B.S Computer Science 6th Semester

Project Report



**Image Steganography**

Submitted By

**Muhammad Hassan Farid**

**Huzaifa Awan**

**Tayyaba Shahid**

DEPARTMENT OF COMPUTING

HAMDARD INSTITUTE OF ENGINEERING & TECHNOLOGY

HAMDARD UNIVERSITY ISLAMABAD

**Introduction:**Image steganography is the practice of concealing information within digital images to ensure the secrecy and integrity of the hidden data. It is a subfield of steganography, which involves hiding information in various media types, such as text, audio, or video. In image steganography, the hidden data is embedded within the pixels of an image, making it imperceptible to the human eye. This report provides an overview of image steganography, its techniques, applications, and challenges.

**Technique Use in Image Steganography:**Image steganography employs several techniques to embed information in images. But we use Least Significant Bit (LSB).**Why Least Significant Bit (LSB) Substitution?**

Least Significant Bit (LSB) image steganography is one of the most widely used techniques for hiding information within digital images. It leverages the concept that slight modifications to the least significant bit of the pixel values in an image are often imperceptible to the human eye. By replacing the LSBs of selected pixels with bits of the hidden data, LSB steganography provides a simple and effective way to conceal information while maintaining the visual integrity of the cover image. This detailed note explores the working principles, advantages, limitations, and potential applications of LSB image steganography.**Working Principles**:The LSB technique operates on the binary representation of the pixel values in an image. In digital images, each pixel's color is represented by a combination of red, green, and blue (RGB) values. Each RGB value typically occupies 8 bits, allowing for 256 possible values (0-255) per color channel.To embed information using LSB steganography, the following steps are typically followed:**a. Selection of Pixels:** A subset of pixels in the image is chosen to carry the hidden data. These pixels should be carefully selected to minimize visual changes and maintain statistical properties of the image.**b. Conversion of Hidden Data:** The information to be hidden, such as text, binary data, or another image, is converted into a bitstream.**c. Embedding Process:** For each selected pixel, the least significant bit(s) are replaced with bits from the hidden data. The number of LSBs used for embedding depends on factors like image size, color depth, and desired capacity. Typically, only one or two LSBs are modified to minimize visual impact.**d. Reconstructed Image:** The modified pixel values are used to reconstruct the stego-image, which appears visually similar to the original cover image.

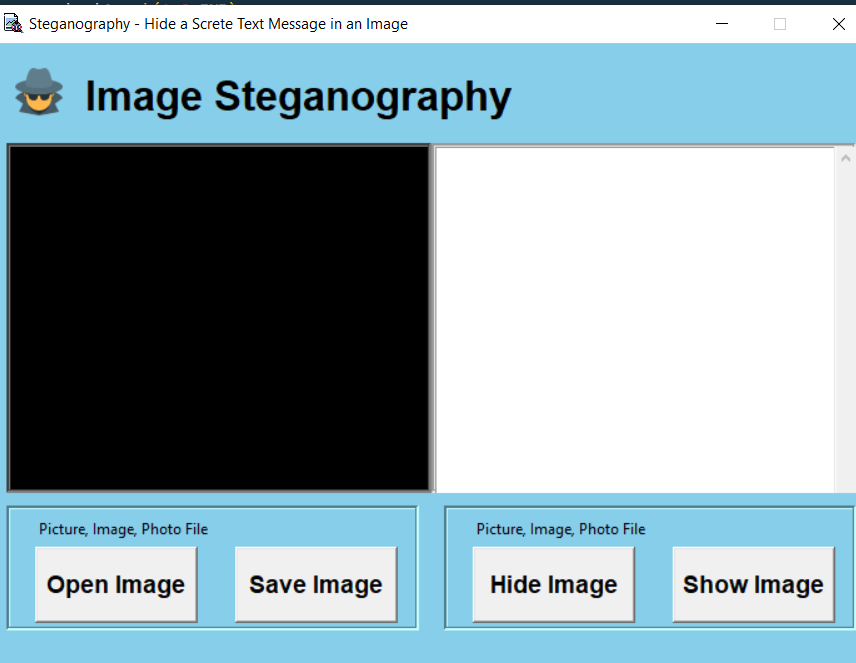
**Advantages of LSB Steganography**LSB image steganography offers several advantages, contributing to its popularity and widespread use:**a. Simplicity:** LSB steganography is relatively easy to implement compared to more complex techniques. It requires minimal computational resources and does not involve complicated algorithms or transformations.**b. Imperceptibility:** The modifications made to the LSBs of the pixel values are usually imperceptible to the human eye. The visual quality of the stego-image remains largely unaffected, ensuring that the hidden data remains undetected.**c. Ease of Extraction:** Extracting the hidden data from an LSB-stego image is a straightforward process. By extracting the LSBs of the selected pixels, the embedded information can be retrieved without the need for a decryption process.

**Limitations and Security Considerations**Despite its advantages, LSB steganography has certain limitations and security considerations that should be taken into account:**a. Limited Capacity:** The amount of data that can be hidden using LSB steganography is directly related to the number of pixels available for modification. Embedding a large amount of data may require a significant number of pixels, which can increase the risk of detection.**b. Susceptibility to Lossy Compression:** LSB steganography can be affected by lossy compression algorithms commonly used in image formats like JPEG. Compression can introduce artifacts and modify pixel values, potentially corrupting the hidden data or making it more detectable.**c. Vulnerability to Steganalysis:** Sophisticated steganalysis techniques can be employed to detect the presence of hidden data within LSB-stego images. Statistical analysis, visual inspection, and other advanced algorithms can help in identifying the modifications made to the LSBs.

**Applications of Image Steganography:**Image steganography finds applications in various domains where secure and covert communication is necessary. Some prominent applications include:

**Confidential Communication:** Image steganography enables confidential communication by hiding sensitive information within images. This is useful in scenarios where encryption alone may raise suspicion, but a seemingly innocent image can safely carry the hidden message.**Copyright Protection:** Digital watermarking, a form of steganography, is used to embed copyright information or ownership details within images. This helps protect intellectual property rights by providing a means to identify the original creator.**Covert Surveillance:** Steganography can be employed to hide surveillance data within images, making it difficult for unauthorized individuals to detect the presence of sensitive information. This can aid law enforcement agencies and security organizations in discreetly gathering evidence.**Challenges and Limitations**Image steganography faces several challenges and limitations that impact its effectiveness and reliability:**a. Capacity and Robustness:** The primary challenge is to hide a significant amount of data within an image without noticeably degrading its quality. Increasing the embedded data's capacity may result in a higher likelihood of detection or distortion of the cover image.**b. Security and Detection:** Steganography aims to provide secure communication, but adversaries can attempt to detect or remove the hidden information. Advanced statistical analysis and steganalysis techniques can be employed to detect steganographic content, potentially compromising the secrecy of the hidden data.**c. Format and Compression**: Different image formats and compression techniques may impact the effectiveness of steganography. Some image formats and compression algorithms can introduce artifacts or loss of information, making the hidden data more susceptible to detection or corruption.

**Output:**

**Step1:** Click on Open image, select the image.

**Step 2:** After selection, write a message ad then click on Hide Image and then click on save image. The text will be hide in the image and create a new image (Hidden.png).

**Step 3:** Click on Open Image and Select the image that we save (Hidden.png), and then click on Show image, the Hidden Text will be Show.

**Conclusion:**Image steganography is a powerful technique for secure and covert communication, copyright protection, and surveillance applications. It offers a means to hide information within digital images, making it challenging for unauthorized individuals to detect or access the hidden data. However, the effectiveness of image steganography depends on the chosen technique, the amount of data to be hidden, and the security measures employed. As technology advances, new challenges and countermeasures will continue to emerge, requiring ongoing research and development in the field of image steganography.

**Source Code:**

from tkinter import \*from tkinter import filedialog from PIL import Image,ImageTkimport osfrom stegano import lsbroot=Tk()root.title("Steganography - Hide a Screte Text Message in an Image")root.geometry("700x500+400+180")root.resizable(False,False)root.configure(bg='skyblue')# open image functiondef showimage(): global filename filename=filedialog.askopenfilename(initialfile=os.getcwd(), title="Select Image File", filetype=(("PNG File","\*.png"), ("JPG File","\*.jpg"), ("All file","\*.text") )) img = Image.open(filename) img=ImageTk.PhotoImage(img) lb1.configure(image=img,width=250,height=250) lb1.image=img # Hide Text function def Hide(): global secret message=text1.get(1.0,END) secret = lsb.hide(str(filename), message) # Show image functiondef Show(): clear\_message = lsb.reveal(filename) text1.delete(1.0,END) text1.insert(END, clear\_message) # Save image fuction def save(): secret.save("hidden.png")#iconimage\_icon=PhotoImage(file='icon.png')root.iconphoto(False,image\_icon)#logologo=PhotoImage(file="Logo2.png")Label(root,image=logo,bg="skyblue").place(x=10,y=13)Label(root,text="Image Steganography",bg="skyblue",fg="black",font="arial 25 bold").place(x=70,y=20)#first Framef = Frame(root,bd=3,bg="black",width="340",height=280,relief=GROOVE)f.place(x=10,y=80)lb1 = Label(f,bg="black")lb1.place(x=40,y=10)#second frameframe2 = Frame(root,bd=3,width=340,height=280,bg="white",relief=GROOVE)frame2.place(x=350,y=80)text1=Text(frame2,font="Robote 20", bg="white",fg="black",relief=GROOVE,wrap=WORD)text1.place(x=0,y=0,width=320,height=280)scrollbar1 = Scrollbar(frame2)scrollbar1.place(x=320,y=0,height=300)scrollbar1.configure(command=text1.yview)text1.configure(yscrollcommand=scrollbar1.set)#third frameframe3=Frame(root,bd=3,bg="skyblue",width=330,height=100,relief=GROOVE)frame3.place(x=10,y=370)Button(frame3,text="Open Image",width=10,height=2,font="arail 14 bold",command=showimage).place(x=20,y=30)Button(frame3,text="Save Image",width=10,height=2,font="arail 14 bold",command=save).place(x=180,y=30)Label(frame3,text="Picture, Image, Photo File",bg="skyblue",fg='black').place(x=20,y=5)#Fourth Frameframe4=Frame(root,bd=3,bg="skyblue",width=330,height=100,relief=GROOVE)frame4.place(x=360,y=370)Button(frame4,text="Hide Image",width=10,height=2,font="arail 14 bold",command=Hide).place(x=20,y=30)Button(frame4,text="Show Image",width=10,height=2,font="arail 14 bold",command=Show).place(x=180,y=30)Label(frame4,text="Picture, Image, Photo File",bg="skyblue",fg='black').place(x=20,y=5)root.mainloop()