Secret Communication Using Cryptography and Steganography

Youssef Mohamed Abdelshahid
ID: 2022/05890

December 28, 2024

Introduction

In this project, we address the critical issue of secure communication in the realm of network security. The primary goal is to propose and implement a novel technique that combines RSA encryption and Least Significant Bit (LSB) steganography [3]. By merging these two methods, the system ensures both confidentiality and stealth in message transmission. The RSA algorithm encrypts the message, providing a robust layer of security, while LSB steganography embeds the encrypted message into an image, rendering it inconspicuous to potential attackers.

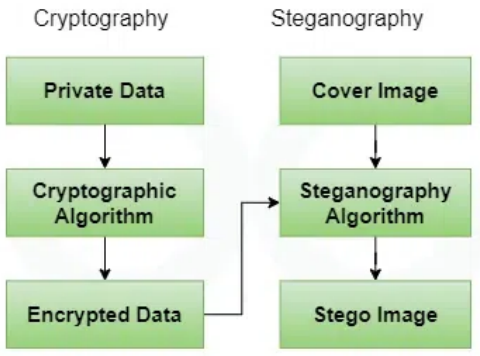

Figure 1: Diagram summarizing the target of the project

Methodology

The methodology is divided into several well-defined phases:

1. Image Acquisition

The user uploads an image to serve as the carrier for the encrypted message. Upon upload, the system performs the following validations to ensure the image meets the requirements for LSB steganography:

- **Supported Format Check:** The image format is checked to ensure compatibility with OpenCV-supported formats, such as '.png', '.jpg', or '.bmp'.
- **Image Integrity Check:** The system verifies whether the uploaded file is a valid and readable image.
- **Capacity Check:** The system calculates the maximum number of bytes that can be embedded in the image based on its dimensions and ensures the size of the entered message (including delimiters) can be accommodated.

The user also inputs the message to be embedded, which is then encrypted and processed for embedding.

2. Message Encryption (RSA Algorithm)

- **Key Generation:** A pair of public and private keys is generated using the RSA algorithm [1].
- **Message Encryption:** The plaintext message is encrypted with the public key, converting it into a secure ciphertext.
- **Hexadecimal Conversion:** The ciphertext is converted into a hexadecimal format for compatibility with the embedding process [1].

3. Data Embedding (LSB Steganography)

The encrypted message in hexadecimal form is embedded into the least significant bits of the pixel values in the carrier image. To mark the end of the embedded data, a delimiter sequence is appended [2].

4. Data Extraction and Decryption

The embedded message is extracted from the image by retrieving the modified bits. The extracted data is decrypted using the RSA private key to recover the original plaintext message.

5. Visual Integrity Check

The steganographic image is compared with the original image to ensure that the embedding process has not visibly altered the image quality.

Testing and Results

Testing Scenarios

The system was tested under various conditions, including:

- Embedding short and long messages in images of varying resolutions and formats.
- Extracting and decrypting messages from steganographic images to verify accuracy.
- Evaluating the impact of embedding on the visual quality of the carrier image.

Visual Comparison

To illustrate the effectiveness of the LSB steganography method, the figure below compares the original image with the steganographic image after embedding the encrypted message.

Results Analysis

- **Accuracy:** The encrypted messages were successfully embedded, extracted, and decrypted without any data loss.
- **Stealth:** Visual inspection confirmed that the steganographic images were indistinguishable from the original images.
- **Efficiency:** The embedding and extraction processes were computationally efficient, making the system suitable for real-time applications.

```
Image Steganography:
1. Encode the data
2. Decode the data
3. Exit
Select the option: 1

Encoding...
Upload the image to encode:
Choose Files turtle.png
• turtle.png(image/png) - 136690 bytes, last modified: 6/16/2024 - 100% done
Saving turtle.png to turtle.png
The shape of the image is: (240, 240, 3)
Enter data to be encoded: secret message
Maximum Bytes for encoding: 21600
Encoded image saved as turtle_stego.png

Image Steganography:
1. Encode the data
2. Decode the data
3. Exit
Select the option: 2

Decoding...
Upload the steganographic image to decode:
Choose Files turtle_stego.png
• turtle_stego.png(image/png) - 89302 bytes, last modified: 12/28/2024 - 100% done
Saving turtle_stego.png to turtle_stego (1).png
Decoding message...
Decoded message is: 2d2195e6fb990478e383f3655f0cea89b38ccb9842d90d2fba974395acb4576cb846321323597289aeafceecbad4a3ce51c867f62920f5171f34a6dbea6ace55
Decrypted message: secret message
```

Figure 2: Testing the code

Image before steganography

Image after steganography

Figure 3: Comparison of Original and Steganographic Images

Conclusion

The project successfully demonstrates a secure communication system that integrates RSA encryption and LSB steganography. This dual-layered approach enhances both the confidentiality and stealth of the transmitted messages, making it highly suitable for applications requiring advanced network security.

References

[1] RSA Algorithm: https://www.javatpoint.com/rsa-encryption-algorithm

[2] LSB Steganography: https://en.wikipedia.org/wiki/Steganography

[3] Crypto-Steganography research: https://www.researchgate.net/publication/269524396_A_Crypto-Steganography_A_Survey

[4] Python OpenCV Documentation: https://docs.opencv.org/