A Report on

Implementation of LSB Steganography using AES

By

Shambwani Rohit Pradeep	2021H1030055G
Siddharth Vinze	2021H1030024G
Himanshu Kumar Suman	2021H1030066G
Talati Rathin Nikhilkumar	2021H1030046G

Under the Guidance of
Dr. Sanjay K. Sahay

Birla Institute of Technology & Science, Pilani
K K Birla Goa Campus
2021-22

ABSTRACT

Steganography is the art of concealing a message in another message which provides security through obscurity. It supplements cryptography as it doesn’t attract attention and provides secrecy as opposed to cryptography which provides privacy. In image steganography, we hide the important message to be sent in images through some techniques. The prominent techniques include hiding the data into the Least Significant Bit of an image done in linear form. But cracking this traditional LSB is possible due to the large number of tools available online. In this report, we present an improvement to traditional Least Significant Bit-based image steganography by using a Pixel Locator Sequence to randomize the pixels in which data is to be stored rather than storing it in the last bit linearly. This provides increased security from attackers. We use Advanced Encryption Standard (AES) to enhance data transfer security by encoding both the Pixel Sequence and the message and decoding it in a reverse way for getting the message or plaintext back to the receiver. Usage of AES will provide double encryption for both the pixel sequence and the message to be transmitted.

TABLE OF CONTENTS

INTRODUCTION

When we say steganography, the main objective is to hide the message and not to encrypt or decrypt it as is done in traditional cryptography. When steganography is done with images, we term it as Image Steganography, where the main objective is to hide the secret or confidential message in an image sent from sender to receiver. The project or the assignment comprises three main components: Least Significant Bit substitution, Advanced Encryption Standard, and Pixel Locator Sequence.

The Least Significant Bit substitution technique is one of the most popular techniques used in Image Steganography. It is a technique concerned with the spatial domain. Very simple and easy to implement, the technique is very popular as it is very deceptive to human eye where the user is unable to find the real difference between real image and stego image.The technique is extensible to higher number of least significant bits like 2,4 and 8 bits but as the number of bits that are to be substituted increases the noise in the image increases and we don't want that much of noise that clearly helps the attacker to easily distinguish between the two images.The traditional Least Significant Bit technique is very prone to attacking as attacker may be able to find the right data if it isn't encrypted properly.To improve this,the project or assignment uses AES technique to enhance the security of traditional techniques.

AES stands for Advanced Encryption Standard and is a block cipher. It comprises of 128-bit block size and key sizes which are 128,192,256 bits are different and used as per requirement.Block cipher relies on two of the main things that is confusion and diffusion.AES does it both with its four main layers which are Subbytes Layer,Shift Rows Layer,Mix Columns Layer and at last a round key is added in Round Key layer and this whole process is repeated for some number of rounds depending on the key size.The decryption process in AES is done in exact reverse order as the encryption process of four layers.In, traditional Least Significant Bit technique we know where the data is stored as it is done in a linear manner. So the assignment aims to introduce the Pixel Locator Sequence which will help determine the pixel in which data is to be entered or to be hidden.

Pixel Locator Sequence is a random generation of a pixel sequence which is generated uniquely for each image.A sender can also manually generate this sequence, It is by the help of this sequence we are able to randomize our confidential data across the image rather than traditional Least Significant Bit technique where data or message is spread linearly. This sequence also helps in the decoding process where it acts as a key for getting the plaintext back at the receiver's end. All of the above mentioned techniques are combined to give us the system that serves all in all an improved version of what exists in traditional techniques.The AES provides double encryption along with Pixel Locator Sequence that randomly distributes the the message instead of traditional way of iterating linearly

LITERATURE SURVEY

- Traditional LSB Techniques
  There exists two methods for hiding a message inside digital images which are as follows:
  - Least Significant Bit Method: In this case,the least significant bit of every pixel of the image is replaced by the bit of the message that is to be hidden.
  - Most Significant Bit Method: In this case,complementary to LSB technique the most significant bit of every pixel of the image is replaced by the bit of the message that is to be hidden.

- Methods of Steganography
  - One Bit Stego:In this case the resultant stego image is obtained by manipulation of one or more bits in a pixel of the initial image.
  - Two Bit Stego:In this case, manipulation of two Least Significant Bits of one of the colors involved in RGB pixels is done to hide the message bits.
  - Three Bit Stego:In this case, manipulation of three Least Significant Bits of one of the colors involved in RGB pixels is done to hide the message bits.
  - Four Bit Stego:In this case, manipulation of four Least Significant Bits of one of the colors involved in RGB pixels is done to hide the message bits.
  - Color Cycle Stego:This method increases the complexity of finding the hidden data by cycling through the values of colors in each pixel.

- Algorithm to embed the text in a color image using LSB Steganography
  - Step 1: Storing the pixels of an image into an image array by reading the image and transforming it.
  - Step 2: Convert the message that is to be sent into binary format
  - Step 3:Storing the binary format of the image into a message array.
  - Step 4:Replace the pixel chosen from the array of image with the characters from the message array and place it in the LSB of the pixel.
  - Step 5: Resultant image will be the image that will contain the hidden data.

- LSB Steganography using Secret Key
  - In this case, a secret key is added to the cover image to increase the security of the hidden information.
  - So the resultant steganographic image will comprise the initial image along with secret key and hidden text.
  - The secret key is converted into a circular array of 1 dimension bit stream.
  - The secret key along with the Red Matrix is used to make the decision for placing the hidden information into a green or blue matrix.

SCOPE

This project includes hiding the data in the image. But the difference is that it uses the Pixel Locator Sequence (PLS). The basic idea remains the same: use the least significant bit (LSB). Computing the data in a linear format is tweaked so that we get a robust security for our steganographic image. Encryption technique and a steganographic technique go hand in hand. And this project has used the Advanced Encryption Standard technique (AES) to encode the overall data after the data has been hidden in the image.

Now the steganography can be done in audio, text, video, images etc. but the scope of this project only covers the image part of the encoding and decoding. For the AES encryption technique we have used the python library instead of reinventing the wheel and also the AES is not in the scope of our project. The only focus of our project has been the LSB steganography using the PLS algorithm. We will also use a Graphical User Interface which will work as follows: takes the text from the user which he/she has to encode. Upload the image on which the encoding has to take place. Our algorithm will work in the backend and generate the required image and the secret data will be hidden in the form of noise generated in the image. The application of our algorithm is as follows:

- Confidentiality and secretive communication between two parties.
- Protection from the data alteration.
- Access control system for digital content distribution.
- Unifying two data into one in the media database system.
- Protecting the copyright content from getting viral on the internet.
- Used in the implementation of selective extraction.
- Protecting the fragility of the embedded data.
- Hiding data in the plain sight.

The use case of our project is as follows:

- Can be sold as a proprietary product for an organization where secretive information can be passed along the company’s top officials contained in an innocuous email or a flash message. In this way even if someone from the rival company tends to spy on the information; he/she won’t be able to retrieve the information. Decoding might be impossible as the hacker won’t be able to decode as he/she doesn’t know the LSB sequence.
- Another use case is copyright infringement. Professional photographers can encode the IP address into the image. So if the image is leaked and used inappropriately; the image can be decoded to extract out the IP address. This will give the photo owner the freedom to take the appropriate actions.

PROPOSED MODEL

Our project mainly focuses on enhancing the security of LSB-based steganography by using PLS during the encoding and decoding process of LSB so that text is hidden in a randomly distributed sequence of a pixel inside the image.

LSB-based steganography is done by iterating over the pixel in systematic order. PLS will now allow us to cover the pixel in systematic order. PLS will now allow us to encrypt the text before encoding it into the pixel. To make the decoding possible we will have to send this sequence along with the stego image. So, we must encrypt the PLS using AES before sending it over to the receiver. The receiver will receive a stego image along with an encrypted PLS file. At first, the PLS is decrypted using AES, and then our algorithm iterates over the PLS and from each pixel decodes the text which is hidden in the LSB of that pixel. The text that we get from this decoding is an encrypted text that further needs to be decrypted to get the original text.

The PLS created will be used to distribute the data through pixels across the image. Before iterating over the image pixels we need the information of the pixel in the i, j format because we consider an image to be made up of [N]×[M] matrix where N and M are whole numbers. This is easily achievable if we know N and M. Here N and M represent the number of pixels in the vertical direction and horizontal directions respectively. Let X be the location of the pixel from the PLS where the data has to be stored. The row and column value of X can be easily calculated using

$$row = X/M, \; column = X\%M$$

Algorithm: LSB encoding

```
Result: Text encoded into the image
image[][], PLS[], i = 0, textlist[];
while i ≤ Tn-1 do
    while k ≤ textlist[i].size() do
        text[] = textlist[i];
        pix [] = Append RGB values of next three pixels from PLS;
        while j ≤ 7 do
            if text[j]=='0' and pix[j]%2!=0 then
                pix[j]–;
                j++;
            else
                if text[j]=='1' and !(pix[j]&1)
                if pix[j]==0 then
                    pix[j]++;
                    j++;
                else
                    pix[j]–;
                    j++;
                end
            end
        end
        k++;
    end
    i++;
end
```

Algorithm: LSB decoding

```
Result: Extracting the encrypted text from image
image[][], PLS[], i = 0, data = ”;
while true do
    pix [] = Append RGB values of next three pixels from PLS;
    binstr = ”;
    while j ≤ 7 do
        if pix[j]%2==0 then
            binstr += ’0’;
        else
            binstr += ’1’;
        end
        j++;
    end
    data += char(ASCII(binstr));
end
```

IMPLEMENTATION

We mainly used Python as our programming language to implement the LSB encoding, LSB decoding, PLS and AES. Furthermore, we used the following python libraries:

1. **Pillow (a fork of PIL):** Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aid in editing, creating, and saving images.
2. **PBKDF2:** PBKDF2 is a simple cryptographic key derivation function, which is resistant to dictionary attacks and rainbow table attacks. It is based on iteratively deriving HMAC many times with some padding.
3. **PYAES:** A pure-Python implementation of the AES block cipher algorithm and the common modes of operation (CBC, CFB, CTR, ECB, and OFB).
4. **NumPy:** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
5. **OpenCV:** OpenCV is a huge open-source library for computer vision, machine learning, and image processing.

The steganography system should embed the secret message into the image such that the visual quality of the image is not perceptibly changed. So, we used Peak Signal to Noise Ratio(PSNR) and Mean Square Error(MSE) as our metrics.

A larger PSNR value implies lower distortion.

$$PSNR = 20.\log_{10}\frac{MAX_I}{\sqrt{MSE}}$$

MSE measures the average of the square of the ”error” with the error being the amount by which the estimator differs from the quantity to be estimated.

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

We have also optimized our algorithm for the LSB substitution and the PLS algorithm. To be specific, the complexity of randomness has been reduced. The time taken for the pixels to be randomly allocated throughout the pixels is now done in order of **O(n)**.

Whitebox and blackbox testing has been done on the pixel locator sequence. For the whitebox testing we took a variety of images of different extensions. Some of the images were corrupted (the reason for the corruption varied from incorrect storage of the images or manipulation of some of the bits in the image). The PLS worked as expected; for the incorrectly stored images the code throws an exception. For the manipulated bits, the images are properly encoded.

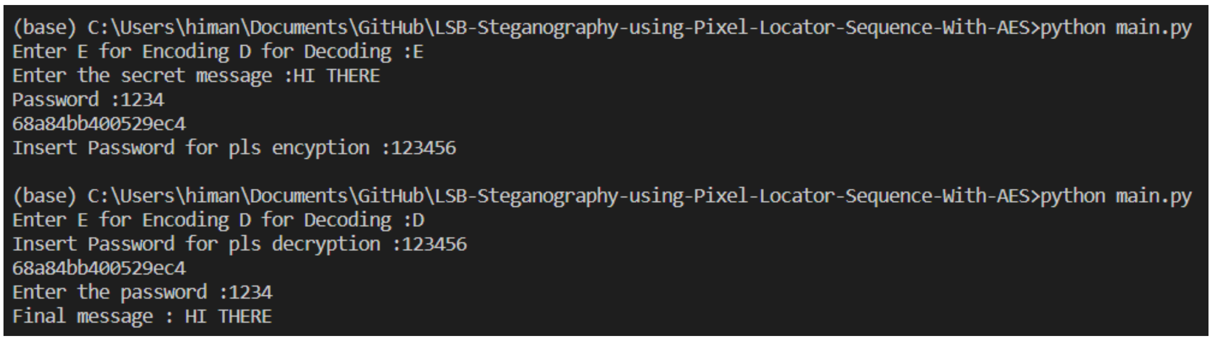
Created a single page application in which the user can upload the image and enter the text he wishes to enter in the image. In the same page the user will get the encoded image. This functionality will happen without reloading the page. This encoding will happen asynchronously and all the processing of the image will be done in the backend. The user can then download the encoded image and use it accordingly. This is basically a user interface which can be integrated with any application. Just for the purpose of this project, we've just kept it as a single page website.

RESULTS

Input Image:

Output Image:

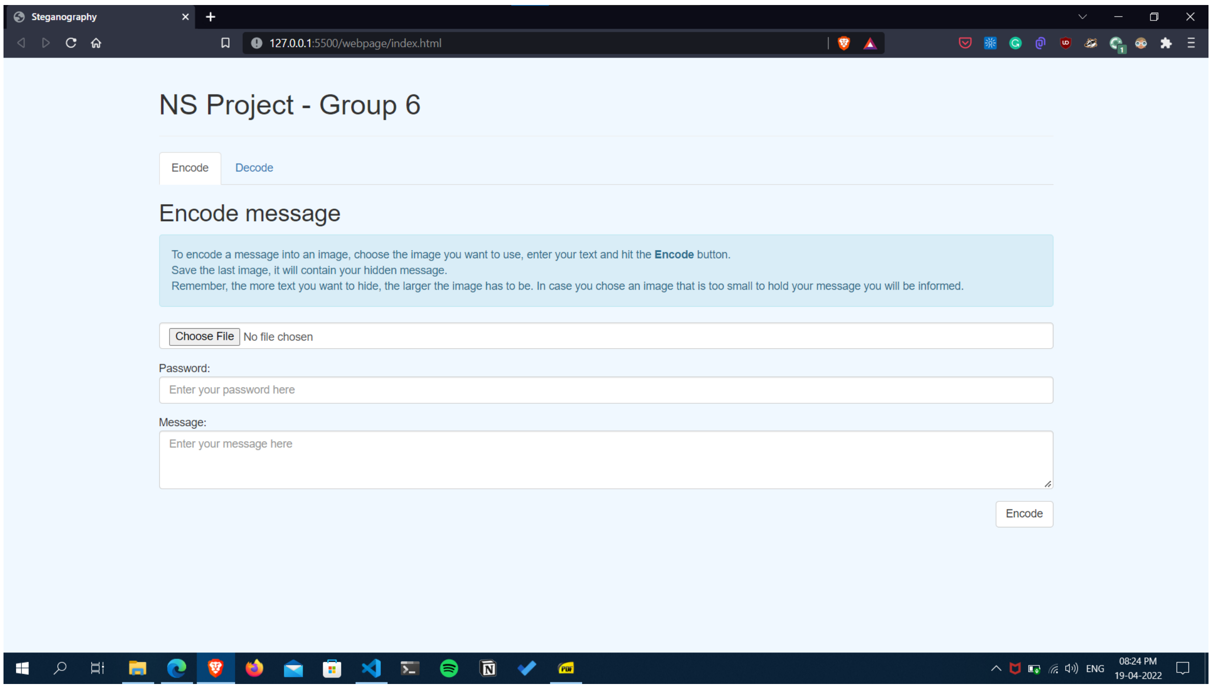
Code Output:



```
(base) C:\Users\himan\Documents\GitHub\LSB-Steganography-using-Pixel-Locator-Sequence-With-AES>python main.py
Enter E for Encoding D for Decoding :E
Enter the secret message :HI THERE
Password :1234
68a84bb400529ec4
Insert Password for pls encyption :123456

(base) C:\Users\himan\Documents\GitHub\LSB-Steganography-using-Pixel-Locator-Sequence-With-AES>python main.py
Enter E for Encoding D for Decoding :D
Insert Password for pls decryption :123456
68a84bb400529ec4
Enter the password :1234
Final message : HI THERE
```

Webpage Integration:



NS Project - Group 6

Encode | Decode

Encode message

To encode a message into an image, choose the image you want to use, enter your text and hit the **Encode** button.
Save the last image, it will contain your hidden message.
Remember, the more text you want to hide, the larger the image has to be. In case you chose an image that is too small to hold your message you will be informed.

Choose File No file chosen

Password:
Enter your password here

Message:
Enter your message here

Encode

REFERENCES

- S. M. Masud Karim, M. S. Rahman and M. I. Hossain, ”A new approach for LSB based image steganography using secret key,” 14th International Conference on Computer and Information Technology (ICCIT 2011), Dhaka, 2011, pp. 286-291, doi: 10.1109/ICCITechn.2011.6164800.
- [3] K. Joshi and R. Yadav, ”A new LSB-S image steganography method blend with Cryptography for secret communication,” 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, 2015, pp. 86-90, doi: 10.1109/ICIIP.2015.7414745.
- [4] O. Elharrouss, N. Almaadeed and S. Al-Maadeed, ”An image steganography approach based on k-least significant bits (k-LSB),” 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2020, pp. 131-135, doi: 10.1109/ICIoT48696.2020.9089566.
- [5] X. Li, B. Yang, D. Cheng and T. Zeng, ”A Generalization of LSB Matching,” in IEEE Signal Processing Letters, vol. 16, no. 2, pp. 69-72, Feb. 2009, doi: 10.1109/LSP.2008.2008947.