the cover image pixel. The procedure was as follows:

Suppose the message binary was 010 and the binary of one pixel image in each layer was:

01010111 01010011 01010111

If the message and binary image were known, subtitute the message at the end of the binary or the 1st LSB binary image as follows:

01010110 01010011 01010110

After the pixel binary completed it would be converted into a matrix and made into an image again.

2.2 Image

Image was a photo or two-dimensional dwasplay that could describe the vwasualization of objects. The image could be grouped into print or digital form. Digital images could be converted into an array of numbers, while printed images must first be converted to digital if they want to be processed.

Digital image was a collection of thousands of very small dots and each of these dots has a certain color. The small boxes were called pixels, the number of pixels in an image could also determine the size of the image. Resolution was the number of pixels per certain. If the pixel could determine the size of the image, the resolution could determine the image quality. Intensity was the number of colors contained in an image. Intensity also has many terms such as: 256 colors, black and white (black & white), high color, grayscale, and 16 million colors (true color).

The maximum number of colors in an image depends on the file type (extension). File types with a jpg extension could accommodate a maximum color of 16 million colors, files with a .gif extension could accommodate a maximum color of 265 colors (Prabowo, Abdullah and Manik, 2018).

2.3 Image RGB

Color was identified in three-dimensional color space, For types of hardweare-oriented color space include RGB (Red Green Blue), CMY (Cyan Magenta Yellow) and YIQ, while user-oriented color space types include HLS (Hue Saturation Luminance), HCV, HSV (Hue Saturation value), HSB, MTM, CIE-LAB, and CIELUV .Most image file formats (JPEG, BMP, GIF) use the RGB color space. RGB color space was defined based on the values of the axes R, G and B.

There were millions of colors in this nature. If calculated based on variations in the value of R, G and B that were owned by a color, then there would be 256 x 256 x 256 pieces of color. Each value of R, G and B which was owned by a color varies between 0-255. (Karma, 2020).

2.4 Image Quality Testing

Peak Signal to Noise Ratio (PSNR) and Mean Squwere Error (MSE) was the most common parameter used to measure the image quality testing (stegoimage). PSNR told the similarity between the original image and the image from the insertion and was the opposite of MSE which was the damage value of a stego-image. Mathematically PSNR in formulated (1) and MSE in formulated (2):

$$PSNR\ (dB) = 10log_{10}\left[\frac{P^2}{MSE}\right] \tag{1}$$

$$MSE = \frac{1}{RC}\sum_{i=0}^{R-1}\sum_{j=0}^{C-1}\left(X_{i,j} - X'_{i,j}\right)^2 \tag{2}$$

Which was:
P = maximum pixel value,
R = number of pixel in each row,
C = number of pixel in each column,
i ,j = row and column numbers,
$X_{i,j}$ = original image
$X'_{I,j}$ = stego-image.

Analysis obtained results showed that PSNR was reduced when secret information size was increased because of more pixel in cover image was changed (more Noise) (Shehab and Abdulkadhim, 2018).

2.5 Linear Congruential Generator (LCG) Algorithm

Linear Congruential Generator (LCG) algotrithm was a simple pseudo-random number generator which used to determine the next random number based-on previously generated one. This approach had a potential repetition on the generated numbers whenever the numbers selected as parameter values were not appropriately chosen (Sitompul *et al*, 2018), Mathematically LCG will shown in formulated (3):

$$Xn = (a\ (Xn\text{-}1) + b) \bmod m \tag{3}$$

Where:
Xn-1 = previous random number
X = i-th random number
A = multiplier constant
B = constant increase (increase)
M = modulus constant

The period of LCG algorithm was not greater than the modulus (m), the modulus was the maximum threshold for randomizing numbers. So, the larger the size of the container image, the more messages could be inserted. (Hernandes, et al, 2019).

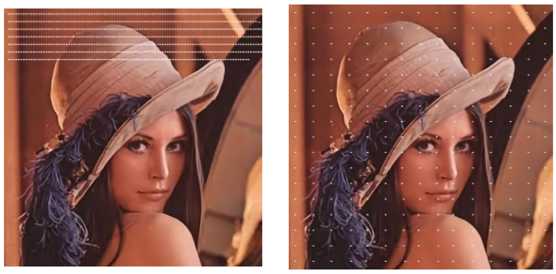
3 METHODOLOGY

The two LSB modifications have something in common where the modifications are made based on input. The first modification of the LSB is carried out based on the container image, especially on bits 2 and 3 of the LSB. While the second modification is carried out based on the length of the message to be inserted into the image. The difference between the two modifications is the position of pixel that are inserted message, and the number of bits that are inserted on insertion process.

3.1 Position of Pixel

In LSB modification with bit inverse, message insertion is done on successive image pixels (resulting in a decrease in image quality in a certain area). LSB modification based on the message length, the insertion is done at random pixels using Linear Congruential Generator (LCG) Algorithm where the constant modulus value is the number of pixels in an image and the multiplier value with an increment of random numbers between 1-100.

Sitompul *et al* (2018) have conducted a study on random message insertion with a smaller MSE yield and a larger PSNR. Figure 1 is an illustration of the pixel position inserted with the message:



(a) (b)

Figure 1: Illustration of the pixel position inserted with the message on LSB Modification with Bit Inverse (a) and LSB modification with insertion based on the length of the message (b).

From Figure 1, it is clear that the sequential insertion of messages makes a decrease in image quality in a certain area whereas random insertion results in only a slight decrease in image quality (inconspicuous and easily recognized by people without access rights.

3.2 Insertion Process

3.2.1 LSB Modification with Bit Inverse

First, insert the message using the LSB algorithm then the prepared message is in the inverse bit. If bits 2 and 3 LSB are the same then inverse is not required. The insertion process is carried out in the following stages: First, change the message and cover image to binary, if the message bit is 010 and the image pixel is:

01010111 01010011 01010111

then the message bit in the last binary of each pixel layer is inserted or position 1 LSB:

01010110 01010011 01010110

After the insertion is complete, the 2nd and 3rd bits will be visible. If the bits are the same, the message (bit 1 LSB) will not change, whereas if the bits are different, the message (bit 1 LSB) will be inverse, like:

01010110 01010010 01010110

After checking each pixel where the message has been inserted, the cover image binary will be converted back into an image.

3.2.2 LSB Modification with Insertion Based on Message Length

The first step in the modification is to calculate the length of the message then convert the message into binary form before the inverse binary message is carried out. The length of the message is modulo with 2, if the result of modulo is 0 then 2 message bits will be inserted at each pixel layer, whereas if the result of modulo is 1 then 1 message bit will be inserted. After the message has been processed, determine the pixels to be inserted using the LCG algorithm, according to the results of Zyaraa (2017) research use Linear Congruential Generator (LCG) Algorithm will increase the security of stegoimage.

Then convert the pixels into binary form. An example of this modification if the result of modulo is 0, the message bit is 101 and the binary of the selected image pixel is:

01010011 01010011 01010011

The message will be inverted first to 010 then insert 2 bits at once in each pixel layer at positions 1st and 2nd LSB, then the insertion results are:

01010001 01010010 01010011

Meanwhile, if the result of the message modulo is 1 then insertion of 1 bit will be carried out in each pixel layer at position 1st LSB so, the insertion result will be:

01010010 01010011 01010010

After all insertion is complete, the binary of the container image will be changed again and the image will turn into a composite image and secret messages.

4 TESTING RESULT

In the experiment, the message used in the form of text or images must be determined before inserting the message. The requirement in testing is that the message length cannot be greater than the cover image (the resolution of the flax image used is 512 x 512, with lena (i) baboon (ii) and pepper (iii)). Image 2 is the image used in the study:


(i)


(ii)


(iii)

Figure 2: Cover Image.

4.1 Text Message Insertion

Following are the results of inserting a text message with a character length of 2.110 and a number of bits of 16.880. Modification of the LSB with insertion based on the length of the message insertion is done as much as 2 bits per byte, the second experiment was carried out with a character length of 2.109 and a number of bits of 16.872, where the LSB modification with substitution based on the length of the message insertion is 1 bit per byte. Then the message is tested on the three cover images. In this study, a system where after the message and cover image are entered, it will automatically generate MSE, PSNR, and insertion time has been developed as in Figure 3:

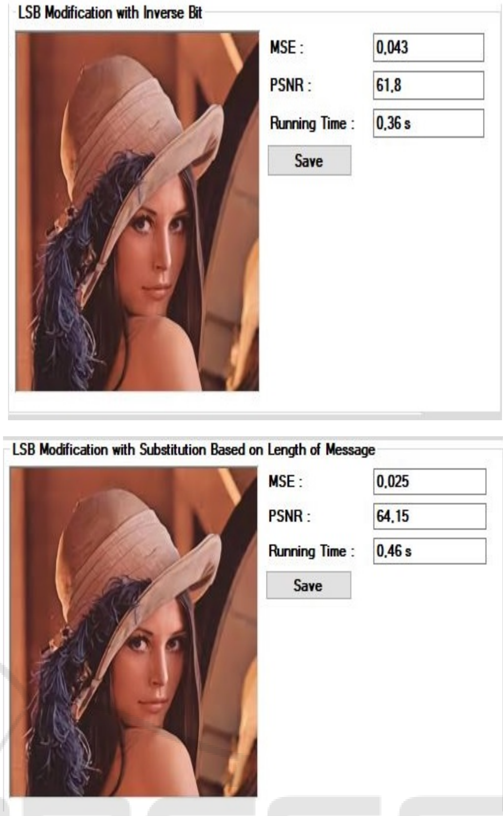


Figure 3: Encoding process for text substituion in system.

MSE, PSNR and running time from figure 3 will collect and analys, there would be a reduction in image quality and This could be seen from the Mean Squared Error (MSE), so the higher the MSE in an image, the lower the quality of the resulting image and the lower the Peak Signal to Noise Ratio (PSNR), the image the resulting would resemble the original image. The results of the MSE and PSNR results can be seen in Table 1:

Table 1: Comparison of stegoimage Quality of Text Message Insertion.

Cover Image	Length of Message	LSB Modification with Bit Inverse		LSB Modification with Subtitution Based On Message Length	
		MSE	PSNR dB	MSE	PSNR
Lena	2109	0,043	61,8 dB	0,011	67,72 dB
Lena	2110	0,043	61,8 dB	0,025	64,15 dB
Baboon	2109	0,043	61,8 dB	0,011	67,72 dB
Baboon	2110	0,043	61,8 dB	0,025	64,15 dB
Pepper	2109	0,043	61,8 dB	0,011	67,72 dB
Pepper	2110	0,043	61,8 dB	0,025	64.15 dB
Average			61,8 dB		66,29 dB

Based on Table 1, it can be seen that if each experiment LSB modification with subtitution based on mesaasge length is better than LSB modification with bit inverse, it means that the use of Linear Congruential Generator (LCG) Algorithm has a very significant impact on the development carried out. Furthermore, Table 2 also shows the time required to insert the message.

Table 2: Comparison of Encoding Time on Text Message Insertions.

Cover Image	Length of Message	LSB Modification with Inverse Bit	LSB Modification with Subtitution Based On Message Length
Lena	2109	0,31 s	0,54 s
Lena	2110	0,36 s	0,46 s
Baboon	2109	0,34 s	0,89 s
Baboon	2110	0,32 s	0,45 s
Pepper	2109	0,33 s	0,54 s
Pepper	2110	0,29 s	0,40 s
Average		0,33 s	0,55 s

From Table 2, it can be seen that the time required for LSB modification with substitution based on the length of the message looks longer because the modifications made must do pixel randomization before message insertion. Meanwhile, LSB modification with bit inverse does not need to randomize pixels and only inserts so that it takes less time.

4.2 Image Insertion

Image insertion is not much different from text insertion, so that in the experiment, an image length of 100 and a number of bits 240000 where LSB modification by insertion according to the length of the message was carried out 2 bit each layer pixel. The second experiment was carried out with an image length of 105 and the number of bits 264600, where LSB modification by insertion according to the length of the message was carried out 1 bit each layer pixel.

Then the message is tested on the three cover images. In this study, a system where after the message and cover image are entered, it will automatically generate MSE, PSNR, and insertion time has been developed as in Figure 4:

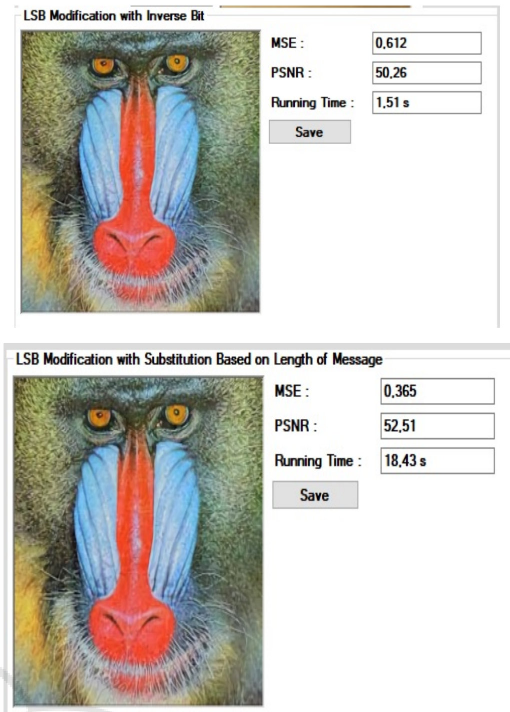


Figure 4: Encoding process for image insertion in system.

The comparison of stegoimage quality as evidenced by calculating mse and psnr will be shown in Table 3.

Table 3: Comparison of stegoimage Quality of Image Message Insertion.

Cover Image	Length of Message	LSB Modification with Bit Inverse		LSB Modification with Subtitution Based On Message Length	
		MSE	PSNR	MSE	PSNR
Lena	105	0,67	49,85 dB	0,17	55,88 dB
Lena	100	0,61	50,27 dB	0,36	52,52 dB
Baboon	105	0,67	49,84 dB	0,17	55,88 dB
Baboon	100	0,61	50.26 dB	0,37	52,51 dB
Pepper	105	0,67	49,86 dB	0,17	55,88 dB
Pepper	100	0,61	50,26 dB	0,36	52,52 dB
Average			50,01 dB		54,20 dB

From Table 3, it can be seen that the quality of the message inserted 1 bit has a higher PSNR. This happens because the resulting image reduction is quite small, only 1 bit compared to 2-bit insertion which gives more image degradation. Even so, the LSB modification substitution based on the length of the message with are still better than the LSB modification with bit inverse. The time will be shown in Table 4.

Table 4: Comparison of Encoding Time on Image Message Insertion.

Cover Image	Length of Message	LSB Modification with Bit Inverse	LSB Modification with Subtitution Based On Message Length
Lena	105	0,61 s	61,09 s
Lena	100	1,59 s	62,29 s
Baboon	105	0,57 s	189,40 s
Baboon	100	1,51 s	18,43 s
Pepper	105	0,66 s	81,70 s
Pepper	100	1,92 s	40,71 s
Average		1,14 s	75,60 s

The time is not much different from the insertion of text, only a little longer (because the modifications made must randomize the pixels before inserting the message). Meanwhile, LSB modification with bit inverse does not need to randomize the embedded pixels so that it takes less time.

4.3 Result Analysis

Based on the research that has been carried out, the modification made by the author with the insertion based on the length of the message provides better stegoimage quality to LSB modification with the inverse bit. This is evidenced by the average Peak Signal to Noise Ratio (PSNR) as a variable which states the similarity between the original image and the higher stegoimage, both at the time of text message insertion (66.29 dB> 61.8 dB) or image insertion (54.20 dB> 50.01 dB). In the use of steganography techniques, comparison of image quality before and after message insertion is very important. If a stegoimage file has a large enough file size and the image quality is too bad, other people who know about the use of steganography techniques will be suspicious of the image.

The LSB modification by insertion based on message length has the advantage of stegoimage quality. This is because the used of Linear Congruential Generator (LCG) Algorithm in randomizing the pixels to be inserted into the message. From those results, it can be concluded that the developed modifications were very satisfactory as seen from the higher quality of the stegoimage. While deficiencies were found only during processing, however, they are still within reasonable limits.

5 CONCLUSION

The use of the LCG algorithm in LSB modification with message length insertion has a very significant impact (as evidenced by the higher PSNR value than LSB modification with the inverse bit insertion) both at text message insertion (66.29 dB> 61.8 dB) and image insertion (54.20 dB) > 50.01 dB). For further research, the authors plan to compress the stegoimage to facilitate the sending of images containing secret messages. Experiments carried out on the condition that the decoded stegoimage must not delete messages with exactly the same order as the original message. Furthermore, a given stegoimage tolerance to noise will be tested.

REFERENCES

Al-Aidroos, N. M., Bahamish, H. A., 2019, Image Steganography Based on LSB Matching and Image Enlargement, First International Conference of Intelligent Computing and Engineering (ICOICE). IEEE

Bhardwaj, R., Sharma, V., 2016., Image Steganography Based on Complemented Message and Inverted bit LSB Insertion, *Procedia Computer Science,* vol. 93, pp.832-838.

Gunawan, I., Sumarno., 2018, *Penggunaan Algoritma* Kriptografi Steganografi Least Significant Bit *Untuk Pengamanan Pesan Teks dan Data Video*, *J-SAKTI,* vol.2, pp.57-65.

Hernandes, A., Hartini, Sartika, D., 2019. Steganografi *Citra Menggunakan Metode* Least Significant Bit (LSB) *Dan* Linear Congruential Generator (LCG), *JATWASI,* vol. 5, pp. 137-150.

Karma, I., G., M, 2020. Determination and Measurement of Color Dwassimilarity. *International Journal of Engineering and Emerging Technology,* vol. 5(1) pp.67-71.

Prabowo, D.A., Abdullah, D., Manik, A., 2018*, Deteksi Dan Perhitungan Objek Berdasarkan Warna Menggunakan* Color Object Tracking, *Journal Pseudocode*, vol.5, pp.85-91.

Sari, R.D., Siahaan, A. P. U., 2018, Least Significant Bit Comparison between 1-bit and 2-bit Insertion, *International Journal for Innovative Research in Multi Disciplinary Field*, vol. 4(10), pp. 110-113.

Shehab, J. N., Abdulkhadim, H. A., 2018. Image Steganography Based on Least Significant Bit (LSB) and 4-Dimensional Lu and Liu Chaotic System, *International Conference on Advanced Science and Engineering (ICOASE)*, pp. 274 - 279. IEEE

Shtayt, B.A., Zakaria, N. H. B., Harun, N. H., 2020, A Comprehensive Review of Spatial Domain-Based Image Steganography Techniques.*International Journal of Advanced Science and Technology*, vol. 29 (7), pp. 13311 -13329.

Singh, A., Chauhan, M., Shukla, S., 2018, Comparison of LSB and Proposed Modified DWT Algorithm for Image Steganography. *International Conference on Advances in Computing, Communication Control and Networking (ICACCCN2018)* pp. 889-893. IEEE

Sitompul, O.S., Naibaho, F.R., Situmorang, Z., Nababaan, E.B., 2018, Steganography with Highly Random Linear Congruential Generator for Security Enhancement. *International Conference on Informatics and Computing (ICIC).* IEEE.

Zyaraa. A. H., 2017, Suggested m.thod for encryption and hiding image using LCG and LSB, *Journal of University of Babylon for Pure and Applied Sciences* pp.1-16.