# ****Internship Report****

## ****Project Title:****

**Steganographic File Integrity Checker**

### ****Intern Details****

**Name:** Ishan Chowdhury

**Intern ID:** 159

**Team Name:** SkullFaced

## ****1. Introduction****

In today’s digital world, file sharing is an integral part of communication. However, files are often **tampered with** or **corrupted** during transfer, which may lead to data loss, security breaches, or misinformation.

To address this, we have developed a **Steganographic File Integrity Checker** — a simple yet effective tool that **embeds a file's SHA-256 hash** inside an image using **steganography**.

By hiding the hash within an image, we ensure that **file verification data remains concealed** from unauthorized access, thereby improving both **security** and **integrity**.

## ****2. Objective****

The main objectives of this project are:

To ensure **data integrity** by verifying files for tampering.

To use **steganography** for securely hiding hash values.

To develop a **Python-based CLI tool** that is simple, secure, and effective.

To provide a demonstration for embedding, extracting, and verifying file hashes.

## ****3. Technologies Used****

| **Technology** | **Purpose** |
| --- | --- |
| **Python 3.13** | Core programming language |
| **Pillow (PIL)** | Image processing and steganography |
| **hashlib** | SHA-256 hash generation |
| **unittest** | Automated testing |
| **Command Line** | CLI-based project interface |

## ****4. System Requirements****

### ****Hardware Requirements****

Processor: Intel/AMD Dual Core or higher

RAM: 4 GB minimum

Storage: 100 MB free space

### ****Software Requirements****

Operating System: Windows / Linux / macOS

Python 3.10+

Required Python libraries: pillow

## ****5. Project Architecture****

Stenographic\_File\_Integrity\_Checker/

│

├── stegcheck.py # Main CLI tool

├── steg\_png.py # Steganography functions

├── hash\_utils.py # Hash generation & verification

├── test\_stegcheck.py # Automated unit tests

├── sample.txt # Sample file for testing

├── cover.png # Example cover image

├── README.md # Documentation

└── Internship\_Report.pdf # Final report

## ****6. Project Workflow****

### ****Step 1 — Embed a File Hash****

We calculate the SHA-256 hash of a file and **embed** it into an image file.

**Command:**

python stegcheck.py embed cover.png stego.png sample.txt

**Output:**

✅ Hash successfully embedded into stego.png

### ****Step 2 — Extract Hash****

We **extract** the hidden hash from the modified image.

**Command:**

python stegcheck.py extract stego.png

**Output:**

Extracted hash: b2956a2d698bac01e855fb3dd3c4813eda8f8ac7988869eccb7a222ab283ae64

### ****Step 3 — Verify File Integrity****

We verify whether the file has been **tampered** with.

**Command:**

python stegcheck.py verify stego.png sample.txt

**Output if file is unchanged:**

✅ File integrity verified. No tampering detected.

**Output if file is tampered:**

❌ File integrity verification failed! Tampering detected.

## ****7. Testing****

We used **unittest** to automate testing of core functions.

**Run tests:**

python test\_stegcheck.py

**Expected Output:**

...

----------------------------------------------------------------------

Ran 3 tests in 1.45s

OK

## ****8. Results****

Successfully implemented **steganographic embedding** of file hashes.

Achieved **accurate detection** of file tampering.

Verified correctness through **unit tests**.

## ****9. Applications****

**Data Integrity**: Ensures files are unchanged during transfers.

**Digital Forensics**: Verifies authenticity of digital evidence.

**Secure Communications**: Hides verification data inside images.

## ****10. Future Enhancements****

Support for **JPEG, BMP, and GIF** image formats.

Add **GUI-based interface** for better usability.

Implement **AES encryption** of file hashes before embedding.

Build **cloud integration** for online verification.

## ****11. Conclusion****

The **Steganographic File Integrity Checker** provides a secure, efficient, and user-friendly way to verify **data integrity** without exposing sensitive information.

This project demonstrates how **steganography** and **hashing** can be combined to develop a robust integrity-checking system.

## ****12. References****

[Python Documentation](https://docs.python.org/3/)

[Pillow Library](https://pillow.readthedocs.io/)

[SHA-256 Hashing](https://en.wikipedia.org/wiki/SHA-2)

## ****13. Intern Declaration****

I, **Ishan Chowdhury** (Intern ID **159**), hereby declare that the project entitled **“Steganographic File Integrity Checker”** has been completed by me under the guidance of **DigiSuraksha** as part of my internship with **Team SkullFaced**.

**Date:** 24/08/2025