SYMBIOSIS SKILLS AND ROFESSIONAL

UNIVERSITY KIWALE, PUNE

INNOVATIVE PROJECT REPORT

On

From Pixels to Cipher: Image Encryption Mechanism

SUBMITTED BY

Aaditya Jain {2201106030}

Puniyat Jangir {2201106034}

[THIRD YEAR]: B. TECH CSIT (CYBER SECURITY)

AY-2024-25

UNDER THE GUIDANCE OF

Prof. Parul Bhanarkar

School of CSIT (Cyber Security)

Symbiosis Skills and

Professional University

Internal Examiner External Examiner

SCHOOL OF CSIT (CYBER SECURITY)

INNOVATIVE PROJECT REPORT SUBMISSION CERTIFICATE

This is to certify that Mr. Aaditya Jain & Puniyat Jangir has successfully submitted Innovative Project (SEM [V], AY:2024-25) entitled “ From Pixels to Cipher: Image Encryption Mechanism ” Under my supervision, in the partial fulfilment of [B.TECH] CSIT (Cyber Security), Symbiosis Skills and Professional University.

Date: 18/11/2024 Place:

Guide:

Prof. Parul Bhanarkar

School of CSIT (Cyber Security)

Symbiosis Skills and

Professional University

DECLARATION

I affirm that the Innovative Project report titled “From Pixels to Cipher: Image Encryption Mechanism” being submitted in partial fulfilment of the requirements for the award of the Degree of B. TECH CSIT (Cyber Security), is the original work carried out by me. Any part of this work is not submitted for award of any other degree or diploma, either in this or any other Institution. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I declare that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

I, the student of Symbiosis skills and professional University Studying in 3rd year CSIT- Cyber Security (Sem V) express my heartfelt thanks to Symbiosis Skills and Professional university for providing me the opportunity to undertake this 6-month internship. I am also thankful to my mentor “Prof. Parul Bhanarkar” without whose support I would not have been equipped with the sufficient knowledge to undertake this training.

(Signature)

Aaditya Jain

2201106030

DECLARATION & ACKNOWLEDGEMENT

I affirm that the Innovative Project report titled “From Pixels to Cipher: Image Encryption Mechanism” being submitted in partial fulfilment of the requirements for the award of the Degree of B. TECH CSIT (Cyber Security), is the original work carried out by me. Any part of this work is not submitted for award of any other degree or diploma, either in this or any other Institution. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I declare that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

I, the student of Symbiosis skills and professional University Studying in 3rd year CSIT- Cyber Security (Sem V) express my heartfelt thanks to Symbiosis Skills and Professional university for providing me the opportunity to undertake this 6-month internship. I am also thankful to my mentor “Prof. Parul Bhanarkar” without whose support I would not have been equipped with the sufficient knowledge to undertake this training.

(Signature)

Puniyat Jangir

2201106034

Index

|  |  |  |  |
| --- | --- | --- | --- |
| Chapter |  |  | Page No. |
| 1 | Introduction |  |  |
|  | 1.1 | Abstract |  |
|  | 1.2 | Problem Statement |  |
|  | 1.3 | Key Features & Innovations |  |
|  | 1.4 | Literature review |  |
| 2 | Technical Contents |  |  |
|  | 2.1 | Methodology |  |
|  | 2.2 | Tools and Technologies Used |  |
|  | 2.3 | Model / Architecture |  |
| 3 | Details of the implementation |  |  |
|  | 3.1 | Design and Development Steps / Diagram |  |
|  | 3.2 | Testing |  |
|  | 3.3 | Output/Results |  |
| 4 | Conclusion & Future Scope |  |  |
|  | 4.1 | Conclusion |  |
|  | 4.2 | Future Scope |  |
| 5 | References |  |  |

1. **Introduction**

**1.1 Abstract**

In today’s digital world, protecting sensitive data, including images, from unauthorized access is paramount. Image encryption has become a critical technique for securing visual information. This project focuses on the development of a system designed to encrypt and decrypt images using the Advanced Encryption Standard (AES) algorithm in Cipher Block Chaining (CBC) mode. The encryption process ensures that the image remains confidential, and only those with the correct decryption password can access the original content. The project aims to provide an easy-to-use graphical interface for image encryption and decryption tasks, ensuring a seamless user experience for both technical and non-technical users.

The encryption algorithm is built on the widely used AES standard, which is known for its robustness and efficiency. It employs a 256-bit key derived from a user-inputted password. The system also handles various image formats such as PNG, JPEG, and BMP, ensuring broad compatibility. This report discusses the methodology, tools, and technologies used, the design and development of the encryption system, and the testing and validation of its functionality.

**1.2 Problem Statement and Objective**

As more data is being shared digitally, the need for securing sensitive information has grown significantly. Visual data, such as images, contains personal and private content that must be protected from unauthorized access. This project aims to solve the problem of image confidentiality by implementing a secure and efficient image encryption system based on AES in CBC mode. The main objectives are:

- To develop an easy-to-use encryption and decryption system for images.

- To ensure that the encryption algorithm is robust, secure, and reliable.

- To maintain compatibility with multiple image formats to make the system versatile.

- To provide a graphical user interface (GUI) to make the system user-friendly.

By achieving these objectives, the system will ensure that sensitive image data remains secure while providing an intuitive interface for end users.

**1.3 Key Features and Innovations**

The system developed in this project comes with several unique features and innovations that set it apart from other image encryption tools. These features are designed to enhance the security, usability, and versatility of the system:

**1. AES Encryption Algorithm**: The system uses AES, a widely accepted encryption standard, in CBC mode to provide a high level of security. CBC mode ensures that each block of encrypted data is dependent on the previous block, adding a layer of complexity to the encryption process.

2. **User-Defined Password for Key Generation**: The encryption key is generated from a password entered by the user. This ensures that the encryption key can vary, providing a more personalized level of security for each encryption.

3. **Graphical User Interface (GUI):** The system features a GUI developed using Tkinter, which allows users to select images for encryption and decryption, enter passwords, and view results without needing to interact with the underlying code. This makes the system accessible to users with no technical background.

4. **Multi-Format Image Support**: The system supports a wide range of image formats, including PNG, JPEG, and BMP, ensuring compatibility with most common image types.

5. **Padding for Security**: To ensure the encrypted image data is correctly aligned, the system uses padding to make the data length a multiple of 16 bytes, as required by the AES algorithm in CBC mode.

6. **Real-Time Encryption/Decryption**: The encryption and decryption processes are performed in real-time, allowing users to quickly secure their images or restore them to their original form.

**1.4 Literature Review**

Image encryption has been a topic of significant research, particularly in the context of securing digital data. Early methods for encrypting images often relied on basic substitution ciphers and were found to be vulnerable to attacks. Over time, more sophisticated algorithms such as the Data Encryption Standard (DES) and AES have been introduced, offering much stronger protection.

AES, in particular, has become the standard for secure encryption due to its robustness and efficiency. It operates on fixed-size blocks (128 bits) and supports multiple key sizes (128, 192, or 256 bits). The use of AES in CBC mode adds further security by chaining the encryption of each block with the previous one, making it difficult for attackers to predict patterns in the encrypted data.

Several studies have explored the use of AES in image encryption, demonstrating its ability to provide secure and efficient encryption for both small and large images. However, challenges such as padding the image data to ensure it fits within the block size requirements of AES and managing the encryption keys have been common hurdles.

This project builds upon existing research by addressing these challenges and implementing a robust solution that includes key management, padding, and a user-friendly interface for image encryption and decryption.

**2. Technical Contents**

**2.1 Methodology**

The methodology for this project follows a systematic approach to encrypt and decrypt images using AES. The process begins by selecting an image using the GUI. The selected image is then opened and converted into raw byte data. A password entered by the user is used to generate a secure AES key. The password is processed through a key derivation function to ensure it has the correct length for AES encryption.

The image data is encrypted using AES in CBC mode, where each block of the image data is encrypted independently, with each subsequent block depending on the previous one. The resulting encrypted data is then saved as a new image file in the same format and dimensions as the original. The decryption process follows a similar approach, reversing the encryption steps to restore the original image.

The system ensures that the data is padded to a 16-byte boundary, which is required by the AES algorithm in CBC mode. If the data length is not a multiple of 16 bytes, padding is added before encryption and removed during decryption.

**2.2 Tools and Technologies Used**

The system was developed using the following tools and technologies:

- **Python 3.x**: The primary programming language used to implement the image encryption system.

- **Tkinter:** A standard library in Python for creating the GUI, which allows users to select images, enter passwords, and view encrypted or decrypted images.

- **Pillow (PIL)**: A Python Imaging Library (PIL) that handles image operations, such as opening, converting, and saving images.

- **PyCryptodome**: A Python library that provides cryptographic functions, including AES encryption and decryption, in CBC mode.

- **Operating System**: The system was developed and tested on Linux-based platforms, ensuring compatibility with popular distributions such as Ubuntu.

**2.3 Model / Architecture**

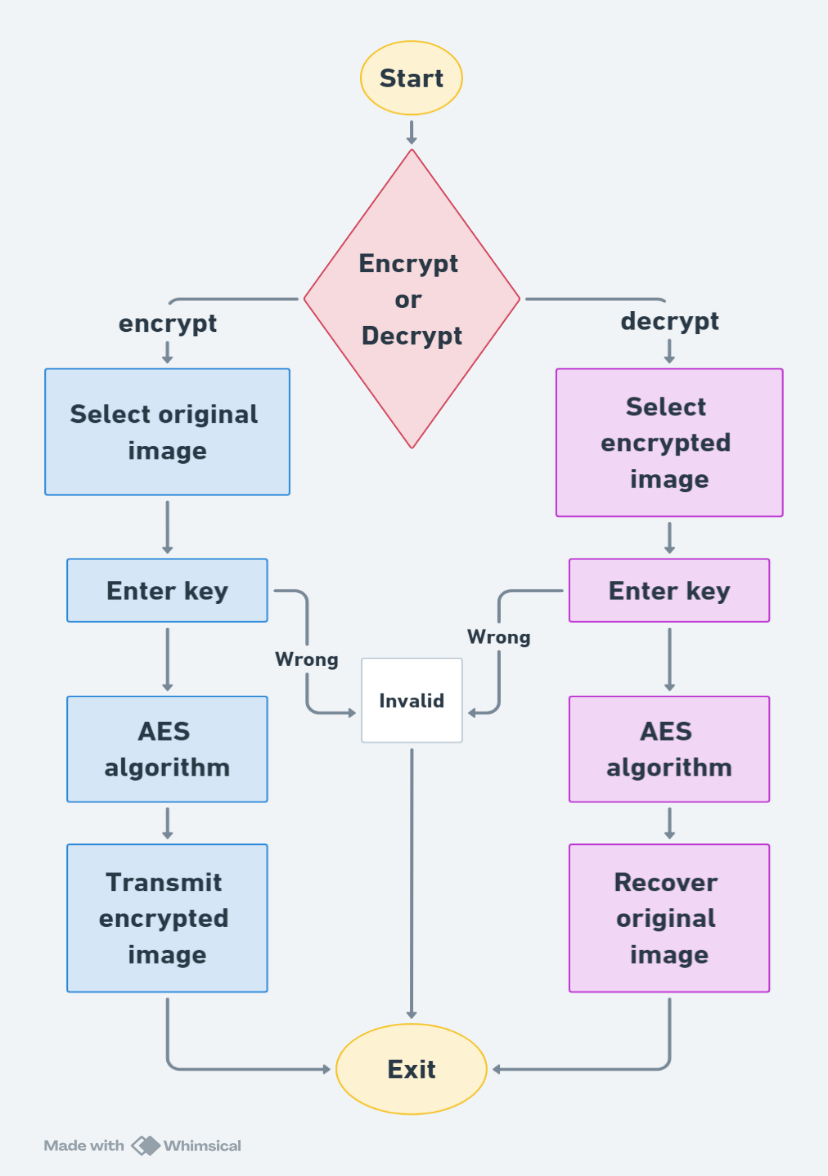
The system architecture consists of three primary components: the user interface, the encryption module, and the decryption module.

1. **User Interface**: The Tkinter-based GUI allows users to interact with the system. They can choose an image to encrypt or decrypt, provide a password, and view the results.

2. **Encryption Module**: The encryption module takes the user-provided password and generates an AES key. The image data is then encrypted using AES in CBC mode.

3. **Decryption Module**: The decryption module reverses the encryption process, using the same password and key to restore the original image data.

The process flow is illustrated below in a simple diagram:



3. **Details of the Implementation**

**3.1 Design and Development Steps / Diagram**

The design of the system follows a modular approach, where each component handles a specific task in the encryption/decryption process. The development steps are as follows:

1. **Image Selection**: The user selects an image from their system using the file dialog in the GUI.

2. **Password Input**: The user enters a password for generating the encryption key.

3. **Image Conversion**: The selected image is converted into byte data for encryption.

4. **AES Encryption**: The byte data is encrypted using AES in CBC mode.

5. **Encrypted Image Output**: The encrypted data is saved as a new image file with the same format and dimensions.

6. **Image Decryption**: The encrypted image is decrypted by reversing the AES encryption process.

7. **Decrypted Image Outpu**t : The original image is restored and displayed.

**3.2 Testing**

The system was tested using a variety of image formats, including PNG, JPEG, and BMP, to ensure compatibility. The following testing steps were performed:

- **Functional Testing:** Ensured that the system correctly encrypts and decrypts images with various formats.

- **Security Testing:** Verified that the encryption method provides adequate security by testing with different passwords.

- **Usability Testing**: Ensured that the GUI is intuitive and easy for users to interact with.

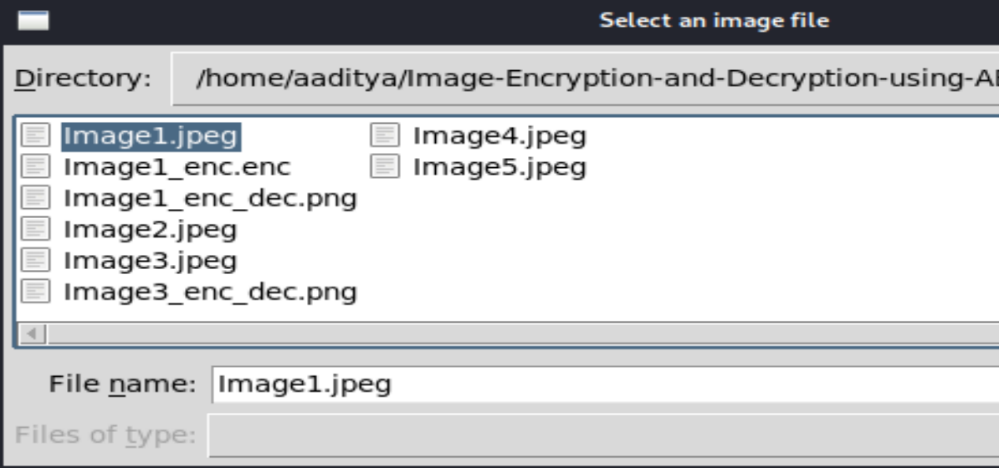
**- Performance Testing**: Assessed the system's performance with large image files to ensure that it works efficiently.

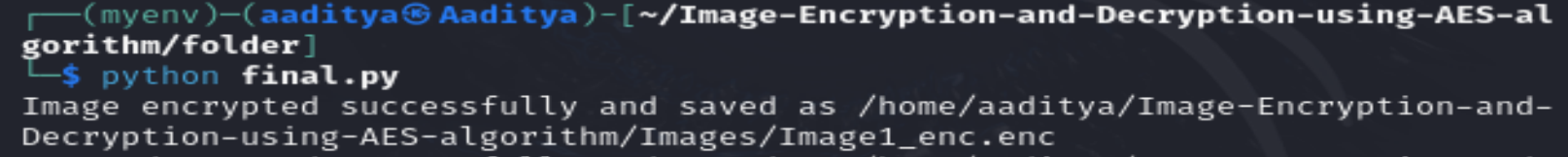
**3.3 Output/Results**

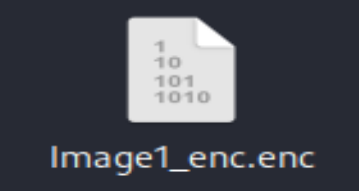
The output of the system includes:

- Encrypted Image : Successfully encrypted images that can only be decrypted using the correct password.



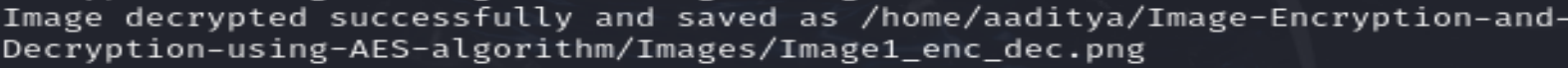


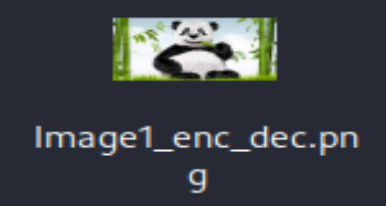




- Decrypted Image : Restored original images after decryption.







- The system was tested with different types of images, including those with varying dimensions and file sizes, and it performed efficiently without any errors.

**4. Conclusion and Future Scope**

**4.1 Conclusion**

The image encryption system developed in this project effectively addresses the need for securing image data using AES encryption in CBC mode. The system provides a user-friendly

GUI for easy interaction, while ensuring robust encryption and decryption processes. The encryption system successfully maintains image quality while ensuring confidentiality, making it a valuable tool for protecting sensitive visual data.

**4.2 Future Scope**

The system can be expanded in the following ways:

- **Cloud Integration**: Integrating with cloud storage services could provide secure online storage for encrypted images.

- **Support for More Formats**: Adding support for more image formats and even other types of media (audio, video).

- **Performance Optimization**: Implementing multi-threading or parallel processing to enhance performance, especially with larger images.

- **Advanced Security Features**: Adding more complex key management and using hybrid encryption (e.g., combining AES with RSA for key exchange) could improve security.