COMP 8505
Assignment 1
Report

Harman Dhillon
A00994245
Jan 26th, 2026

Purpose

Accept command line input that specifies the PNG image, a file to hide or extract and an encryption key. The program will then securely encrypt the data and embeds it into a PNG image using the Least Significant Bit (LSB) steganography or reverses the process to recover the original file.

Requirements

Task	Status
Reads a file (e.g. text, image, audio) containing the message to be hidden.	Fully implemented
Encrypts the message using a symmetric encryption algorithm.	Fully implemented
Embeds the encrypted data into the least significant bits of the pixel values in a PNG image.	Fully implemented
Extracts the encrypted data from the PNG image.	Fully implemented
Decrypts the extracted data to retrieve the original message.	Fully implemented
Outputs the decoded message to the console or saves it to a file.	Fully implemented

Platforms

tested on:

- Omarchy 3.1.1

Language

- ISO C17
- Compiles with Cmake

Documents

- Design
- Testing
- User Guide

These documents can be found in the report folder of this submission.

Findings

I noticed that stego images are often significantly larger than their originals. That is especially true when the original image is highly compressable, such as one that is all white or all black. Even though I am only changing the Least Significant Bit (LSB), these changes introduce noise that the PNG compression algorithms do not account for. Because of this, the stego images data becomes less efficient leading to larger file sizes.

Despite modifications to pixel data, there was no visible difference when observing the original PNG image and their stego counterpart even though I changed the LSB for all 3 parts of the pixel (RGB).

The maximum embeddable payload is constrained by the image resolution and color format. For RGB images, one bit per color channel per pixel is available, while RGBA images provide the same usable capacity due to the alpha channel being intentionally excluded. Payloads exceeding the available capacity are correctly detected and rejected, preventing partial or corrupted embeddings.

Overall, the system successfully meets its design goals by securely embedding encrypted data into PNG images while maintaining visual integrity.