CAPSTONE PROJECT

SECURE DATA HIDING IN IMAGE USING STEGANOGRAPHY

Presented By
Student Name : Pradnya Jagtap
College Name & Department :Don Bosco Institute of Technology

edunet foundation

OUTLINE

- Problem Statement
- Technology used
- Wow factor
- End users
- Result
- Conclusion
- Git-hub Link
- Future scope

PROBLEM STATEMENT

The problem involves securely hiding sensitive data within an image using steganography to ensure confidentiality and prevent unauthorized access. Traditional encryption methods make hidden data noticeable, whereas steganography conceals it within image pixels without altering the visual appearance. The challenge is to develop an efficient algorithm that embeds and extracts data while maintaining image quality and security.

TECHNOLOGY USED

Libraries Used:

Standard Libraries: stdio.h, stdlib.h, string.h

Custom Header: "steganography.h"

Platforms:

Operating System: Windows

Additional Points:

Steganography Method: Least Significant Bit (LSB) Encoding

Data Handling: Embeds and extracts text data from images

WOW FACTORS

1️⃣ LSB-Based Steganography – Hides data in the Least Significant Bit, ensuring security without visible image changes.

2️⃣ Lossless BMP Processing – Works with uncompressed BMP images to maintain data integrity.

3️⃣ Lightweight & Fast – Written in C for high efficiency, outperforming Python/Java-based tools.

4️⃣ Simple CLI Interface – Easy command-line usage for embedding and extracting data, ideal for automation.

END USERS

1️⃣ Cybersecurity Professionals – Use steganography for secure data transmission and covert communication.

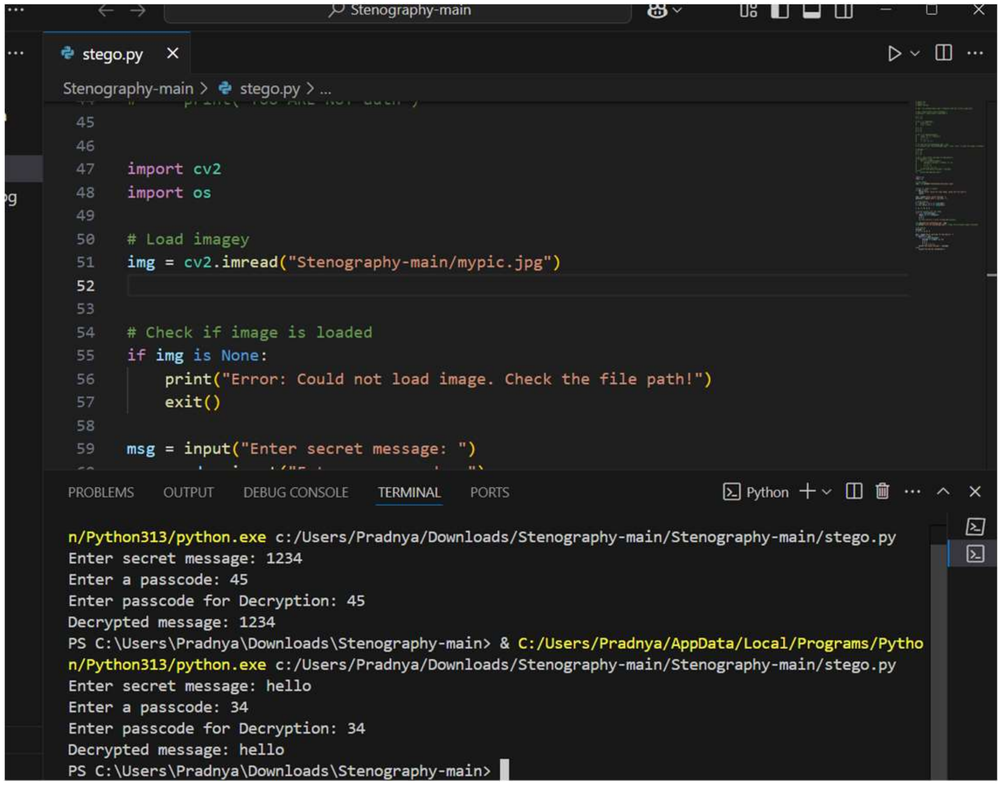
2️⃣ Forensic Experts – Extract hidden information from images for digital investigations.

3️⃣ Government & Defense Agencies – Securely embed sensitive data in images for intelligence and confidential operations.

4️⃣ Researchers & Academics – Study and improve steganographic techniques for data security and cryptography.

5️⃣ Privacy-Conscious Users – Individuals who want to protect personal or confidential information from unauthorized access.

RESULTS



stego.py
Stenography-main > stego.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python

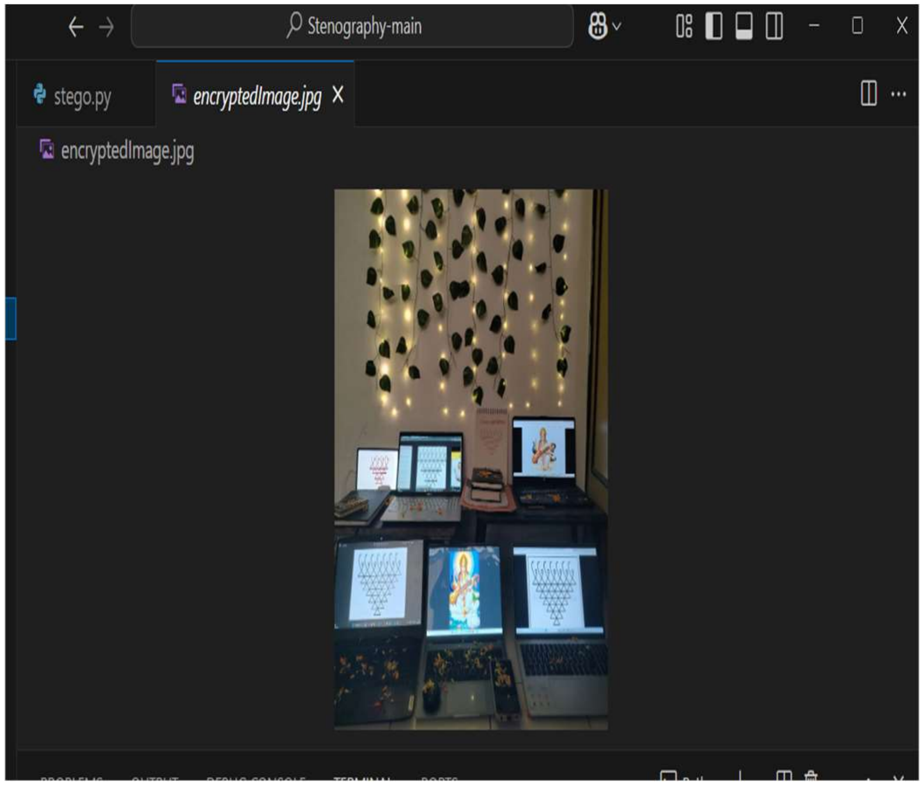
```python
import cv2
import os

# Load imagey
img = cv2.imread("Stenography-main/mypic.jpg")


# Check if image is loaded
if img is None:
    print("Error: Could not load image. Check the file path!")
    exit()

msg = input("Enter secret message: ")
```

```
n/Python313/python.exe c:/Users/Pradnya/Downloads/Stenography-main/Stenography-main/stego.py
Enter secret message: 1234
Enter a passcode: 45
Enter passcode for Decryption: 45
Decrypted message: 1234
PS C:\Users\Pradnya\Downloads\Stenography-main> & C:/Users/Pradnya/AppData/Local/Programs/Pytho
n/Python313/python.exe c:/Users/Pradnya/Downloads/Stenography-main/Stenography-main/stego.py
Enter secret message: hello
Enter a passcode: 34
Enter passcode for Decryption: 34
Decrypted message: hello
PS C:\Users\Pradnya\Downloads\Stenography-main>
```



CONCLUSION

This project successfully addresses the challenge of securely hiding data within an image using LSB-based steganography. By modifying the Least Significant Bit (LSB) of pixel values in BMP images, it ensures that sensitive information remains hidden without noticeable visual changes. The approach maintains data integrity, works efficiently with minimal dependencies, and provides a simple CLI interface for embedding and extracting data.

While the current implementation supports BMP files, it can be extended to support other formats like PNG or use DCT-based methods for JPEG. This project provides a lightweight, fast, and secure solution for data confidentiality, making it useful for cybersecurity, forensics, and private communication.

GITHUB LINK

- https://github.com/Pradnyajagtap-3660/Stenganography.git
- Description:
- This project implements Least Significant Bit (LSB) steganography to hide and extract secret messages within BMP images.

- Repository Contents:
- 📂 Source Code – C programs for embedding and extracting hidden messages.
- 📂 Sample Images – BMP images used for testing steganography.
- 📂 Documentation – Project report, explanations, and implementation details.
- 📂 Presentation – PPT slides summarizing the project.

edunet foundation

FUTURE SCOPE(OPTIONAL)

- 🖥️ Graphical User Interface (GUI) – Develop a user-friendly GUI to make embedding and extracting messages easier.
- 📂 Drag and Drop Support – Allow users to select images and messages through a simple drag-and-drop interface.
- 🛠️ Customization Options – Provide settings for users to adjust the encoding method and choose different bit depths.
- 📊 Real-time Preview – Show a preview of the image before and after steganography is applied.

THANK YOU

edunet
foundation