PROJECT 1

Steganography Tool - StegTool for Image/File Hiding

Submitted By,

Friendsy Sajeev

Objective:

To create a simple GUI-based tool for hiding text or files inside images using steganography techniques.

Abstract

This project presents a simple yet effective **steganography tool** designed to conceal textual information within digital images using the **Least Significant Bit (LSB)** technique. Developed using Python, the tool integrates libraries such as **PIL (Pillow)** and **Tkinter** to provide a user-friendly graphical interface. Users can upload a cover image, enter a secret message, and embed the message into the image with minimal quality loss. The tool also supports the extraction of hidden messages from modified images. It is capable of handling lossless image formats like **PNG** and **BMP**, ensuring data integrity. The primary goal of this tool is to offer a basic, secure method of information hiding that demonstrates the core principles of steganography in a practical, hands-on application.

Technologies Used:

- **Python**
- **PIL (Pillow)** – for image manipulation
- **stepic** (or custom LSB logic) – for encoding/decoding
- **Tkinter** – for GUI
- **Optional:** cryptography – for encrypting/decrypting hidden data

How It Works:

1. Encoding (Hiding Data):

- Convert the input text or file into **binary**.
- Use **Least Significant Bit (LSB)** method to embed this binary data into the pixels of a cover image.
- Save the modified image as the **stego-image**.

2. Decoding (Extracting Data):

- Extract the LSBs from the stego-image's pixels.
- Reconstruct the binary data and convert it back into text (or decrypt if encrypted).

Features:

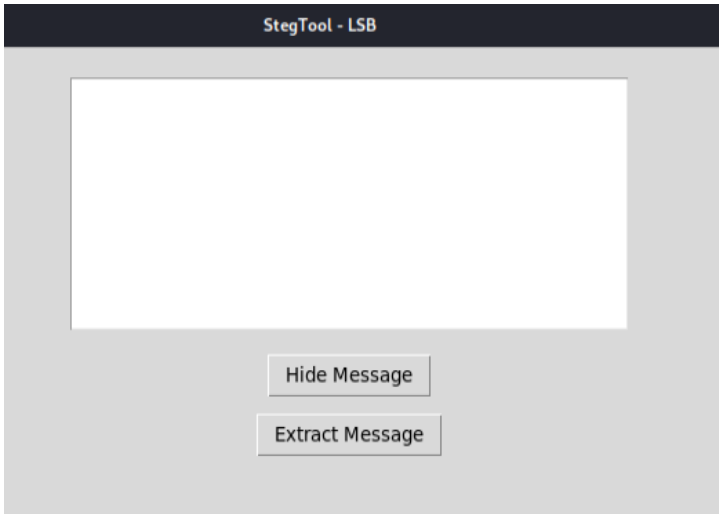
- **Hide Text or File** in PNG/BMP image
- **Extract Hidden Data** from stego-images
- **Optional Encryption** of text before embedding
- **Simple GUI** with drag-and-drop support
- **Supports** formats like .png, .bmp (lossless)

GUI Features (Tkinter):

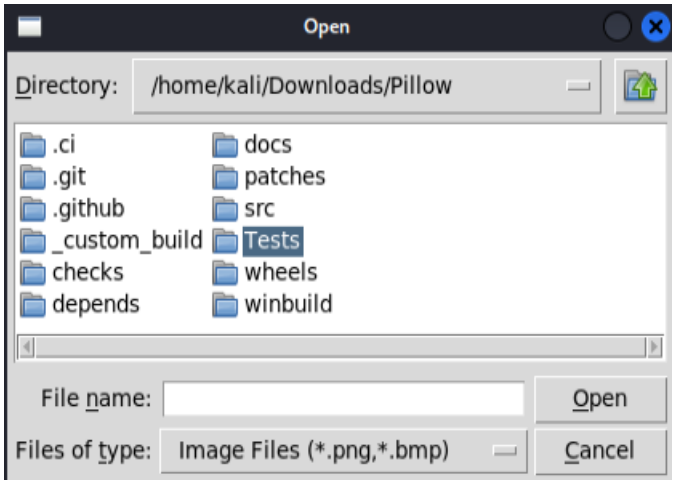
- Upload **cover image**
- Enter or upload **message/file to hide**
- Button to **encode** and save output image
- Button to **decode** from an image

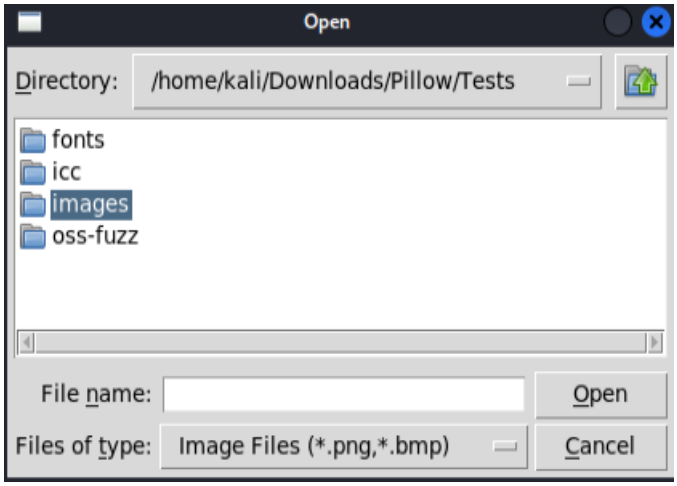
- Status messages (success, error)

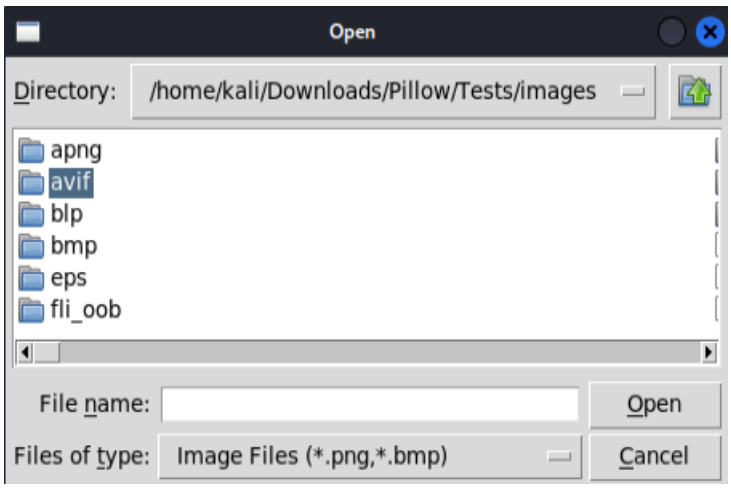
SCREENSHOTS:

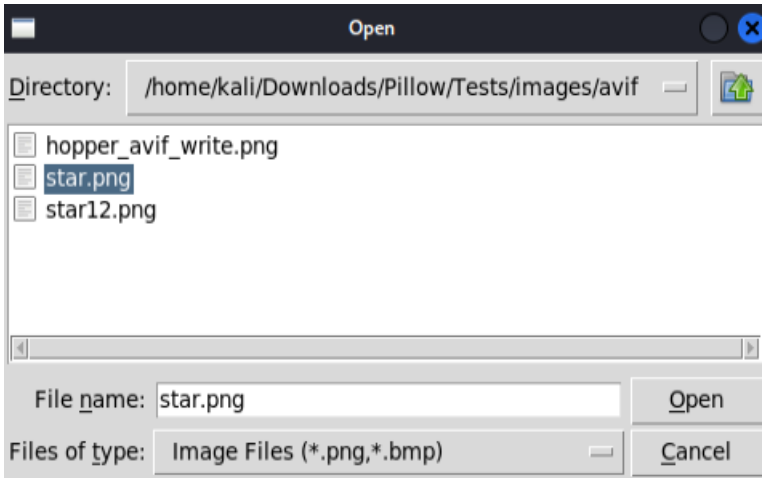


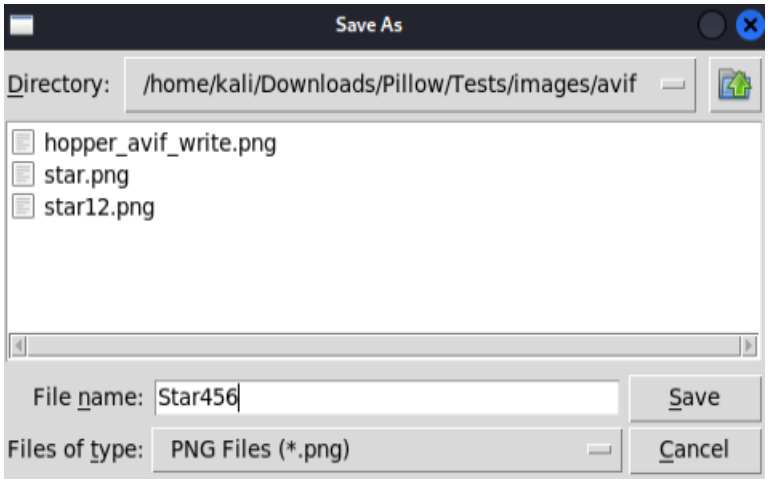
To hide a message:


Open
Directory:
/home/kali/Downloads/Pillow
.ci
.git
.github
_custom_build
checks
depends
docs
patches
src
Tests
wheels
winbuild
File name:
Open
Files of type:
Image Files (*.png,*.bmp)
Cancel


Open
Directory:
/home/kali/Downloads/Pillow/Tests
fonts
icc
images
oss-fuzz
File name:
Open
Files of type:
Image Files (*.png,*.bmp)
Cancel

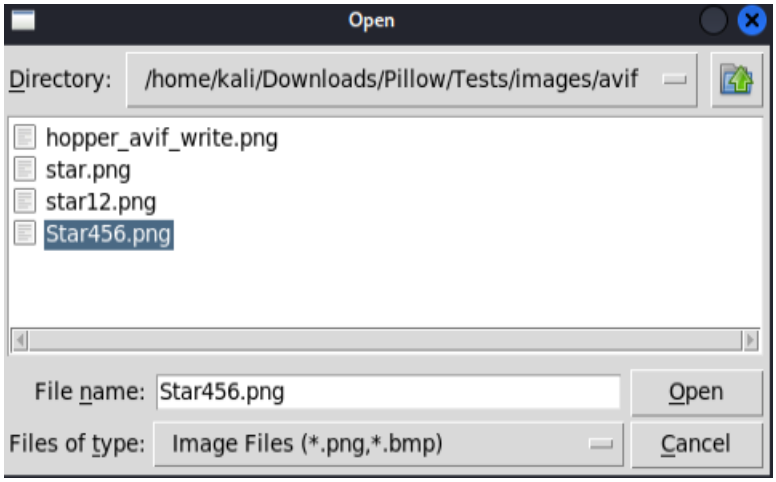

Open
Directory:
/home/kali/Downloads/Pillow/Tests/images
apng
avif
blp
bmp
eps
fli_oob
File name:
Open
Files of type:
Image Files (*.png,*.bmp)
Cancel

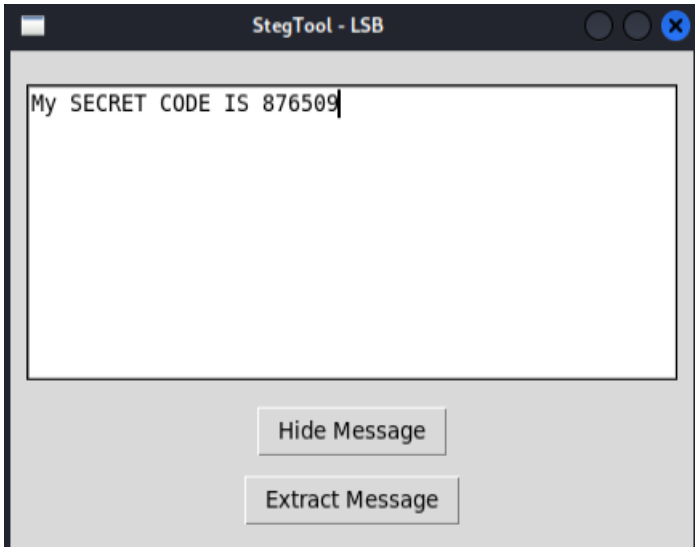




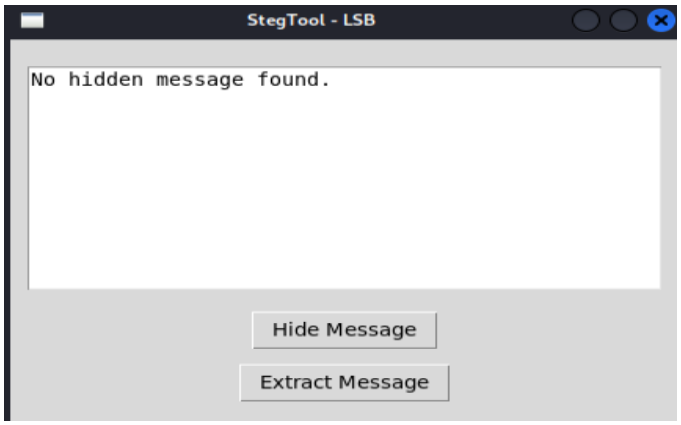
Message saved in Star456.png.

To Extract the message:





And if we open the wrong file – a ‘no hidden message found’ opens.



Conclusion

This project demonstrates how steganography can be used to securely hide messages within images using the LSB technique. The tool provides a simple and user-friendly interface, making data hiding both accessible and practical. It highlights the importance of information security through creative and efficient means.