Abstract – In this paper a process to Increase security of hidden data in Image and Prevent data extraction has been presented. We will encrypt data by use data hiding key and hide data using LSB technique. In the LSB approach, the basic idea is to replace the Least Significant Bits (LSB) of the cover image with the Bits of the messages to be hidden without destroying the property of the cover image significantly. The LSB-based technique is the most challenging one as it is difficult to differentiate between the cover-object and stego-object if few LSB bits of the cover object are replaced. By using the key, the chance of getting attacked by the attacker is reduced.

Keywords- GUI, LSB, PSEUDO, PSNR, SNR

I. INTRODUCTION

Steganography is derived from the Greek word steganographic which means covered writing. It is the science of secret communication. The goal of steganography is to hide the existence of the message from unauthorized party. The modern secure image steganography presents a task of transferring the embedded information to the destination without being detected by the attacker. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet. For hiding secret information in images, there exist a large variety of steganographic techniques some are more complex than others and all of them have respective strong and weak points. In this paper we purposed an image based steganography that Least Significant Bits (LSB) techniques and pseudo random encoding technique on images to enhance the security of the communication. In the LSB approach, the basic idea is to replace the Least Significant Bits (LSB) of the cover image with the Bits of the messages to be hidden without destroying the property of the cover image significantly. The LSB-based technique is the most challenging one as it is difficult to differentiate between the cover-object and stego-object if few LSB bits of the cover object are replaced. In Pseudo-Random technique, a random-key is used as seed for the Pseudo-Random Number Generator is needed in the embedding process. Both the techniques used a stego-key while embedding messages inside the cover image. By using the key, the chance of getting attacked by the attacker is reduced.

To simplify the process we have designed the technique in graphical user interface (GUI). By using GUI an encoder does not have to learn the program of the technique or to rewrite the code for it, user can simply do this by GUI technique. [5]

II. RESULTS AND ANALYSIS

- Data hide in image1

Data is hiding in the image1 with the size of 16.5 kb and pixel 305*165.

![Fig 1: Data hide](image1.png)
The fig 1 shows that data hiding successful in input image.

![Fig 2: Data retrieve successfully](image2.png)
The fig 2 shows that hidden data is retrieved successfully from the encrypted image.

![Figure 2: Histogram of image 1](image)

The fig 3 shows the histogram of the input image. We calculate the different noises for image 1 that affect the quality of image. The Peak-SNR value by salt and pepper noise is 21.4862 and the SNR value is 14.1796 and the Peak-SNR value by poisson noise is 29.1308 and the SNR value is 21.8243.

- Data hide in image 2

Data is hiding in the image1 with the size of 11.7 kb and pixel 236*213.

![Figure 4: Data hide in image 2](image)

The fig 4 shows that data is hide successfully in image 2.

The fig 5 shows that data is successfully retrieve from the encrypted image.

![Figure 5: Data successfully retrieve](image)

We calculate the different noises for the image 2. The values of the Peak-SNR value by salt and pepper noise is 20.8544 and the SNR value is 17.3141. The Peak-SNR value by poisson noise is 28.0250 and SNR value is 24.4847.

- Image hide in image 3

Data is hiding in the image1 with the size of 6.15 kb and pixel 294*171.
The fig 7 shows that data is hide successfully in image 3.

The fig 8 shows that data is successfully retrieve from the encrypted image.

The fig 9 shows the histogram for the image 2.

We calculate the different noises for the image 2 and these noises are:
- The Peak-SNR value by salt and pepper noise is 22.0686 and the SNR value is 18.1186
- The Peak-SNR value by poisson noise is 26.5308 and the SNR value is 22.5808.

- Data hide in image 4

Data is hiding in the image 1 with the size of 6.15 kb and pixel 220*229

The fig 10 shows the data hide in image 4 successfully.

The fig 11 shows that data is successfully retrieve from the encrypted image.
The fig 12 shows the histogram for the image 2.

We calculate the different noises for the image 2 and these noises are:
The Peak-SNR value by salt and pepper noise is 20.3643 and the SNR value is 19.3867.
The Peak-SNR value by poisson noise is 27.1010 and the SNR value is 26.1234.

- Data hide in image 5

Data is hiding in the image 1 with the size of 4.02 kb and pixel 225*225

The fig 13 shows that data is hide successfully in image 5.

The fig 16 shows the histogram of image 5.

The Peak-SNR value by salt and pepper noise is 20.6723 and the SNR value is 19.6854.
The Peak-SNR value by poisson noise is 26.5873 and the SNR value is 25.6004.

V. CONCLUSIONS

Steganography is an effective way to hide sensitive information. In this paper we have used the LSB Technique on images to obtain secure stego-image. Our results indicate that the LSB insertion after encryption of Data for various size of images gives better results. The image resolution doesn't change much and is negligible when we embed the message into the image but image size will increase...
because of data hiding. So, it is not possible to damage the data by unauthorized personnel. Overall we can conclude that data security has been improved as attacker cannot extract encrypted Data.

We also calculate the PSNR and SNR ratio for pepper and salt noise and poisson noise for different size of images. We conclude that psnr and snr may varies with the size of image and every image have different psnr and snr values for each noise.

IV. REFERENCES


