REPORT

On

Steganography Tool for Image

Introduction:

Steganography, derived from Greek words meaning "covered writing," is the art of hiding information within another medium so that its very presence is concealed. Unlike encryption, which scrambles a message to make it unreadable, steganography hides the message itself to avoid suspicion. Digital image steganography leverages the vast amount of data contained in images to embed secret information without visible alterations to the human eye. This project focuses on implementing a practical steganography tool that encodes text data within common image formats like PNG and BMP by manipulating the least significant bits (LSB) of the image pixels. As digital communication grows, secure and covert transmission of information becomes more critical, making steganography an important technique in information security, digital watermarking, and privacy protection. This project not only demonstrates the fundamental concepts but also provides an accessible user-friendly interface for embedding and extracting hidden text messages.

Abstract:

The project develops a robust Python-based steganography application with a graphical user interface designed for efficient and secure secret message embedding and extraction within digital images. Using the Least Significant Bit (LSB) method, the application modifies the pixel data of images in PNG and BMP formats to hide textual data in a way that is imperceptible to the naked eye or standard image viewers. The GUI, built with Python's Tkinter, features responsive design elements to ensure adaptability to varying window sizes, enhancing usability across devices. Threading enables smooth background execution of intensive operations, maintaining application responsiveness. The project highlights the practical implementation of steganography for covert communication with simplicity, demonstrating how digital files can be used as silent carriers for sensitive data. This tool is valuable for privacy-conscious users and educational purposes within cybersecurity fields.

Tools Used:

- **Python:** Programming language used to implement the project.
- **Tkinter:** Python standard GUI toolkit to build the user interface.

- **PIL/Pillow:** Python Imaging Library to handle image processing tasks.
- **NumPy:** For efficient manipulation of image pixel data arrays.
- **Threading module:** To run embedding and extraction operations without freezing the GUI.

Steps Involved in Building the Project:

- **Design the GUI layout** using Tkinter with input boxes, buttons, and decorative elements.
- **Implement image loading** with file dialogs for the user to select PNG/BMP images.
- **Implement text embedding** by converting input text to binary and modifying the least significant bit of image pixels accordingly.
- **Save the stego-image** after embedding and clear the input text box.
- **Implement text extraction** by reading the least significant bits from the stego-image until a termination byte is found, then reconstruct the hidden text.
- **Add threading** to keep the application responsive during embedding/extraction.
- **Make GUI responsive** to window resizing by handling resize events and adjusting widget positions.
- **Test the application** for various inputs and images to ensure functionality and usability.

Conclusion:

The developed steganography tool provides an effective method of hiding text within images using the LSB technique. The graphical interface makes steganography accessible to users without specialized technical knowledge. Its support for multiple image formats and responsive design enhances user experience. This project validates the practical application of steganography in secure communication and can be extended in the future to support different file types or incorporate encryption for added security.