



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering (SCOPE)

B. Tech. Winter 2020-2021

CSE2005: Operating Systems

**Secure File Sharing System Using Image
Steganography and Cryptography Techniques.**

Course Faculty

Braveen.M

Batch Member:

K V SAI RAGHAVENDRA

(19BCE1178)

Table of Contents

S. NO.	TITLE	PAGE NUMBER
1	Abstract	3
2	Introduction	4
3	Motivation	4
3	steps	5
4	Algorithms used	6
5	Source Code	9
6	Snapshots	9
7	Advantages	13
8	Conclusion	14

Abstract

Information security is one of the most challenging problems in today world. One of the reasons that intruders can be successful is the most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. In order to secure the transmission of secret data over the internet various methods has been there.

This project proposes a system which uses Image steganography combined with cryptography. User gives a key and the image and selects the file. The user information(file data) is encrypted using this key and then this cipher text is encoded in the image. There are two security layers that is a steganography hiding layer and the second is an encryption layer protecting the information.

Introduction

In this project, we aim to create a Application where in users can hide their file data in any image of their choice. The main goal of this project is to embed textual information into a image file and also the text information is encrypted so we also get the advantage of cryptography. The Application consists of two major parts namely the encoding part and the decoding part. In the encoding section a user can input the image of his choice into which the user text information is encoded. The text information is also encrypted before encoding it into the image, the user can then download the encoded image. In the decoding section the user can input the decoded image and the decrypted text information is displayed to the user in a new file.

Steganography is the art of hiding information in digital media through the techniques of embedding hidden messages in such a way that no one except the sender and the intended receiver(s) can detect the existence of the messages.

One of the reasons why the attackers become successful in intrusion is that they have an opportunity to read and comprehend most of the information from the system. Intruders may reveal the information to others, misuse or modify the information, misrepresent them to an individual/ organization or use them to plan even some more severe attacks. One of the solutions to this problem is through the use of steganography and cryptography.

Motivation

The internet allows for easy transfer of information over large areas. This is both a blessing and a curse since the receiver and the anonymous intruders can see our information. Encrypting data has been the most popular approach to protecting information but this protection can be broken with enough computational power. An alternate approach to encrypting data would be to hide it inside it in a image making it only a plane image is there without knowing there is data beneath the image. This two layers of security ensures the transfer of data in a secured way.

Steps

A basic user interface layout is created like selecting the file , selecting the Image and file data is displayed, encrypted file data appears and the image which has been selected is also shown and the encrypted image can be downloaded into the laptop and send it to the receiver. In receiver side he can select the decrypted image and the data is shown in window. He can save the data into the new file.

Sender's Side:

- 1.First the user selects the file which he want to store it in image.
- 2.Then the user will select the image.
- 3.Now the user will enter the key with the encryption algorithm using the key now the data will encoded into the cipher text.
- 4.Then the user selects the encode button in which the cipher text is encoded Into the image and the image is given to the receiver.

Receiver's Side:

- 1.Here the receiver will selects the steganographed image and then selects the decode button and the cipher text will be visible to him.
- 2.To see the normal text he want to know the key which the sender has used to encrypt it.
- 3.After entering the key the receiver will be able to see the original text and Save it in a new file.

Algorithms Used:

Encryption Algorithm:

```
String key, message;  
    String cipher="";  
    for(i=0;i<message.length();i++)  
    {  
        if(j>key.length()-1){  
            j=0;  
        }  
        int temp=message.charAt(i) ^ key.charAt(j);  
        cipher=cipher+String.format("%02x", (byte)temp);  
        j++;  
    }  
System.out.println(cipher);
```

Example of Encryption:

message: raghavendra, key=sai
 $r^s = 114^{115} = 1 = 01$
 $a^a = 97^{97} = 0 = 00$
 $g^i = 103^{105} = 14 = 0e$
 $h^s = 104^{115} = 27 = 1b$
continue.....
cipher=01000e1b001f160f0d0100
raghavendra-> 01000e1b001f160f0d0100

Decryption Algorithm:

```
String key,cipher;  
String hexToDeci = "";  
for (int i = 0; i < cipher.length(); i+=2) {  
    // splitting hex into a pair of two  
    String output = cipher.substring(i, (i+2));  
    int decimal = Integer.parseInt(output, 16);  
    hexToDeci += (char)decimal;  
}  
String decrypText = "";  
int keyItr = 0;  
for (int i = 0; i < hexToDeci.length(); i++) {  
    // XOR Operation  
    int temp = hexToDeci.charAt(i)^key.charAt(keyItr);  
    decrypText += (char)temp;  
    keyItr++;  
    if(keyItr >= key.length()){keyItr = 0; }  
}
```

Example:

Cipher="01000e1b001f160f0d0100", key="sai"
01=1== $1^s=1^{115}=114=\text{char}(114)=r$
00=0== $0^a=0^{97}=97=\text{char}(97)=a$
0e=14= $14^i=14^{105}=103=\text{char}(103)=g$
1b=27 $27^s=27^{115}=104=\text{char}(104)=h$
Continues.....
01000e1b001f160f0d0100→raghavendra

Least Significant Bit Algorithm:

- 1.Convert the image into pixels.
- 2.Convert the image pixels into bytes.
- 3.Store the length of the text in image pixels.
- 4.Convert the text into Ascii Numbers and into bytes.
- 5.Traverse through each pixel of the array and do
- 6.Convert the pixel into binary
- 7.Get the next bit to be embedded form the text
- 8.Select the last bit and replace with text
- 9.Repeat until the text gets over.

Example Of LSB Method

- A grid for 3 pixels of a 24-bit image can be as follows:

```
(00101101 00011100 11011100)
(10100110 11000100 00001100)
(11010010 10101101 01100011)
```

- When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

```
(00101101 00011101 11011100)
(10100110 11000101 00001100)
(11010010 10101100 01100011)
```

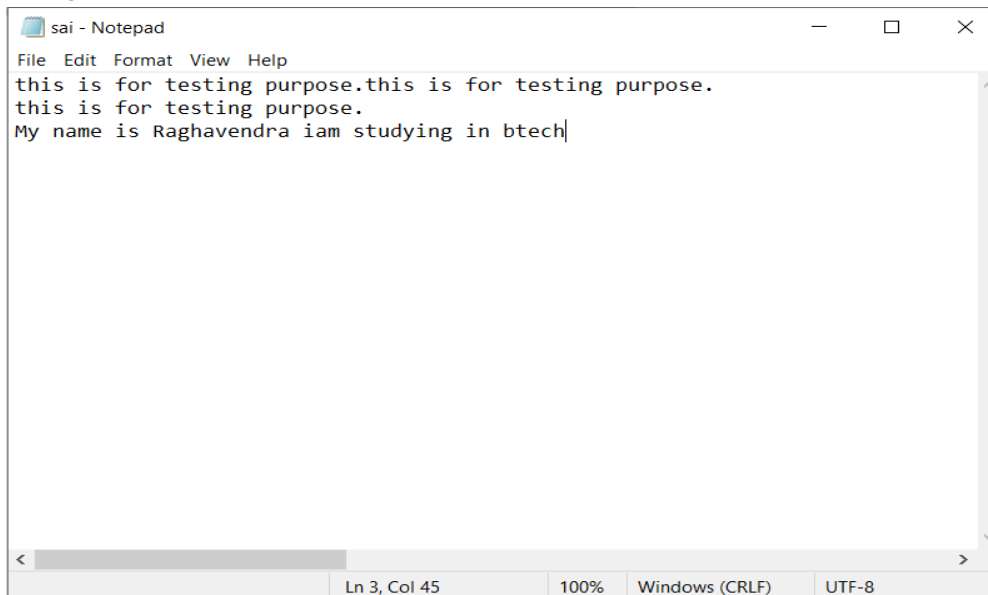

Source Code

For Detailed source visit Google Drive Link:

<https://drive.google.com/drive/folders/1VElQFM2epzT06DjN4Pci2KP8P14HQrqY?usp=sharing>

Results / Snapshots

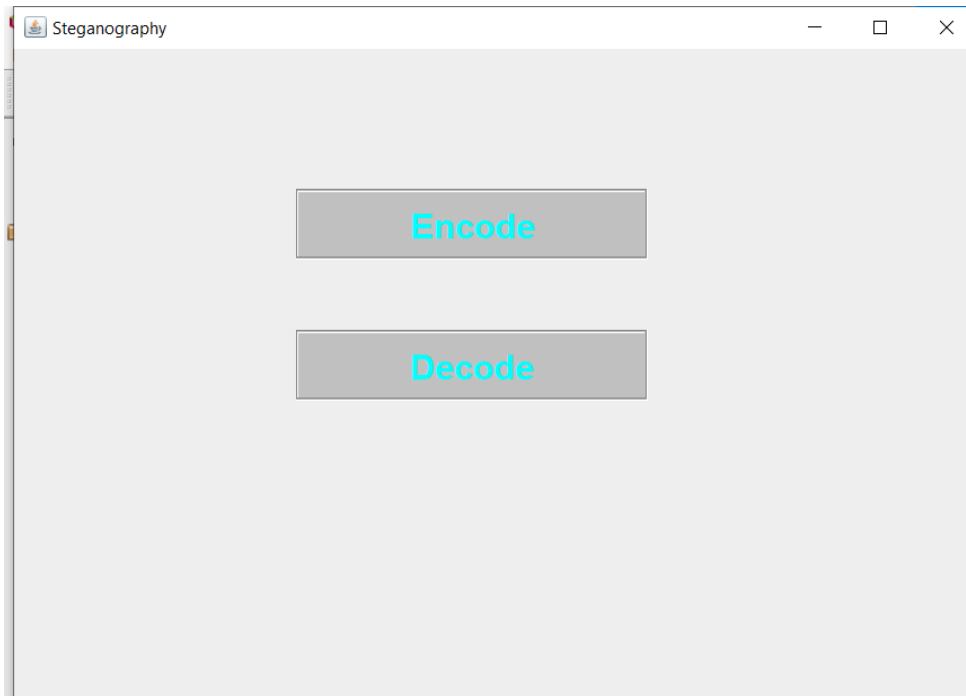
Original File:

A screenshot of a Notepad window titled 'sai - Notepad'. The window has a menu bar with 'File', 'Edit', 'Format', 'View', and 'Help'. The text area contains the following code:

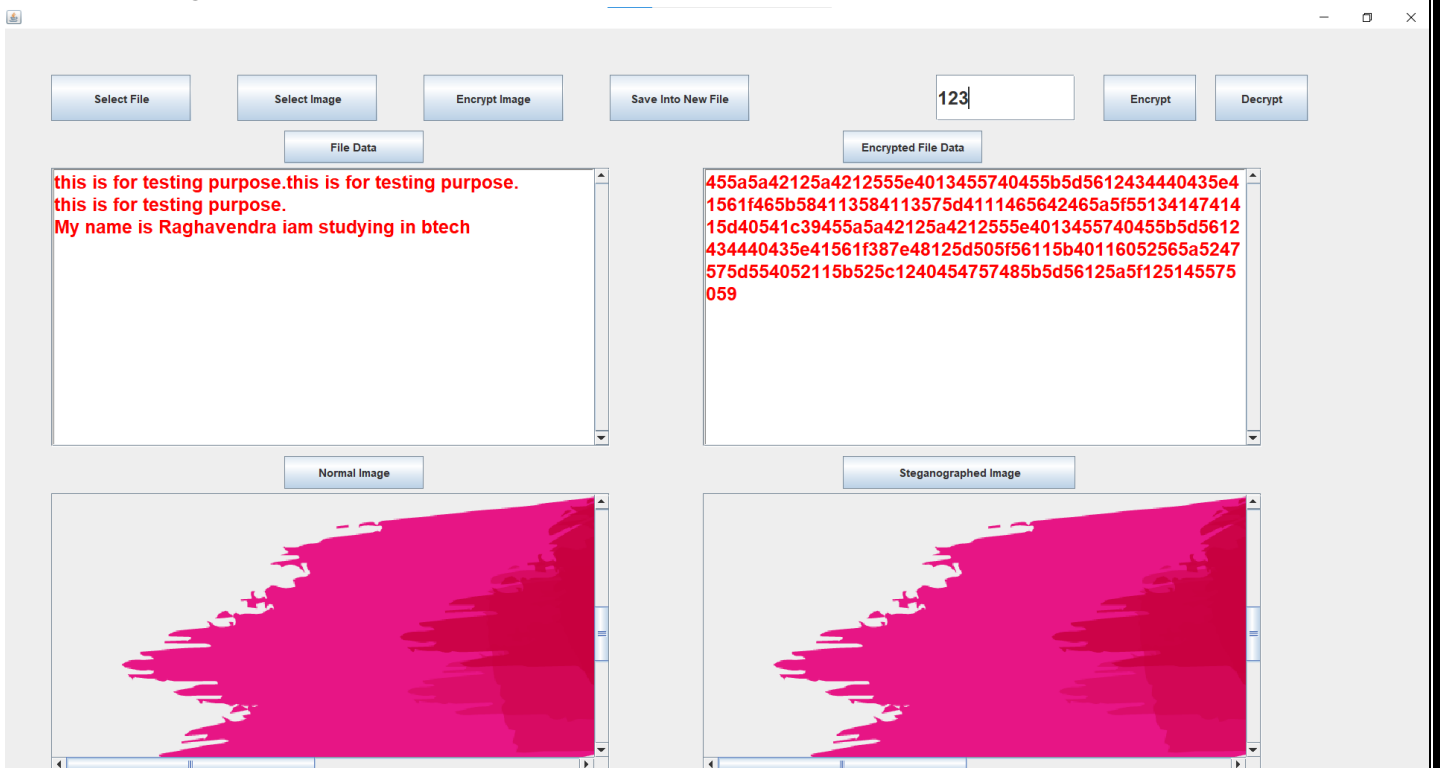
```
this is for testing purpose.this is for testing purpose.  
this is for testing purpose.  
My name is Raghavendra iam studying in btech|
```

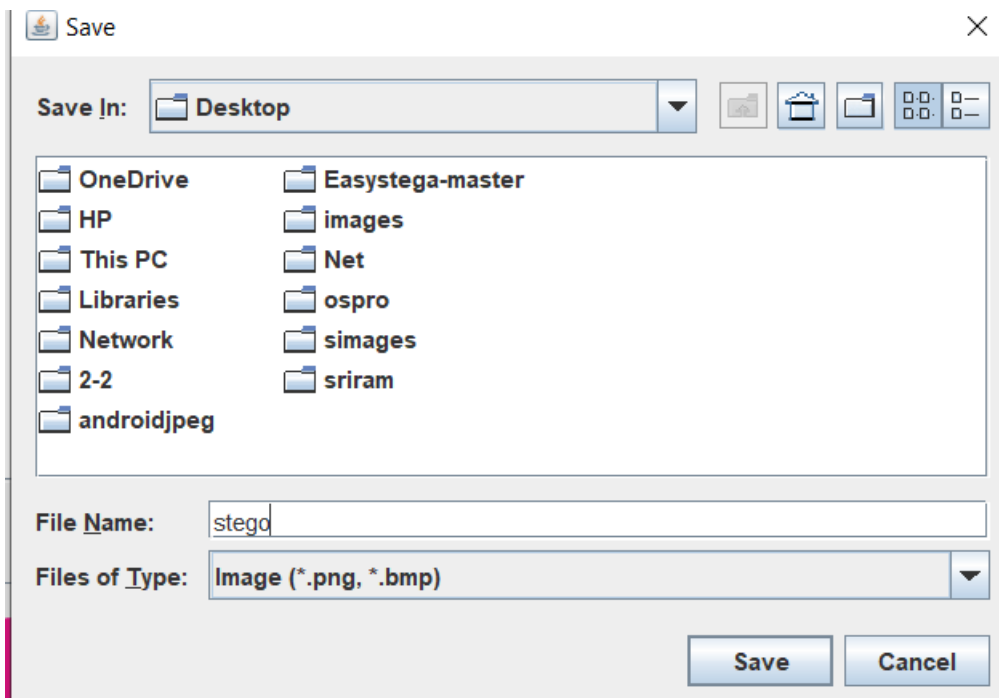
The status bar at the bottom shows 'Ln 3, Col 45', '100%', 'Windows (CRLF)', and 'UTF-8'.

Home Page:



Encode Page:





Normal Image:



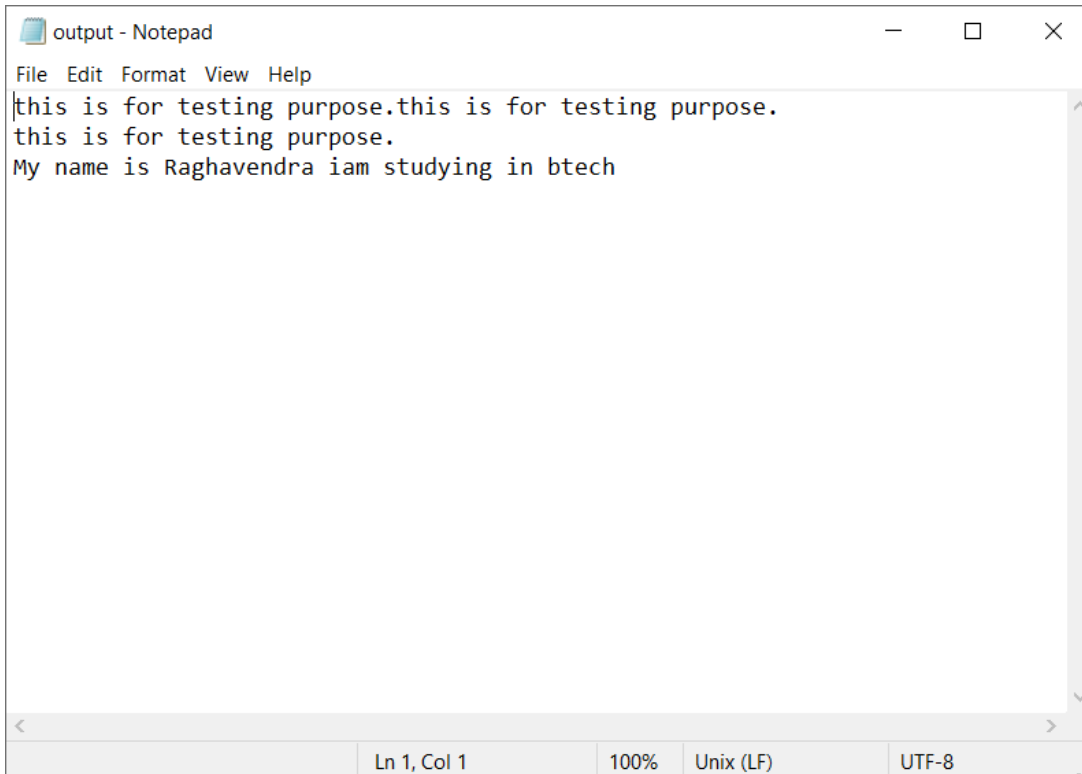
Encrypted Image:



Decode page:

A screenshot of a web application interface for decoding a steganographed image. The interface is light gray and contains several buttons and text areas. At the top, there are five buttons: "Select File", "Select Image", "Decrypt Image", "Save Data Into Ne...", and "Decrypt". Below these buttons are two text areas: "Decrypted File Data" and "Encrypted File Data". The "Decrypted File Data" area contains the text: "this is for testing purpose.this is for testing purpose.
this is for testing purpose.
My name is Raghavendra iam studying in btech". The "Encrypted File Data" area contains a long string of hexadecimal characters: "455a5a42125a4212555e4013455740455b5d5612434440435e41561f465b584113584113575d4111465642465a5f5513414741415d40541c39455a5a42125a4212555e4013455740455b5d5612434440435e41561f387e48125d505f56115b40116052565a5247575d554052115b525c1240454757485b5d56125a5f125145575059". Below these text areas is a button labeled "Steganographed Image". At the bottom of the interface is a large image viewer showing a red brushstroke, which is the original image hidden within the steganographed image.

Output File:



Advantages of this system:

1. Files are Securely shared over the network without the data being displayed.
2. Two Layers of Security which enhances the Performance of the System.
3. Usage of Cryptography and Least Significant Bit Algorithms makes it more efficient.
4. This protection of files inside the images can be used in Our Laptop and Pcs.

Conclusion:

We have successfully developed a desktop application for image steganography combined with cryptography techniques where the user can insert a encrypted file data into the image using the cryptography and least significant bit algorithm and similarly user can decode the information from the image using the same algorithm. Though the implementation is specific to image file by the system works as a generic system supporting media such as audio and video, but the respective steganographic techniques must be implemented.