REPORT

Sharayu Jadhav S. Y. BSc. Cyber Security

**Steganography Tool for Image/File Hiding**

**1. Project Overview**

**Objective:** Develop a Python-based GUI application for hiding and extracting text/files within images using steganography techniques.

**Key Features:**

* **Encoding / Decoding** using LSB replacement in lossless formats (PNG, BMP)
* File upload/download dialogs and drag-and-drop support
* Optional encryption for hidden content
* Support for embedding both text messages and small files

## 2. Technical Approach

## LSB Encoding Algorithm

* Convert message (or file) to binary.
* Iterate over the image pixels in groups of three.
* For each pixel’s RGB component, adjust the least-significant bit according to the message bit.
* Use an LSB flag in the 9th byte to mark end of data

.Decoding

* Traverse pixels in the same order.
* Extract LSBs and reassemble bytes.
* Detect end-of-data via flag bit and reconstruct the original content .

## 3. GUI Architecture (Tkinter)

* Main window with buttons for "Encode" and "Decode".
* File selection dialogs via tkinter.filedialog.
* Text input/output widgets for embedding messages.

## 4. Optional Encryption Integration

**Workflow:**

1. Encrypt original message/data.
2. Embed the ciphertext bytes in the image.
3. During extraction, decrypt using the secret key.

## 5.Testing & Validation

* **Text:** Test with varying lengths.
* **Files:** Embed documents (e.g., .txt/.pdf).
* **Image integrity:** Visually compare original vs. stego.
* **Decode:** Verify correct message/file recovery

## 6. Conclusion

* Transparent LSB-based embedding and recovery of hidden data in PNG/BMP
* User-friendly GUI for encoding, decoding, and file handling
* Secure optional encryption
* Foundation for future advanced techniques