
Security & Forensic Lab-1 Report

Wednesday, 03.02.2021

P Rahul Bharadwaj, 17CS02003

Problem Statement

To write a program that hides the secret message in a cover image using Least Significant Bit (LSB) steganography Algorithm in such a way that it minimizes the loss in visual quality of the cover image after the message embedding is done.

Approach:

LSB Algorithm:

The idea behind LSB embedding is that if we change the last bit value of a pixel, there won't be much visible change in the color. This approach has the advantage that it is the simplest one to understand, easy to implement and results in stego-images that contain embedded data as hidden.

The disadvantage of Least Significant Bit is that it is vulnerable to steganalysis and is not secure at all. So as to make it more secure, the least significant bit algorithm is modified to work in a different way. This proposed approach simply does not pick up least significant bits of pixel in a sequence but it intelligently selects pixel positions for embedding the secret message.

Software Requirements

- open-cv
 - Python
 - Numpy
-

Encryption Algorithm

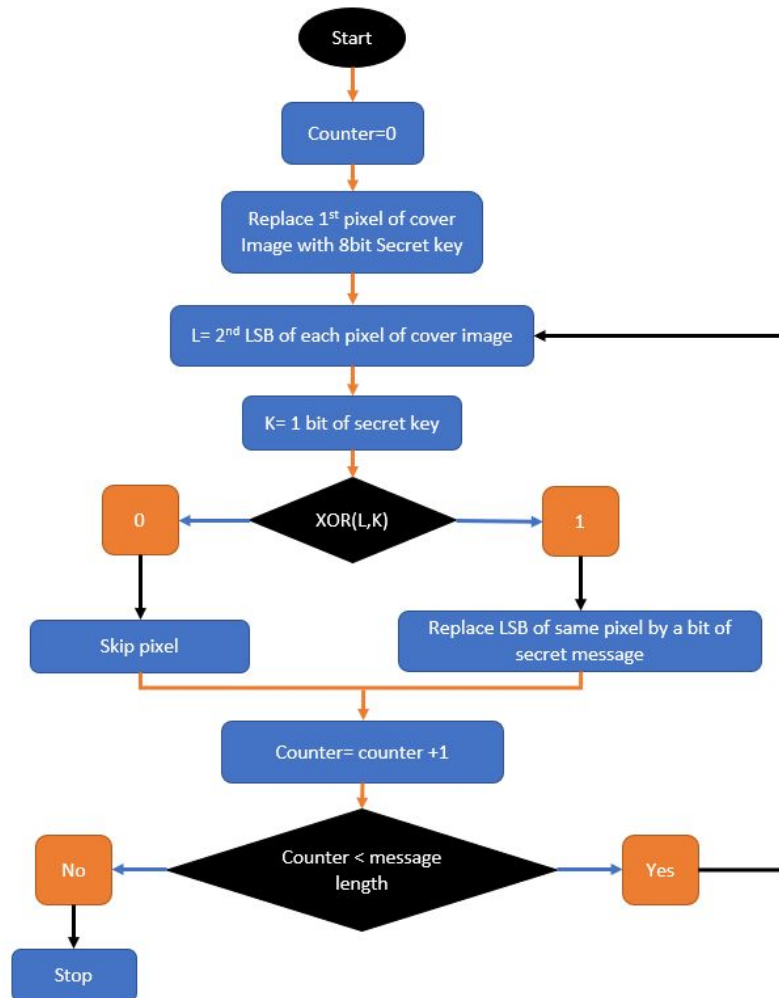


Fig.1 Encryption Algo for hiding the message

To hide secret text messages we have to take a cover image which is divided into three matrices (Red, Green, Blue) .Then we will use an 8-bit secret key. Here 8-bit secret key and second LSB of each pixel of cover image used for decision making to replace hidden information into cover image. In the next step we will apply XOR operation between one bit of 8-bit secret key and 2nd LSB of cover image pixel. If the XOR result of this operation is 1 then we will hide one bit of our secret key into the LSB of the same pixel of the cover image, Otherwise we will not hide any bit in that pixel. The substitution process will be continued depending on the length of the message.

Decryption Algorithm

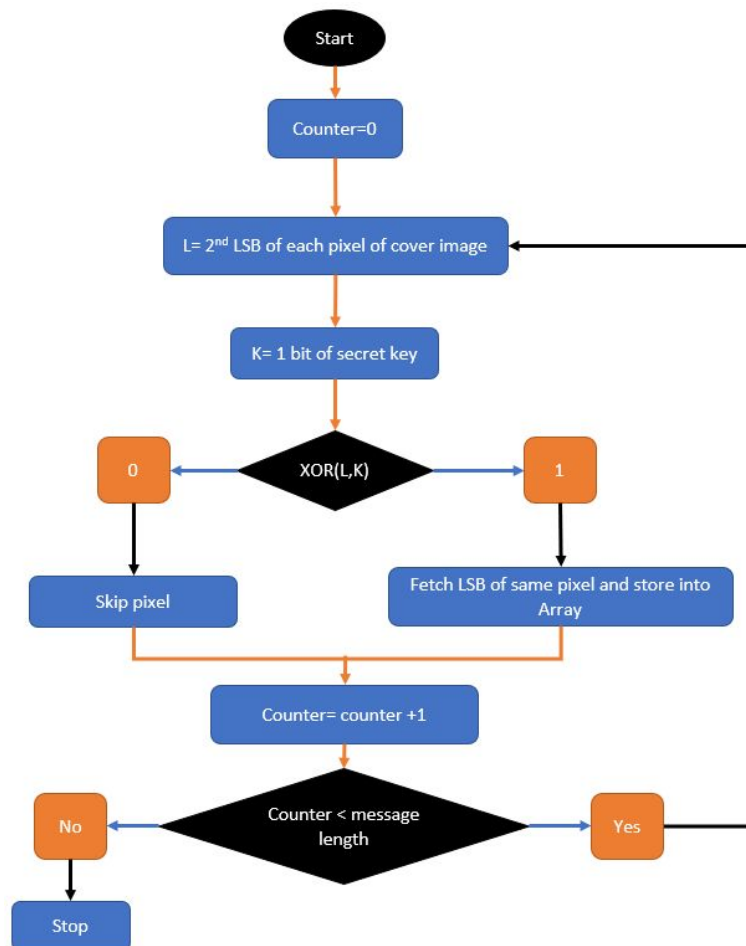
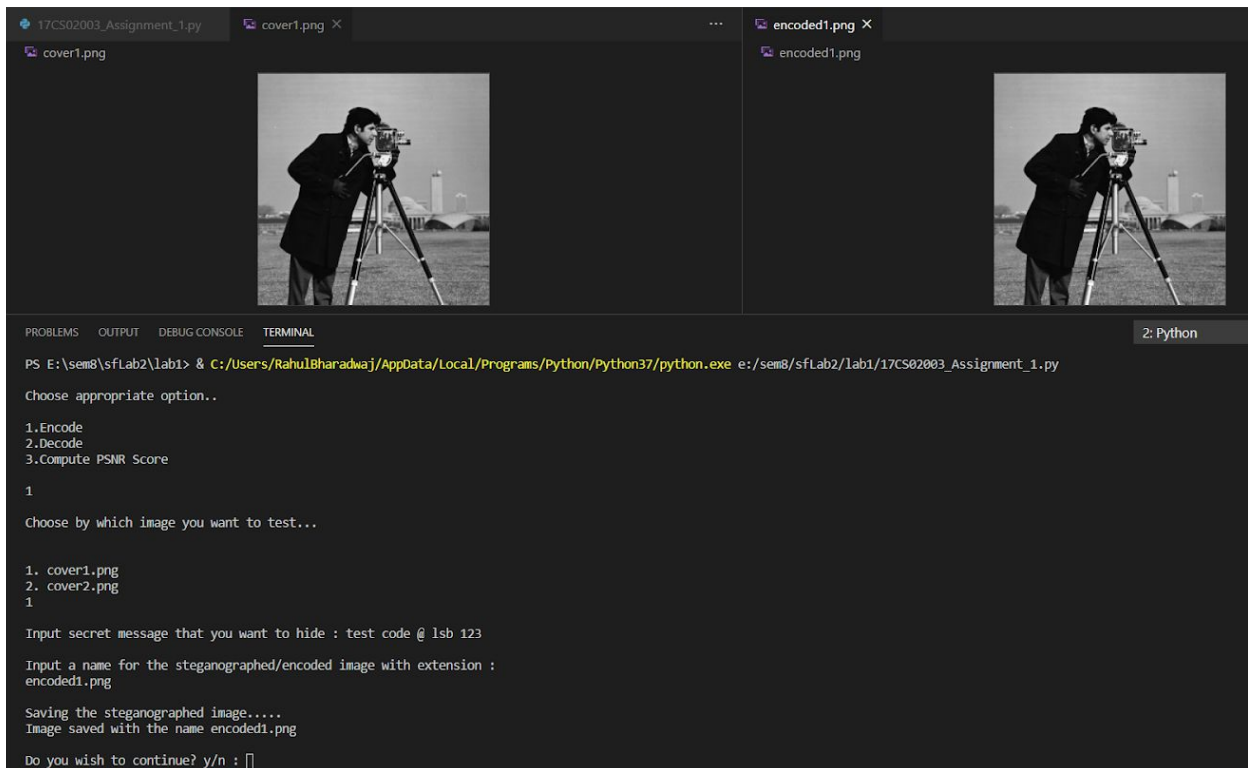


Fig.2 Decryption technique to retrieve hidden message

To recover the secret text message from the stego image, we have to take a stego image which is divided into three matrices (Red, Green, Blue). Now the 8-bit secret key and second LSB of each pixel of cover image will be used for decision making to fetch hidden information from stego image. Our first step is to perform XOR operation between 2nd LSB of stego image pixel and 8-bit key. If the XOR result of the above operation is 1 then we will fetch LSB of the same pixel of stego image and store it in an array, Otherwise we will skip that pixel. The fetching process will be continued depending on the length of secret message. After fetching all the bits we split the array bits into a sequence of 8bits and then convert them into characters by use of ASCII values.

Output:



```

17CS02003_Assignment_1.py  cover1.png  encoded1.png
cover1.png
encoded1.png

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python
PS E:\sem8\sfLab2\lab1> & C:/Users/RahulBharadwaj/AppData/Local/Programs/Python/Python37/python.exe e:/sem8/sfLab2/lab1/17CS02003_Assignment_1.py

Choose appropriate option..

1.Encode
2.Decode
3.Compute PSNR Score

1

Choose by which image you want to test...

1. cover1.png
2. cover2.png
1

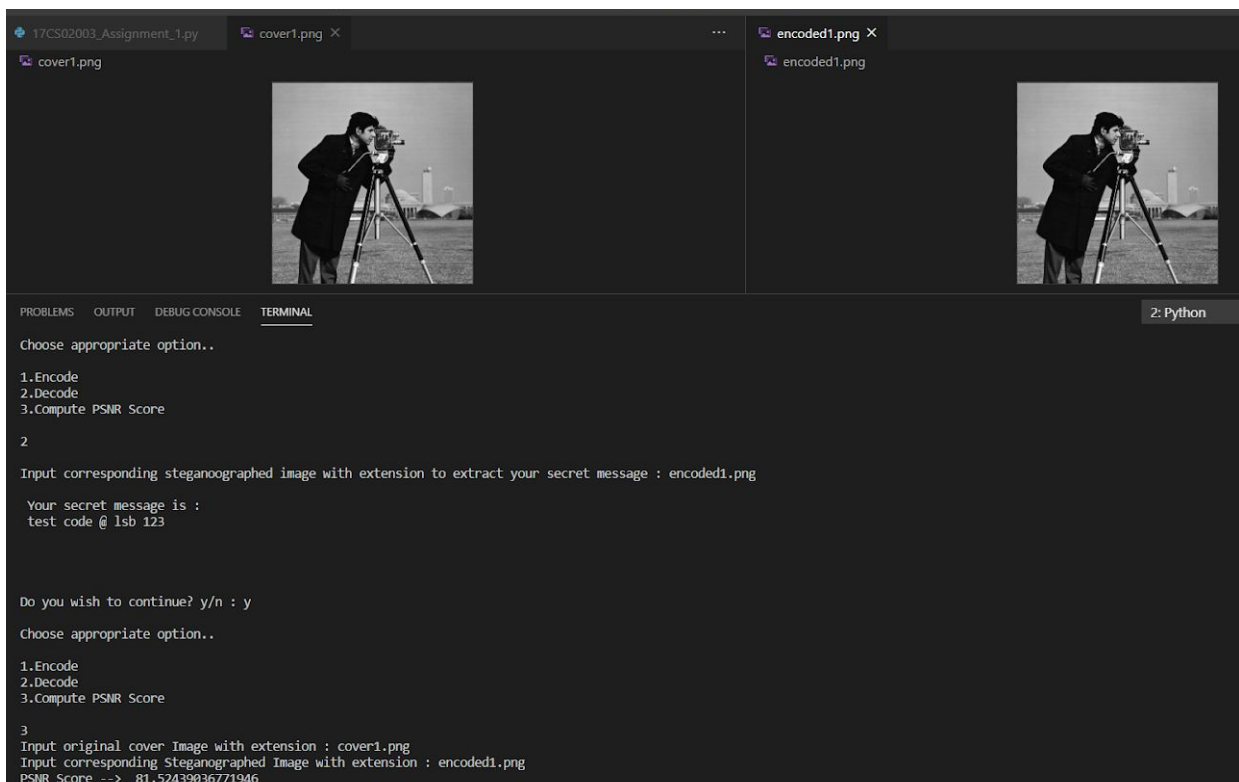
Input secret message that you want to hide : test code @ 1sb 123

Input a name for the steganographed/encoded image with extension :
encoded1.png

Saving the steganographed image.....
Image saved with the name encoded1.png

Do you wish to continue? y/n : 

```



```

17CS02003_Assignment_1.py  cover1.png  encoded1.png
cover1.png
encoded1.png

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

Choose appropriate option..

1.Encode
2.Decode
3.Compute PSNR Score

2

Input corresponding steganographed image with extension to extract your secret message : encoded1.png

Your secret message is :
test code @ 1sb 123

Do you wish to continue? y/n : y

Choose appropriate option..

1.Encode
2.Decode
3.Compute PSNR Score

3

Input original cover Image with extension : cover1.png
Input corresponding Steganographed Image with extension : encoded1.png
PSNR Score --> 81.52439036771946

```

And the quality of embedding is measured by computing psnr score.

Input & Steganographed Images



Fig.3 Cover Image -1



Fig.4 Steganographed Image -1



Fig.5 Cover Image -2



Fig.6 Steganographed Image-2

References

- A Steganography Algorithm for Hiding Secret Message inside Image using Secret Key
- LSB based Image steganography