National Institute of Technology Patna

Department of Computer Science and Engineering

Secure Email With Image Compression

Anurag Bhanu (2206190)

Raunak Raj Tiwari (2247012)

Garima Kushwaha (2206302)

Supervisor: Dr. Kakali Chatterjee

Network Security Project (CS64153)

CONTENTS

I. Introduction

In today's digital world, securely transmitting sensitive data—such as personal messages and images—is essential due to rising cyber threats and surveillance risks. This project, "Secure Image Transfer with Server Relay," presents a secure communication system that leverages hybrid cryptography to ensure confidentiality, integrity, and authenticity in data exchange.

The system combines RSA (asymmetric encryption) for secure key exchange with AES (symmetric encryption) for efficient message and image encryption. To further optimize performance, images are converted to PNG format and compressed losslessly, ensuring reduced data size without compromising image quality.

A standout feature is the relay server, which acts as a passive middleman between sender and receiver. It forwards encrypted data without decrypting it, improving modularity and scalability while preserving end-to-end security.

Additionally, the use of digital signatures ensures that messages are authentic and tamper-proof. The sender signs the payload using their RSA private key, and the receiver verifies it using the sender's public key, providing robust protection against forgery and unauthorized changes. The components developed in this project include:

- A key generation module for RSA keys.
- A sender module that handles message input, image compression, AES encryption, RSA key wrapping, and digital signing.
- A relay server module that securely passes encrypted data between sender and receiver.
- A receiver module that decrypts the data, verifies the digital signature, and displays the received message and image.

Together, these modules demonstrate a full end-to-end secure image and message transfer solution that is cryptographically secure, efficient, and user-friendly.

I. Objective

The primary objectives of this project are:

- To securely transmit text messages and image files between two parties (Alice and Bob) over a network.
- To implement hybrid encryption using RSA for secure key exchange and AES for efficient encryption of message and image data.
- To digitally sign the original payload using RSA to ensure data authenticity and prevent tampering.
- To compress image data before transmission using lossless compression, optimizing the bandwidth and reducing file size.
- To convert images from JPEG or other formats to PNG, ensuring lossless quality and consistent handling during transmission.
- To create a simple and efficient server relay system that intermediates the secure exchange between sender and receiver.

II. Technical Overview

This project implements a secure email communication system using various cryptographic and networking techniques. The architecture is built on the following core technical components:

Cryptographic Concepts:

- **AES (Advanced Encryption Standard):** Used for encrypting the actual message and image payload. It provides fast, secure symmetric encryption.
- **RSA (Rivest–Shamir–Adleman):** Used for securely transmitting the AES key. RSA also supports digital signatures.
- **Digital Signatures:** Sender signs the plaintext using their private key. The receiver verifies this using the sender's public key to ensure data authenticity.

Communication Setup:

- **Server Relay:** A central server accepts connections from both the sender and the receiver. It facilitates direct data transfer between them without reading or altering the content.
- **Socket Programming:** Used for handling the network communication between client-server and peer-to-peer data transfer.

III. Implementation

The implementation is done using Python and is hosted on GitHub: https://github.com/anuragbhanu/Email_With_Image_Compression

A. Modules

- **KeyGenerator.py:** Generates RSA key pairs and stores them securely.
- **Server.py:** Simulates an SMTP-like mail server to route messages.
- **Sender.py:** Encrypts and signs the message and sends it.
- **Receiver.py:** Decrypts and verifies the received message.

B. Working Procedure

1) **Key Generation** `KeyGen.py`:
   - Generates RSA key pairs (public and private) for both the sender (Alice) and receiver (Bob).
   - Keys are saved in PEM format under a keys/ directory.
2) **Sender** `Sender.py` to:
   - Prompts the user to enter a message and optionally attach an image.
   - Compresses the image and formats the payload with the message.
   - Encrypts the entire payload using AES, and then encrypts the AES key using Bob's RSA public key.
   - Signs the original payload using Alice's private key.
   - Sends the encrypted key, IV, signature, and payload to the server.
3) **Receiver** `Receiver.py` to:
   - Connects to the server and receives all transmitted data.
   - Decrypts the AES key using Bob's private key and decrypts the payload using AES.
   - Verifies the digital signature using Alice's public key.
   - Extracts and displays the message; if an image is present, saves it locally.

IV. Result

The system successfully demonstrates secure email communication.Upon execution, the project performs the following:

- **Key Generation:** RSA key pairs generated and stored in PEM format.
- **Message Composition:** Email encrypted with AES, signed with RSA.Allows the sender to securely send a message and optionally an image.
- **Transmission:** Server routed messages via TCP/IP to recipient.
- **Decryption and Verification:** Receiver decrypted content and verified signature.
- **Security Goals Met:** Confidentiality, integrity, authentication, and non-repudiation were achieved.
- **Output:**
  - Confirmation of connection and transfer success.
  - Message display at the receiver's end.
  - Image saved locally as `received_image.png`

V. Conclusion

This project effectively demonstrates secure file transmission over a network using modern cryptographic techniques. The use of a hybrid RSA-AES encryption scheme leverages the performance of symmetric encryption and the secure key exchange benefits of asymmetric encryption. The server relay ensures that both parties can communicate even without knowing each other's IP addresses, promoting decoupling and abstraction.
Key takeaways:

- Only the recipient can decrypt the message.
- Sender authenticity is verified.
- Tampering is detected.
- Sender cannot deny sending the message.

This architecture addresses SMTP's weaknesses using modern cryptographic protocols and demonstrates the viability of secure communication using Python and socket programming.

VI. Future Work

Potential enhancements include:

- **GUI Interface:** Add a graphical interface for user-friendly interaction.
- **End-to-End Encryption without a relay:**Implement direct peer-to-peer communication using NAT traversal.
- **Support for larger files or other file types:** Add chunking and file-type handling for general secure file transfer.
- **TLS-based server communication:** Secure the relay server communication itself using TLS/SSL.
- **Multithreaded Server:** Handle multiple sender-receiver pairs concurrently.

REFERENCES

[1] W. Stallings, ”Cryptography and Network Security: Principles and Practice,” Pearson Education.
[2] B. Schneier, ”Applied Cryptography”, Wiley.
[3] RFC 5751, ”Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2”, https://tools.ietf.org/html/rfc5751
[4] Anurag Bhanu, ”Secure Email System,” GitHub, https://github.com/anuragbhanu/Email_With_Image_Compression