Image Encryption Based on AES Algorithm

Group Members
Khoubaib Bourbia
Islem Chouayakh
Maha Jdidi

Overview

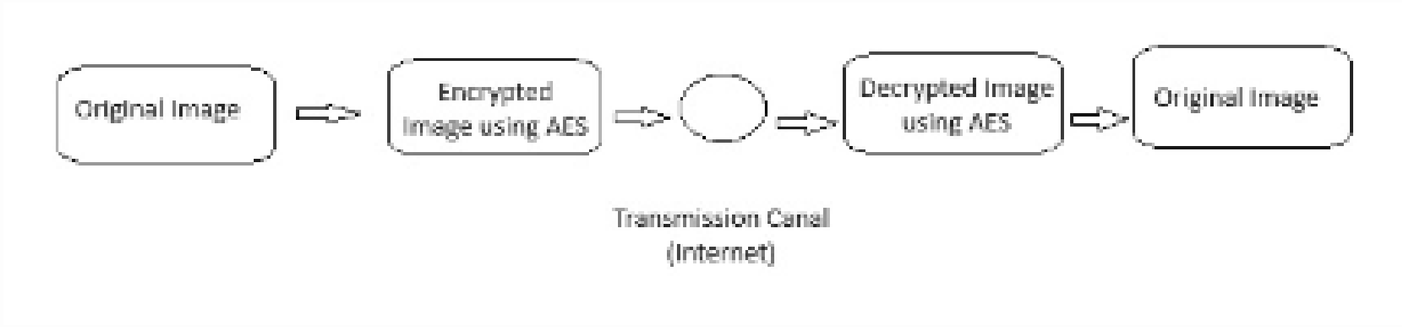
Introduction

In recent years, with the rapid development of network connection technologies, people around the world have been able to use the internet more and more frequently, increasing the volume of information and data exchanged. This raises the issue of data security. Images are sent over an insecure transmission channel from different sources, some image data contains secret data, and some images themselves are highly confidential, so it is essential to protect them from attack.

To solve this problem, we will use AES in the context of image encryption, given its ability to protect against unauthorized access, data tampering and other malicious activities. AES uses a block cipher structure with variable key lengths, offering a high degree of flexibility and scalability to meet the specific security requirements of different types and sizes of image data.

Scope

The project aims to develop a secure transfer of images between sender and receiver. Image should be encrypted before it is sent on a network and it should be correctly decrypted on the receiver side.



Design and Implementation constraints:

- the preferred programming language here is python
- Encryption and Decryption should be done using AES algorithm
- Original Image must be preferably in a common format (.jpeg/.png format).

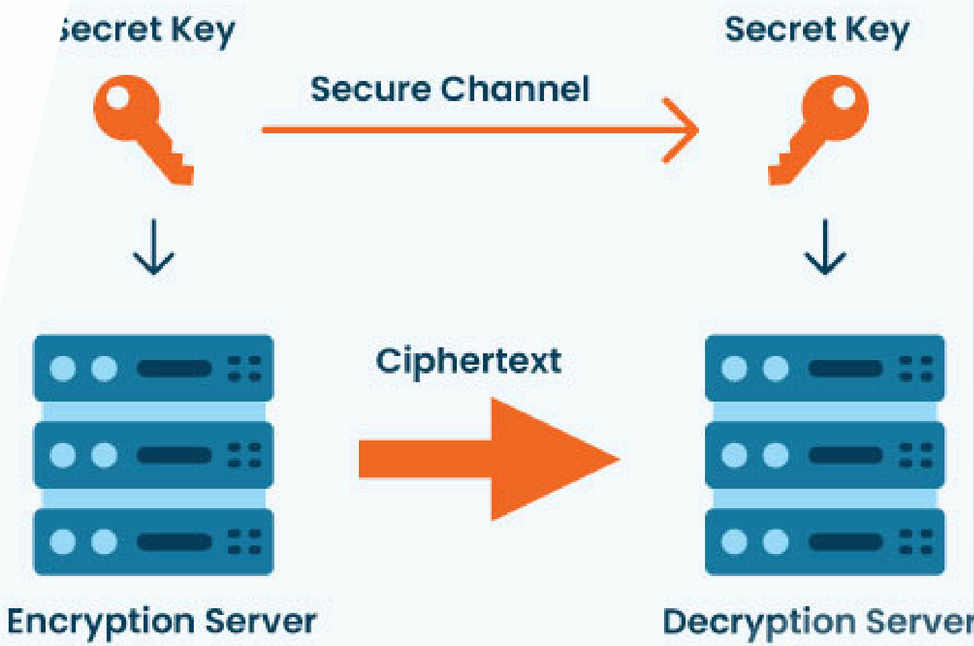
Functional Requirements:

1-Encryption of Image:

- The system shall encrypt the given image file using the AES encryption algorithm.
- Encrypted images shall be saved in an unreadable format to ensure data security.

2-Decryption of Encrypted Image:

- The system shall decrypt the received encrypted image file using the AES decryption algorithm.
- Decrypted image shall be identical to the original input image, ensuring data integrity.



AES Algorithm Working

Non-Functional Requirements:

1-Data Security:

- The system shall utilize strong encryption keys derived from user-provided passwords using PBKDF2 to enhance security.

2-Transmission Security:

- Encrypted images shall be securely stored or transmitted over any communication channel.
- Appropriate measures shall be taken to prevent unauthorized access to the transmitted or stored data.

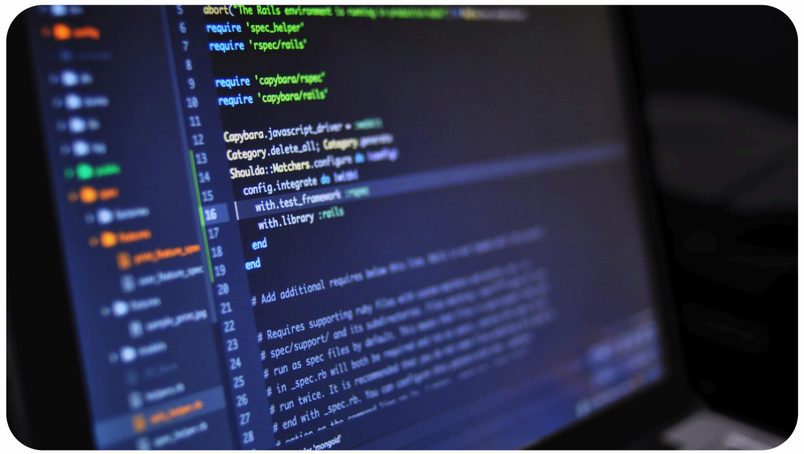
3-Performance:

- Encryption and decryption processes shall be optimized for efficiency to minimize processing time and resource utilization.

Overall System Requirements:

- The system shall handle image encryption and decryption operations without requiring a user interface.
- All cryptographic operations shall be performed internally by the system based on predefined parameters.
- The system shall maintain data confidentiality and integrity throughout the encryption and decryption processes.
- Detailed documentation shall be provided to ensure proper understanding and usage of the system.

Main Concepts:



Main components

AES Algorithm
Key Derivation Function (PBKDF2)
Block Cipher Mode
Initialization Vector (IV)
Padding

-AES Algorithm

-Key Derivation Function (PBKDF2):

- BKDF2 is a key derivation function that derives a cryptographic key from a password and a salt.
- By iteratively applying a pseudorandom function (such as HMAC-SHA1) to the password and salt, PBKDF2 strengthens the derived key and increases its resistance against brute-force attacks.
- The iteration count parameter allows for tuning the computational cost of key derivation, balancing security and performance considerations

Block Cipher Mode:

- AES uses or operates in different block cipher modes like ECB,CBC,CTR
- The modes tell us how our image block is encrypted

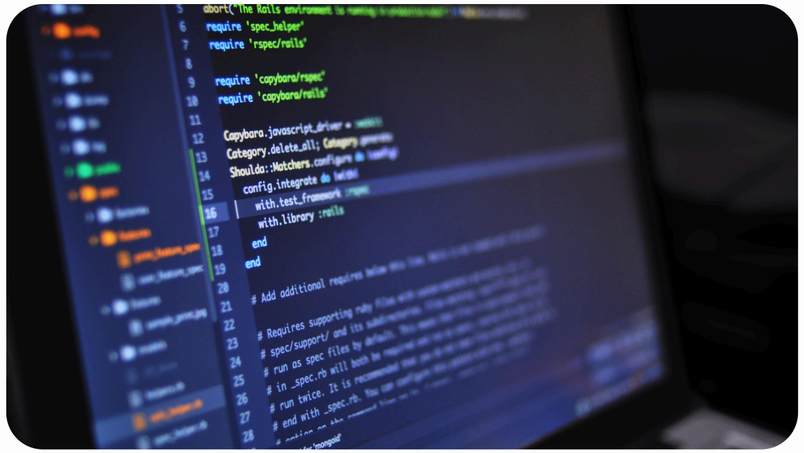
Initialization Vector (IV):

- An Initialization Vector (IV) is a random value used to introduce randomness into the encryption process and ensure that identical plaintext blocks do not produce identical ciphertext blocks.
- The IV is XORed with the plaintext before encryption in modes like Cipher Block Chaining (CBC) to mitigate against certain cryptographic vulnerabilities such as pattern recognition.

Padding:

- Padding is employed to adjust the plaintext data to a size that is a multiple of the block size required by the encryption algorithm.
- Common padding schemes include PKCS#7 padding, where the value of each added byte is set to the number of bytes added, ensuring unambiguous removal of padding during decryption.

Main Concepts:



Functional Flow

Encryption process

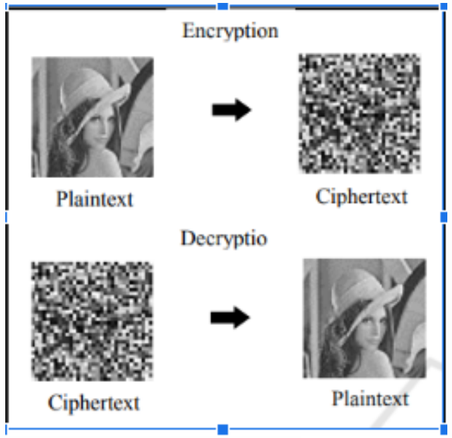
Decryption Process

- Encryption process

- Generate a random salt to enhance the security of the key derivation process.
- Derive an encryption key from the user-provided password and salt using PBKDF2.
- Initialize an AES cipher in a suitable mode of operation (e.g., CBC) with a randomly generated IV.
- Read the binary data of the input image file into memory.
- Apply padding to ensure that the plaintext data is a multiple of the block size.
- Encrypt the padded plaintext using the AES cipher, producing ciphertext.
- Write the salt, IV, and ciphertext to the output file for storage or transmission.

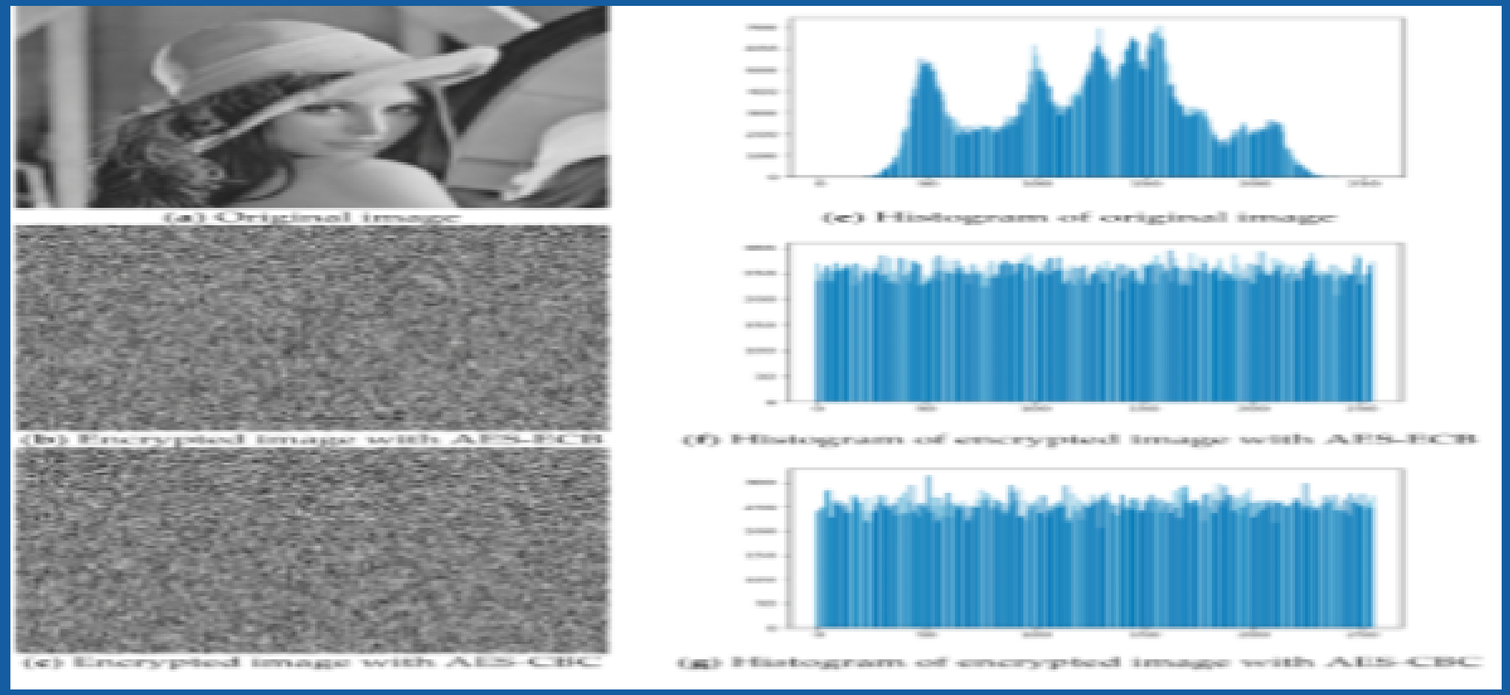
-Decryption Process

- Read the salt, IV, and ciphertext from the encrypted image file.
- Derive the decryption key from the user-provided password and salt using PBKDF2.
- Initialize an AES cipher in the same mode of operation (e.g., CBC) with the retrieved IV.
- Decrypt the ciphertext to obtain the padded plaintext data.
- Remove the padding to restore the original plaintext image data.
- Write the decrypted image data to an output file for further processing or display.



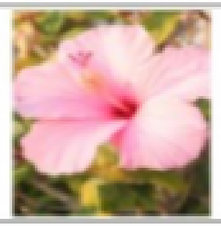
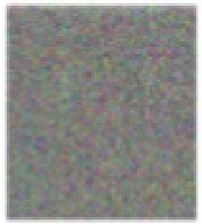
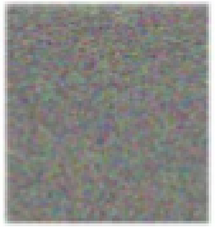
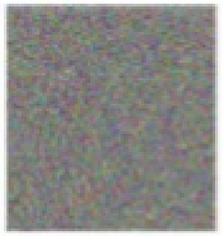
Why AES for Image Encryption:

According to the findings, the AES method provides superior picture encryption quality, as seen by the histogram's greater number of converging columns:



Why AES for Image Encryption:

AES vs. RSA:

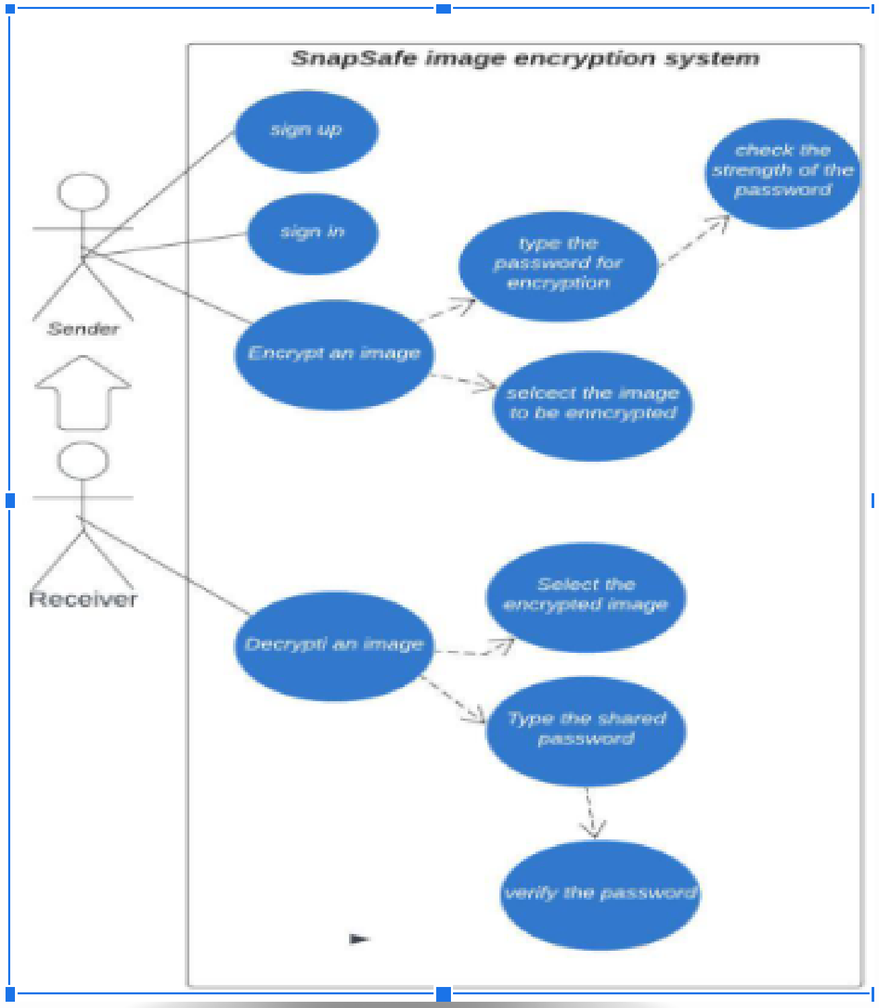
Algorithm	Flower	Nature	Lion
Original			
RSA			
AES			

Objective of using AES in image encryption

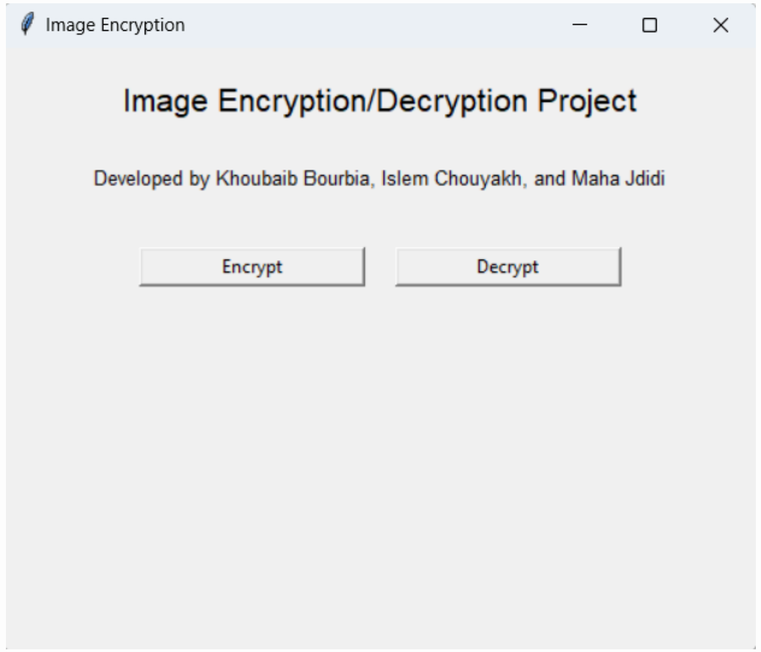
- Develop an intuitive user interface for interacting with the system.
- Encrypt image files using AES in CBC mode for robust security. Save encrypted images in an unreadable format to maintain data confidentiality.
- Decrypt the received encrypted image files using AES in CBC mode and restore the original image to maintain data integrity.
- Verify password strength to ensure users provide strong passwords for encryption. Allow users to re-enter passwords during decryption in case of incorrect passwords.
- Use PBKDF2 for key derivation, enhancing the security of the encryption key. Protect data during transmission and storage to prevent unauthorized access.
- Optimize encryption and decryption processes for efficiency.
- Implement error handling for incorrect passwords or corrupted files. Provide clear error messages for a smooth user experience.

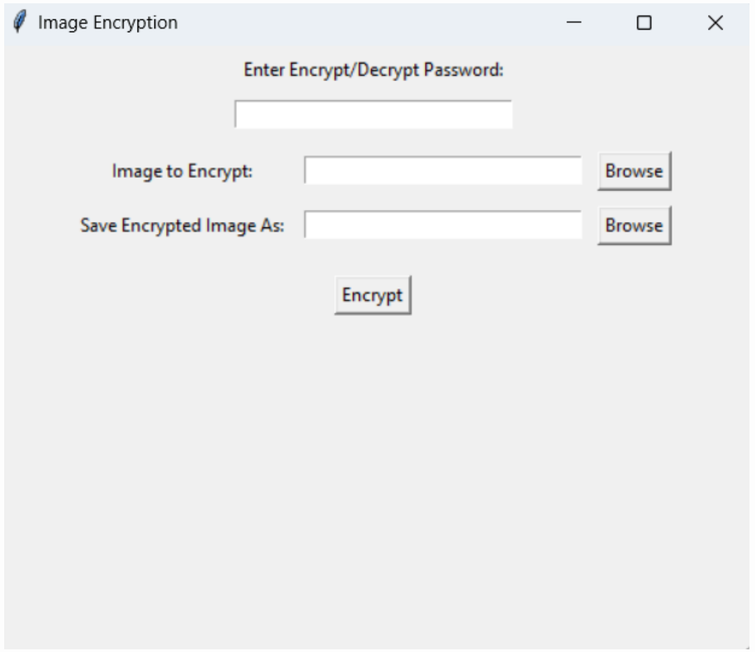
==>These objectives aim to provide a secure, efficient, and user-friendly solution for encrypting and decrypting images while maintaining data security and integrity.

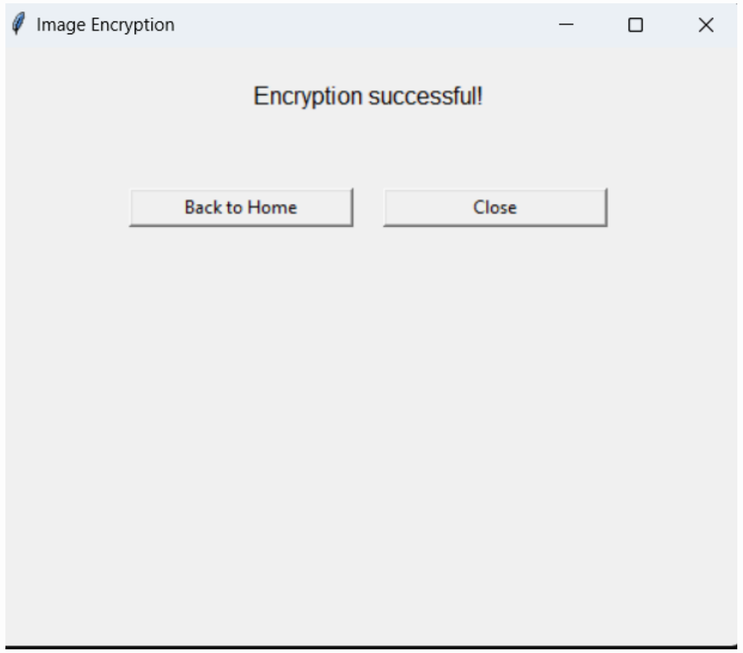
Project design



Project design







Project tools

- **Cryptography & PyCrypto:** provide support for various cryptographic algorithms, such as AES, RSA, and SHA**.**
- **PySeCrypt:** provides an easy-to-use interface for encrypting and decrypting messages using various cryptographic algorithms, such as AES and Blowfish.
- **PBKDF2:** is a key derivation function that uses a pseudorandom function to derive keys from password**s**
- **Passlib:** is used for password hashing, password storage, and password strength checking
- **Tkinter:** provides support for creating GUI applications. It provides various widgets such as buttons, labels, and text boxes, which can be used to create a user interface for your steganography project.

CONCLUSION

We have successfully developed a program that encrypts and decrypts the image files accurately. This will help in minimising the problem of data theft and leaks of other sensitive information. The file that we obtained after encryption is very safe and no one can steal data from this file. So, this file can be sent on a network without worrying

THANK YOU!