Proof of Concept (PoC) – Stenographic File Integrity Checker

Name: Chanchal Teli
Intern ID: 125

This PoC demonstrates a File Integrity Checker that hides cryptographic hashes (SHA256) inside cover files (images) using steganography. The tool allows generating file hashes, embedding them into images, and later extracting and verifying them.

Required Modules
1. hashlib → for SHA256 hashing of files.
2. Pillow (PIL) → for image manipulation and steganography.

Install using:

pip install pillow

Main Parts of the PoC Code
1. Hashing Function → Computes SHA256 hash of the target file.
2. Embedding Function
   → Encodes the hash into the least significant bits (LSB) of image pixels.
3. Extraction Function
   → Retrieves hidden bits back from the image to reconstruct the hash.
4. Verification Function
   → Compares the extracted hash with the current hash to check file integrity.

How to Run (Step by Step)

1. Save the Python script as stenochecker.py.
2. Place a target file (e.g., report.pdf) and a cover image (e.g., cover.png) in the same folder.
3. Open a terminal/command prompt in that folder.
4. To embed the hash:

```
python stenochecker.py embed report.pdf cover.png
```

→ Output: Hash embedded into stego_output.png

5. To verify integrity:

```
python stenochecker.py verify report.pdf stego_output.png
```

Example Output

Case 1 – Intact File
[*] Extracted Hash: 9f86d081884c7d659a2fe...
[*] Current File Hash: 9f86d081884c7d659a2fe...
[+] File integrity verified. No modification detected.

Case 2 – Modified File
[*] Extracted Hash: 9f86d081884c7d659a2fe...
[*] Current File Hash: 1a4d55aaf4275f...
[-] Integrity check failed! File has been modified.

Importance of This PoC
- Ensures file authenticity & integrity without maintaining separate manifest files.
- Provides a covert mechanism (steganography) to store reference hashes in images.
- Useful in digital forensics, enterprise security, and document authenticity checks.- Even a 1-bit modification in the target file will be detected.