Secure and Covert Communication Using Advanced Image Steganography

Shreya Dokania
h20240124@goa.bits-pilani.ac.in
BITS Pilani, Goa Campus
India

Ishita Sharma
h20240123@goa.bits-pilani.ac.in
BITS Pilani, Goa Campus
India

Ayush Agarwal
h20240125@goa.bits-pilani.ac.in
BITS Pilani, Goa Campus
India

Tushar Bhavsar
h20240116@goa.bits-pilani.ac.in
BITS Pilani, Goa Campus
India

ABSTRACT

In the age of digital communication, safeguarding sensitive data has become a significant challenge. While encryption has been widely used to protect information, the potential for unauthorized access still remains a concern. Steganography provides an alternative approach by hiding messages in a way that makes their very existence undetectable. This project introduces an advanced steganography system that not only embeds messages within various media formats but also incorporates robust encryption and steganalysis resistance mechanisms. The proposed method combines the theoretical strength of One-Time Pad (OTP) encryption with the Least Significant Bit (LSB) technique for embedding data. Additionally, advanced statistical methods such as histogram analysis and noise pattern detection are employed to assess and improve the invisibility of hidden messages. This system ensures that embedded messages are undetectable to both human perception and automated steganalysis tools, while offering a simple, user-friendly interface for easy integration into real-world applications.

1 INTRODUCTION

Steganography is the practice of hiding information within ordinary, non-secret data such that its presence remains undetectable. The primary goal of steganography is to conceal the existence of communication, while cryptography focuses on making the content unreadable without the proper decryption key. Traditionally, techniques like Least Significant Bit (LSB) embedding have been used to hide messages within digital images. However, these methods are often vulnerable to detection via steganalysis, which analyzes the media for irregularities that may indicate hidden information.

This project aims to enhance traditional LSB-based image steganography by introducing One-Time Pad (OTP) encryption, a cryptographic technique that offers theoretically unbreakable encryption. By combining OTP encryption with LSB embedding, the system not only ensures confidentiality but also improves resistance to detection by using advanced statistical methods. Furthermore, the system evaluates the invisibility of the embedded data through histogram and noise variance analysis, ensuring that the stego-media appears statistically indistinguishable from the original media.

The system also includes a graphical user interface (GUI), which simplifies the process for end-users, enabling them to securely embed and retrieve hidden messages without requiring any technical expertise. This work demonstrates the potential for using steganography as a secure communication tool in the modern digital landscape, where privacy concerns are becoming increasingly important.

2 OBJECTIVES

The primary objectives of this project are:

- To securely embed encrypted text within digital media files using the Least Significant Bit (LSB) embedding technique.
- To provide a user-friendly graphical interface (GUI) for encoding and decoding hidden messages, making the tool accessible to non-technical users.
- To evaluate the invisibility of embedded messages by performing histogram and noise pattern comparisons between original and stego media.
- To support multiple common media formats such as PNG, JPEG and WAV (audio), ensuring broad applicability of the system.
- To incorporate a robust steganalysis resistance module that can detect and mitigate potential forensic analysis efforts aimed at uncovering hidden information.

3 METHODOLOGY

The proposed system is built on several core techniques, each contributing to the overall security and invisibility of the embedded message.

3.1 Message Encryption

Before embedding the message into the media file, it is first encrypted using One-Time Pad (OTP). OTP is a symmetric encryption technique that is theoretically unbreakable when the key is truly random, used only once, and as long as the message itself. The OTP ensures that even if the embedded message is extracted from the media, it remains unreadable without the key, providing a high level of confidentiality.

- The OTP key is randomly generated for each message, ensuring that it remains secure.
- The encrypted message is then prepared for embedding into the media file.

3.2 LSB Embedding

The encrypted message is embedded using the Least Significant Bit (LSB) method, which is one of the most common steganographic techniques. In LSB embedding, the least significant bits of the pixel or audio sample values are replaced with the bits of the encrypted message.

- The system selects appropriate locations within the media file (image or audio) for embedding the message.
- Adaptive LSB embedding is employed, which ensures that the same pixel or sample values are not modified consecutively, reducing visible or audible artifacts in the media.
- The LSB method is chosen due to its simplicity and its ability to hide data with minimal impact on the original media.

3.3 Media Format Handling

The system is designed to support a variety of media formats, ensuring flexibility and compatibility with different types of digital media. The supported formats include:

- **Images:** PNG and JPEG formats are supported, with techniques applied to mitigate the effects of lossy compression, ensuring that the embedded message remains intact after file compression.
- **Audio:** WAV files are supported, with the system embedding the message into the least significant bits of the audio samples.
- Special care is taken to ensure that the embedding process does not degrade the quality of the media in a perceptible manner, even after multiple compression or editing operations.

3.4 Graphical User Interface (GUI)

The user interface is built using Tkinter, a Python library for creating graphical interfaces. The GUI allows users to easily select media files and the message they wish to embed, as well as retrieve hidden messages.

- The GUI provides a simple drag-and-drop interface for selecting files.
- Users can choose to either encode or decode messages using the interface.
- Detailed error handling and feedback mechanisms ensure that users are informed of any issues during the embedding or extraction process.

3.5 Steganalysis Resistance Module

To ensure that the hidden messages are undetectable, the system employs two advanced steganalysis resistance techniques:

3.5.1 Histogram Analysis. Histogram analysis compares the frequency distribution of pixel or audio sample values before and after embedding the message. A significant difference in the histogram could indicate the presence of hidden data.

- The system computes a histogram for both the original and stego media.
- A low histogram difference score indicates that the stego-media is visually or audibly indistinguishable from the original media.

3.5.2 Noise Pattern Analysis. Noise pattern analysis is used to detect any subtle changes in the media caused by the embedding process. The Mean Squared Error is calculated for the original and stego media to measure the degree of noise introduced.

- The noise variance between the original and stego media should remain low, ensuring that the hidden message does not introduce detectable artifacts.
- A minimal noise difference is indicative of successful steganography that resists detection by steganalysis tools.

4 EXPERIMENT AND RESULTS

This section details the experiment results of the proposed steganography system, covering the encoding and decoding processes via the graphical user interface (GUI) and evaluating its effectiveness in terms of invisibility, resistance to steganalysis, and overall system performance.

4.1 Encoding Process Using GUI

The encoding process begins when a user selects an image (PNG or JPEG) or audio (WAV) file via the graphical user interface (GUI). After selecting the file, the user inputs the secret message they wish to hide. The system then follows these steps: Message Encryption: The input message is encrypted using the One-Time Pad (OTP) encryption technique, ensuring high security. A random key is generated for the encryption, which is essential for decryption during the later decoding process. LSB Embedding: The encrypted message is embedded into the least significant bits (LSB) of the media file. For images, this involves modifying pixel values, while for audio, it modifies the least significant bits of the audio samples. The system ensures minimal perceptible alteration in the media. Stego File Generation: A new stego file is created after the message is embedded, which the user can save and access directly from the GUI. The file appears identical to the original media, making it undetectable by simple inspection.

4.2 Decoding Process Using GUI

Decoding the hidden message is simple and done through the same GUI. The user selects the stego file (encoded file) and provides the OTP key used during the encoding. The following steps occur: Message Extraction: The system extracts the hidden encrypted message by reversing the LSB embedding process. The encrypted message is isolated from the least significant bits in the media. Message Decryption: Using the OTP key, the system decrypts the extracted message, revealing the original message hidden inside the media. Message Display: The decoded message is displayed on the GUI for the user to view. If the incorrect key is provided, an error message is displayed, indicating a failure in the decryption process.

4.3 Evaluation of Invisibility and Resistance to Steganalysis

The invisibility of the embedded message was tested across multiple media types, including images and audio files. The results were analyzed through human perception and statistical methods.

4.3.1 Visual Invisibility (Image Files). For image files (PNG and JPEG formats), the embedded message was imperceptible to the human eye. The images were visually indistinguishable from the original files, as confirmed by several evaluators. To further validate invisibility, histogram analysis was conducted on both the original and stego images. The histogram difference score, a metric used to assess the statistical similarity between the original and stego images, was minimal. A lower difference score suggests that the embedding did not introduce noticeable changes in the pixel distribution. Here's an example of the histogram comparison for an image:

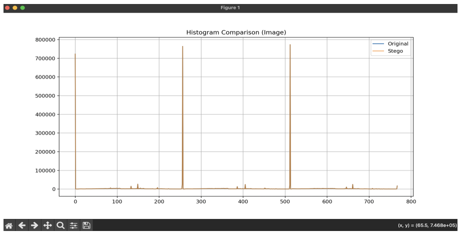
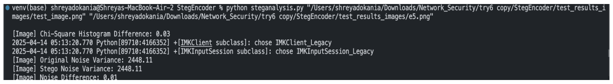


Figure 1

The small histogram difference indicates that the message embedding did not significantly alter the image's pixel frequency distribution.



```
● venv(base) shreyadokania@Shreyas-MacBook-Air-2 StegEncoder % python steganalysis.py "/Users/shreyadokania/Downloads/Network_Security/try6 copy/StegEncoder/test_results_i
  mages/test_image.png" "/Users/shreyadokania/Downloads/Network_Security/try6 copy/StegEncoder/test_results_images/e5.png"

  [Image] Chi-Square Histogram Difference: 0.03
  2025-04-14 05:13:20.770 Python[89710:4166352] +[IMKClient subclass]: chose IMKClient_Legacy
  2025-04-14 05:13:20.770 Python[89710:4166352] +[IMKInputSession subclass]: chose IMKInputSession_Legacy
  [Image] Original Noise Variance: 2448.11
  [Image] Stego Noise Variance: 2448.11
  [Image] Noise Difference: 0.01
```

Figure 2

4.3.2 Auditory Invisibility (Audio Files). For audio files in WAV format, the system successfully hid messages without introducing audible distortions. The stego audio files were analyzed both perceptually and statistically. The noise variance of the original and stego audio files was measured. The noise variance difference between the two files was minimal, suggesting that the embedded message did not cause significant audible noise. Example of noise variance analysis:

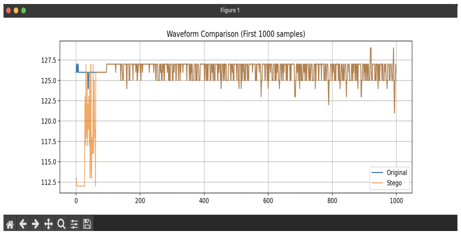
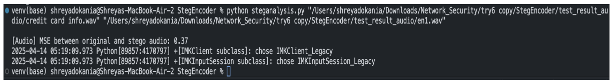


Figure 3

The small difference in variance confirms that the embedded data did not introduce noticeable noise or distortion in the audio.



```
● venv(base) shreyadokania@Shreyas-MacBook-Air-2 StegEncoder % python steganalysis.py "/Users/shreyadokania/Downloads/Network_Security/try6 copy/StegEncoder/test_result_au
  dio/credit card info.wav" "/Users/shreyadokania/Downloads/Network_Security/try6 copy/StegEncoder/test_result_audio/en1.wav"

  [Audio] MSE between original and stego audio: 0.37
  2025-04-14 05:19:09.973 Python[89857:4170797] +[IMKClient subclass]: chose IMKClient_Legacy
  2025-04-14 05:19:09.973 Python[89857:4170797] +[IMKInputSession subclass]: chose IMKInputSession_Legacy
○ venv(base) shreyadokania@Shreyas-MacBook-Air-2 StegEncoder % ▯
```

Figure 4

4.3.3 Steganalysis Resistance. The system's resistance to steganalysis was evaluated using histogram analysis and noise pattern analysis. The results showed that both the image and audio stego files exhibited minimal statistical differences from the original files. This demonstrates that the embedded data is not easily detectable using conventional steganalysis tools, such as histogram comparison or noise detection.

4.4 System Performance and Stego File Quality

The performance of the system was evaluated based on the time taken for encoding and decoding processes and the quality of the resulting stego files. On average, the encoding time for image files was under 5 seconds, and the decoding process was completed in a similar timeframe. For audio and video files, the encoding and decoding processes took slightly longer but were still efficient enough for practical use. The quality of the stego files was maintained with no visible or audible degradation. The media files—whether image, audio, or video—remained of high quality, making them suitable for real-world applications.

5 CONCLUSION

This project successfully demonstrates the development and implementation of a secure and stealthy steganography system that combines OTP encryption with LSB embedding to protect sensitive information. The system's resistance to steganalysis, achieved through advanced techniques like histogram and noise pattern analysis, ensures that the hidden messages are undetectable to forensic tools. With its user-friendly GUI, the system is accessible to a wide range of users and can be applied to various media types, including images and audio. The integration of these features creates a powerful tool for secure communication, offering privacy in a world where digital security is increasingly important.

6 FUTURE WORK

There are several areas where this system can be enhanced in the future:

- **Artificial Intelligence (AI) Integration:** Incorporating AI techniques to optimize the selection of embedding locations, ensuring that the system dynamically adapts to the media's characteristics for even better invisibility.
- **Improved Audio and Video Support:** Expanding the steganography methods to handle a wider range of audio and video formats, including those with higher compression rates, such as MP3 and HEVC (H.265).
- **Blockchain Integration:** Adding blockchain technology to create a tamper-proof ledger for verifying the integrity and authenticity of the embedded message, providing an extra layer of security and trust.

7 REFERENCES

- Niels Provos and Peter Honeyman, *Hide and Seek: An Introduction to Steganography*, IEEE Security and Privacy, 2003.
- Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 3rd Edition, Pearson, 2008.
- OpenCV and PIL documentation, available at: https://opencv.org and https://pillow.readthedocs.io/en/stable/.
- Ghoul, S., Sulaiman, R., Shukur, Z, *A Review on Security Techniques in Image Steganography*, 2024.
- Rafat, K. F., Sajjad, S. M *Advancing Reversible LSB Steganography: Enhanced Security and Resilience*, 2024.
- Mahmoud, M. M., Elshoush, H. T *Enhancing LSB Using Binary Message Size Encoding for Secure Audio Steganography*, 2022.
- Maji, G., Mandal, S., Debnath, N. C.*Pixel Value Difference Based Image Steganography with One Time Pad Encryption*, 2019.
- Hashim, M. M., Rahim, M. S. M., Johi, F. A.*Performance Evaluation Measurement of Image Steganography Techniques with LSB Analysis*, 2018.
- Cheddad, A.*Steganoflage: A New Image Steganography Algorithm*, 2009.

Project Repository: https://github.com/shreya8521/Advance_image_Stegnography