PROJECT REPORT ON

IMAGE ENCRPTY

Submitted by

A.BHAVISHYA (ST#IS#6119)

Under the Supervision of

UPENDRA
Senior Security Analyst

KRISHNA
Junior Security Analyst

Registered And Head OfficeD.NO: 11-9-18, 1st Floor,
Majjivari Street,
Kothapeta,Vijayawada - 520001.

+91 9550055338 / +91 7901336873

contact@suprajatechnologies.com

Table of Contents

1.Introduction

1.1 Overall Description

In the last few years the security and integrity of the data is the most important concern. Now a day's almost all the data is transferred over the computer networks and it has increased the attacks over the network. Before transmitted data it has to be encrypted and store so that it cannot be attacked by various attackers. Encryption is a process of hiding the data, where it converts the original text into cipher text. Encryption uses different algorithm to encrypt the data into different form. Cryptographic Algorithm uses a set of keys with the different characters for both encryption and decryption. By using key the plain text is converted to the cipher text and decryption is done by converting back the plaintext from the cipher text. Cryptography is a process of transmitting and storing data in a form that it is read only by authorized users. Cryptography is a science of protection of data by encoding it into unreadable form. It is useful way of protecting the important sensitive information by using mathematical form algorithm for both encryption and decryption process. The encryption and decryption process depend on the key value. The strength of the algorithm is how difficult it is to determine the key value and get the original text. The algorithm is majorly divided into two types symmetric and asymmetric depending on the keys. If same keys are used for both encrypting and decrypting then it is called symmetric algorithm. Symmetric algorithm is further divided into stream and block ciphers. A stream cipher is done on the single byte of data, where as the block a cipher is done on the block of data. Asymmetric algorithm uses two different keys one for encryption and both for decryption. The key should be kept secret so that the message should be not be decrypted. The purpose of cryptography is to provide Authentication (proving the one's identity), Non-repudiation (the receiver should know the sender should not be faking), Integrity (data should be correct, accuracy, and trustworthiness), and Privacy/confidentiality (message is read by only the intended receiver)

2. Existing System

The variation in the characteristics of the multimedia data such as correlation among the pixels and high redundancy of the image. Therefore there were some limits where same techniques cannot be used for protection all type of multimedia data. The traditional encryption algorithms may not use to encryption the image directly because of these reasons:

- As the size of image will be not same as the text it may varies. Hence the traditional encryption algorithm may take longer time to encrypt and decrypt the image compare to text.
- There is condition which says that the text encryption both decrypted and original text must be equal but it can be never true for the image. For decryption of image the small distortion is also accepted by the human perceptive.
- Computational time is high in exiting system.
- High computing power is required.
- For networking Systems it is not efficient

. • Security is also a major issue.

3.Proposed System

There should be a reliable storage and transmission of digital image where it has been served such as multimedia systems, medical and military imaging systems. The security of image is the most critical problem, due to increase in the growth of internet, cell phones and multimedia technology in the society. This project is to propose a secure image encryption and decryption form by using AES algorithm. The AES algorithm is widely used in the applications of daily life, such as smart cards, cell phones, automated teller machines and WWW servers. AES encrypts a plaintext to a cipher text, which can be decrypted to the original plaintext by using common private key. The cipher text is made very different form so that it should not have any idea of the original plain text. For image encryption and decryption, the AES encrypt the image in different form using the key which should have no idea of original form. After decrypting it, it should be in the original form. The encryption of image should be strong so that it should not be known by the intruders. Strengths of AES: • AES is extremely fast compared to other block ciphers. • As the round transformation is parallel for the design, which makes the important for the hardware to allow it for fast execution. • AES was designed to be agreeable to pipelining. • There is no arithmetic operations for the cipher, so there is no bias towards the big or little endian architectures. • AES is fully self-supporting. • AES is not based on obscure or not well understood processes.

4.System Design

4.1 Feasibility Study

Image encryption is the process of securing digital images to prevent unauthorized access or viewing. This feasibility study evaluates the potential implementation of an image encryption system, focusing on economic, technical, and social feasibility.

- Design a robust and scalable architecture for image encryption.
- Consider factors such as encryption algorithms, key management, and integration with existing systems.

Security Considerations:

- Evaluate encryption algorithms for strength and resistance to attacks.
- Implement secure key management practices to safeguard encryption keys.
- Address potential vulnerabilities such as side-channel attacks or implementation flaws.

4.1.1 Economic Feasibility

Cost-Benefit Analysis:

- **Costs**: Initial investment in development, hardware, and software.
- **Benefits**: Reduced risk of data breaches, potential regulatory compliance benefits, and enhanced reputation for security.

- **ROI**: Calculate the return on investment based on the projected benefits versus costs.

4.1.2 Technical Feasibility

The project works by encrypting the given image using AES algorithm so that this image can be sent securely over the network. At the receiver side, the receiver has code for decrypting the image so that he can get the original image. This helps in sending confidential and sensitive information securely over the internet. Main application of this can be very helpful in medical and military fields**.**

4.1.3 Social Feasibility

**The system shall encrypt the given image to an unreadable format. This is done using
AES encryption function.
**The system shall decrypt the received encrypted image to a readable format. This is
done using AES decryption function. The output image should be same as the original
image.
**The system ensuresthat the image is securely sent over any transmission medium. Third
party system cannot make modifications to the file being sent since unauthorised access
is not supported.

4.2 Input And Output Design

4.2.1 Input Design

Encrypt Button Click: Initiates the encryption process.

Select Image File: User selects an image file for encryption.

Enter Encryption Key: Optionally, user can manually enter an encryption key.

Send Decryption Key via Email: User can choose to send the decryption key via email and provide recipient and sender email addresses.

Browse Encrypted Image File: For decryption, user browses and selects the encrypted image file.

Enter Decryption Key: User enters the decryption key to decrypt the selected encrypted image.

Output Design:

Success or Error Messages: System displays success or error messages after each operation.

Decryption Key Email Sent Confirmation: Confirmation message displayed upon successful email delivery of the decryption key.

Decrypted Image Display: After successful decryption, the decrypted image is displayed.

Encryption Key Generation: If user doesn't provide an encryption key, a random key is generated and displayed.

Email Entry Fields: If sending decryption key via email, fields for recipient and sender email addresses are provided

4.2.2 Objective

• Encryption of an Image to unreadable format
• Decryption of encrypted image to original image
• Secure transfer of an image over the network such as internet
• Ensure no modifications are made while transferring over the network.

4.2.3 Output Design

5.Implementation

5.1 Module Description

➢ **Email Module:** If the user chooses to send the decryption key via email, this module facilitates the sending of emails. It interacts with an SMTP (Simple Mail Transfer Protocol) server to send the key securely to the specified recipient email address.

- **User Interface (UI):** The UI component provides the graphical interface for users to interact with the application. It includes buttons, input fields, and display areas for images and messages.

- **Error Handling:** This component manages error handling and displays appropriate error messages to the user in case of failures during encryption, decryption, or email sending processes.

- **Event Handling:** The application handles user events such as button clicks, file selections, and text input, triggering corresponding actions like encryption, decryption, or email sending.

- **Main Control Flow:** Coordinates the flow of control among different modules and components. It manages the sequence of operations based on user actions and ensures the smooth functioning of the application.

- **Encryption Module:** Responsible for encrypting the selected image using the AES (Advanced Encryption Standard) algorithm. It generates a random encryption key or uses the key provided by the user.

- **Decryption Module:** Handles the decryption process for encrypted images. It requires the decryption key provided by the user to decrypt the image using the AES algorithm.

- **File I/O Operations:** Responsible for reading and writing image files to/from the file system. It retrieves the image file selected by the user for encryption or decryption and saves the encrypted or decrypted image back to the file system

5.2 System Architecture

- **Backend Modules:**

- **Encryption Module:** Responsible for encrypting image files using the Fernet encryption algorithm from the cryptography library.

- **Decryption Module:** Handles the decryption of encrypted image files using the same encryption algorithm.

- **Email Module:** Manages the sending of decryption keys via email using the smtplib library for SMTP communication.

➢ **Client Side (GUI Application):**

The client side consists of the graphical user interface (GUI) built using Tkinter, a standard GUI toolkit in Python..Users interact with the GUI to perform encryption, decryption, and email sending operations.The GUI communicates with the backend modules to execute the requested operations

6.Algorithm Implementation

Symmetric Encryption

Symmetric encryption algorithms use the same secret key for both encryption and decryption.

Symmetric encryption algorithms come in two different varieties:

1. Block ciphers.
2. stream ciphers.

→Block Ciphers

A block cipher encrypts data in fixed-size chunks. For example, the Advanced Encryption Standard (AES) uses a block length of 128 bits.

If the plaintext is shorter than the block length, then it is padded out to the desired length before encryption. At the other end, the recipient of the message will decrypt it and then remove the padding to restore the original message. If a plaintext is longer than the block length, then it is broken up into multiple different chunks for encryption. A block cipher mode of operation defines how these chunks are related to one another.

Each mode of operation has its pros and cons. For example, Electronic Code Book (ECB) mode is the simplest mode of operation. With ECB, each block is encrypted completely independently.

**The algorithm used in this project is AES

Symmetric Encryption: AES is a symmetric encryption algorithm, meaning the same key is used for both encryption and decryption. This makes it efficient for encrypting and decrypting large amounts of data.

Block Cipher: AES operates on fixed-size blocks of data. For AES, each block size is 128 bits (16 bytes).

Key Lengths: AES supports key lengths of 128, 192, or 256 bits. The longer the key, the stronger the encryption.

Rounds: AES consists of multiple rounds of processing, with each round applying a series of substitution, permutation, and mixing operations to the data.

AES Decryption Process:

Key Expansion: Similar to encryption, the decryption key is derived from the original key.

Initial Round: The ciphertext is combined with the final round key using a bitwise XOR operation.

Rounds (Inverse): Decryption involves applying the inverse operations of the encryption rounds in reverse order. This includes:

- **InverseSubBytes:** Each byte of the data is replaced with a corresponding byte from the inverse substitution box.Inverse Substitute Bytes is the inverse of the byte substitution transformation, in which the inverse S-box is applied to each byte of the State. It is reverse process of Substitute byte transform. This is obtained by applying the inverse of the affine transformation followed by taking the multiplicative inverse in GF (2^8). There is an inverse s-box table for substitute the value.
- **InverseShiftRows:** The bytes in each row of the data matrix are shifted cyclically to the right by different offsets.Inverse shift row transformation Inverse Shift Rows is the inverse of the Shift Rows transformation. The bytes in the last three rows of the State are cyclically shifted over different numbers of bytes. The first row, r = 0, is not shifted. The bottom

three rows are cyclically shifted by Nb-shift(r, Nb) bytes, where the shift value shift(r,Nb) depends on the row number

- **InverseMixColumns:** Each column of the data matrix is multiplied by the inverse of the MixColumns matrix.Inverse Mix Columns is the inverse of the Mix Columns transformation. Inverse Mix Columns operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF(2^8) and multiplied modulo x^ 4 + 1 with a fixed polynomial (x)
- **AddRoundKey:** The round key for the current round is combined with the data using a bitwise XOR operation.

Final Round: The final round is similar to the other rounds but without the InverseMixColumns step.

Output: The output of the final round is the decrypted plaintext, which should match the original input data.

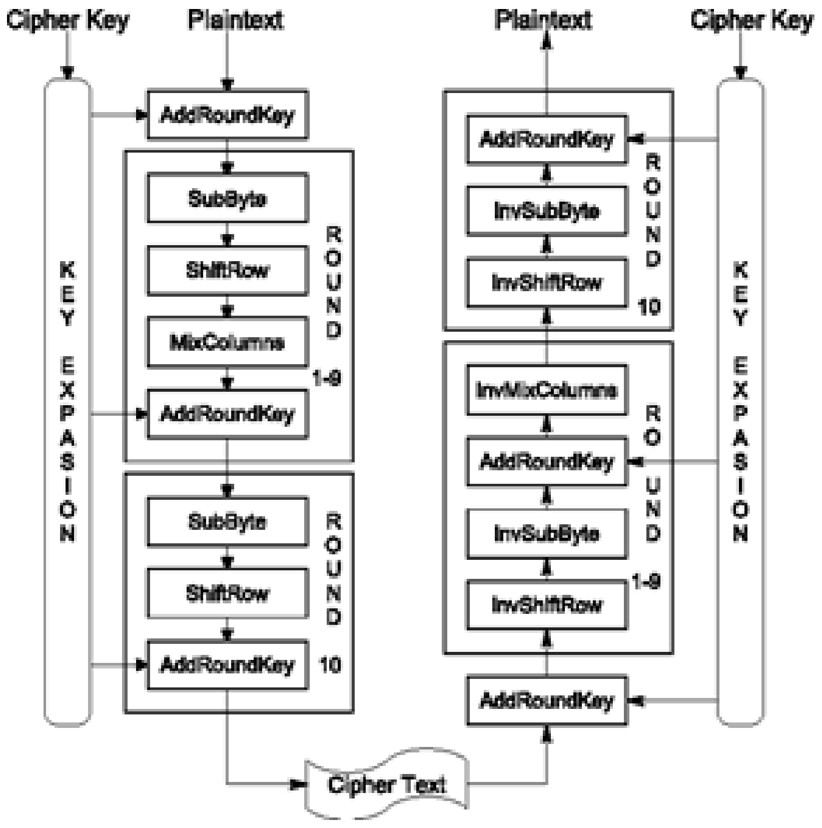
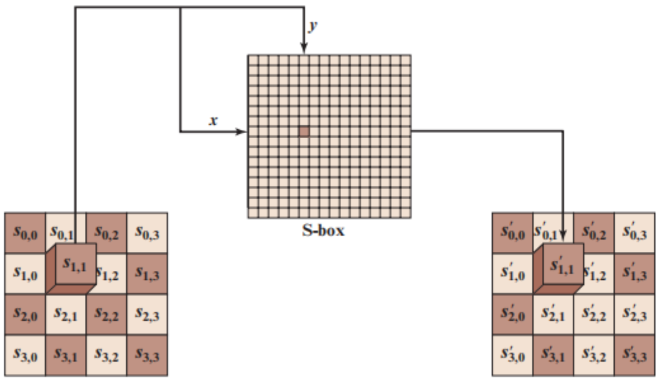


Figure 2: Block diagram AES Encryption and Decryption

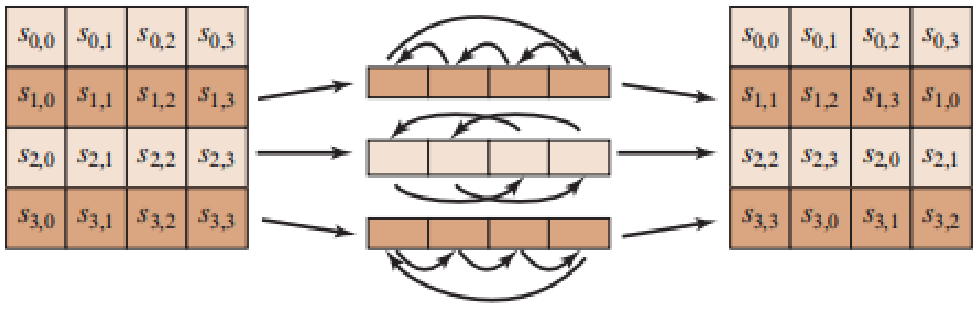
Rounds: Multiple rounds of substitution, permutation, and mixing operations are applied to the data. Each round consists of the following steps:

- **Sub Bytes:** Each byte of the data is replaced with a corresponding byte from a substitution box (S-box), which is a fixed table generated from the encryption key.The Substitute bytes transformation is a non-linear byte substitution that operates independently on each byte of the State using a substitution table S-box. The operation of substitute byte.

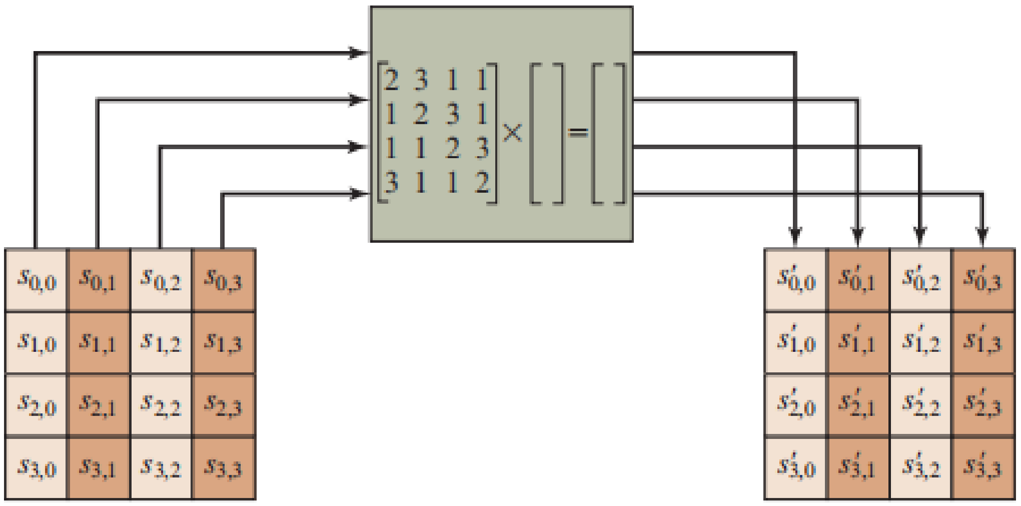


(a) Substitute byte transformation

-
- **ShiftRows:** The bytes in each row of the data matrix are shifted cyclically to the left by different offsets.In the Shift Rows transformation, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes. The first row, r = 0, is not shifted.This has the effect of moving bytes to "lower" positions in the row while the "lowest" bytes wrap around into the "top" of the row.

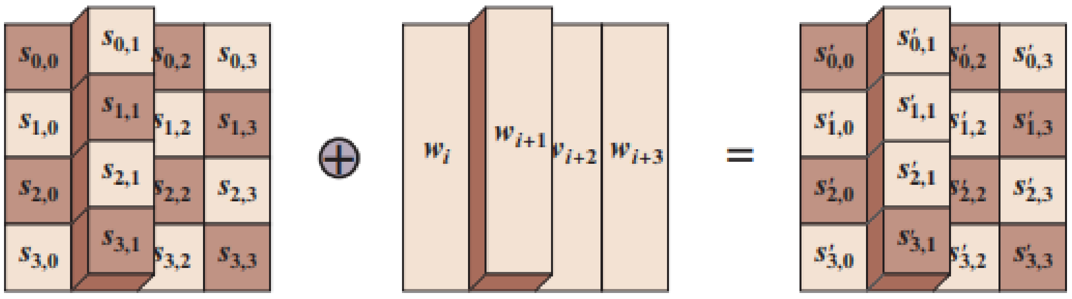


-
- **MixColumns:** Each column of the data matrix is multiplied by a fixed matrix to ensure diffusion.The Mix Columns transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF(2^8) and multiplied modulo x 4 + 1 with a fixed polynomial a(x), given by a(x) = {03}x ^3 + {01}x^ 2 + {01}x + {02} . The resultant columns are shown in the figure below. This is the operation of mix columns.



➢

➢ **AddRoundKey:** The round key for the current round is combined with the data using a bitwise XOR operation.In the Add Round Key transformation, a Round Key is added to the State by a simple bitwise XOR operation. The Round Key is derived from the Cipher key by means of key schedule process. The State andRound Key are of the same size and to obtain the next State an XOR operation is done per element: b (i, j) = a (i, j) ⊕ k (i, j)

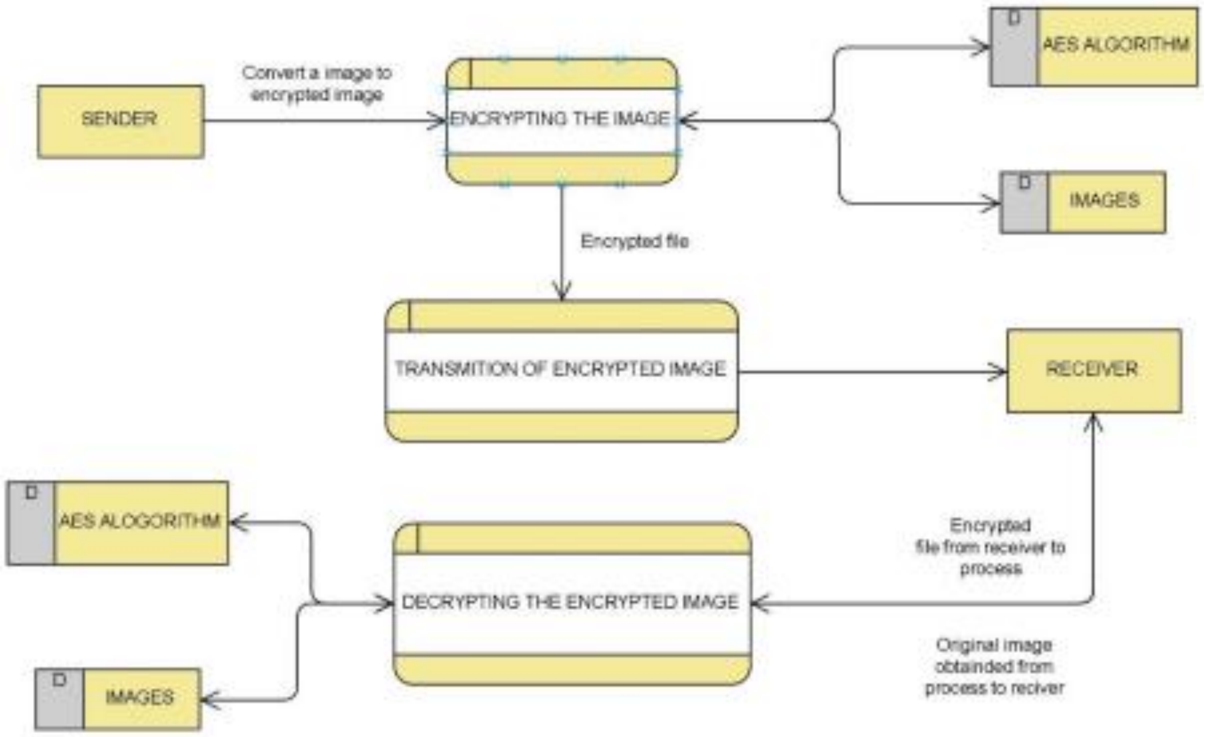


4.1.4 Final Round: The final round is similar to the other rounds but without the MixColumns step.

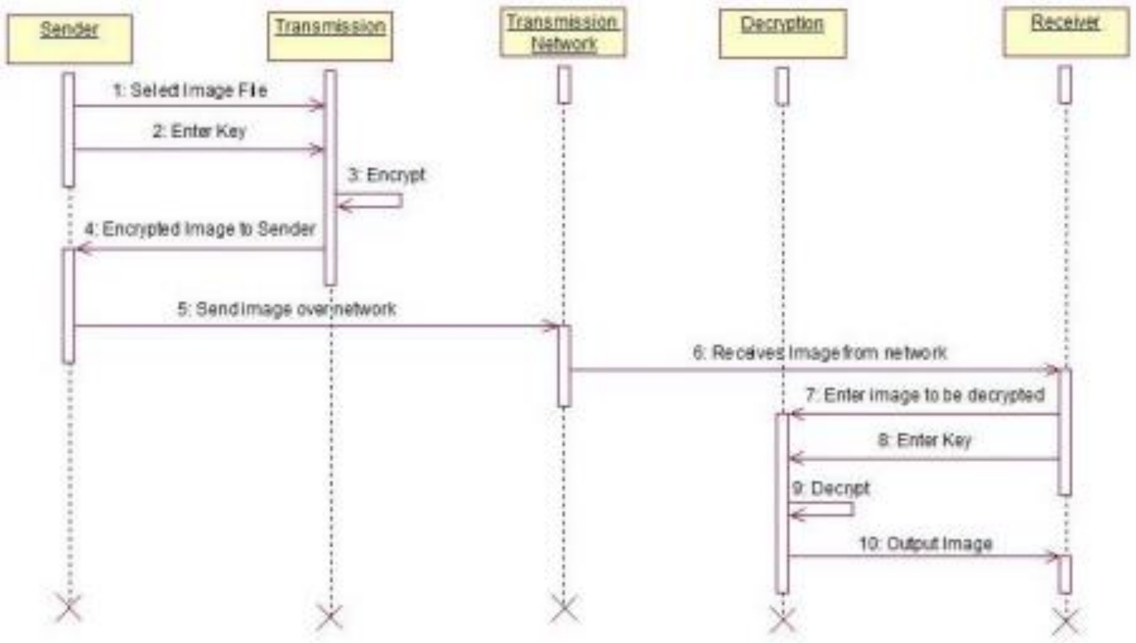
4.1.5 Output: The output of the final round is the ciphertext, which is the encrypted form of the original plaintext.

7.System Design

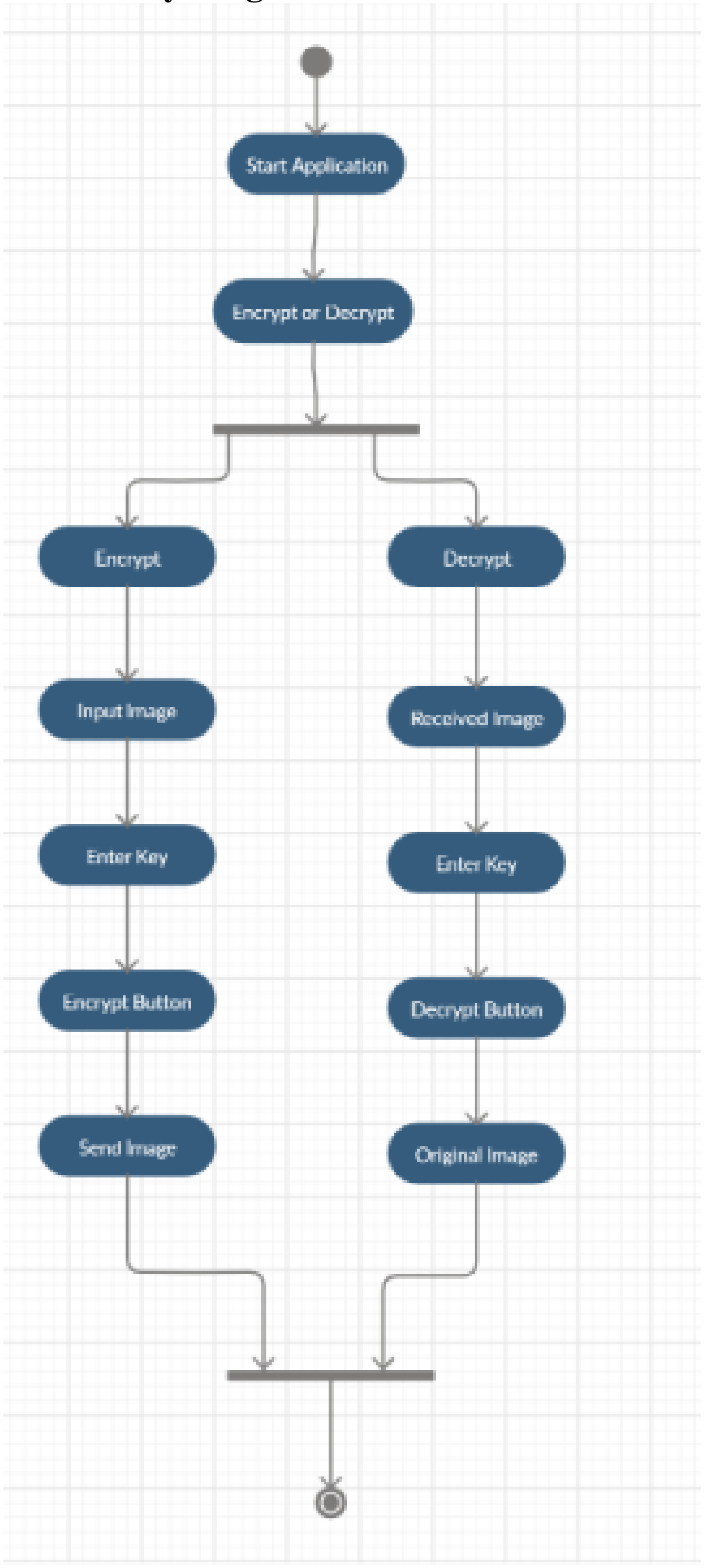
7.1 Dataflow Diagrams


SENDER
Convert a image to encrypted image
ENCRYPTING THE IMAGE
AES ALGORITHM
IMAGES
Encrypted file
TRANSMITION OF ENCRYPTED IMAGE
RECEIVER
AES ALOGORITHM
DECRYPTING THE ENCRYPTED IMAGE
Encrypted file from receiver to process
Original image obtainded from process to reciver
IMAGES

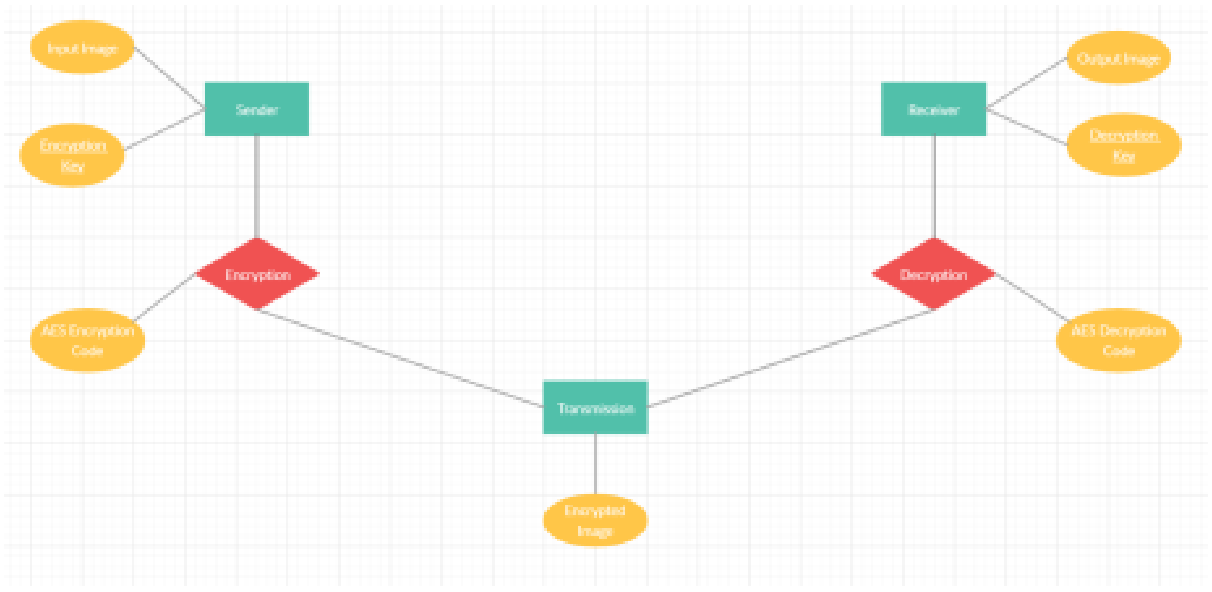
7.2 Sequence Diagram



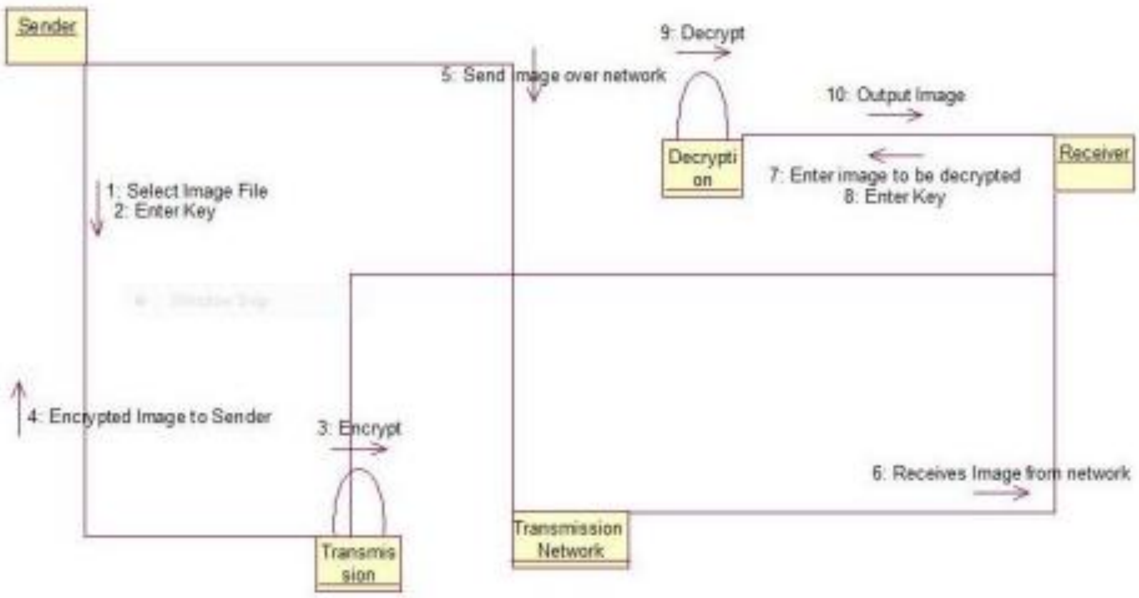
7.3 Activity Diagram



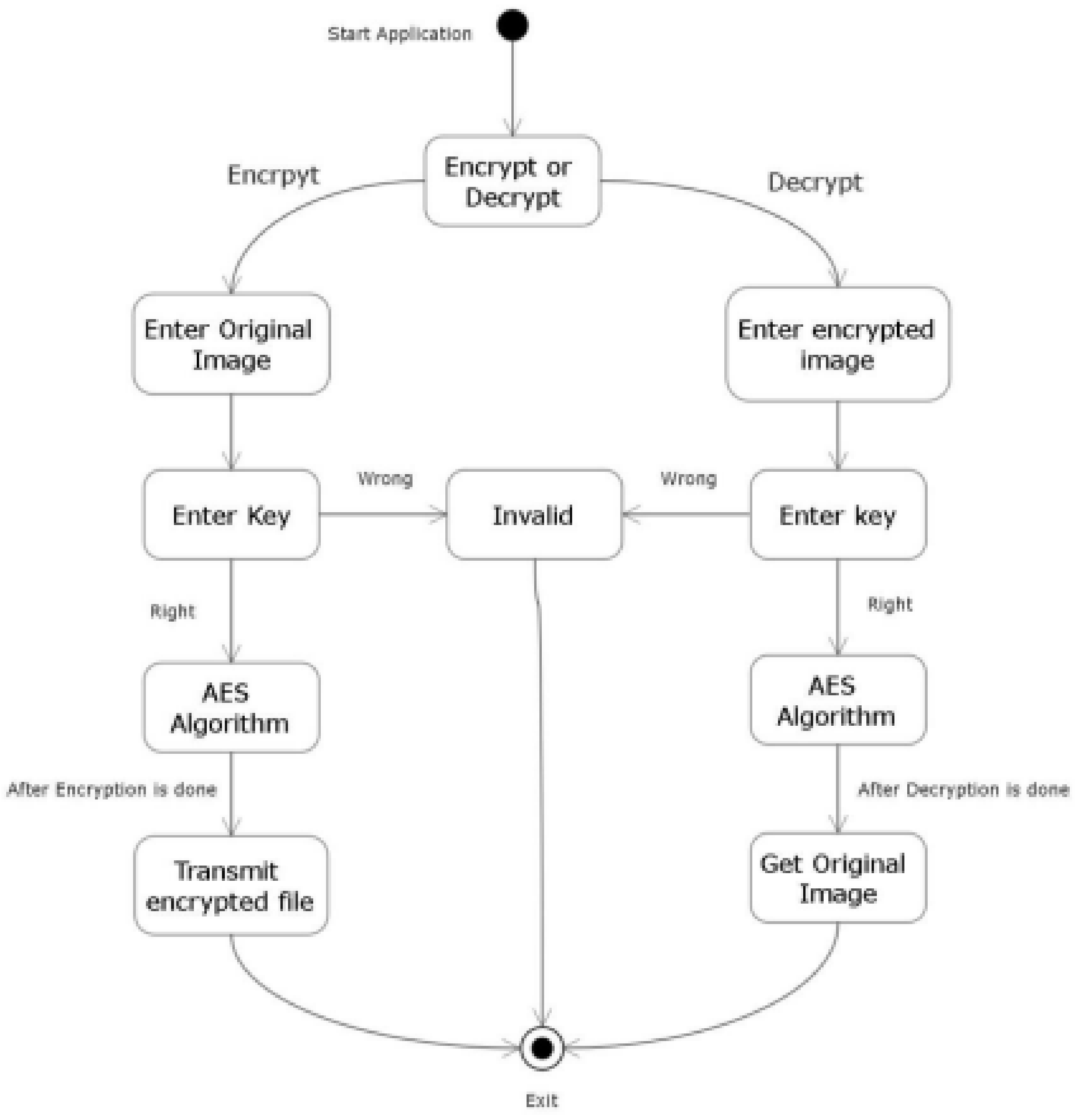
7.4 E-R Diagram



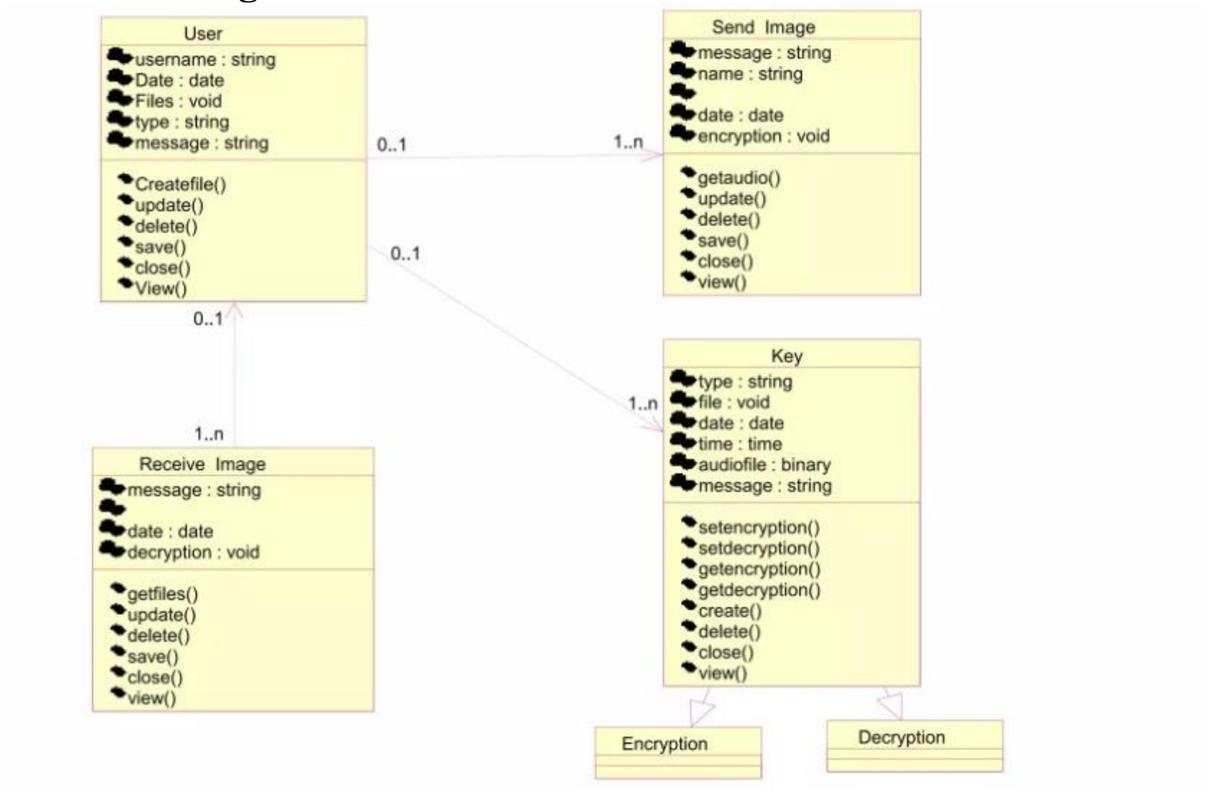
7.5 Collaborative Diagram



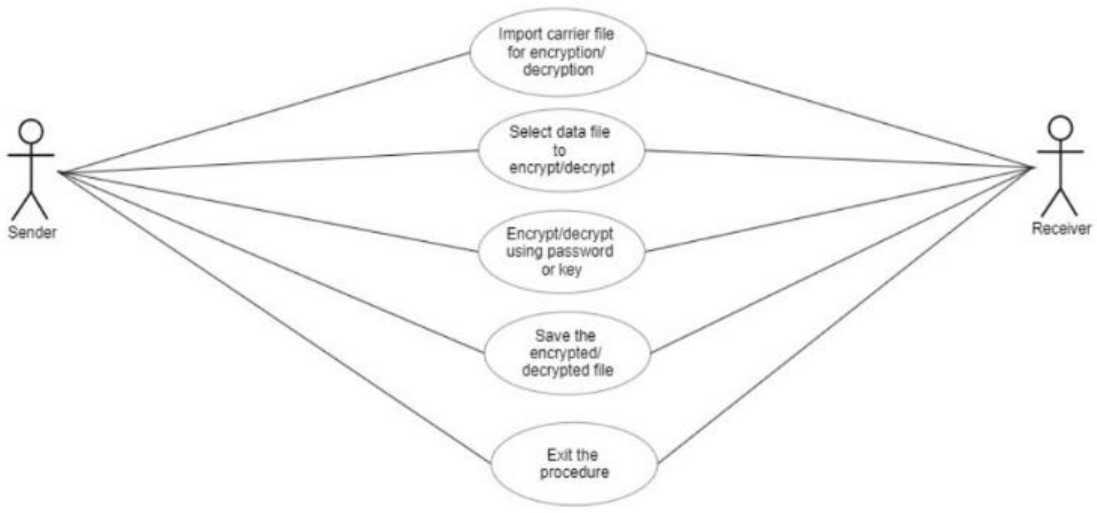
7.6 Statechart Diagram



7.7 Class Diagram



7.8Usecase Diagram


Import carrier file for encryption/ decryption
Select data file to encrypt/decrypt
Encrypt/decrypt using password or key
Save the encrypted/ decrypted file
Exit the procedure
Sender
Receiver

8.Requirement Specification

8.1 Functional Requirements

- **Encryption Functionality:** Allow users to encrypt image files using a specified encryption key.
- **Decryption Functionality:** Enable users to decrypt encrypted image files using the

corresponding decryption key.

- **Email Integration:** Provide an option to send decryption keys via email.
- **Error Handling:** Display informative error messages for various scenarios, such as file selection errors or encryption failures.

8.2 Non-Functional Requirements:

- **Security:** Ensure robust encryption algorithms are used to protect image data.
- **User Interface:** Design an intuitive and visually appealing GUI for ease of use.
- **Performance:** Optimize encryption and decryption processes for efficient operation.
- **Reliability:** Ensure the application functions reliably under different operating conditions.
- **Compatibility:** Ensure compatibility with multiple image file formats and email providers.
- **Scalability:** Design the application to handle a large number of image files and encryption keys if needed.

Constraints:

- **Dependent Libraries:** The application relies on external libraries such as cryptography and PIL, which must be installed.
- **Email Configuration:** Proper configuration of SMTP settings is required for sending decryption keys via email.

Assumptions:

- Users have basic knowledge of encryption concepts and the importance of keeping encryption keys secure.
- Users have access to a functional SMTP server for sending emails.

Future Enhancements:

- **Batch Processing:** Allow users to encrypt or decrypt multiple images simultaneously.
- **Password Protection:** Implement password-based encryption for an additional layer of security.
- **Integration with Cloud Storage:** Enable users to store encrypted images securely in cloud storage services.
- **Image Preview:** Provide a preview feature for viewing images before and after encryption.
- **Enhanced Error Handling:** Implement more informative error messages and logging

capabilities.

8.3 Software Requirements:

Operating Systems Supported:

The ImageEncrypt application is developed using Python and Tkinter, making it compatible with multiple operating systems, including:

- Windows
- macOS
- Linux distributions

Technologies and Languages Used to Develop:

- **Python:** The core programming language used for developing the application logic and functionality. Python provides extensive libraries for GUI development, cryptography, and email handling.

- **Tkinter:** Tkinter is the standard GUI toolkit included with Python, used for creating the graphical user interface of the ImageEncrypt application. It provides widgets, such as buttons, entry fields, and labels, for building interactive interfaces.

- **Cryptography:** The cryptography library in Python is utilized for implementing encryption and decryption functionalities. It offers secure algorithms and protocols for cryptographic operations, ensuring the confidentiality of image data.

- **PIL (Python Imaging Library):** PIL is a library used for image processing and manipulation in Python. In the ImageEncrypt application, PIL is employed for loading, saving, and processing image files during encryption and decryption operations.

- **smtplib:** The smtplib module in Python is used for sending emails via the Simple Mail Transfer Protocol (SMTP). It facilitates the integration of email functionality into the application, allowing users to send decryption keys via email.

- **Standard Library Modules:** Various Python standard library modules are also used for tasks such as file handling, error handling, and user input validation, contributing to the overall functionality and reliability of the application.

Development Environment:

The ImageEncrypt application can be developed and executed using any text editor or integrated development environment (IDE) that supports Python programming.

Popular choices for Python development environments include:

- PyCharm
- Visual Studio Code
- IDLE (Integrated Development and Learning Environment)
- Jupyter Notebook

8.4 Hardware Requirements:

- **System:** Any computer capable of running Python applications.
- **Hard Disk:**

Adequate space for storing image files, encryption keys, and temporary files.

A few megabytes of disk space should suffice.

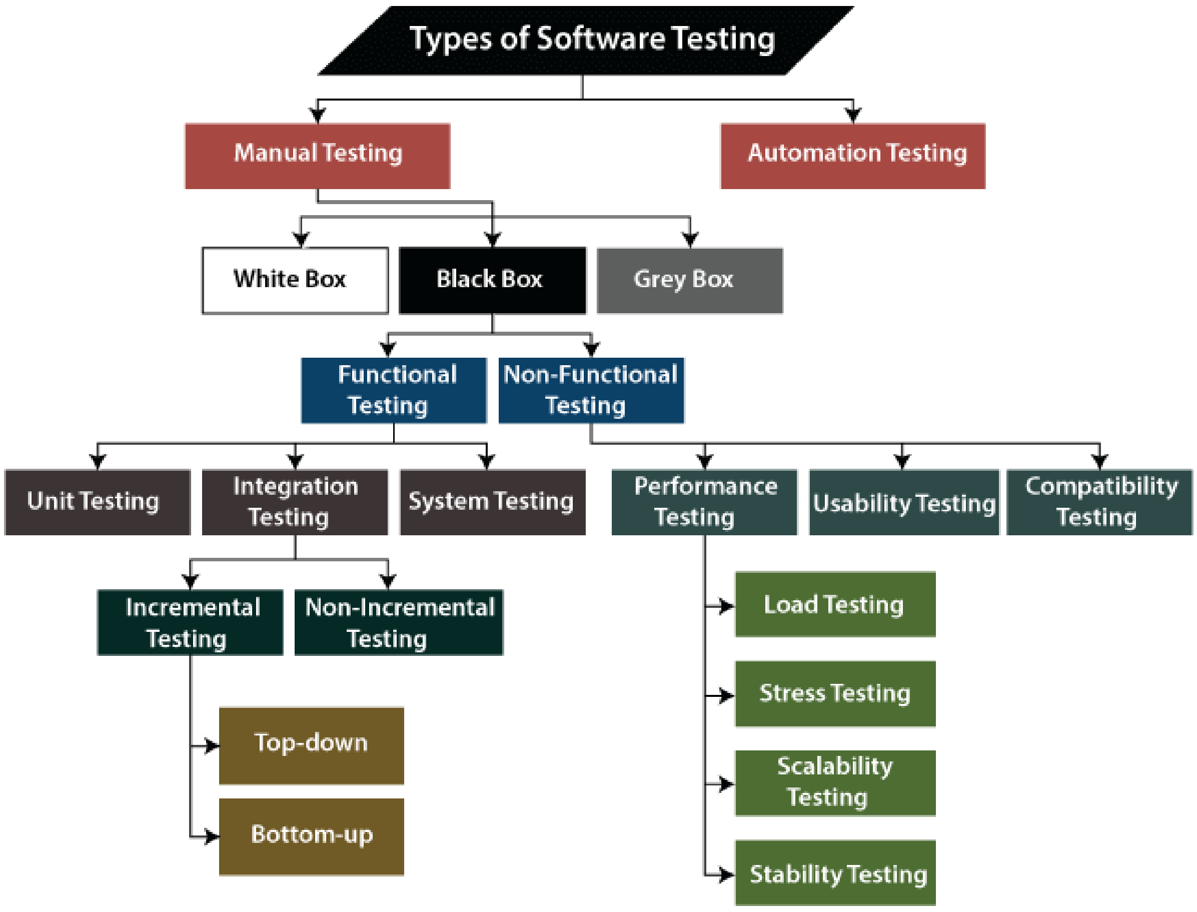
- **RAM (Random Access Memory):**

Minimum of 1GB RAM recommended.

More RAM may improve performance, especially with larger image files.

9.System Testing

9.1 Types of Testing



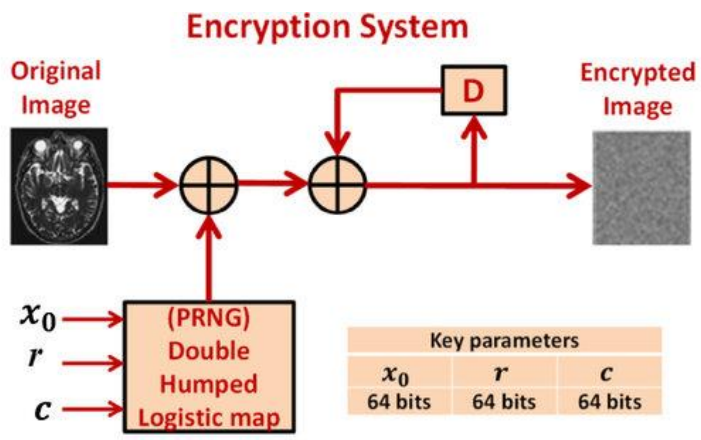
9.1.1 Integration Testing

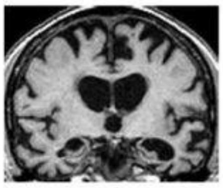
- Ensure encryption and decryption processes integrate seamlessly.Test GUI interaction with encryption/decryption modules.

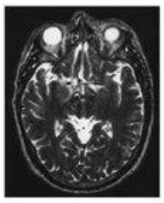
9.1.2 Unit Testing

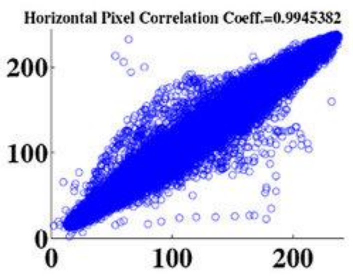
Functional testing of integrations is essential to verify that data arrives into your application from external sources accurately and writes to external sources correctly. Additionally, best practice suggests including error handling for all connectors.

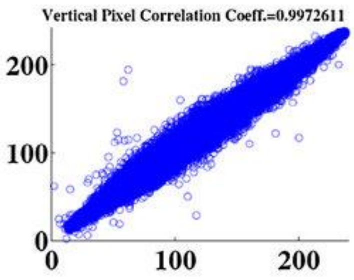
- Test encryption and decryption functions with known inputs.
- Verify key generation functionality.Mock SMTP server to test email sending

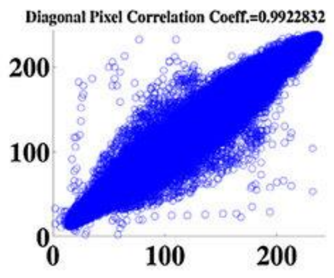


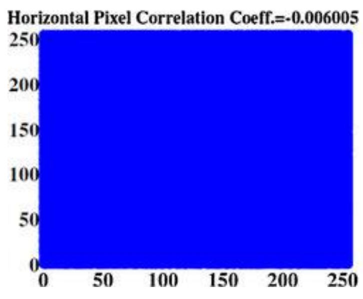


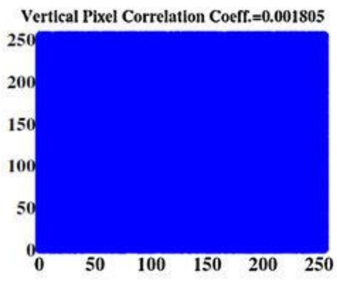


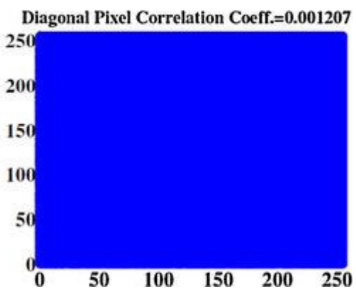












9.1.3 Functional Testing

Positive testing with valid inputs. Negative testing with invalid inputs and edge cases.Boundary testing for input parameters.

9.2 Integration Testing

Part of testing the functionality of your integrations is ensuring that the data that comes into your application is accurate, secure, and does not introduce regression issues. For example, in a connector that fetches personal details from the database, if an employee makes a change in the database and your system pulls the updated information, other system functionality should not break when returning the updated data.

9.3 Test Results

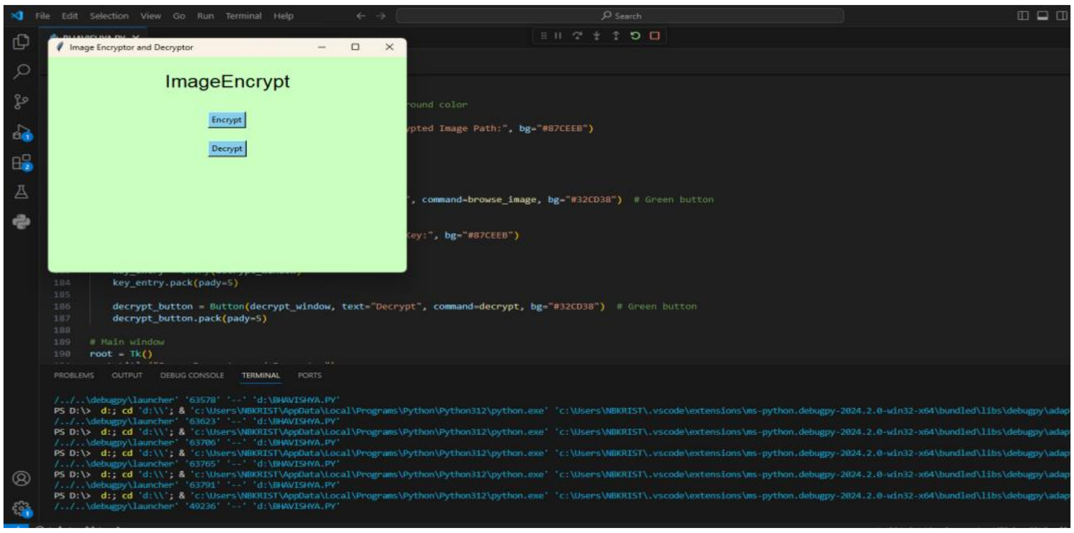
All the test cases mentioned above are passed succefully. No defects encountered.

9.4 Acceptance Testing

9.4.1 Test Results

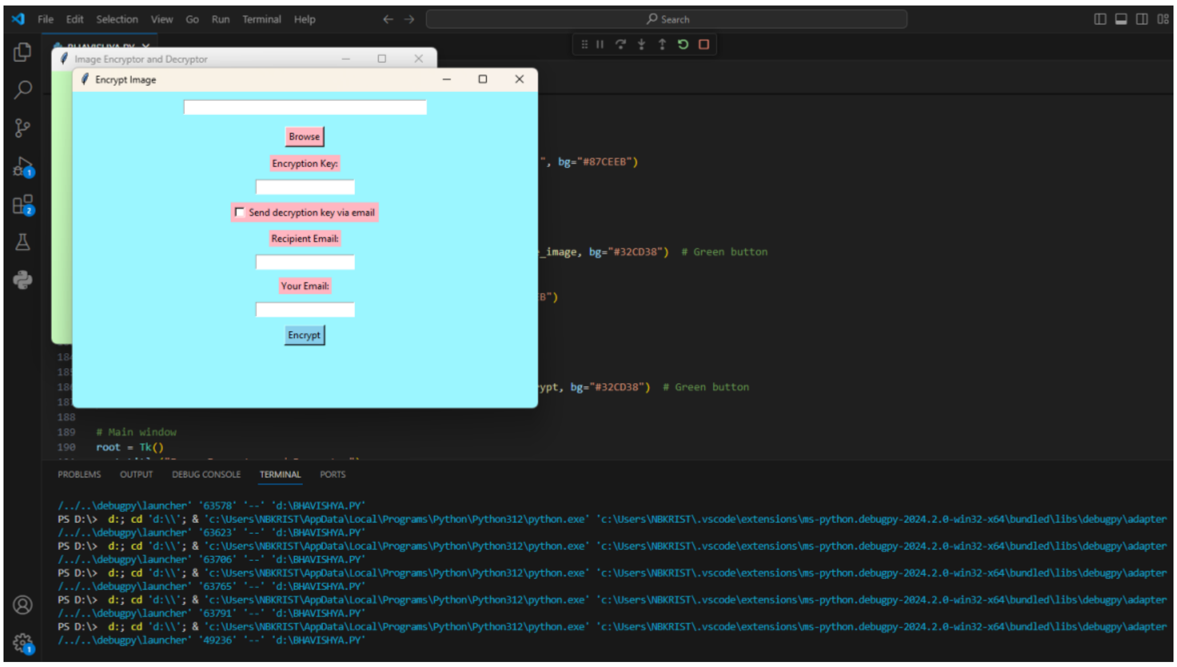
All the test cases mentioned above are passed succefully. No defects encountered.

EXECUTION:



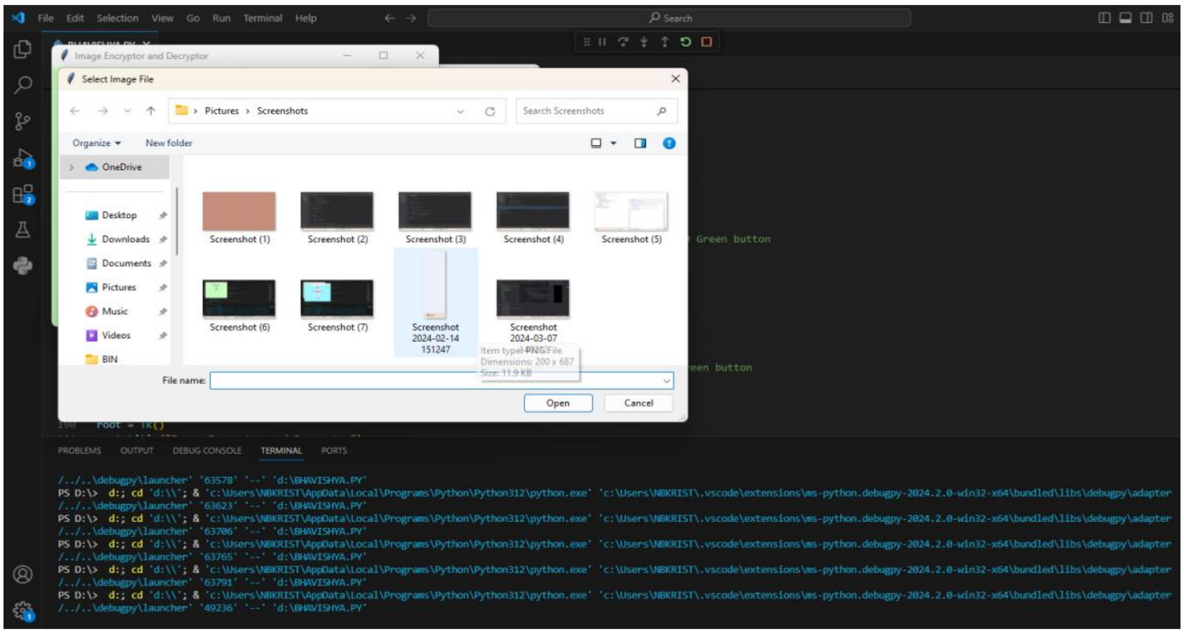
```
round color
ypted Image Path:", bg="#87CEEB")
", command=browse_image, bg="#32CD38")  # Green button
Key:", bg="#87CEEB")
    key_entry.pack(pady=5)

    decrypt_button = Button(decrypt_window, text="Decrypt", command=decrypt, bg="#32CD38")  # Green button
    decrypt_button.pack(pady=5)

# Main window
root = Tk()
```

```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adap
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adap
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adap
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adap
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adap
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```

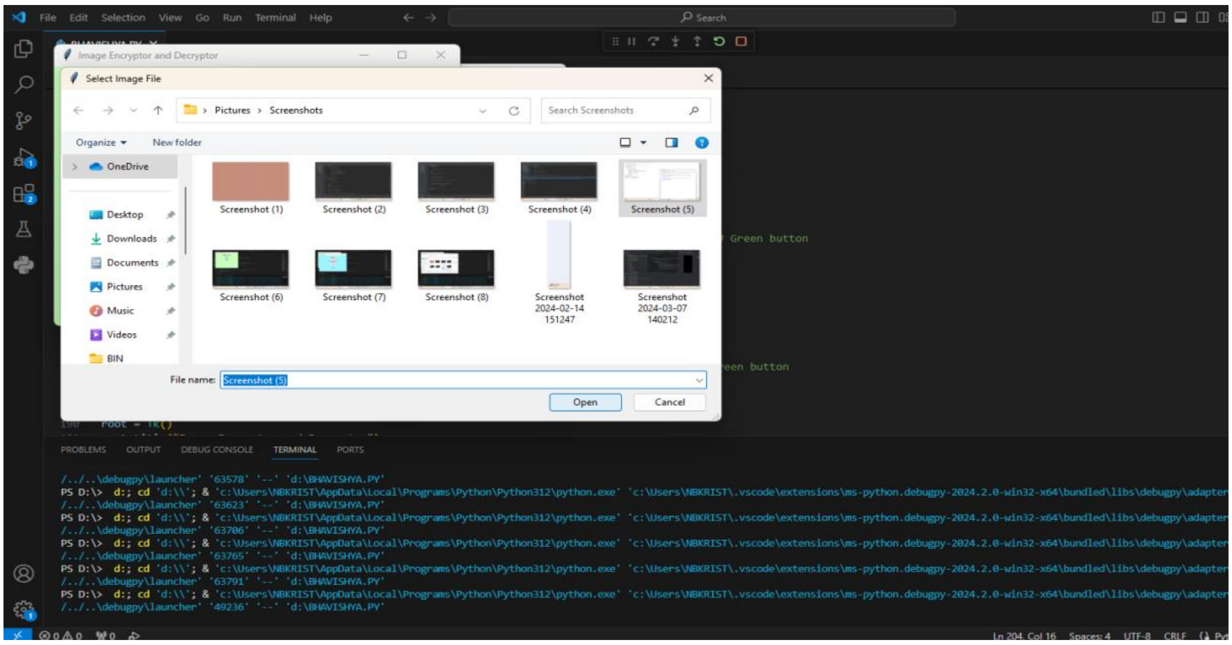


```
", bg="#87CEEB")
_image, bg="#32CD38")  # Green button
B")
ypt, bg="#32CD38")  # Green button

# Main window
root = Tk()
```

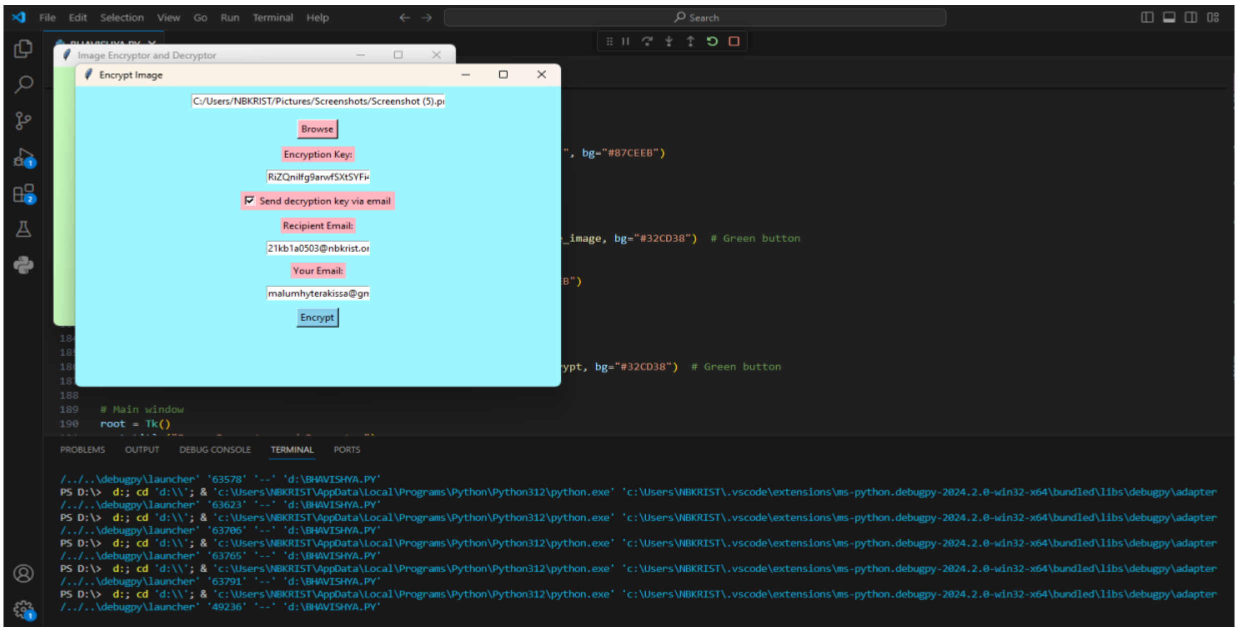
```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```



```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```



```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```


File Edit Selection View Go Run Terminal Help
Search
Image Encryptor and Decryptor
Encrypt Image
Browse
Encrypt
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

C:/Users/NBKRIST/Pictures/Screenshots/Screenshot (5).p

Encryption Key: RiZQnilfg9arwfSXtSYFi

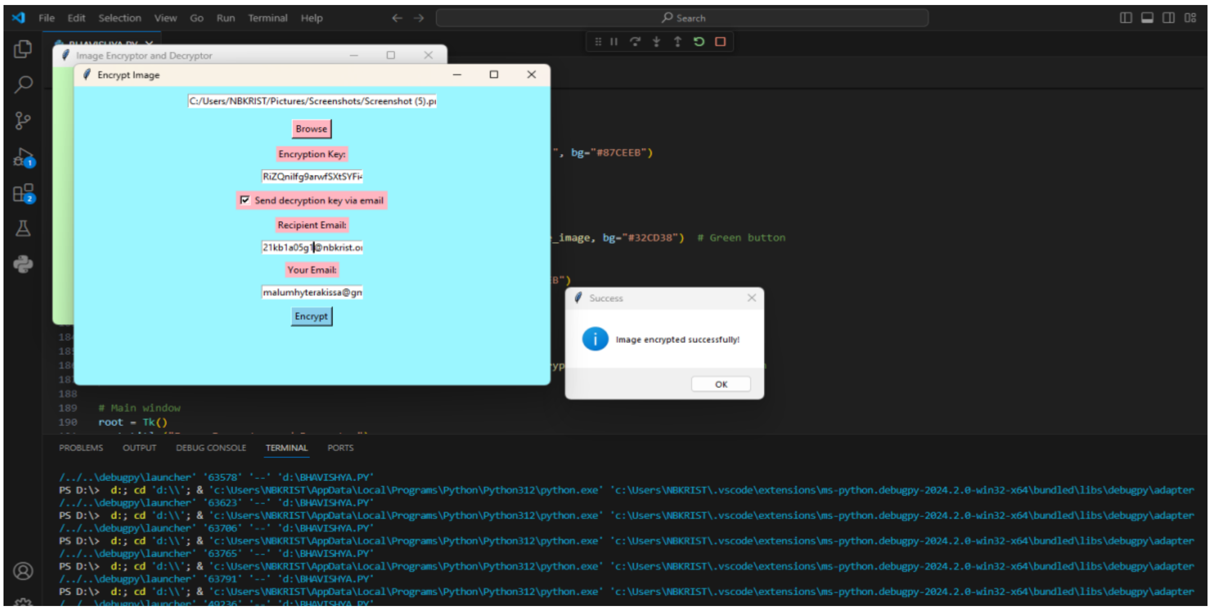
☑ Send decryption key via email

Recipient Email: 21kb1a0503@nbkrist.o

Your Email: malumhyterakissa@gn

```
", bg="#87CEEB")
_image, bg="#32CD38")  # Green button
B")
ypt, bg="#32CD38")  # Green button

# Main window
root = Tk()
```

```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```


File Edit Selection View Go Run Terminal Help
Search
Image Encryptor and Decryptor
Encrypt Image
Browse
Encrypt
Success
OK
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

C:/Users/NBKRIST/Pictures/Screenshots/Screenshot (5).p

Encryption Key: RiZQnilfg9arwfSXtSYFi

☑ Send decryption key via email

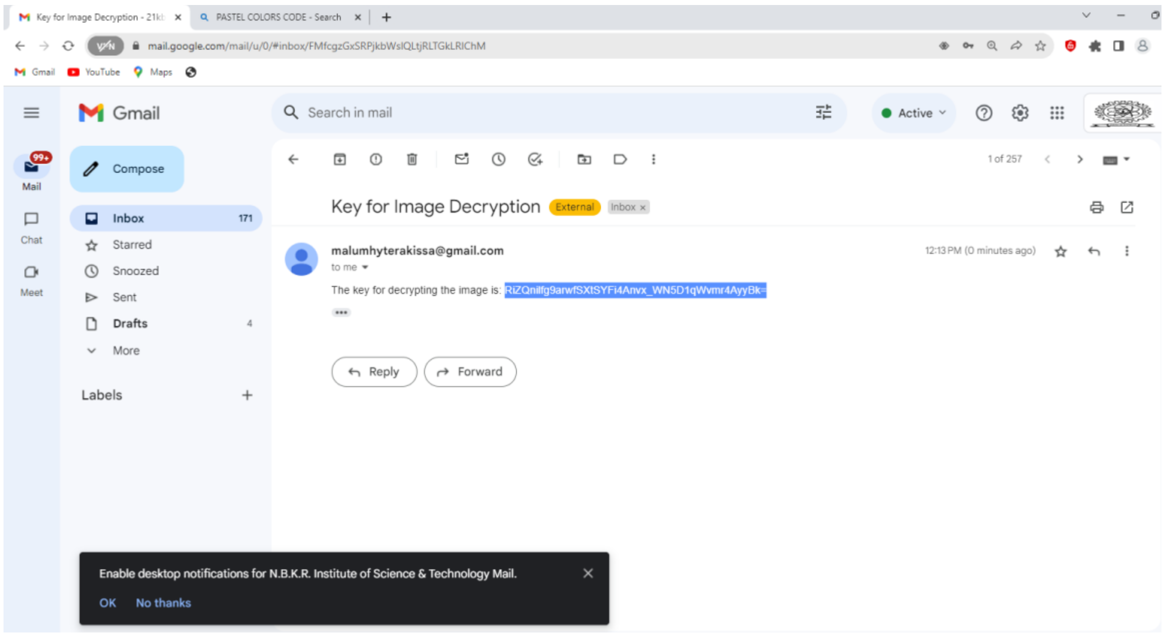
Recipient Email: 21kb1a05g1@nbkrist.o

Your Email: malumhyterakissa@gn

Image encrypted successfully!

```
", bg="#87CEEB")
_image, bg="#32CD38")  # Green button
B")

# Main window
root = Tk()
```

```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```



Key for Image Decryption External Inbox

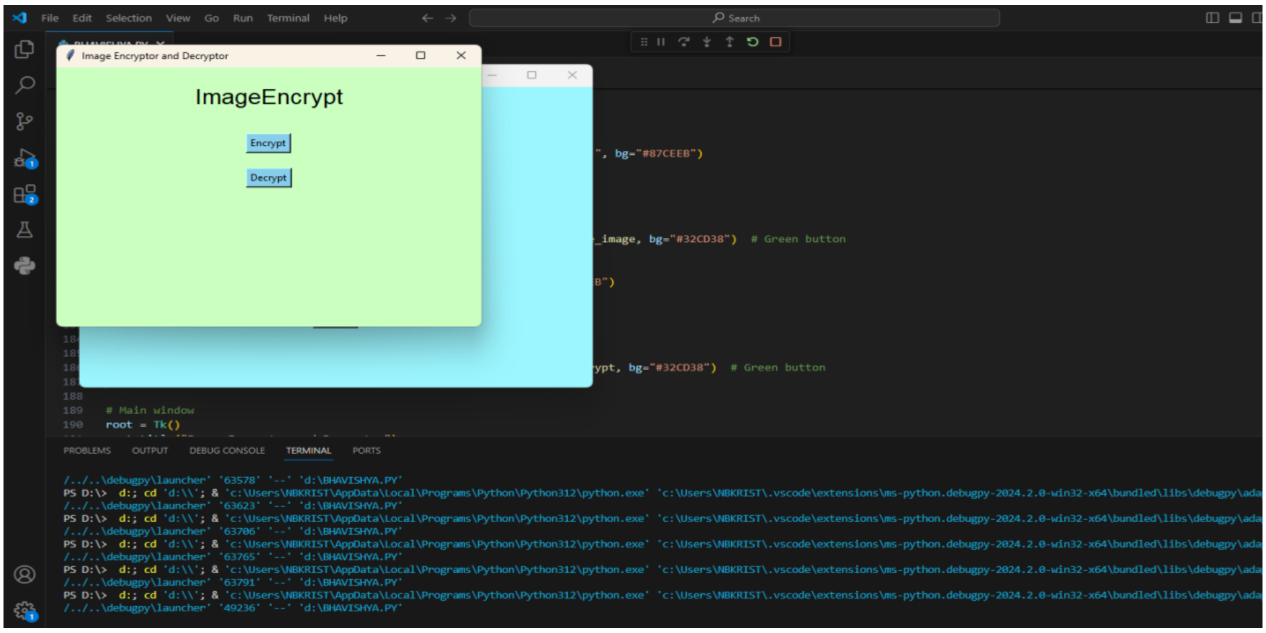
malumhyterakissa@gmail.com

to me

12:13 PM (0 minutes ago)

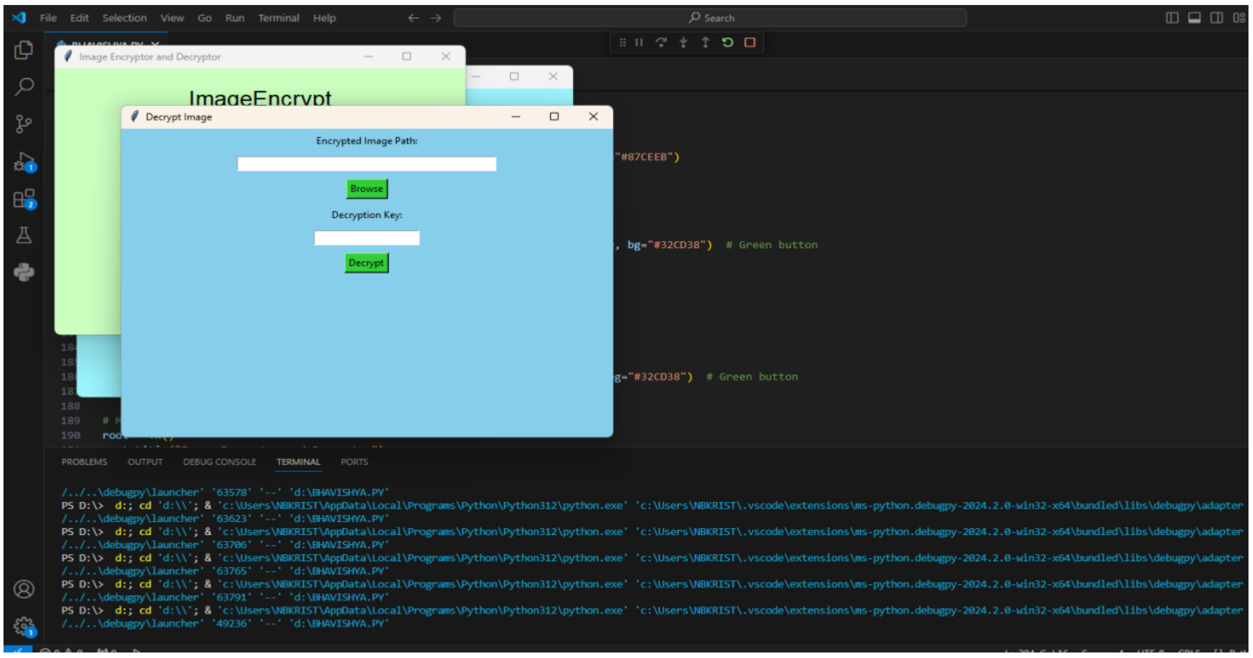
The key for decrypting the image is: RiZQnilfg9arwfSXtSYFi4Anvx_WN5D1qWvmr4AyyBk=

Enable desktop notifications for N.B.K.R. Institute of Science & Technology Mail.



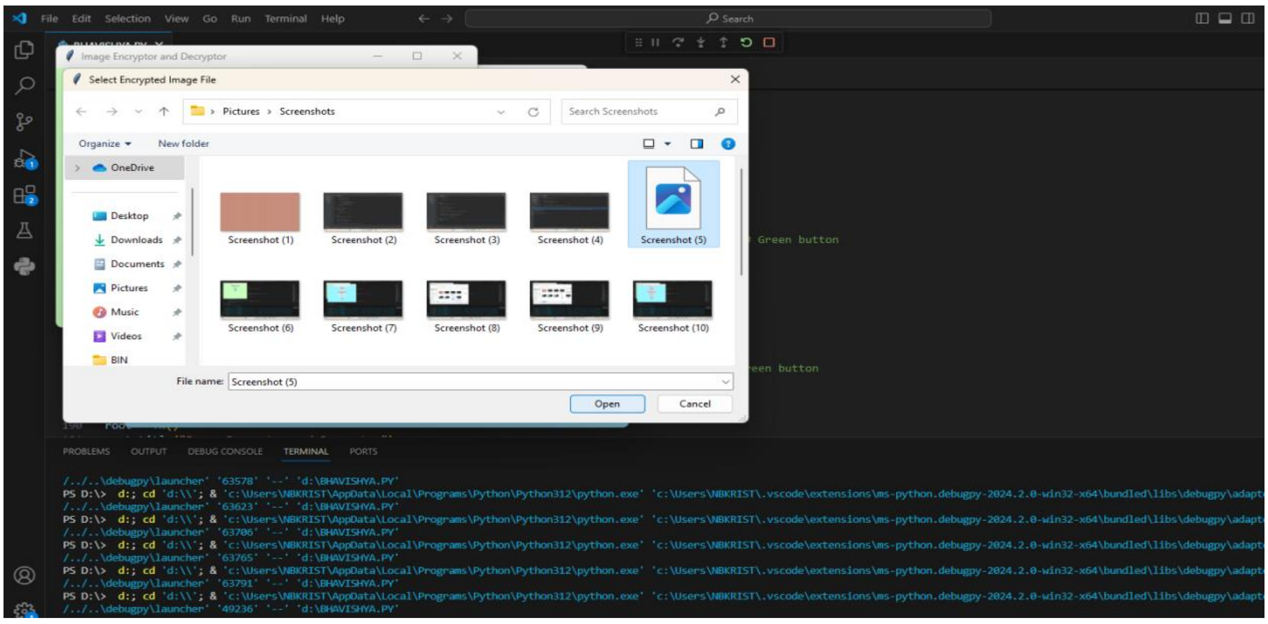
```
", bg="#87CEEB")
_image, bg="#32CD38")  # Green button
B")
ypt, bg="#32CD38")  # Green button

# Main window
root = Tk()
```

```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\ada
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\ada
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\ada
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\ada
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\ada
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```

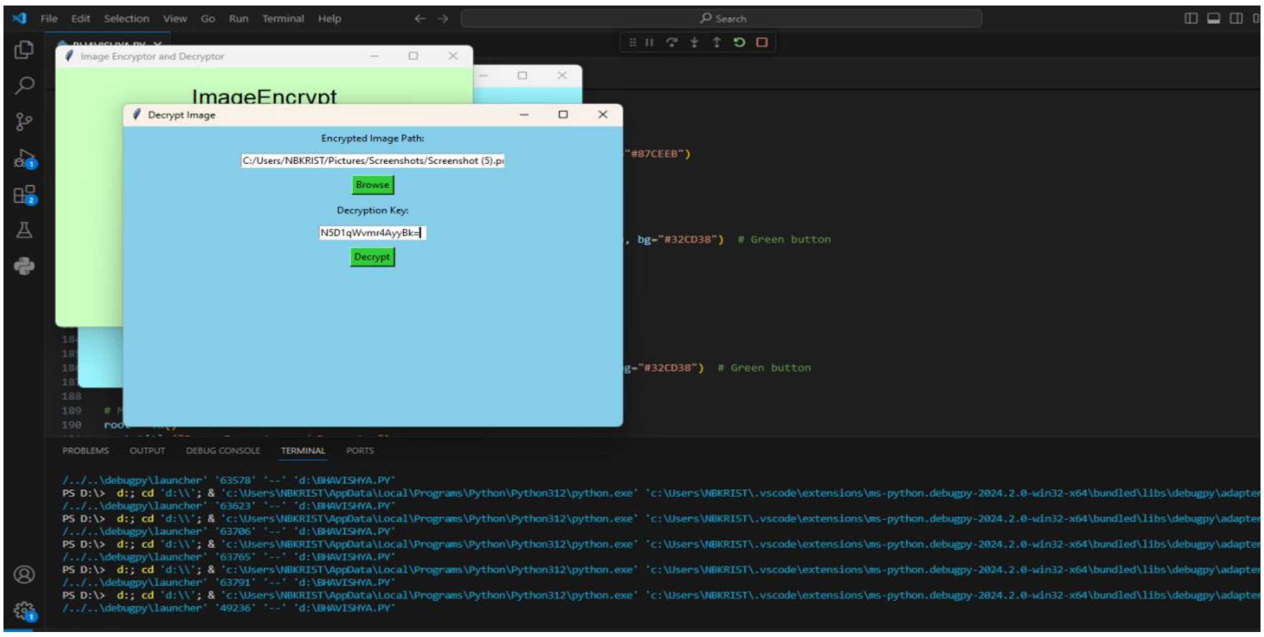


```
"#87CEEB")
, bg="#32CD38")  # Green button
g="#32CD38")  # Green button
```

```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```



```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapt
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapt
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapt
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapt
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapt
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```



Encrypted Image Path: C:/Users/NBKRIST/Pictures/Screenshots/Screenshot (5).p

Decryption Key: N5D1qWvmr4AyyBk=

```
"#87CEEB")
, bg="#32CD38")  # Green button
g="#32CD38")  # Green button
```

```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```

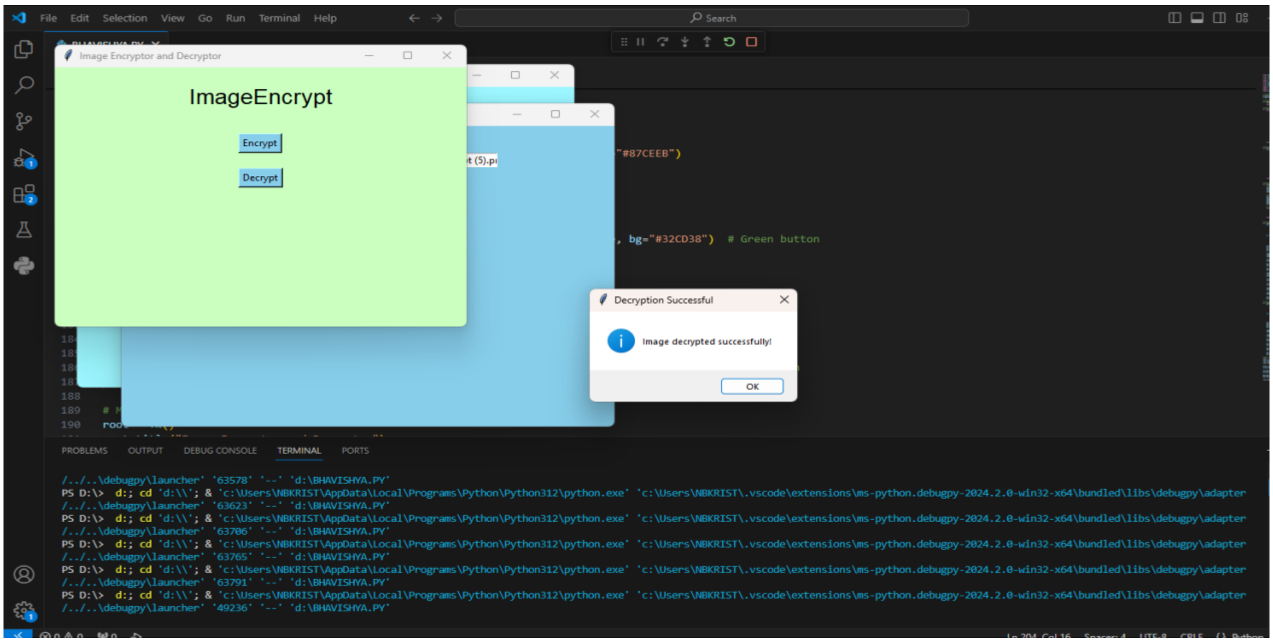
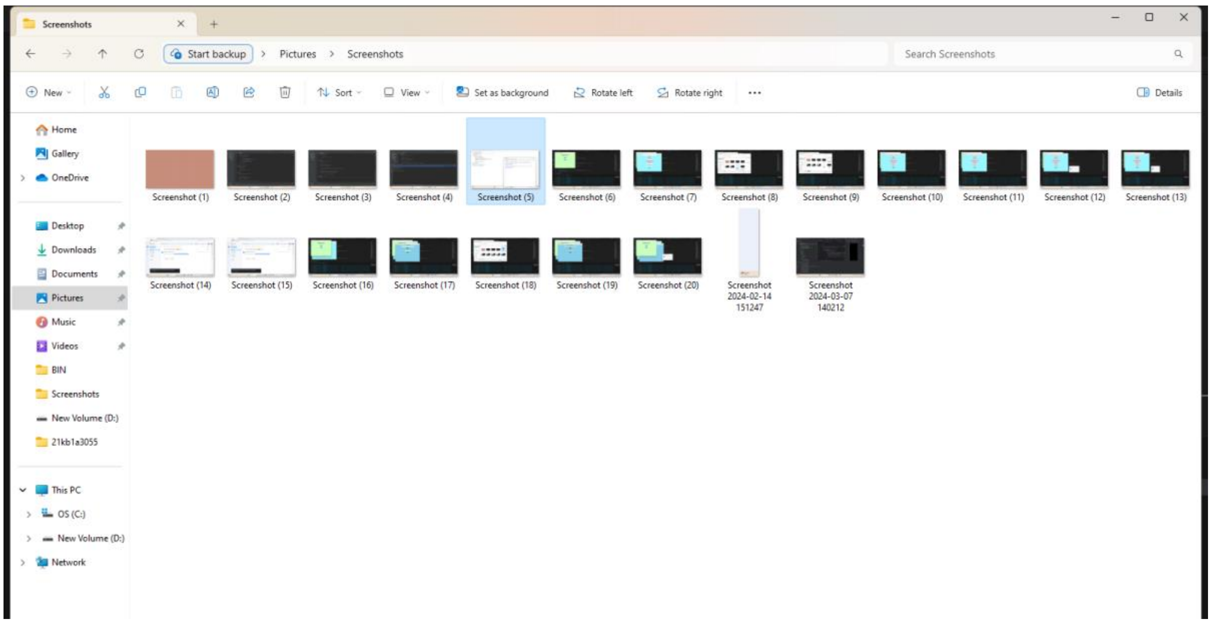


Image decrypted successfully!

```
"#87CEEB")
, bg="#32CD38")  # Green button
```

```
/../..\debugpy\launcher' '63578' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63623' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63706' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63765' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '63791' '--' 'd:\BHAVISHYA.PY'
PS D:\>  d:; cd 'd:\\'; & 'c:\Users\NBKRIST\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\NBKRIST\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundled\libs\debugpy\adapter
/../..\debugpy\launcher' '49236' '--' 'd:\BHAVISHYA.PY'
```



10.Conclusion

Image encryption is a technique in which either a pixel is converted to another pixel or replaced by other pixel in same image (position permutation) or visual transformation based algorithm (Image on image i.e. image as a key, or watermark, based encryption). Here we surveyed some of these algorithms. There are so many technique to make an image secure. Each technique has its own suitability area. Each technique has its own limitations.In this important encryption techniques have been presented and analyzed in order to make familiar with the other encryption algorithms used in encrypting the image which has been transferred over network. The results of the simulation show that every algorithm has advantages and disadvantages based on their techniques which are applied on images. I conclude that all techniques are good for image encryption and give security so that no one can access the image which is in the open network.