The International Journal of Engineering and Information Technology

journal homepage:www.ijeit.misuratau.edu.ly

MISURATA UNIVERSITY

Cryptography and Steganography Based on Hybrid Edge Detection and LSB Comparing with Traditional LSB

Yahia, F.
Elmergib University
ffyahia@elmegib.edu.ly

Abushaala, A.
Misurata University
a.mohamed@it.misuratau.edu.ly

Abstract - **Nowadays the rapid growth of internet and digital communications has been required to be protected from unauthorized users. It is important to secure the information transmitted between the sender and receiver over the communication channels such as the internet, since it is an open environment. This paper proposed an algorithm of two layers of security against attackers, which are cryptography and Steganography for sending information in secrete way. In cryptography, we are using affine hill cipher method; while in steganography we are using Hybrid edge detection with Least Significant Bit (LSB) to hide the message. We combined the two edge detectors which are canny and Laplacian of Gaussian to get the hybrid edge area in the cover image. Our paper shows how we can use edge pixels of the image to hide the bits of the secrete message using LSB method instead of using traditional LSB method. Grey scale images are used for our experiments and a comparison is developed with previous work based on based some performance measurements such as, PSNR (Peak Signal to Noise ratio), MSE (Mean Squared Error). The results indicate that, the image edge area could preserve the imperceptibility of the Stego-image by enhancing the quality of the Stego-image with acceptable values of PSNR and MSE; comparing to using the traditional LSB method. Our best results achieved for MSE and PSNR are 0.0003 and 82.2962 respectively as comparing to traditional LSB which are 0.18 and 55.67 respectively. This paper also proved that the secrete message was extracted successfully without any distortion.**

Index Terms— **Edge detection; canny, Laplacian of Gaussian; Steganography; Cryptography; LSB; Embedding.**

I. INTRODUCTION

Depending most of the organizations nowadays on the information system, gives the opportunity for the competitive organizations to get access to the information system of other organizations [1] [2]. Attackers sometimes attempt to get sensitive information and make use of it such as, telephone conversation, files transferred or electronic email messages; without the intention of altering the system itself. Therefore, the security of these kinds of information should be ensured; especially when this information transmitted through the web by using either emails or social network [1] [2] [11]. Cryptography and steganography are the two most popular ways of sending information in secrete way [1] [2] [11].

Cryptography is called the art of the secrete writing [3] [11]. It distorts or scrambles the message itself by transforming the data (plain text) into unreadable format (cipher text) using encryption function and key [1]. In order to get the information back to the receiver, some decryption function and key is used [1]. Hence, only the authorized person can see and get the message which can be text, image, video or audio.

Steganography comes from the Greek words “Stegos” and “grafia” which mean “covered writing” [1] [2] [3]. Steganography was practiced since long time ago when the Greek historian Herodotus wrote of a nobleman, Histaeus who wanted to communicate with his son-in-law in Greece. He tattooed the message onto a slave’s scalp after shaving his head and then the slave was send with the hidden message when his hair grew back [1]. It is the science of hiding the existence of data inside a suitable carrier object which can be image, text, audio or video [3]. The data hidden can be text, video, audio or image, for example hiding text message inside an image will produce an output image called Stego-image [1] [2]. Hence, if any person sees the cover object, he cannot investigate that there is hidden information inside. The main objective of using steganography techniques is to provide more security for sensitive information.

Received 29 Apr , 2024; revised 11 May, 2024; accepted 15 Mar 2024.
Available online 08 Aug, 2024

II. LITERATURE REVIEW

In recent times, the entire world became connected by networks. So, protecting the information is becoming a significant issue. In fact, a lot of efforts have been made by the researchers around the world to discover and develop ways to keep the information safe. In this section, we will review some studies that used cryptography and steganography for securing data.

A new technique was developed which steganography and cryptography were integrated in order to make highly secured system for data transmission. Rail fence cipher technique was used for cryptography and DCT for Steganography. The system was evaluated by measuring PSNR to the **S**tego-image and the result showed that the PSNR values are much greater than 36 dB; which emphasis the suitability of the proposed system [1].

A new approach developed for hiding secrete data in any type of media such as text, audio or video by taking the benefits of mixing both cryptography and steganography together in one system. First, they encrypted the message using AES algorithm, then they applied the LSB method after making some modification to it, by adding the secrete key in order to make the hiding process non sequential. The results showed that the proposed algorithm has increased the security level [3].

A hybrid Canny-Sobel edge detection method was applied for hiding secrete message in order to get large embedding area [8]. The combination of the two detectors canny and Sobel, was conducted using OR operation. After that the message was embedded in the edge area using LSB method. The result showed that, using both detector canny and Sobel could efficiently increase the edge areas for higher payload capacity of messages and high PSNR for the Stego-image.

In [9] a system was proposed that takes the advantage of combining both techniques cryptography and steganography. Text message was encrypted by using Advanced Encryption Standard algorithm (RSA) and then it was embedded in a 24-bit color image using LSB techniques. The system could provide more security against the attackers.

A new algorithm was suggested for securing data which consists of two layers, encryption and steganography. First they encrypted the message using a secrete key and XOR operations represented in binary and hiding the encrypted message inside a cover image using LSB technique. Some of the evaluations metrics were used such as MSE, PSNR Entropy and histogram analysis to prove the quality of the algorithm. Acceptable results could be achieved for highest values of PSNR and MSE were 55.67 dB and 0.18 respectively [11].

Based on the previous studies discussed above, our proposed algorithm will take the advantage of combining both cryptography and steganography to increase the level of data transmission security. The following section explains the methods and the materials used in this research.

III. MATERIALS AND METHODS

In order to complete and achieve the objectives of this paper a set of methods and well designed methodology has to be conducted. The following sections explain the methods, materials and the methodology used.

A. *Affine Hill Cipher*

Affine hill cipher is a symmetric key cryptography algorithm that was developed as an enhancement to the original hill cipher by mixing it with a nonlinear affine transformation [4]. In fact, the weakness of any symmetric key cryptosystem is that, only a single key is required to bleak the encryption system. For example, in original hill cipher we need to get the key matrix and its inverse respectively for encrypting and decrypting the secrete message. But, the problem arises when the inverse of the key matrix does not exist. So, to overcome all of the above difficulties; affine hill cipher built the idea of taking a matrix as a key for the encryption but for the safety measures the key will not be revealed to the sender. The receiver will figure out the key, which is in fact the non-singular matrix that implements a reflection of key matrix of order 3 in an arbitrary line y= a x +b; the values of a and b are chosen over a set of numbers z_q that corresponding to set of letters, where q is a prime number. This equation will be provided by the sender to generate the key [4]. Hence, the encryption and the decryption formulas will be as following:

- **Encryption formula:**

$$C = PK + B \text{ (mod q)}$$

$$\left(P\begin{pmatrix} k_{11} & k_{12} & \dots\dots & k_{1n} \\ k_{21} & k_{22} & \dots\dots & k_{2n} \\ \dots & \dots & \dots\ \dots & \dots \\ \dots & \dots & \dots\ \dots & \dots \\ k_{n1} & k_{n2} & \dots\dots & k_{nn} \end{pmatrix} + B\right)\text{(mod q)}$$

- **Decryption formula:**

$$P = (C - B)\ K^{-1} \text{ (mod q)}$$

$$P = (C - B)\begin{pmatrix} k_{11} & k_{12} & \dots\dots & k_{1n} \\ k_{21} & k_{22} & \dots\dots & k_{2n} \\ \dots & \dots & \dots\ \dots & \dots \\ \dots & \dots & \dots\ \dots & \dots \\ k_{n1} & k_{n2} & \dots\dots & k_{nn} \end{pmatrix}^{-1} \text{(mod q)}$$

Where K is the key matrix, P is the plain text, C is the cipher text and B is row vector over the set $\mathbf{z_q}$ with the formula
B = (a b 1). It should satisfy gcd (det K (mod q), q) = 1.

Generation of the Key Matrix with Order 3*3

In order to generate a matrix of order 3 that reflects in an arbitrary line = ax + b (non-singular matrix) the following transformation must be followed [4]:

- Translate by (0,-b) so that the line y = ax + b maps to y = ax.

- Reflect through the line y = ax using the householder transformation
- M = I_3 - 2n.n^T

  $$where\ n = \begin{pmatrix} \frac{a}{\sqrt{a^2+1}} \\ \frac{-1}{\sqrt{a^2+1}} \\ 0 \end{pmatrix} n^T = \begin{pmatrix} \frac{a}{\sqrt{a^2+1}} & \frac{-1}{\sqrt{a^2+1}} & 0 \end{pmatrix}$$

  such that $n\, n^T = 1$

  So $M = \frac{1}{a^2+1}\begin{pmatrix} 1-a^2 & 2a & 0 \\ 2a & 2a & 0 \\ 0 & 0 & a^2+1 \end{pmatrix}$

- Translate by (0, b) to undo the earlier translation. Then the key matrix

$$K = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{pmatrix} \frac{1}{a^2+1} \begin{pmatrix} 1-a^2 & 2a & 0 \\ 2a & a^2-1 & 0 \\ 0 & 0 & a^2+1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix}$$

$$K = \frac{1}{a^2+1}\begin{pmatrix} 1-a^2 & 2a & -2ab \\ 2a & a^2-1 & 2b \\ 0 & 0 & a^2+1 \end{pmatrix}$$

So to generate the matrix key for the encryption process, we need to follow the following steps [4]:

- **Step1**: input the key values of a, b $\in z_q$ such that (greatest common divisor) gcd (a^2+1,q)=1
- **Step2**: compute the value of the key

$$K = \frac{1}{a^2+1}\begin{pmatrix} 1-a^2 & 2a & -2ab \\ 2a & a^2-1 & 2b \\ 0 & 0 & a^2+1 \end{pmatrix}$$

Note: if the values of a, b are any integers from the set z_q, then the key matrix is a non-singular. If the values of a, b equals to zero, then K = I_3 where I is the identity matrix of order 3 and q is a prime number [4].

For example: if the input values a = 1, b = 1 and q = 29, hence (gcd) (2,29) = 1. The value of the key is:

$$K = \frac{1}{2}\begin{pmatrix} 0 & 2 & -2 \\ 2 & 0 & 2 \\ 0 & 0 & 2 \end{pmatrix}$$

Algorithm of Affine Hill Cipher Method

The algorithm of affine hill cipher method consists of two tasks, one is the encryption task and the other one is the decryption which is the inverse process of the encryption [4]. Table 1 consists of numbers which are used to replace all the letters of the plain text from A to Z with some of other special characters.

Table (1): The Substitution of the Letters and Characters

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
0	1	2	3	4	5	6	7	8	9
K	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**
10	11	12	13	14	15	16	17	18	19
U	**V**	**W**	**X**	**Y**	**Z**	**!**	**?**	**Space**	
20	21	22	23	24	25	26	27	28	

Encryption Algorithm:

- Break up the plain text into blocks of order 1 * 3. For example if we want to encrypt the message HELLO SIR, we will first replace all the letters H, E, L, L ,O, ,S, I, R with the numbers 7, 4, 4, 11, 14, 30, 18, 8 , 17 respectively according to the numbers given in Table 1. Hence, the blocks of the plain text will be (7 4 4), (11 14 30) and (18 8 17) respectively. If the length of the plain text is not multiple of three, then we will add spaces to the plain text to convert it to multiple of three.
- Generate the key matrix and get the row vector from the equation y = ax + b.
- Determine the cipher text from the equation C = (PK+B) (mod q).
- Write and save the cipher text.

Decryption Algorithm:

- Input the keys **K, B**.
- Determine the plain text from the equation **P = (C-B) K^{-1}** (mod q).
- Write and save the plain text.

B. Least Significant Bit (LSB)

Least Significant Bit (LSB) is a spatial domain steganography technique, which deals directly with the pixels of the image. LSB is based on replacing the smallest bit value of the image pixels with the bit value of the secrete message, this operation is performed in sequential or random order [5]. For example, if a message with a value 10 to be inserted in an image with values {250, 120, 80, 175}; the following steps must be taken [5][12]:

Step1: convert the pixel values of the carrier image into bit numbers, which either 0 or 1.
Cover image: {250, 120, 80, 175}
Image in binary: {1111101**0**, 011110**0**, 0101000**0**, 101011**1**}

Step2: convert the message to be hidden into binary numbers. If the message is character, then it should be converted into ASCII code first and then convert it to binary number. But if the message is decimal number, it will be converted to binary format.
Message: {10}
Message in binary: {1010}

Step3: replace the least significant bit of each image pixel value with each bit value of the message.
Cover image: {1111101**0**, 0111100, 0101000**0**, 101011**1**}
Stego-image: {1111101**1**,011110**0**,0101000**1**, 101011**0**}

Step4: Finally, convert back the bits values of the Stego-image pixels into decimal numbers.
Decimal values of Stego-image: {**251**, 120, **81, 174**}

From the above steps, we notice that there is a small change in the pixel value, {250, 120, 80, 175} to {**251**, 120, **81, 174**}.

C. Canny Edge Detector

Canny edge operator is one of the most widely used methods for detecting edges in an image that was invented by John Canny in 1986 [6]. It is commonly used for detecting edges with small errors in various image processing applications and computer vision [6]. Canny detector depends on a threshold value that determines which pixel should be discarded as noise and which pixel should be consider as a part of the edge [6].

D. Laplacian of Gaussian Edge Detector(LoG)

Laplacian operator calculates the second spatial derivative which passes through zero value of an image, where a rapid change in the intensity value of the pixel occurs. Therefore, this detector could be able to detect edges of an image that has been smoothed using Gaussian filter to reduce its sensitivity to noise [7]. That is why it is called Laplacian of Gaussian, since it combines both the Laplacian of the image for edge detection and the Gaussian filtering for decreasing noise as mentioned in [7]. In another words, the zero crossing detector looks for places where the second derivatives of an image passes through zero value [7]. This means, the points where the sign of the second derivatives change, usually occurs at the edges of the image [7].

IV. METHODOLOGY

Our proposed algorithm will be tested by using three different sizes of grayscale images as shown in Figure 1 below.

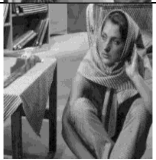
Image Name	Image Sample	Image Size
Lena.jpg		255 x 255
Barbara.jpg		512 x 512
Man.jpg		1024 x 1024

Figure 1. Cover Image Samples

These images are used as a cover media for hiding the secrete message. For our experiments, we use messages with different lengths such as 200 bytes, 500 bytes and 1000 bytes to test if the cover image could keep its quality even with high message payload capacity.

Actually, we are using the same sample and sizes of the images that were used in a previous study [10]; in order to compare our results with it.

First of all the message will be encrypted using affine hill cipher technique. After that, hybrid edge detection technique will be applied to the image, in order to get the edges of the image. These edge pixels are the positions where the encrypted message will be hidden using the technique of LSB.

The proposed algorithm is divided into five processes, which are encryption process, Hybrid edge detection process, hiding process, extraction process and decryption process. All these processes are represented in the following figure.

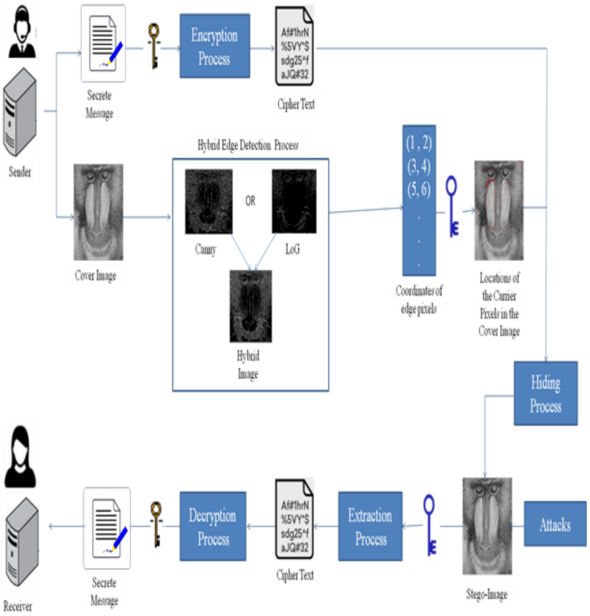


Figure 2. Methodology of combining cryptography and steganography

E. Encryption Process(Affine Hill Cipher)

Encryption process is used to encrypt the secrete message (plain text) into cipher text. The encryption method used in this research is called Affine Hill Cipher, which has been used in previous study has achieved a good result in terms of providing more security against attackers [4]. However, a small modification will be made by us, which is including numbers and some special characters. The total number of characters in substitution table is q which equal to 41 which is must be a prime number as shown in Table 1.

Table 1. The Substitution of the Letters and Characters

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
0	1	2	3	4	5	6	7	8	9
K	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**
10	11	12	13	14	15	16	17	18	19
U	**V**	**W**	**X**	**Y**	**Z**	**!**	**?**	**%**	**$**
20	21	22	23	24	25	26	27	28	29
SPACE	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
30	31	32	33	34	35	36	37	38	39
9	**10**								
40	41								

The following steps show the encryption process:

Step1: The sender will provide the secrete message (plain text).

Step2: Give two values a, b by the sender in order to generate the key matrix **K** and the row vector **B** for the encryption process.

Step3: Encrypt the plain text using the keys **K** and **B** for the encryption algorithm (Affine Hill Cipher).

Step4: Display the output which is called cipher text or encrypted message.

F. Hybrid Edge Detection Process

This process is based on generating hybrid edge detection image by combining both detectors (LOG and Canny) through using an OR operation. The following steps explain how to conduct this process:

Step1: Read the cover image (grayscale image).

Step2: Apply the Canny and Laplacian of Gaussian detectors separately on the cover image and save each result at the variables **C** and **L** respectively.

Step3: Conduct the OR operation on both result images **C** and **L** to produce a hybrid edge detection image **H** as in equation (1). The result of edge detection image is in binary format, where "1" refers to edge pixel value and "0" as a non-edge pixel value.

$$H= C \| L$$

Step4: Save the edge pixels coordinates as a key for the embedding and the extraction processes, in order to determine the locations of the carrier pixels in the cover image.

G. Hiding Process

During this process, the encrypted message is embedded inside the cover image specifically in the edge area pixels which is determined in the previous process. The inputs required for this process are the cover image (grayscale image), Stego-key which is the saved hybrid edge area, and the encrypted message. The details for this process are explained below.

Step 1: Read the cover image.

Step 2: Read the Stego-key, which is the coordinates of the edge pixels to hold the secrete message.

Step 3: Change the secrete message which has been encrypted in the previous encryption process into binary format according to the ASCII code.

Step 4: Reshape the binary message into vector form where each character is represented by 8 bits.

Step 5: Measure the length of the message if it is enough to be inserted to the edge area pixels. Otherwise, the system will provide a notification.

Step 6: Perform the embedding process which is the Least Significant Bit algorithm (LSB), where each bit of the secret message is embedded to the 8th bit of the edge pixel. We repeat this process until all the binary bits of the message are embedded.

Step 7: Save the image that will be resulted from step 6, which is called a Stego-image.

B. Extraction Process

The main idea of this process is obtaining the message perfectly. The required inputs for this process are the Stego-image and the Hybrid edge area pixels coordinates as a key for the extraction. Here are the steps required:

Step 1: Read the Stego-image.

Step 2: Read the Hybrid edge area pixels coordinates as a key for the extraction.

Step 3: Get each edge pixel value and convert it into binary form.

Step 4: Get the least significant bit of each edge pixels area sequentially.

Step 5: Combine each 8 least significant bits of each edge pixel to form one character.

Step 6: Repeat step 5 until we get the final encrypted message.

C. Decryption Process

Finally, this process comes to decrypt the cipher text in order to get back the original text message. The steps required for this process are as the following:

Step 1: Get the Cipher text.

Step 2: Input the keys **K** and **B** for the decryption.

Step 3: Apply the reverse of the encryption algorithm (Affine Hill Cipher) to get the plain text.

Step 4: Finally, the secrete message is read by the receiver.

V. RESULTS AND DISCUSSION

To achieve the goal of this paper, we have developed a prototype that implements our proposed algorithm. For our experiments, we use messages with different lengths such as 200 bytes, 500 bytes and 1000 bytes; to test if the cover image could keep its quality even with high message payload capacity. The quality measurements used are PSNR and MSE which are calculated using the following formulas [10]:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Where m and n is the width and the height of the original matrix image I (i, j) and K is i*j matrix of Stego-image.

$$PSNR = 10.log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

Where MAX is the maximum pixel value of original image I which is equals to 255 for grayscale image.

In addition, the efficiency of our proposed algorithm will be demonstrated by proving that the algorithm can get a perfect message extraction.

From Figure 3 we can see that, there is a noticeable difference between the edge areas detected using different detectors. The edge area obtained by using hybrid technique is clearer than using the others techniques. However, the edge area obtained by applying the canny detector still has more details or edges than using the Laplacian of Gaussian detectors (LoG).

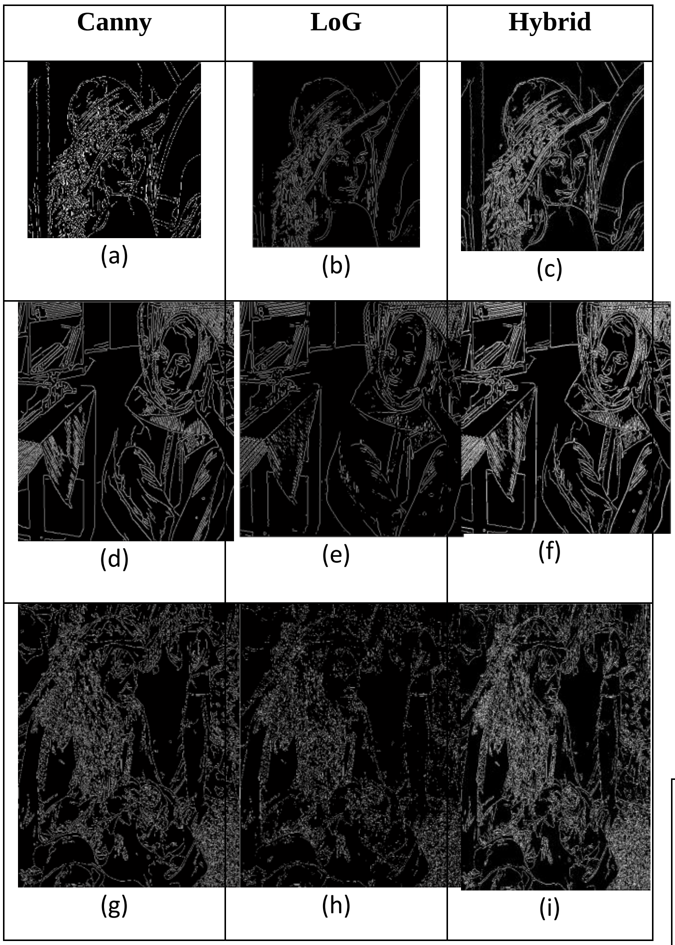


Figure 3. Edge areas of the cover images {(a) original image Lena; (b) Lena-canny; (c) Lena-LoG; (d) Lena-Hybrid (canny-LoG); (e) original image Barbara; (f) Barbara -canny; (g) Barbara -LoG; (h) Barbara -Hybrid (canny-LoG); (i) original image Man; (j) Man -canny; (k) Man -LoG; (l) Man -Hybrid (canny-LoG) }.

Table 2. Number of the edge area pixels based on different edge detection techniques (Canny-LoG-Hybrid)

Image	Number of edge area pixels		
	Canny	LoG	Hybrid
Lena (255 x 255)	6241	4170	8656
Barbara (512 x 512)	28242	17462	38385
Man (1024 x 1024)	107118	67778	146136

The above Table shows the calculations results which determine the number of the edge pixels for each detector. we can also conclude that, the hybrid technique could increase the number of the edge pixels compared to using canny and LoG and separately. Figure 4 show that, our hiding technique (Canny-LoG) in our proposed algorithm has got higher payload capacity.

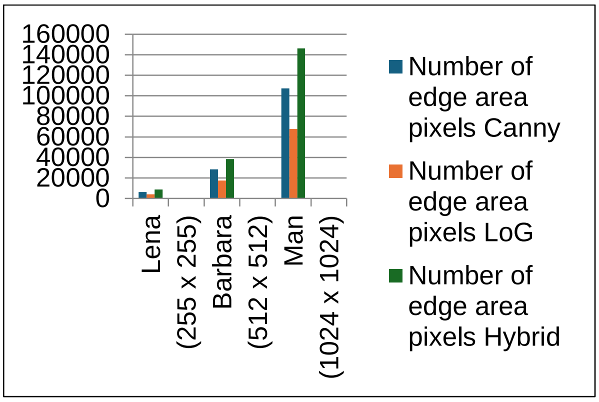


Figure 4. The Number of the Edge Area Pixels Using Different Edge Detectors (canny- LoG- hybrid)

In order to evaluate our proposed algorithm, we need to compare with methods that have been done before. The following Table 3 displays the comparison results that based on the values of the quality measurements PSNR and MSE.

Table 3. Comparison between proposed method and Existing method

Image	Message Length (in bytes)	Proposed Algorithm (Hybrid edge detection with LSB)		Previous Work [10]	
		MSE	PSNR	MSE	PSNR
Lena (255 x 255)	200	0.0058	70.4978	5.52	40.74
	500	0.0308	63.268	13.38	36.90
	1000	0.0620	60.235	26.36	33.95
Barbara (512 x 512)	200	0.0016	75.9454	0.78	49.23
	500	0.0075	69.366	1.96	45.23

	1000	0.0150	66.3787	4.14	41.99
Man (1024 x 1024)	200	0.0003	82.2962	0.18	55.67
	500	0.0018	75.3866	0.42	51.95
	1000	0.0038	72.3632	0.85	48.86

The following Table 4 shows a sample of extraction and decryption results from a Stego-image Lena. That means, our proposed algorithm could success through a perfect message extraction and decryption. In another words, there is no missing information at all while performing our algorithm.

Table 4. Sample of extraction and decryption results from a Stego-image (Lena)

Original Message	Message Length	Message extraction	Message Decryption
Password is 9?!9…	200 bytes	Dbkfjsef….	Password is 9?!9…
If you receive the letter I send you please call me on 926658745 …	500 bytes	gjypzls!=fdlfwkiu:m!7uuf!sst!>eorpzlq!bbfp!fzmbxn!8p!c:!l776894!6g….	If you receive the letter I send you please call me on 926658745 ……
Social Media is a prevalent medium ….	1000 bytes	pt@bjen!8jegj!bb!!7uuf!sst!>eorpzlq!bbfp!fzmbxn......	Social Media is a prevalent medium ….

VI. CONCLUSION

To improve the stage of data security though transmission, combining both hiding techniques such as cryptography and steganography can increase the layer of securing sensitive data. Since, when the data is hidden in a good way, attackers cannot realize that there is hidden information inside it. In addition, the hidden data is encrypted which make it difficult to get the original message. In this paper, we used Affine Hill Cipher technique to encrypt the message which will be hidden. Besides that, we combined two edge detectors which are canny, LoG and use the LSB technique to hide every bit of the secrete message. The results proved that, using the edge pixels of the image to hide messages bits, could maintain the Stego-image quality with high values of PSNR and high embedding capacity. Since the edge area of an image can tolerate any change in the image. We found out that, using our proposed algorithm for hiding data gives a better result than using the traditional LSB such as a previous work [10]; in terms of the values of MSE and PSNR.

REFERENCES

[1] Khalid I. R., Amit K. G., Manisha M.2015 .Study of Cryptography and Steganography System. International Journal of Advanced Trends in Computer Science and Engineering, Vol. 4, Issue 8, p.p. 13685-13687.

[2] Monika and Sudhir Y.2016.Data hiding using Cryptography and Steganography. International Journal of Enhanced Research in Science Technology & Engineering. ISSN: 2319-7463. Vol. 5 Issue 4.

[3] Ahmed S., Talal A.2017. Cryptography and Steganography: New Approach Transactions on Networks and Communications. Vol. 5, No. 6, p.p. 25-38.

[4] M.G.Vara Prasad, P.Pari Purna Chari, K.Pydi Satyam. 2016. Affine hill cipher key generation matrix of order 3 by using reflects in an arbitrary line y=a x+ b. International Journal of Science Technology and Management Vol.5, Issue 08.

[5] Smitha G., Baburaj E.2018. Sobel edge detection technique implementation for image steganography analysis. Biomedical Research. Special Issue: S487-S493.

[6] Yingke F., Jinmin Z., and Siming W.2017. A ne w edge detection algorithm based on Canny idea. 2nd International Conference on Materials Science, Resource and Environmental Engineering.

[7] Lital B., and Rostislav P.2010. Digital Image processing Chapter 10 Image segmentation.Website:http://www.cs.bgu.ac.il/~klara/ATCS111/gonzales_10.1_10.2.pdf.

[8] De Rosal I. M. S., Jumanto J. 2018. An Enhanced LSB-Image Steganography Using the Hybrid Canny-Sobel Edge Detection. Vol. 18. No2. Print ISSN: 1311-9702; Online ISSN: 1314-4081.

[9] Varsha, Dr. Rajender S., C.2015. Data Hiding Using Steganography and Cryptography. International Journal of Computer Science and Mobile Computing, Vol.4 Issue.4. p.p. 802-805.

[10] Ali A., and Abdulmotalib A.2020. A Secure Image Steganography using LSB and Double XOR Operations. IJCSNS International Journal of Computer Science and Network Security. VOL.20 No.5.

[11] Sultana O., Mohammed M., Farij, Evaluation of using Steganography technique to hide text in grayscale digital images. Journal of Academic Research , VOL 19, July 2021.

[12] Tutuk I., et al,. Steganography on Color Images using Least Significant Bit Method. ICONNSMAL pp. 39-48, 2023.