A Novel Image Encryption Scheme Based on a Modified AES Algorithm

Ajay Dileep,Allwin Nebu,Alvin Boby Mathew,Mathew Somy

Absract

This project addresses the critical need for safeguarding sensitive information in digital communication by integrating Image Steganography with a Modified AES-based encryption standard. The approach provides a dual layer of security: encryption transforms the image data into an unreadable format, ensuring confidentiality, while steganography embeds the encrypted data into a carrier image, concealing its presence. This combination ensures both the protection of sensitive data and its invisibility, offering an innovative and robust solution for secure digital communication. MAES encrypts small pixel groups using AES with multiple keys and extra XOR steps, ensuring a uniform histogram and addressing pixel correlation for secure image encryption. It is optimized for images, making it more efficient than standard AES.

Keywords: AES Encryption, Image Steganography, Enhanced Security, Improved Confidentiality, Increased Stealth, Robustness

Hypothesis

Combining Modified AES (MAES) encryption with image steganography will provide a robust two-layer security solution for sensitive data. MAES encryption works by encrypting data using a modified version of the AES algorithm, where small pixel groups are encrypted with multiple keys and additional XOR operations. This enhances security by ensuring that the data is transformed into an unreadable format, making it more resistant to attacks compared to standard AES. The encrypted data is then hidden within an image using steganography, where the data is embedded in the image's pixel values without altering the image's visible appearance. This concealment makes it difficult for attackers to detect the presence of sensitive data, ensuring both the confidentiality and invisibility of the information. This dual approach improves data security by both encrypting the data and hiding it in an image. The optimized MAES encryption ensures efficient and secure handling of image data, addressing pixel correlation and maintaining image quality. As a result, this method provides a more secure, reliable, and efficient way to transmit sensitive information without compromising either data integrity or image appearance.

Methodology

The program begins by prompting the user to input a shared key for encryption and decryption. This key is then padded to a 32-character length to ensure it meets the AES encryption requirements and is encoded into a UTF-8 format. The user is then given the option to either hide data in an image or retrieve hidden data from an image. If the user chooses to hide data, they are asked to specify the type of data (either text or audio), which is read into memory. The selected data is then encrypted using the Modified AES (MAES) algorithm, which divides the data into small pixel-sized chunks, applies multiple keys, and uses additional XOR operations to enhance security. The data is padded to fit the AES block size and encrypted using the modified AES algorithm in ECB mode, resulting in secure, unreadable data.

Once the data is encrypted, the program proceeds to embed the encrypted data into a chosen carrier image. The image is opened and converted into a numpy array to access the pixel values. The encrypted data is then converted into binary format, and each bit of the binary data is embedded into the least significant bit (LSB) of corresponding image pixels. This process ensures the image appears unchanged to the human

eye while secretly containing the hidden encrypted data. After embedding, the modified image (stego-image) is saved to the specified output file path, and the user is informed that the data has been successfully hidden within the image.

If the user later chooses to retrieve the hidden data, the program extracts the encrypted data from the stego-image by reading the LSB of each pixel in the image. The extracted binary data is then grouped into bytes to recreate the original encrypted data. This encrypted data is then decrypted using the same shared key that was used during encryption. The MAES decryption algorithm reverses the encryption process, including the XOR operations and key transformations. Once decrypted, the data is unpadded to restore it to its original form, whether as text or audio. Finally, the decrypted data is saved to a file at the user-specified output path, with the data being either written as a text file or saved as an audio file, depending on the original data type. Through these steps, the program ensures that sensitive data is securely encrypted, embedded, and retrieved, offering both confidentiality and invisibility for secure digital communication.

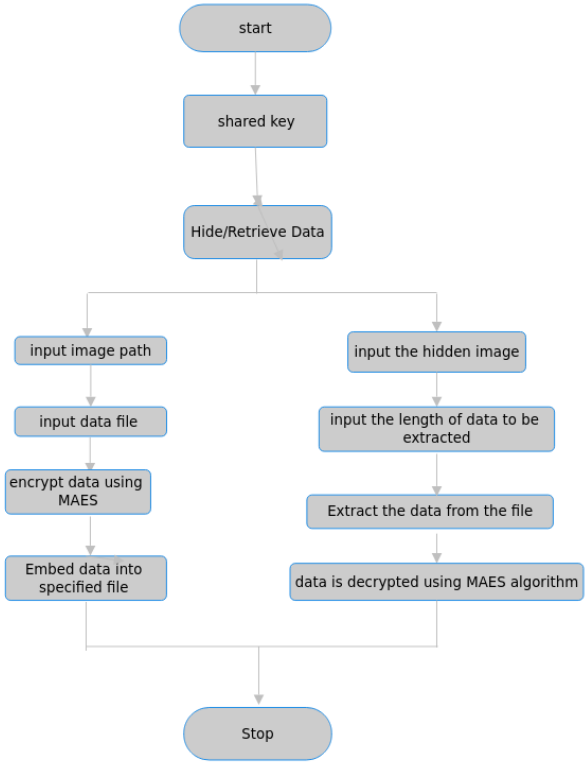

Figure 1: This is an example image.

Feasibility Analysis

0.1 Technical Feasibility

The combination of MAES encryption and image steganography relies on well-established cryptographic techniques and image manipulation methods. The MAES algorithm, being an enhancement of the standard AES encryption, is computationally feasible and suitable for securing small to medium-sized data (text and audio). The steganography process uses the least significant bit (LSB) method to embed encrypted data into an image, a widely used and efficient technique for hiding information. With libraries such as Python Imaging Library (PIL) for image processing and PyCryptodome for AES encryption, the technical implementation is feasible with readily available tools and resources.

Additionally, the system's ability to work with both text and audio files adds to its versatility, making it adaptable for different types of sensitive data. The program requires basic user inputs (file paths and

a shared key), and once the key is provided, the system works autonomously, ensuring that the process is straightforward for users without technical expertise.

0.2 Security Feasibility

The security of the system is largely dependent on the strength of the encryption and the effectiveness of the steganographic technique. MAES encryption adds an extra layer of security over traditional AES by incorporating multiple keys and XOR operations, which makes it more resistant to common cryptographic attacks such as brute force or frequency analysis. The encryption process ensures that even if the stego-image is intercepted, the embedded data remains unreadable without the decryption key.

Steganography hides the encrypted data within an image, making it difficult for attackers to detect the presence of sensitive data. By altering only the least significant bit of the image's pixels, the visual appearance of the image remains largely unchanged, making it harder for an observer to identify that the image contains hidden data. However, the security of the steganographic method can be affected by sophisticated detection algorithms, so the system's effectiveness depends on the size and quality of the carrier image used.

Milestone

- **Jan 16 - Jan 19: Planning** – Objectives, scope, and methodology outlined. Initial project plan developed.
- **Jan 20 - Jan 26: Key Expansion & Encryption/Decryption** – MAES algorithm developed and tested for encryption and decryption.
- **Jan 27 - Feb 2: Embedding/Extracting Data** – Implemented LSB embedding and data extraction methods.
- **Feb 3 - Feb 9: Audio Conversion** – Implemented audio handling for embedding and retrieval.
- **Feb 10 - Feb 16: Main Application Workflow** – Integrated MAES encryption, steganography, and audio handling.
- **Feb 17 - Feb 28: Testing & Integration** – Testing for encryption/decryption accuracy and functionality.
- **Feb 22 - Feb 28: Finalization and Presentation** – Polishing and preparing for final report and presentation.

Work Plan and Task Allocation

- **Allwin Nebu** – Embedding and extraction of image part and audio conversion.
- **Alvin Boby Mathew** – Work on MAES algorithm, lead front-end development, and user interface design.
- **Ajay Dileep** – Lead documentation, user manuals, and technical reports.
- **Mathew Somy** – Oversees testing & integration, contributes to documentation.

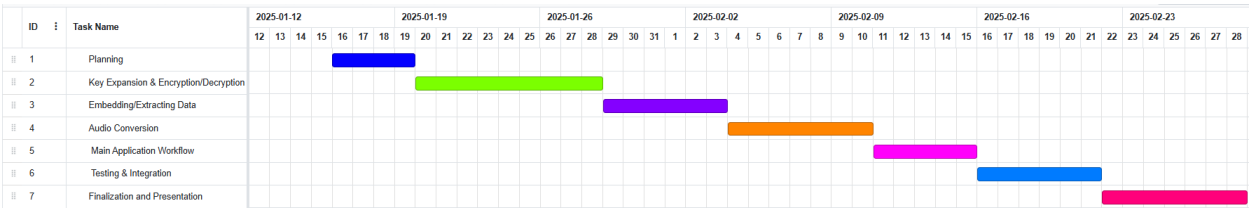

Figure 2: This is an example image.

References

References

[1] Sharma, P. (2020). A new image encryption using modified AES algorithm and its comparison with AES. *International Journal of Engineering Research & Technology (IJERT)*, 9(8), 194–197. Available at: `https://www.ijert.org`