Steganography Tool for Image/File Hiding

Introduction

In today's digital age, data security and privacy are paramount. Steganography is the art of hiding information within other non-secret data, such as images, to avoid detection. This project focuses on building a GUI-based steganography tool that allows users to embed and extract hidden messages or files within image files using the Least Significant Bit (LSB) technique.

Abstract

The objective of this project is to develop a Python-based application that enables secure and invisible communication by hiding textual data inside image files. The tool uses LSB steganography to modify pixel values in a way that is imperceptible to the human eye. A user-friendly interface built with Tkinter allows users to upload images, input messages, and save stego-images. The tool also supports message extraction and optional encryption for added security. This project demonstrates practical applications of image processing, GUI development, and basic cryptography.

Tools Used

- **Python 3.13** – Core programming language
- **Pillow (PIL)** – Image processing library
- **Tkinter** – GUI development
- **VSCode** – Code editor
- **PNG/BMP Images** – Lossless formats used for embedding
- *(Optional)*: AES or XOR encryption for message security

Steps Involved in Building the Project

1. **Environment Setup**
   Installed Python and required libraries (Pillow, Tkinter) in VSCode.
2. **Steganography Logic**

- Converted text to binary using ASCII encoding.
- Embedded binary data into the red channel of image pixels using LSB manipulation.
- Added an EOF marker (1111111111111110) to signal the end of the hidden message.

3. **Image Mode Handling**
    - Converted all images to RGBA format to ensure consistent pixel structure.
    - Supported both RGB and RGBA images to avoid unpacking errors.
4. **Message Extraction**
    - Read LSBs from the red channel of each pixel.
    - Reconstructed the binary string and converted it back to readable text.
5. **GUI Development**
    - Built a Tkinter interface with buttons for image selection, message input, embedding, and extraction.
    - Added file dialogs and message boxes for user interaction.
6. **Testing and Validation**
    - Tested with various image formats and message lengths.
    - Verified that the stego-image looked identical to the original and that messages could be accurately retrieved.

Conclusion

This project successfully demonstrates how steganography can be used to hide sensitive information within images in a secure and user-friendly manner. The tool provides a practical solution for covert communication and data protection. Through this project, key concepts in image processing, binary encoding, and GUI development were explored and implemented. The application is flexible, extendable, and can be enhanced further with encryption, drag-and-drop support, and file hiding capabilities.