IMAGE CRYPTOGRAPHY

A Mini Project-II Report submitted
in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

1. .K.Jahnavi - 19B01A0569
2. K.Chandana - 19B01A0579
3. N.Harshita Naga Sai Chandu - 19B01A0597
4. V.Harshita Sai- 19B01A05A2
5. N.Rishika - 19B01A05C0

Under the esteemed guidance of

Mr.M.Narasimha Raju

VISHNU
UNIVERSAL LEARNING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)
(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202

2021 – 2022

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN (A)

(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

*This is to certify that the Mini Project-II entitled "**Image Cryptography**", is being submitted by **K.Jahnavi, K.Chandana, N.Harshitha Naga Sai Chandu ,V.Harshita Sai, N.Rishika** bearing the **Regd. No. 19B01A0569, 19B01A0579, 19B01A0597, 19B01A05A2, 19B01A05C0** in partial fulfillment of the requirements for the award of the degree of "**Bachelor of Technology** in **Computer Science & Engineering**" is a record of bonafide work carried out by her under my guidance and supervision during the academic year* **2021 – 2022** *and it has been found worthy of acceptance according to the requirements of the university.*

Internal Guide

Head of the Department

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task wouldbe impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing my deep sense of gratitude to **Dr. B. Vishnu Raju**, Chairman of Shri Vishnu Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. G. Srinivasa Rao**, Principal and **Dr. P. Srinivasa Raju**, Vice-principal of SVECW for their constant support and encouragement.

I would also like to thank **Dr. P. Kiran Sree**, Professor and Head of the Department of Computer Science and Engineering, for his constant support.

I am grateful to **Mrs.T.Gayathri**, Project Co-Ordinator for her relentless effort and support to me to keep myself on track throughout the course.

I am grateful to **Dr. P. Kiran Sree (HOD), Mrs. T. Gayatri(Project Co-Ordinator)** and **Mr. M. Narasimha Raju (Project Guide) - PRC Members** for their unfailing encouragement and suggestions, given to me in the course of my project work.

I express my gratitude to **Mr. M. Narasimha Raju,** Assistant Professor, my project guide, forconstantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Finally, a note of thanks to the teaching and non-teaching staff of Dept of Computer Scienceand Engineering, for their cooperation extended to me, and my friends, who helped me directly or indirectly in the course of the project work.

1. **K.Jahnavi - 19B01A0569**
2. **K.Chandana - 19B01A0579**
3. **N.Chandu - 19B01A0597**
4. **V.Harshita Sai – 19B01A05A2**
5. **N.Rishika - 19B01A05C0**

ABSTRACT

Now a day the use of devices such as computer, mobile and many more other devices for communication as well as for data storage and transmission has increased. As a result, there is increase in number of users. Along with these users, there is also increase in number of unauthorized users which are trying to access a data by unfair means. This arises the problem of data security. Images are sent over an insecure transmission channel from different sources, some image data contains secret data, some images itself are highly confidential hence, securing them from any attack is essentially required.

To solve this problem, we are using AES algorithm for encrypting and decrypting image.
So,Image Cryptography is a project dealing with securing images over network. Therefore, no hacker including server administrators and others have access to original message or any type of transmittedinformation through public networks such as internet.

This encrypted data is unreadable to the unauthorized user. This encrypted data can be sent over network and can be decrypted using AES at the receiving end. Hence it ensures secure transmission of image.

Contents

1. INTRODUCTION

IMAGE CRYPTOGRAPHY:

In the current trends, the technologies have been advanced.Most of the individuals prefer using the internet as the primary medium to transfer data from one end to another across the internet. There are many possible ways to transmit data using the internet like: via e-mails, sending text and images, etc. In the present communication world, images are widely in use. However, one of the main problems with sending data over the Internet is the 'security' and authenticity.

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.

Image encryption is a technique that convert original image to another form that is difficult to understand. No one can access the content without knowing a decryption key. Image encryption has applications in corporate world, health care, military operations, and multimedia systems.

AES stands for Advanced Encryption Standard and is a majorly used symmetric encryption algorithm. It is mainly used for encryption and protection of electronic data. It was used as the replacement of DES(Data encryption standard) as it is much faster and better than DES. AES consists of three block ciphers and these ciphers are used to provide encryption of data.

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.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system private key bulk encryption algorithms such as Triple DES are not suitable for transmission of large amounts of data (such as images). Due to the complexity of their internal structure,they are not particularly fast in terms of execution speed and cannot be applied for images in the real time scenario.

Also traditional cryptographic techniques such as DES cannot be applied to images due to the intrinsic properties of images such as bulk data capacity, redundancy and high correlation among pixels.

Here are few disadvantages for using DES :

- It is broken using brute-force search. However, using 3DES mitigates this issue at the cost of increasing execution time.
- DES is also vulnerable to attacks using linear cryptanalysis. However, it takes 247 known plaintexts to break DES in this manner.
- The 56 bit key size is the largest defect of DES and the chips to implement one million of DES encrypt or decrypt operations a second are applicable (in 1993).
- Hardware implementations of DES are very quick.
- DES was not designed for application and therefore it runs relatively slowly.
- In a new technology, it is improving a several possibility to divide the encrypted code, therefore AES is preferred than DES.

2.2 PROPOSED SYSTEM

In the proposed system ,to achieve highly secured image encryption decryption technique we use AES Key expansion for encrypting and decrypting images over network.

ADVANTAGES :

- As it is implemented in both hardware and software, it is most robust security protocol.
- It uses higher length key sizes such as 128, 192 and 256 bits for encryption. Hence it makes AES algorithm more robust against hacking.
- It is most common security protocol used for wide various of applications such as wireless communication, financial transactions, e-business, encrypted data storage etc.
- It is one of the most spread commercial and open source solutions used all over the world.
- No one can hack your personal information.
- For 128 bit, about 2^{128} attempts are needed to break. This makes it very difficult to hack it as a result it is very safe protocol.

Apart from these all benefits, the system objectives also include considerations regarding graphical user interface and other visual aspects.

2.3 FEASIBILITY STUDY

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

Operational Feasibility: Operational feasibility deals with the study of prospects of the system to be developed. This system provides the summarizer of the youtube video using Youtube Transcript API. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility: Economic feasibility or Cost-benefit is an assessment of the economic justification for the computer based project. As hardware was installed from the beginning and for lots of purposes thus the cost on hardware is low. So, the project is economically feasible.

Technical Feasibility: Technical Feasibility is the process of validating the technology assumptions, architecture and design of a product or project.. We used colab notebook for running the code. Thus, the project is technically feasible.

3.SYSTEM REQUIREMENTS & SPECIFICATIONS

3.1 HARDWARE REQUIREMENTS

- System : Intel Core i5&above
- RAM : 256MB
- Hard Disk : 10 GB
- Input Device : Keyboard and Mouse
- Output Device : Monitor or PC

3.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7, 10 or Higher Versions
- Platform : Python IDE
- Programming Language : Python

3.3 FUNCTIONAL REQUIREMENTS

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish

- 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 ensures that 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.

The requirements are usually described in an abstract way. However, functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on.

4. SYSTEM DESIGN

4.1 INTRODUCTION

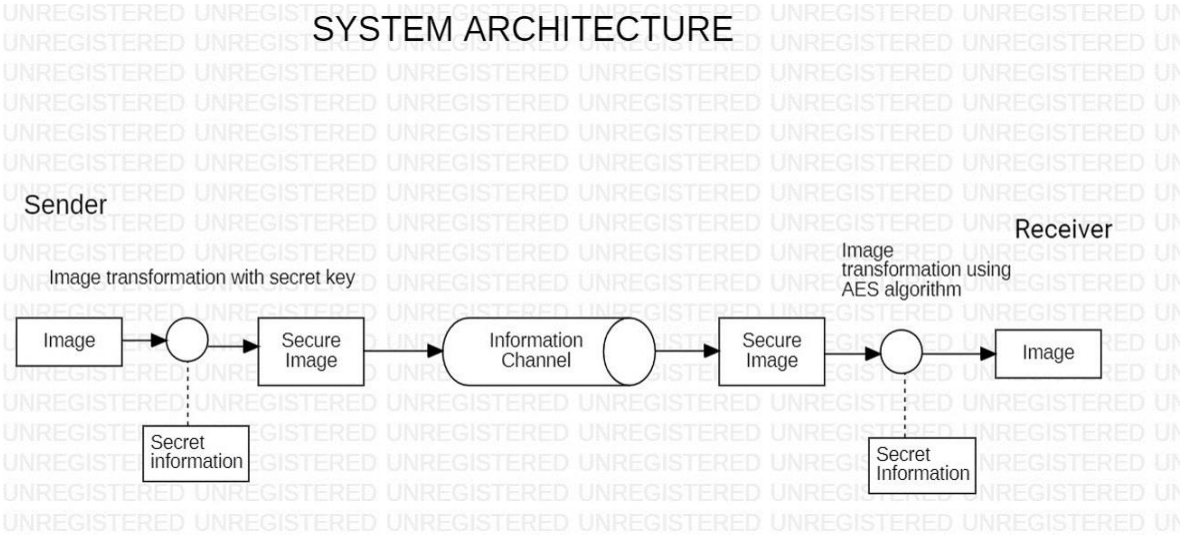
Design is the first step in the development phase of an engineering product or system. Design is the place where quality is considered in the software development. Design is the only way that we can accurately translate user requirements into finished software product or system. Software design serves as the foundation for all the software engineers and software maintenance that steps follow.

DESIGN GOALS:

Few system design goals are as follows:

0. **Performance requirement:** For smooth & efficient encryption, image size must be less than 5MB. Decryption should not take more than 10 seconds.
1. **Platform constraints:** The main target is to encrypt and decrypt the image using private key.
2. **Accuracy and Precision**: Requirements are accuracy and precision of the data given as input as well as produced as output.
3. **Reliability**: External factors do not affect the system. AES algorithm is universally accepted and generates consistent results therefore there are very less chances of errors. Error can occur only if there is a transmission glitch (the probability of which is very rare). So, the system is reliable..
4. **Safety and Security Requirements :** If the decryption takes more than 10 seconds, then discard the message (because the message might have been corrupted during transmission) and ask sender to re-send it. Encryption is done using encryption key. Decryption will happen only when same encryption key is used at the receiver side.
5. **Usability:** It uses python based GUI which is user friendly and provides buttons for easy navigation. A person with basic understanding of computer can easily use this software for encrypting/decrypting image using key.
6. **Testability:** The system is easy to test and find defects. The system is divided into different modules performing specific functions that can be tested individually

4.2 SYSTEM ARCHITECTURE:



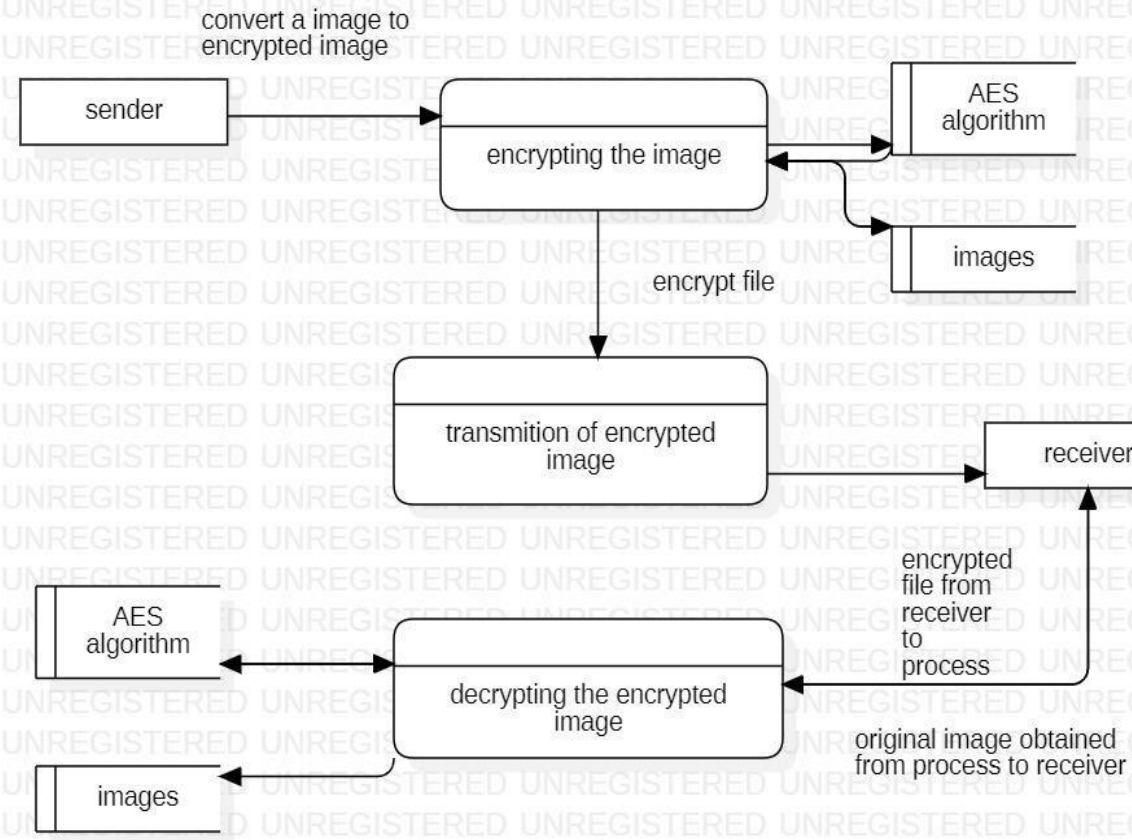
The project works by encrypting the given image using AES algorithm so that this image can be sent securely over the network. At the receiver side, the receiver has code for decrypting the image so that he can get the original image. This helps in sending confidential and sensitive information securely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

4.3 Data flow diagrams (UML Diagrams)

Introduction to UML

A model is an abstract representation of system, constructed to understand the system priority to building or modifying it. A model is a simplified representation of reality and it provides a means for conceptualization and communication of ideas in a precise and ambiguous form. We build models so that we can better understand the system we are developing. The elements are like components which can be associated in different ways to make a complete UML picture, which is known as diagram. Thus, it is very important to understand the different diagrams to implement the knowledge in real life systems.

UML (Unified Modeling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It is a method for describing the system architecture in detail using the blueprint. We use UML diagrams to portray the behavior and structure of a system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.



There are various kinds of methods in software design:

1. Use case Diagram
2. Class Diagram
3. Sequence Diagram
4. Activity Diagram
5. State Chart Diagram
6. Communication Diagram

4.2 UML DIAGRAMS

Use Case Diagram: It represents the functionality of a system by utilizing actors and use cases. It encapsulates the functional requirement of a system and its association with actors. It portrays the use case view of a system.

Following are the purposes of a use case diagram given below:

- It gathers the system's needs.
- It depicts the external view of the system.
- It recognizes the internal as well as external factors that influence the system.
- It represents the interaction between the actors.

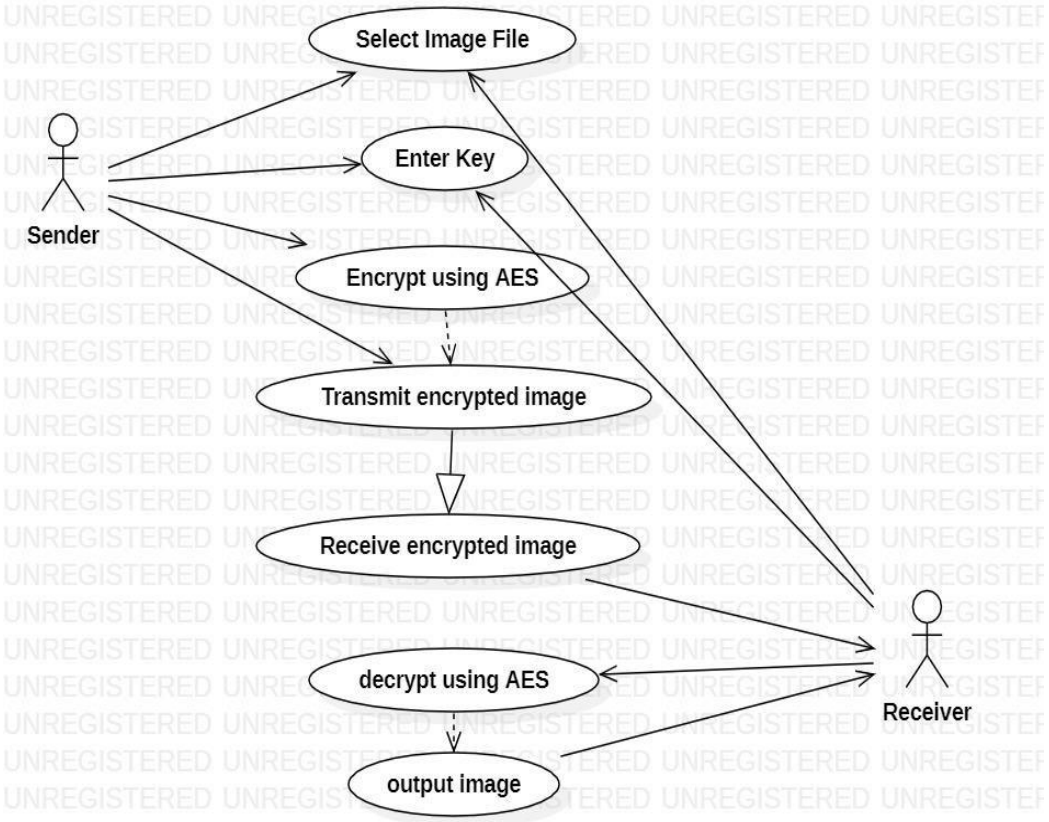


Figure 4.2.1 Use Case diagram for Image Cryptography

Class Diagram: Class diagrams are one of the most widely used diagrams. It is the backbone of all the object-oriented software systems. It depicts the static structure of the system. It displays the system's class, attributes, and methods. It is helpful in recognizing the relation between different objects as well as classes.

Following are the purposes of a class diagram given below:

- It analyses and designs a static view of an application.
- It describes the major responsibilities of a system.
- It is a base for component and deployment diagrams.
- It incorporates forward and reverse engineering.
-

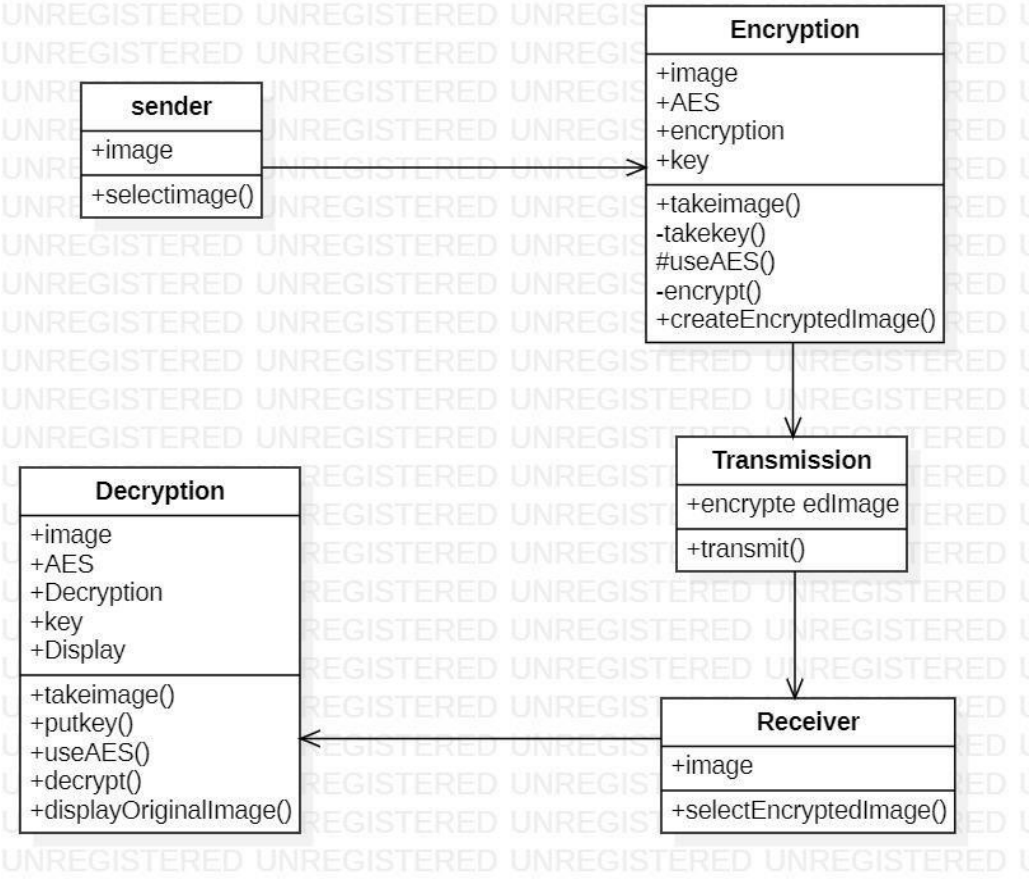


Figure 4.2.2 Class diagram Image Cryptography

State Chart Diagram: It is a behavioral diagram. it portrays the system's behavior utilizing finite state transitions. It is also known as the State Machine diagram. It models the dynamic behavior of a class in response to external stimuli.

Following are the purposes of a state-chart diagram given below:

- For modeling the object states of a system.
- For modeling the reactive system as it consists of reactive objects.
- For pinpointing the events responsible for state transitions.
- For implementing forward and reverse engineering.

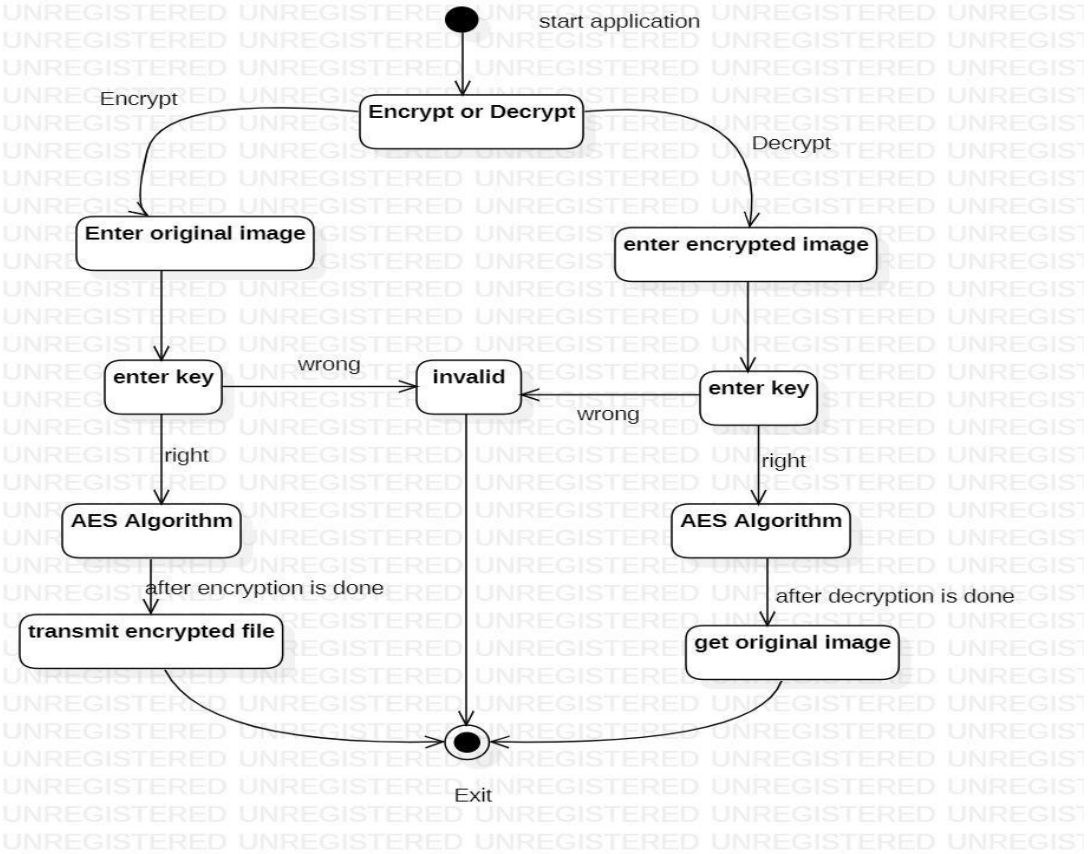


Figure 4.2.3 State Chart diagram for Image Cryptography

Activity Diagram: It models the flow of control from one activity to the other. With the helpof an activity diagram, we can model sequential and concurrent activities. It visually depicts the workflow as well as what causes an event to occur.

Following are the purposes of an activity diagram given below:

- models processes and workflow
- envisions the dynamic behavior of the system
- deals with flow that can be sequential, branched, or concurrent
- has fork and join elements to show special flow

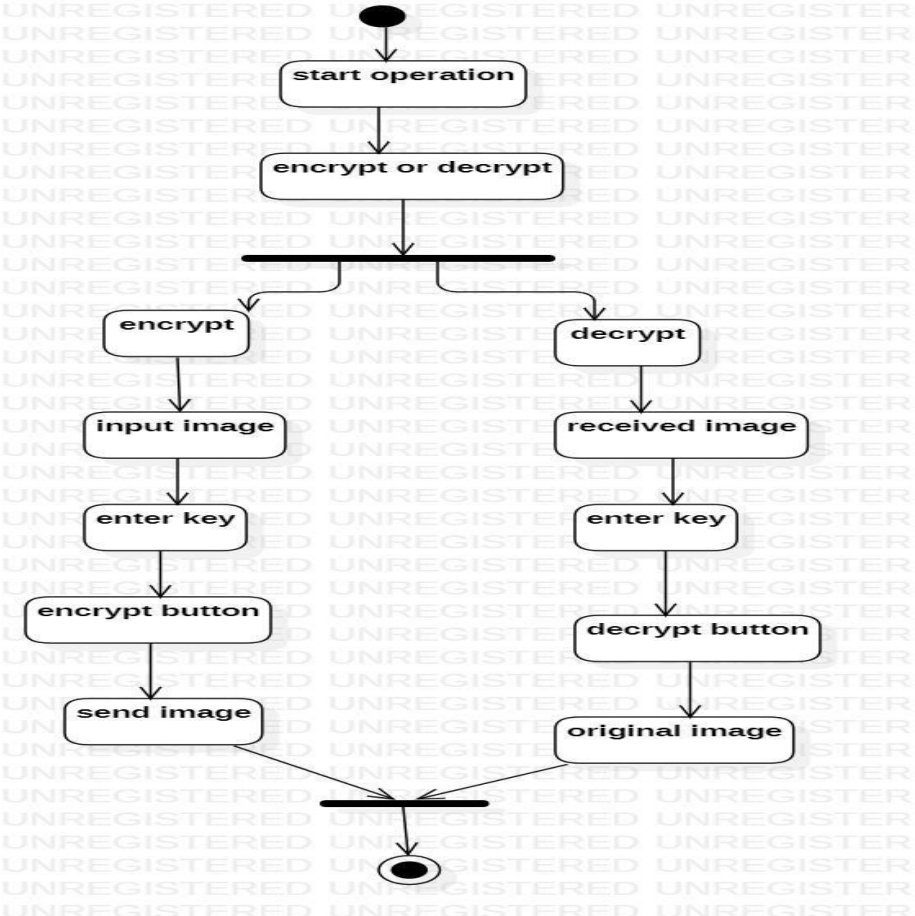


Figure 4.2.4 Activity diagram for Image Cryptography

Sequence Diagram: It shows the interactions between the objects in terms of messages exchanged over time. It delineates in what order and how the object functions are in a system.

Following are the purposes of a sequence diagram given below:

- To model high-level interaction among active objects within a system.
- To model interaction among objects inside a collaboration realizing a use case.
- It either model generic interactions or some certain instances of interaction.

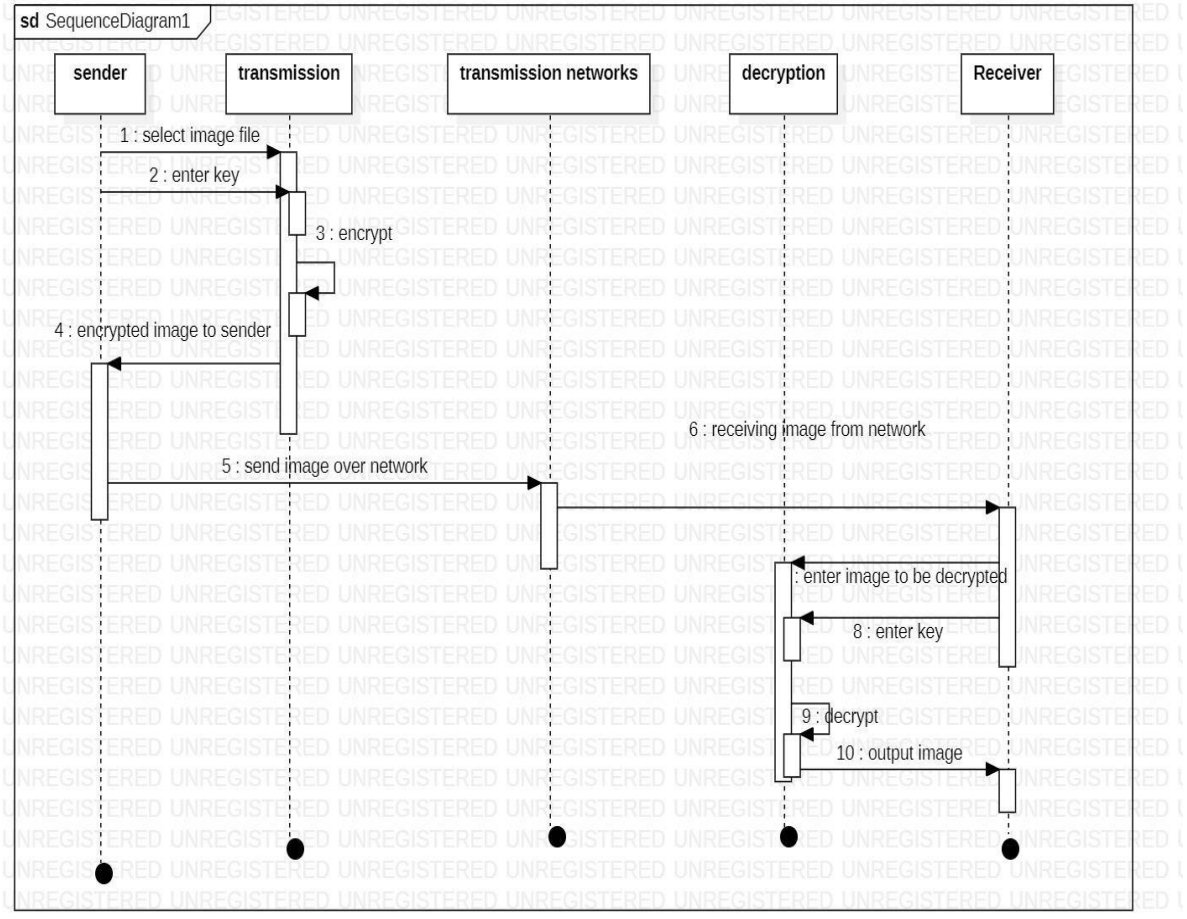


Figure 4.2.5 Sequence Diagram for Image Cryptography

Component Diagram: It portrays the organization of the physical components within the system. It is used for modeling execution details. It determines whether the desired functional requirements have been considered by the planned development or not, as it depicts the structural relationships between the elements of a software system.

Following are the purposes of a component diagram given below:

- It portrays the components of a system at the runtime.
- It is helpful in testing a system.
- It envisions the links between several connections.

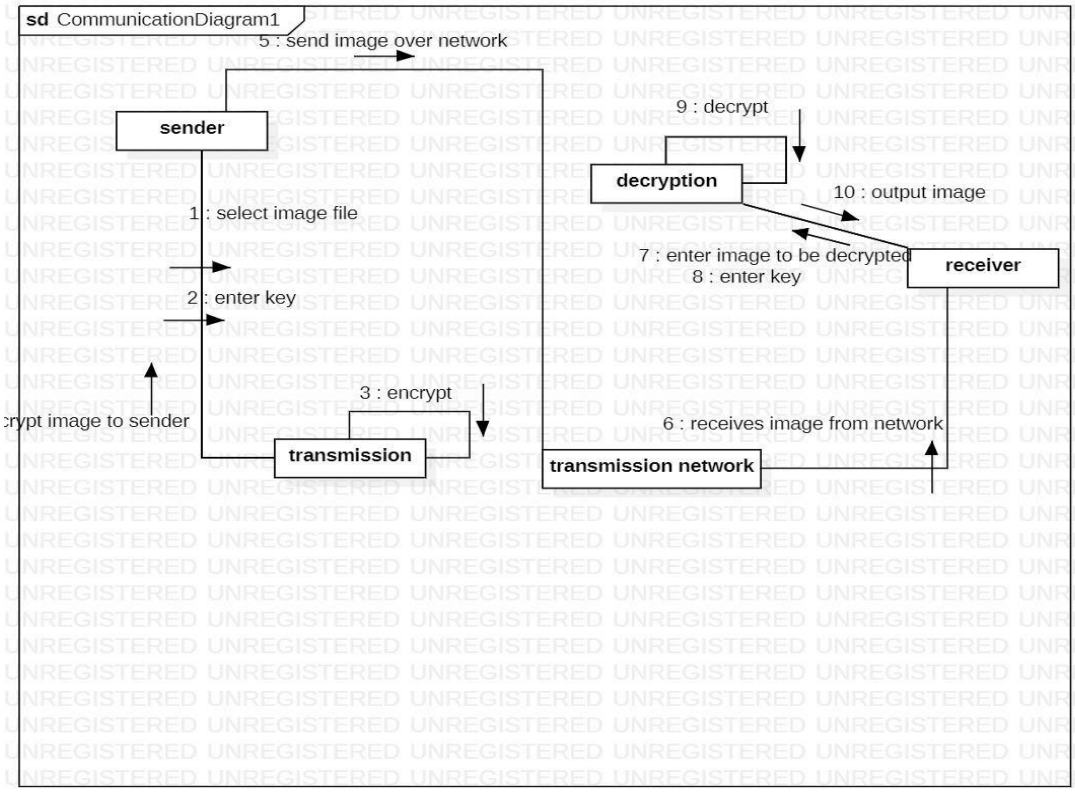


Figure 4.2.6 Component Diagram for Image Cryptography

5.SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

The purpose of system implementation can be summarized as follow making the new system available to the prepared set of users (the deployment), and positioning on- going support and maintenance of the system within the performing organization (the transaction). At a finer necessary to educate the consumer on the use of system, placing the newly developed system into production, confirming that business functions that interact with the system and functioning properly. Transitioning the system support responsibilities involve changing from a system development to the system and maintenance mode of operation, with ownership of the new system moving from the project team to the performing organization.

A key difference between system implementation and all other phases of lifecycle is that all project activities up to this point have been performed in safe, protected and check your environments. It is through the careful planning, execution and management of system implementation activities that the project team can minimize the likelihood of these occurrences and determine appropriate contingency plans in the event of the problem.

Our project explores Implementation of our system consists of encryption and decryption of Images.

5.2 Source code:

ImageCryptography.py - Notepad

File Edit View

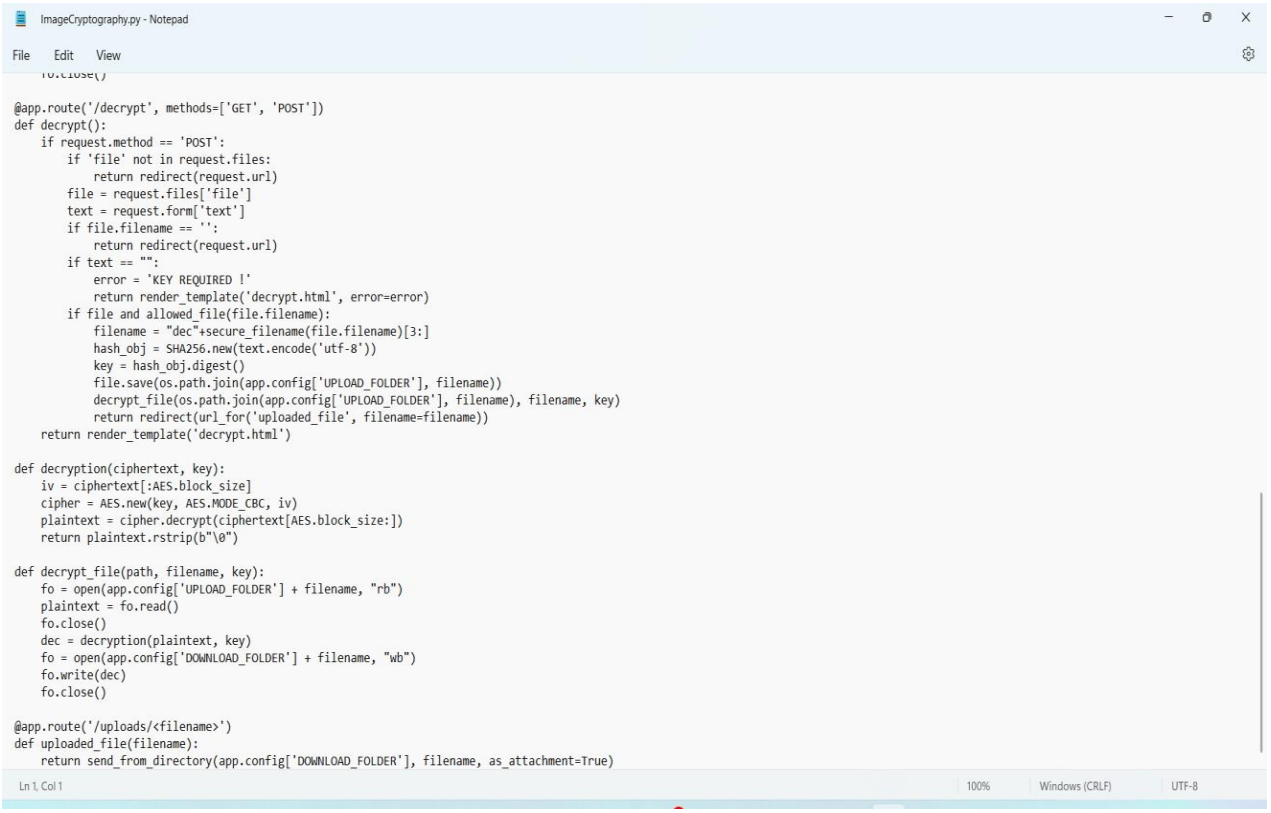
```
from flask import Flask, request, render_template, url_for, redirect, send_from_directory, flash
from werkzeug.utils import secure_filename
import os

from Crypto import Random
from Crypto.Cipher import AES
from Crypto.Hash import SHA256

UPLOAD_FOLDER = os.path.dirname(os.path.abspath(__file__)) + '/uploads/'
DOWNLOAD_FOLDER = os.path.dirname(os.path.abspath(__file__)) + '/downloads/'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}

app = Flask(__name__, static_url_path="/static")
DIR_PATH = os.path.dirname(os.path.realpath(__file__))
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.config['DOWNLOAD_FOLDER'] = DOWNLOAD_FOLDER
app.config['MAX_CONTENT_LENGTH'] = 8 * 1024 * 1024

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/')
@app.route('/home')
def home_page():
    return render_template('home.html')

@app.route('/encrypt', methods=['GET', 'POST'])
def encrypt():
    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files['file']
        text = request.form['text']
        if file.filename == '':
            return redirect(request.url)
        if text == "":
            error = 'KEY REQUIRED !'
            return render_template('encrypt.html', error=error)
        if file and allowed_file(file.filename):
            filename = "enc_"+secure_filename(file.filename)
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

ImageCryptography.py - Notepad

File Edit View

```
            return redirect(request.url)
        if text == "":
            error = 'KEY REQUIRED !'
            return render_template('encrypt.html', error=error)
        if file and allowed_file(file.filename):
            filename = "enc_"+secure_filename(file.filename)
            hash_obj = SHA256.new(text.encode('utf-8'))
            key = hash_obj.digest()
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            encrypt_file(os.path.join(app.config['UPLOAD_FOLDER'], filename), filename, key)
            return redirect(url_for('uploaded_file', filename=filename))
    return render_template('encrypt.html')

def encryption(message, key, key_size=256):
    message = message + b"\0" * (AES.block_size - len(message) % AES.block_size)
    iv = Random.new().read(AES.block_size)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    return iv + cipher.encrypt(message)

def encrypt_file(path, filename, key):
    fo = open(app.config['UPLOAD_FOLDER']+filename, "rb")
    plaintext = fo.read()
    fo.close()
    enc = encryption(plaintext, key)
    fo = open(app.config['DOWNLOAD_FOLDER']+filename, "wb")
    fo.write(enc)
    fo.close()

@app.route('/decrypt', methods=['GET', 'POST'])
def decrypt():
    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files['file']
        text = request.form['text']
        if file.filename == '':
            return redirect(request.url)
        if text == "":
            error = 'KEY REQUIRED !'
            return render_template('decrypt.html', error=error)
        if file and allowed_file(file.filename):
```


ImageCryptography.py - Notepad

File Edit View

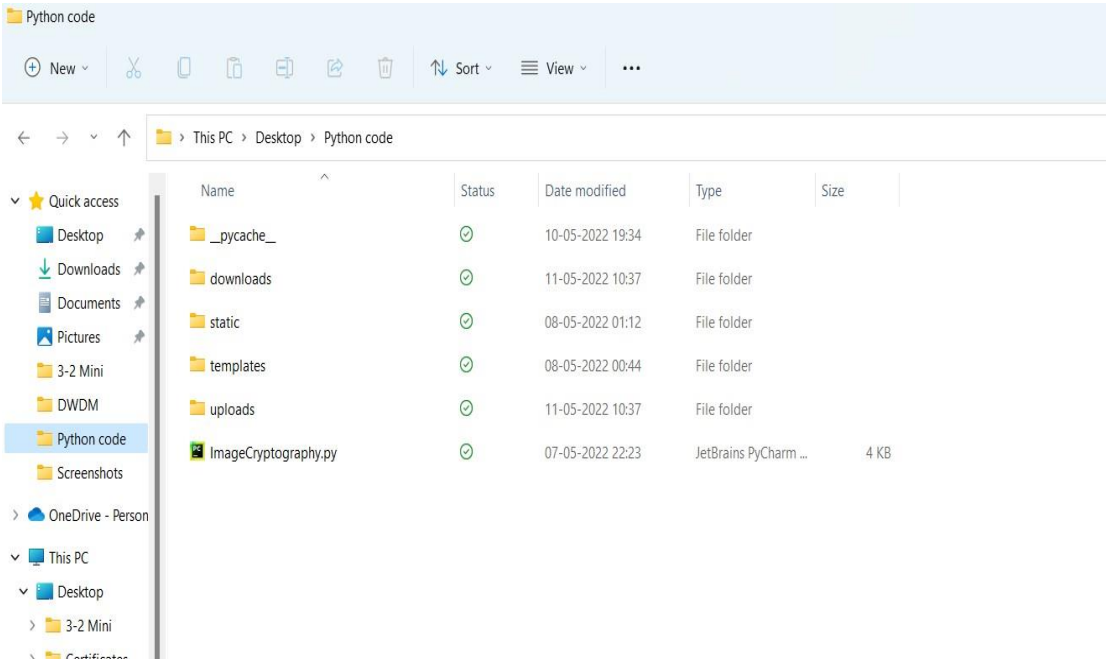
```
    fo.close()

@app.route('/decrypt', methods=['GET', 'POST'])
def decrypt():
    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files['file']
        text = request.form['text']
        if file.filename == '':
            return redirect(request.url)
        if text == "":
            error = 'KEY REQUIRED !'
            return render_template('decrypt.html', error=error)
        if file and allowed_file(file.filename):
            filename = "dec"+secure_filename(file.filename)[3:]
            hash_obj = SHA256.new(text.encode('utf-8'))
            key = hash_obj.digest()
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            decrypt_file(os.path.join(app.config['UPLOAD_FOLDER'], filename), filename, key)
            return redirect(url_for('uploaded_file', filename=filename))
    return render_template('decrypt.html')

def decryption(ciphertext, key):
    iv = ciphertext[:AES.block_size]
    cipher = AES.new(key, AES.MODE_CBC, iv)
    plaintext = cipher.decrypt(ciphertext[AES.block_size:])
    return plaintext.rstrip(b"\0")

def decrypt_file(path, filename, key):
    fo = open(app.config['UPLOAD_FOLDER'] + filename, "rb")
    plaintext = fo.read()
    fo.close()
    dec = decryption(plaintext, key)
    fo = open(app.config['DOWNLOAD_FOLDER'] + filename, "wb")
    fo.write(dec)
    fo.close()

@app.route('/uploads/<filename>')
def uploaded_file(filename):
    return send_from_directory(app.config['DOWNLOAD_FOLDER'], filename, as_attachment=True)
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

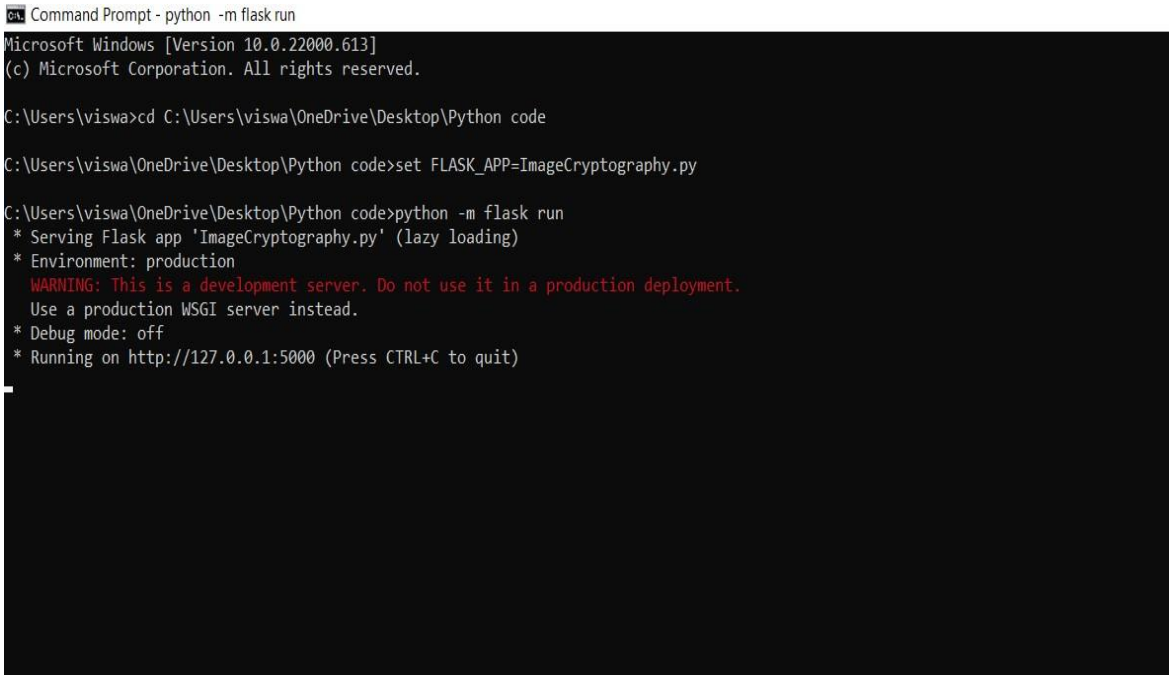
.

5.3 SCREENS :

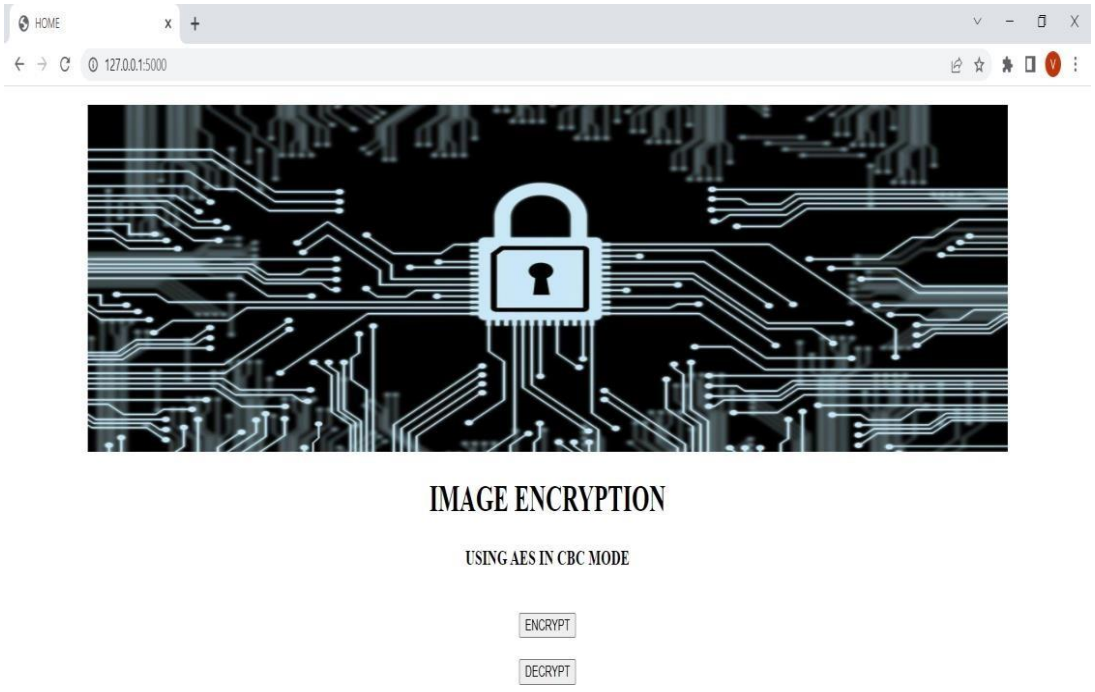
STEP-1 :create a folder on the Desktop which includes all the files related to the project and codes for the templates



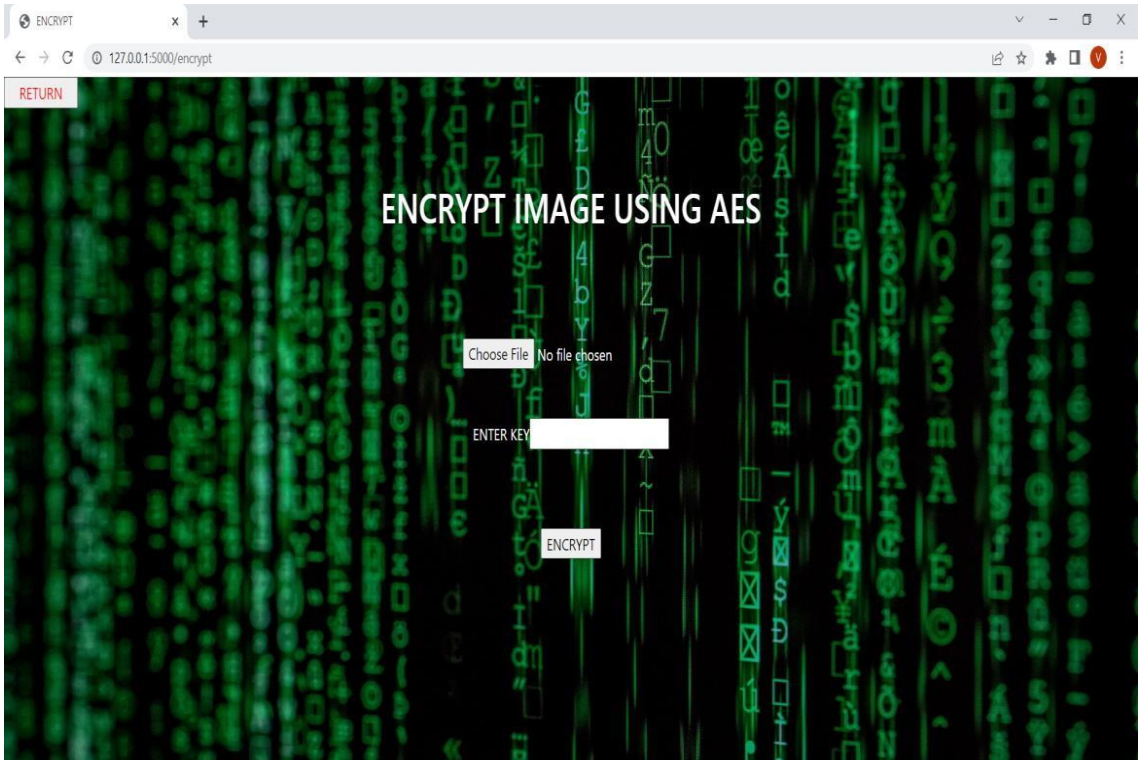
STEP-2: Open a command prompt and run these commands ,then a server is running in the browser.



STEP-3: The page displayed as follows in the browser



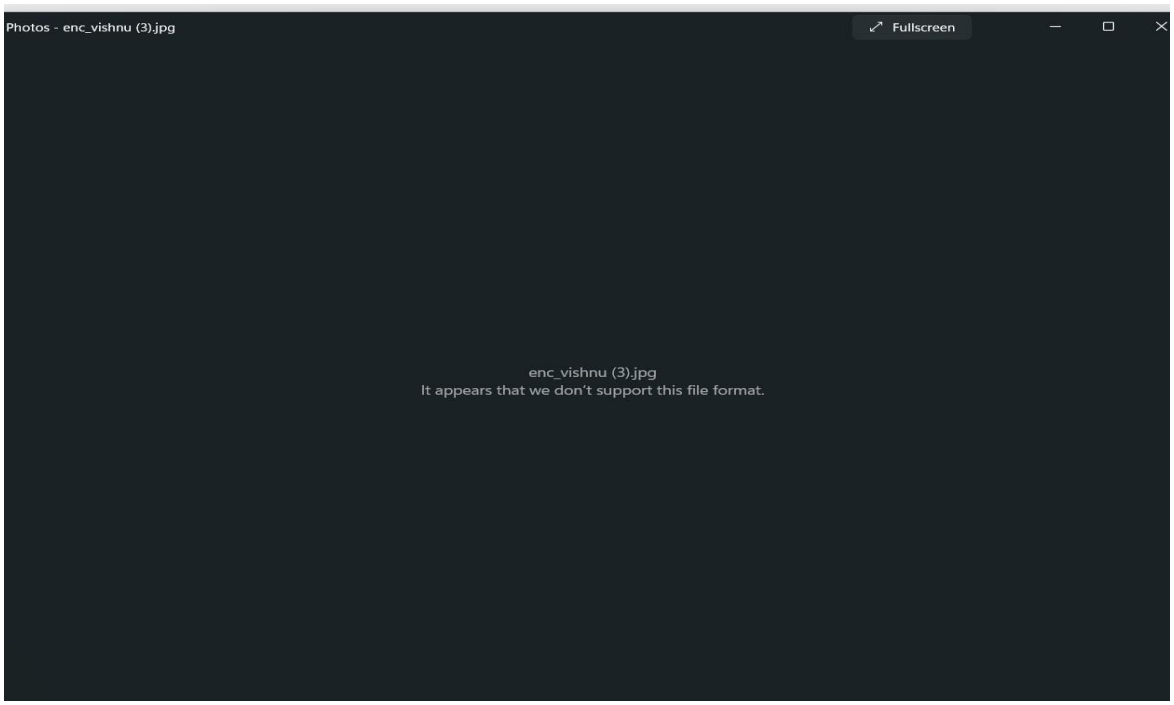
STEP-4: Encrypt web page



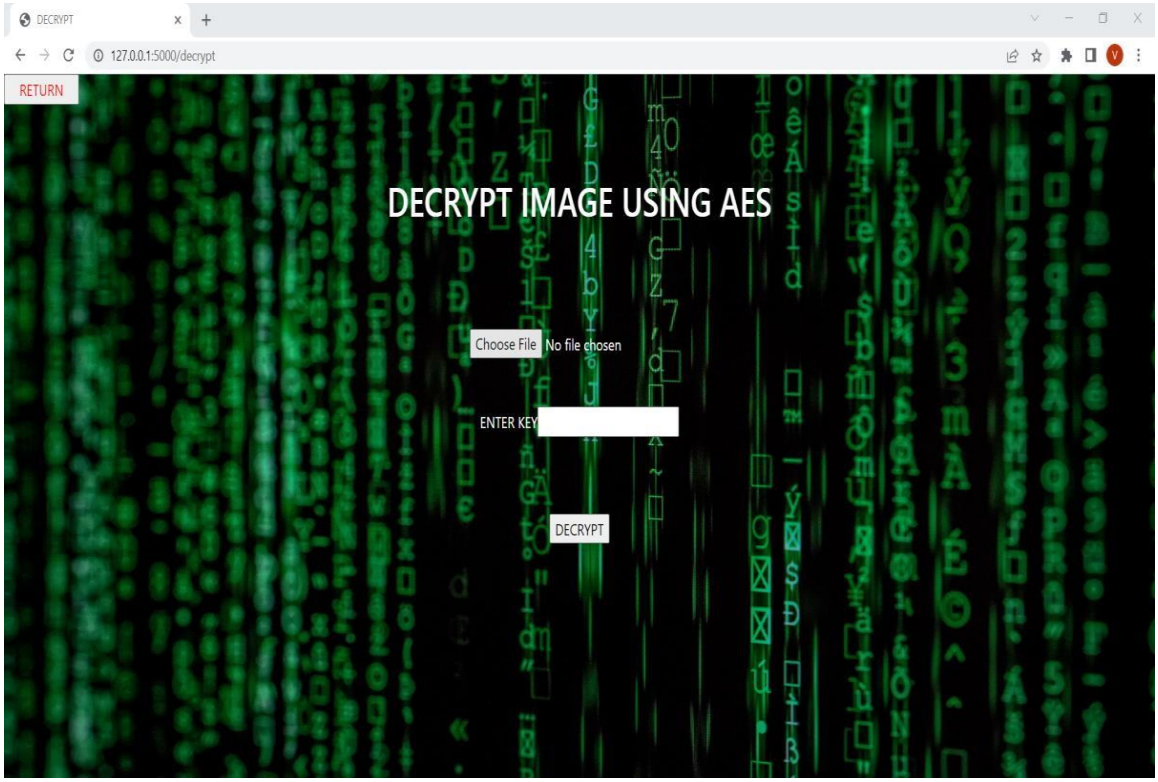
STEP-5 : After file choosen and key is entered .Press encrypt then encrypted image is downloaded into our system.

Original image :

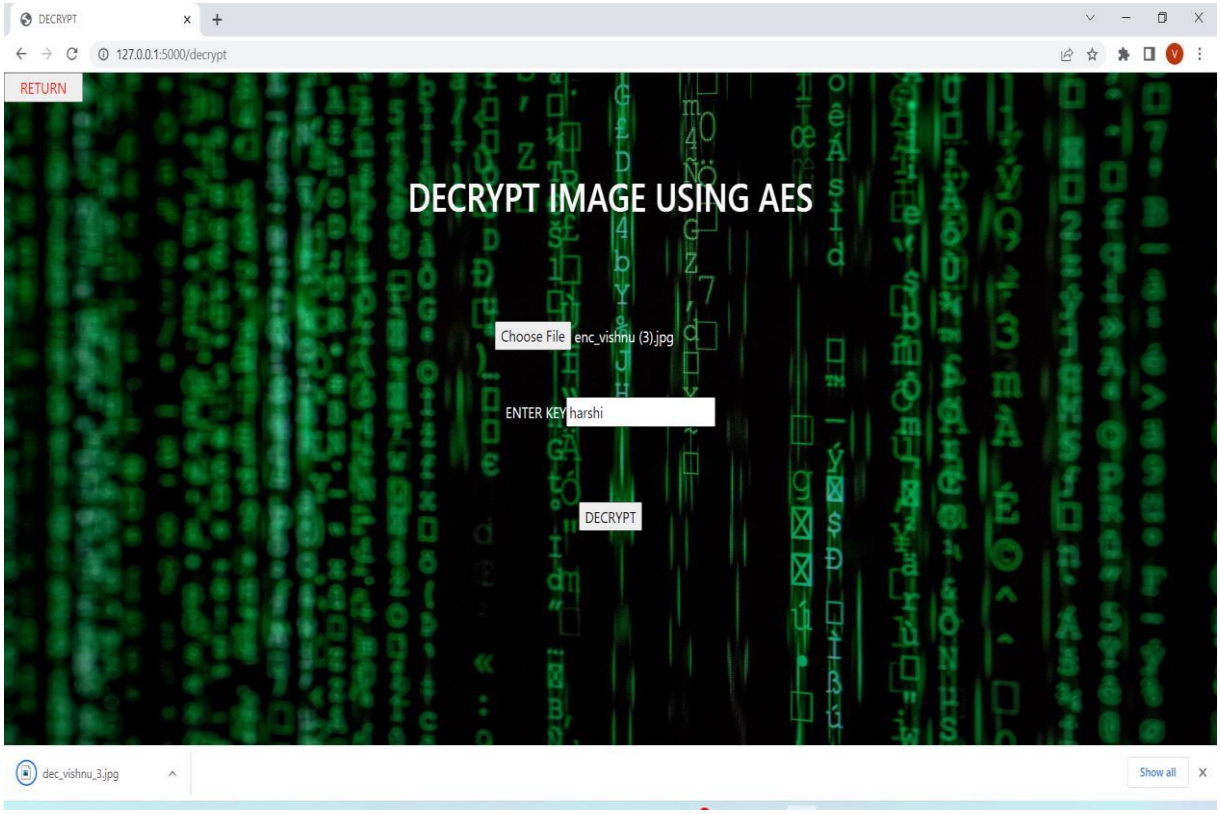
Encrypted Image :



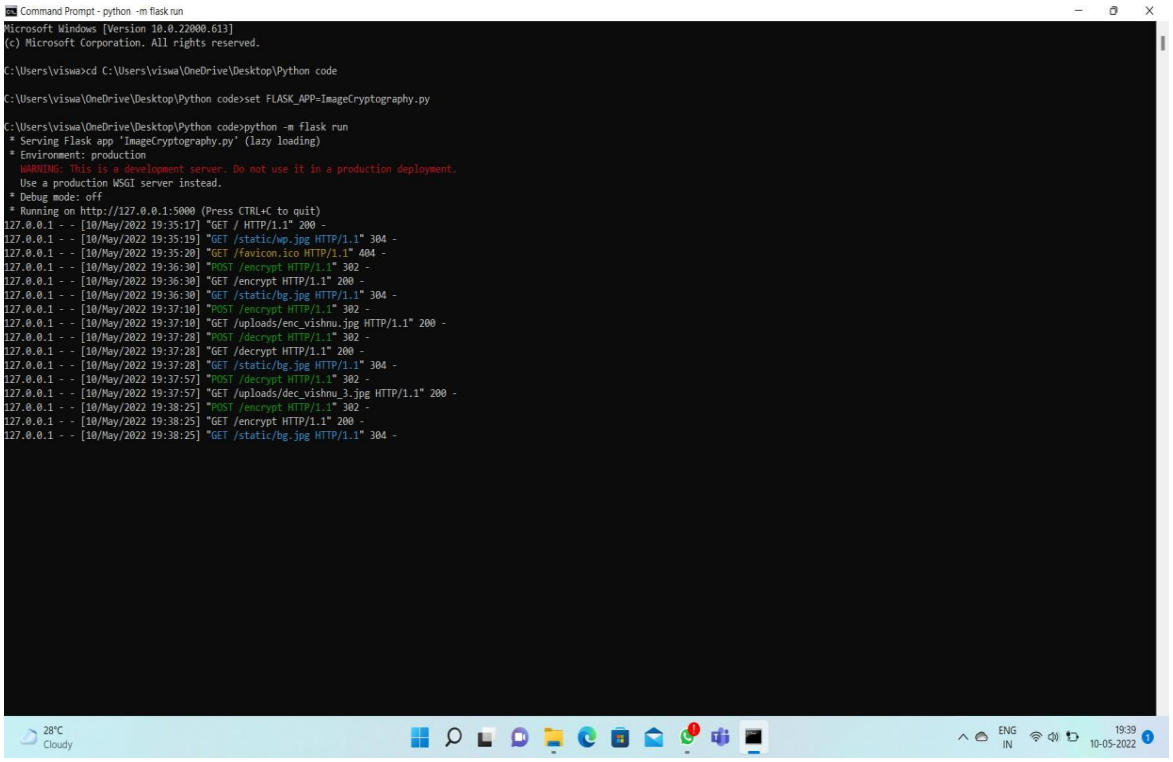
STEP-5: Decrypt webpage



STEP-6: After file choosen and key is which is given at the time of encryption is entered. Press decrypt then decrypted image [Original Image] is downloaded into our system.



Command prompt after encryption and decryption:



6.SYSTEM TESTING

6.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TESTING METHODS

The following are the Testing Methodologies:

1. Unit Testing
2. Integration Testing.
3. User Acceptance Testing.
4. Output Testing.
5. Validation Testing.

Unit Testing Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design

The following are the types of Integration Testing:

1. **Top Down Integration** :This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner. In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. **Bottom-up Integration :**This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

i) The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.

ii) A driver (i.e.) the control program for testing is written to coordinate test case input and output.

iii) The cluster is tested.

iv) Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches test each module individually and then each module is module is integrated with a main module and tested for functionality.

OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking Validation

checks are performed on the following fields.

Text Field

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the

output. The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves. It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program. The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications. The package "Virtual Private Network" has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

6.3 TESTING STRATEGY

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation

UNIT TESTING

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module. In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement of this project.

7. 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 proposed algorithm offers high encryption quality with minimal computational time and the key sensitivity and key space of the algorithm is very high which makes it resistent towards Brute force attack and statistical crypto analysis. And the time taken for encryption is relatively less .

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. Our developed solution is a small contribution that can be very helpful for military or medical fields in future times.

8.BIBLIOGRAPHY

- **AES Key:**

  https://www.geeksforgeeks.org/advanced-encryption-standard-aes/

- **Cryptography:**

  https://en.wikipedia.org/wiki/Cryptography

- **Types of Keys:**

https://www.cryptomathic.com/news-events/blog/classification-of-cryptographic-keys-functions-and

- **Flask:**

  https://www.fullstackpython.com/flask.html

- https://www.codegrepper.com/code-examples/html/how+to+link+html+pages+in+different+folders
- **Algorithms:**

  https://www.ibm.com/docs/en/zos/2.4.0?topic=security-encryption-algorithms

- https://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption
- https://www.sciencedirect.com/topics/engineering/image-processing

9.APPENDIX

9.1Introduction to python

Python is an open source , high-level programming language developed by Guido van Rossum in the late 1980s and presently administered by Python Software Foundation. It came from the ABC language that he helped create early on in his career. Python is a powerful language that you can use to create games, write GUIs and develop web applications. It is a high-level language. Reading and writing codes in Python is much like reading and writing regular English statements. Because they are not written in machinereadable language, Python programs need to be processed before machines can run them. Python is an interpreted language. This means that every time a program is run, its interpreter runs through the code and translates it into machine readable byte code. Python is an object-oriented language that allows users to manage and control data structures or objects to create and run programs. Everything in Python is latest version of Python in fact, first class. All objects, data types, functions, methods, and classes take equal position in Python. Programming languages are created to satisfy the needs of programmers and users for an effective tool to develop applications that impact lives, lifestyles, economy, and society. They help make lives better by increasing productivity, enhancing communication, and improving efficiency. Languages die and become obsolete when they fail to live up to expectations and are replaced and superseded by languages that are more powerful. Python is a programming language that has stood the test of time and has remained relevant across industries and businesses and among programmers, and individual users. It is a living, thriving, and highly useful language that is highly recommended as a first programming language for those who want to dive In to and experience programming. Advantages of Using Python Here are reasons why you would prefer to learn and use Python over other high-level languages

Readability

Python programs use clear, simple, and concise instructions that are easy to read even by those who have no substantial programming background. Programs written in Python are, therefore, easier to maintain, debug, or enhance.

Higher productivity

Codes used in Python are considerably shorter, simpler, and less verbose than other high level programming languages such as Java and C++. In addition, it has well-designed built-in features and standard library as well as access to third party modules and source libraries. These features make programming in Python more efficient.

Less learning time

Python is relatively easy to learn. Many find Python a good first language for learning programming because it uses simple syntax and shorter codes. Python works on Windows, Linux/UNIX, Mac OS X, other operating systems and small form devices. It also runs on microcontrollers used in appliances, toys, remote controls, embedded devices, and other similar devices.

Installing Python in Windows

To install Python, you must first download the installation package of your preferred version from this link: https://www.python.org/downloads/ On this page, you will be asked to choose between the two latest versions for Python 2 and 3: Python 3.5.1 and Python 2.7.11. Alternatively, if you are looking for a specific release, you can scroll down the page to find download links for earlier versions. You would normally opt to download the latest version, which is Python 3.5.1. This was released on December 7, 2015. However, you may opt for the latest version of Python 2, 2.7.11. Your preferences will usually depend on which version will be most usable for your project. While Python 3 is the present and future of the language, issues such as third-party utility or compatibility may require you to download Python 2.

9.2 INTRODUCTION TO FLASK :

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[2] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.