Cambodia Academy of Digital Technology

Course: **Cryptography**

Lecture: **Mr. Meas Sothearath**

Student: **Sorn Sotheara**

Group: 2

ID: IDTB100310

IMAGE STEGANOGRAPHY SYSTEM

---

1. INTRODUCTION

In the modern digital era, information is exchanged rapidly through the internet, which increases the risk of data leakage and unauthorized access. While cryptography is commonly used to protect information by encrypting it, encrypted data can still attract attention because it is obvious that the message is protected.

Steganography is a technique that hides secret information inside another file such as an image, audio, or video, so that the existence of the message is not easily noticed. In this project, image steganography is used to hide secret text messages inside digital images using the Least Significant Bit (LSB) method.

This project focuses on implementing a simple image steganography system using Python. The system allows users to hide and retrieve secret messages from images through a command-line interface.

2. OBJECTIVES

The main objectives of this project are:

- To understand the concept of image steganography
- To hide secret text messages inside digital images
- To extract hidden messages from stego-images

- To apply cryptography-related concepts such as data hiding
- To implement a simple and user-friendly command-line program

3. SYSTEM OVERVIEW

The Image Steganography System allows users to embed secret text into an image and later retrieve it using the same program. The system operates through a Command-Line Interface (CLI), where users can choose to either encode or decode a message.

The system uses the Least Significant Bit (LSB) technique, where each bit of the secret message is stored in the least significant bit of image pixel values. This approach ensures that the image looks almost identical to the original image after encoding.

4. TECHNOLOGIES USED

- **Programming Language:**
  - ✓ Python 3.10+
- **Libraries Used:**
  - ✓ Pillow (PIL) – for image processing and pixel manipulation
- **Technique Used:**
  - ✓ Least Significant Bit (LSB) image steganography

5. PROJECT STRUCTURE

The project consists of the following files:

- ***main.py*** – Main program that provides menu options
- ***steg_encode.py*** – Handles hiding secret messages in images
- ***steg_decode.py*** – Handles extracting hidden messages from images
- ***images/*** – Folder for storing encoded images
- ***README.md*** – Project documentation

6. COMMAND-LINE INTERFACE (main.py)

The command-line interface provides a simple menu with two main options:

1. Hide a secret message inside an image
2. Extract a hidden message from an image

The user interacts with the program by entering:

- Image path
- Secret message (for encoding)
- Output image name

This interface is suitable for beginners and helps users understand how steganography works step by step.

7. ENCODING PROCESS (steg_encode.py)

The encoding process works as follows:

1. The input image is loaded and converted to RGB format
2. The secret message is converted into binary data
3. A special <END> marker is added to indicate message termination
4. Each bit of the binary message is embedded into the least significant bit of the red channel
5. The modified image is saved as a new encoded image

The encoded image appears visually unchanged to the human eye.

8. DECODING PROCESS (steg_decode.py)

The decoding process follows these steps:

1. The encoded image is loaded
2. The least significant bits of pixel values are read
3. Binary data is reconstructed into characters
4. The process stops when the <END> marker is detected
5. The original hidden message is displayed to the user

9. SECURITY ANALYSIS

The system provides basic security by hiding messages inside images, making them difficult to detect by casual observers. Since the message is not encrypted, the main protection comes from data concealment rather than strong cryptographic security.

This project focuses on learning steganography concepts rather than implementing advanced encryption techniques.

10. LIMITATIONS

- The hidden message is not encrypted
- Message size depends on image resolution
- Large messages may not fit in small images
- Advanced steganalysis tools may detect hidden data

11. FUTURE IMPROVEMENTS

Possible future enhancements include:

- Encrypting the message before hiding it
- Adding password-protected decoding
- Using multiple color channels (RGB)
- Developing a graphical user interface (GUI)
- Supporting audio or video steganography

12. CONCLUSION

This project successfully demonstrates how image steganography can be used to hide secret messages inside digital images. By using the Least Significant Bit (LSB) technique, information can be concealed without noticeably altering the image.

Although the system uses a simple approach, it effectively explains the core principles of steganography and secure data hiding. The project helped improve understanding of cryptography-related concepts and practical implementation using Python.