# Project Report on Steganography

## 1. Introduction

Information security has become one of the most important concerns in today's digital era.
While encryption protects data by making it unreadable to unauthorized users, it still attracts attention.
Steganography provides an additional layer of security by hiding secret information within a cover medium such as an image,
audio, or video file, making the existence of the hidden data undetectable to an attacker.

In this project, we implement Image Steganography using Python.
The technique hides secret text messages inside images using the Least Significant Bit (LSB) method.

## 2. Objectives

- To study the concept of steganography and its importance in cybersecurity.
- To implement a Python-based tool that can hide and extract secret text in an image.
- To demonstrate secure communication without altering the visual appearance of the image.

## 3. Methodology

1. Cover Medium Selection: A digital image (PNG/JPG format) is used as the cover file.
2. Message Embedding: The secret text is converted into binary and embedded into the least significant bits (LSB) of the image pixels.
3. Message Extraction: The modified image is processed to extract the hidden binary data, which is then converted back into text.
4. Verification: The extracted message is compared with the original message to verify accuracy.

## 4. Tools & Technologies Used

- Programming Language: Python 3.x
- Libraries:
  - Tkinter → For GUI

  - Stegano → For LSB steganography operations
  - PIL (Pillow) → For image processing
- Platform: Kali Linux / Windows
- IDE: VS Code, PyCharm, or Nano (Linux editor)

**5. Implementation**

Encoding Process:
- The user selects an image and enters a secret message.
- The program embeds the message using LSB method.
- The output is a stego-image (modified image) that visually looks the same.

Decoding Process:
- The program takes the stego-image as input.
- It extracts the hidden binary values from pixel LSBs.
- The binary is converted back into readable text.

**6. Results**

- The secret message was successfully hidden inside an image.
- The image quality remained unchanged to the human eye.
- The hidden text was correctly extracted without data loss.

Screenshots of execution (example):
1. GUI for entering message and selecting image.
2. Output stego-image generated.
3. Hidden message extracted successfully.

**7. Applications**

- Secure communication between two parties.
- Digital watermarking for copyright protection.
- Protecting sensitive military or government data.
- Preventing unauthorized access to confidential information.

**8. Limitations**

- Large messages cannot be hidden in small images.
- If the image undergoes compression or editing, hidden data may be lost.
- Attackers with steganalysis tools may still detect hidden content.

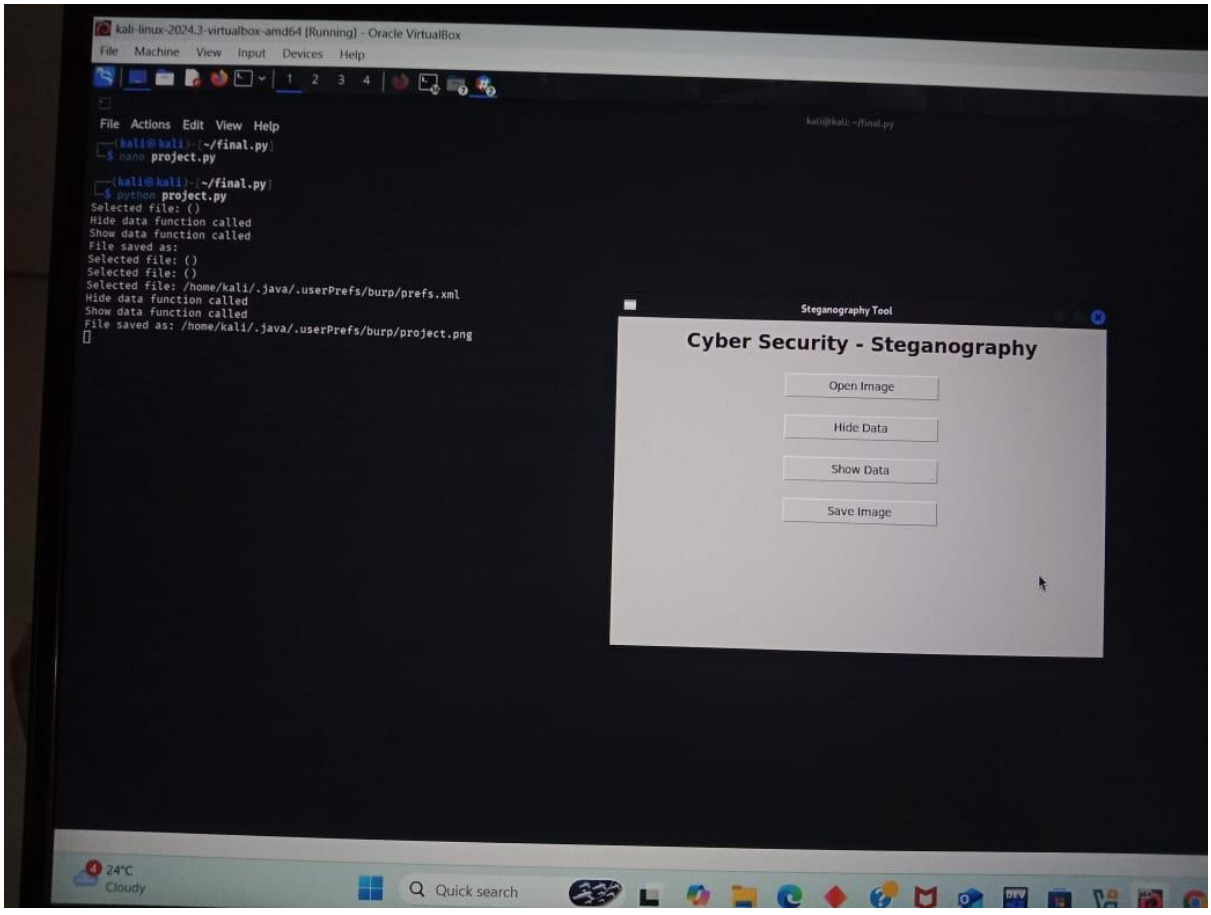
**9. Conclusion**

Steganography is a powerful technique for concealing secret information within cover media.
In this project, we successfully demonstrated Image Steganography using LSB method with the help of Python.
The hidden messages were embedded and retrieved accurately, proving the effectiveness of the method.
This project highlights how steganography can play a vital role in modern information security.

**10. Future Scope**

- Implementing audio and video steganography.
- Combining encryption with steganography for double security.
- Developing a mobile application for real-time secure communication.

## 11.Screenshot Output



12.code

```
from tkinter import *
from tkinter import filedialog
import os

# Function to open an image file
def open_image():
    file_path = filedialog.askopenfilename(
        initialdir=os.getcwd(),
        title="Select File",
        filetypes=(
            ("PNG file", "*.png"),
            ("JPG file", "*.jpg"),
            ("All file", "*.*")
        )
    )
    print("Selected file:", file_path)


# Function to hide data in image (to be implemented later)
def hide_data():
    print("Hide data function called")
```

```
# Function to show hidden data from image (to be implemented later)
def show_data():
    print("Show data function called")



# Function to save the steganography image
def save_image():
    file_path = filedialog.asksaveasfilename(
        defaultextension=".png",
        filetypes=(
            ("PNG file", "*.png"),
            ("JPG file", "*.jpg"),
            ("All file", "*.*")
        )
    )
    print("File saved as:", file_path)



# ---------------- Tkinter GUI ----------------
root = Tk()
root.title("Steganography Tool")
root.geometry("600x400")
root.config(bg="lightgray")
```

```
# Heading
Label(
    root,
    text="Cyber Security - Steganography",
    font="Impact 18 bold",
    fg="black",
    bg="lightgray"
).pack(pady=10)

# Buttons
Button(root, text="Open Image", width=20,
command=open_image).pack(pady=10)
Button(root, text="Hide Data", width=20,
command=hide_data).pack(pady=10)
Button(root, text="Show Data", width=20,
command=show_data).pack(pady=10)
Button(root, text="Save Image", width=20,
command=save_image).pack(pady=10)

# Run main loop
root.mainloop()
```