Report on Image Steganography Tool

Introduction

In today's digital world, protecting information is very important. One way of securing data is steganography, which hides secret messages inside ordinary files so that nobody knows they even exist. Unlike encryption, which makes messages unreadable but noticeable, steganography keeps the presence of the message invisible. This project presents an Image Steganography Tool developed in Python. The tool uses a graphical interface so that even beginners can easily hide (encode) text messages inside images and later retrieve (decode) them. It is a practical example of cybersecurity and privacy in action, allowing users to secure their sensitive information without raising suspicion.

Abstract

The Image Steganography Tool is a simple yet effective application designed to hide secret text messages inside image files (PNG or BMP). It works by changing the Least Significant Bits (LSB) of the image's pixel values — a process invisible to the human eye. The tool is built using Python, with Tkinter for the GUI and Pillow (PIL) for image processing. It provides two main features: 1. Encode – Embed a hidden message inside an image and save it as a new stego-image. 2. Decode – Extract the hidden message back from a stego-image. The project shows how steganography can be applied for secure communication, educational purposes, or research in the field of information hiding.

Tools Used

1. Python 3.x – Main programming language. 2. Tkinter – To create the graphical user interface (GUI). 3. Pillow (PIL) – For image handling and manipulation. 4. OS Module – To handle file operations and validation.

Steps Involved in Building the Project

1. Understanding Steganography – The project uses the Least Significant Bit (LSB) method. Secret messages are converted into binary format and stored inside the pixel data of the image. 2. Message Encoding – Convert each character of the secret text into an 8-bit binary. Append a special marker (1111111111111110) to indicate the end of the message. Modify the RGB values of the image pixels by replacing their least significant bits with the binary message bits. Save the modified image as a new file. 3. Message Decoding – Read the pixels of the stego-image. Extract the least significant bits from the pixel values. Convert them back into text until the special end marker is detected. 4. Graphical User Interface (GUI) – A simple window is created using Tkinter. Users can browse and select an image file, type a secret message, encode and save the new image, and decode and view hidden messages. Error handling and user notifications are included for smooth operation.

Conclusion

The Image Steganography Tool demonstrates how sensitive data can be hidden inside images in a way that is invisible to human eyes. It combines simplicity and security, making it useful for beginners in cybersecurity as well as professionals interested in data hiding. With its user-friendly interface, this project is a good example of how theoretical concepts like steganography can be turned into practical applications.