PIXEL BASED ENCODING OF IMAGES

Mini Project submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Thota Himaja Sree (221810305060)
Thalluri Manoj (221810305059)
Y Vishnu Vardhan Reddy (121810303031)
T V L N Samhigna (221810305058)

Under the Guidance of

Dr. S. Phani Kumar

Head of the Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM (Deemed to be University), HYDERABAD

October 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM (Deemed to be University)

DECLARATION

We submitted this mini-project entitled "**Pixel Based Encoding of Images**" to GITAM School of Technology, GITAM Deemed to Be University, Hyderabad campus in partial fulfilment of the requirements for the award of the degree of "Bachelor of Technology" in "Computer Science Engineering".

I declare that it was carried out independently by us under the guidance of Dr. S. Phani Kumar, HOD-GITAM School of Technology, GITAM Deemed to Be University, Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

October 2021

Thota Himaja Sree (221810305060)

Thalluri Manoj (221810305059)

Y Vishnu Vardhan Reddy (121810303031)

T V L N Samhigna (221810305058)

Department of Computer Science & Engineering

GITAM
(Deemed to be University)

CERTIFICATE

This is to certify that the project entitled **“Pixel Based Encoding of Images”** has been submitted by Thota Himaja Sree (221810305060), Thalluri Manoj (221810305059), T. V. L. N. Samhigna (221810305058), and Y. Vishnu Vardhan Reddy (121810303031) to the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering. The mini project has been approved as it satisfies the academic requirements.

Internal Guide **Head of the Department**

External Examiner

ACKNOWLEDGEMENT

Without the assistance of several people, this endeavor would not have been a success. We'd want to express our gratitude to the individuals who contributed to the project in a variety of ways, as well as those who have provided us with exceptional support since its inception.

Prof. N. Siva Prasad, our distinguished Pro-Vice-Chancellor, and Dr. N. Seetharamaiah, Principal, GITAM Hyderabad, have provided invaluable assistance and encouragement.

We are grateful to Prof. S. Phani Kumar, Head of the Department of Computer Science and Engineering, for giving us the opportunity to work on this project and for his encouragement.

We would like to offer our heartfelt gratitude to S. Phani Kumar, HOD, Department of Computer Science and Engineering, for their important assistance, moral support, and suggestions that contributed to the project's success. We're also grateful to all of the Computer Science and Engineering department's employees who helped make this project a success. We'd want to express our gratitude to our parents and friends for their direct and indirect assistance, encouragement, and moral support during this seminar.

Thota Himaja Sree (221810305060)

Thalluri Manoj (221810305059)

Y Vishnu Vardhan Reddy (121810303031)

T V L N Samhigna (221810305058)

TABLE OF CONTENTS

1. INTRODUCTION
   1.1 Image Processing
   1.2 What is Image processing
   1.3 Encryption
   1.4 Decryption
   1.5 Proposed Algorithm

2. SYSTEM ANALYSIS
   2.1 Requirement Analysis
      2.1.1 Hardware Requirements
      2.1.2 Software Requirements
   2.2 Feasibility Analysis
      2.2.1 Economical Feasibility
      2.2.2 Technical Feasibility

3. SYSTEM DESIGN
   3.1 System Description
   3.2 Tools Used

4. IMPLEMENTATIONS
   4.1 Modules
   4.2 Languages

5. TESTING
   5.1 Software Testing
   5.2 Testing Strategies
      5.2.1 Unit Testing
      5.2.2 Black Box Testing
      5.2.3 White Box Testing
      5.2.4 Integration Testing
      5.2.5 System Testing
      5.2.6 Acceptance Testing
   5.3 Test Approach
      5.3.1 Bottom Up Approach
      5.3.2 Top Down Approach
   5.4 Test Cases

6. SYSTEM TESTING RESULTS

7. CONCLUSION & FUTURE SCOPE

8. BIBLIOGRAPHY

ABSTRACT

An Image Encryption algorithm is useful to protect the digital images from cyber-attacks. A pixels-based keys encryption algorithm uses an XOR operator to the rows and columns of the image using two secret keys. This algorithm results in lossless image encryption, and it is highly secured as two keys are associated with the encryption of each and every pixel of the image. The decryption of the image is not possible without using those keys.

The project comprises of four modules. The first module deals with the generation of the keys in a text file. The second module deals with the encryption of the given image. The third module deals with the decryption of the image. The final module deals with the creation of a Graphical User Interface for controlling the operations through the project and generating the automated Python code for that GUI.

This project uses PIL tool for processing the image, and, numpy tool to generate vectors containing the keys. PIL stands for Python Image Processing Library. Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

1. INTRODUCTION

1.1 Image Processing

Images define the world, each image has its own story, it contains a lot of crucial information that can be useful in many ways. This information can be obtained with the help of the technique known as **Image Processing**.

It is the core part of computer vision which plays a crucial role in many real-world examples like robotics, self-driving cars, and object detection. Image processing allows us to transform and manipulate thousands of images at a time and extract useful insights from them. It has a wide range of applications in almost every field.

1.2 What is Image Processing?

As the name says, image processing means processing the image and this may include many different techniques until we reach our goal.

The final output can be either in the form of an image or a corresponding feature of that image. This can be used for further analysis and decision making.

1.3 Encryption

A process of converting Plain Text into Cipher Text is called as Encryption. Cryptography uses the encryption technique to send confidential messages through an insecure channel. The process of encryption requires two things- an encryption algorithm and a key. An encryption algorithm means the technique that has been used in encryption. Encryption takes place at the sender side.

1.4 Decryption

A reverse process of encryption is called as Decryption. It is a process of converting Cipher Text into Plain Text. Cryptography uses the decryption technique at the receiver side to obtain the original message from non-readable message (Cipher Text). The process of decryption requires two things- a Decryption algorithm and a key. A Decryption algorithm means the technique that has been used in Decryption. Generally the encryption and decryption algorithm are same.

1.5 Proposed algorithm

Pixel based keys algorithm:

Image encryption techniques try to convert an image to another one that is hard to understand [2]. On the other hand, image decryption retrieves the original image from the encrypted one. There are various image encryption systems to encrypt and decrypt data, and there is no single encryption algorithm satisfies the different image types. They protect the secret information by converting the secret information to some unintelligible form using a key. By using a key, we protect the secret information by converting the secret information to some incomprehensible form. We get back information through encrypted information should be converted back to original information. On the Basis of key, the encryption algorithm can be classified into two categories.

They are

(i) **Symmetric key encryption**- This algorithm uses same key for both encryption and decryption and

(ii) **Asymmetric key encryption**-This algorithms uses different keys for encryption and decryption.

Asymmetric key algorithm has very higher computational costs than Symmetric key encryption algorithms which have comparatively lower cost. Asymmetric key algorithms are most time prohibitive for multimedia data.

But the characteristic of multimedia data is totally different from text data. All multimedia data has got a lot of redundancy but text data does not possess any redundancy.

The pixel value of a location is highly correlated to values of its neighboring pixels. Like, a sound sample is correlated to its next sample and its previous samples. This correlation proves to be attack points to any standard encryption algorithm. Because they can predict the values of neighboring pixels or next sound sample by finding out pixel value at a location or one sound sample with reasonable accuracy.

Nearly all the available encryption algorithms like .DES, AES, RSA and IDEA are used for text data. Act of them DES, AES, RSA and IDEA can achieve high security, it is not be suitable for images and videos encryption due to the intrinsic characters of images and videos. So we need some other technique for encrypt image and videos. For large data size and high redundancy, encryption special requirements and different encryption algorithms is needed. The image encryption algorithms divided into three major groups: (I) position permutation based algorithm (ii) value transformation based algorithm (iii) visual transformation based algorithm .several encryption algorithms are based on chaotic maps. In this paper, we propose image encryption using Random Scrambling and XOR operation. Affine transform that is based on shuffling the image pixels and they encrypting the resulting image using XOR operation. We used 32 bit key that is good for practical purposes.

2. SYSTEM ANALYSIS

2.1 Requirement Analysis

2.1.1 Hardware Requirements

1. It requires a minimum of 2.16 GHz processor.
2. It requires a minimum of 4 GB RAM.
3. It requires 64-bit architecture.
4. It requires a minimum storage of 500GB.

2.1.2 Software Requirements

1. It requires a 64-bit Windows Operating System.
2. Python Qt Designer for designing user interface.
3. PIL for image processing.
4. Pyuic for converting the layout designed user interface (UI) to python code.
5. Python language for coding.

2.2 Feasibility Analysis

As the name implies, a feasibility study is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

2.2.1 Economical Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision makers determine the positive economic benefits to the organization that the proposed project will provide. Our project is economically feasible because in this we have used "UBUNTU", "PYTHON", "PYQT" designer tool and "PYUIC" which are all available as an open source.

2.2.2 Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity. Technical feasibility also involves evaluation of the hardware, software, and other technology requirements of the proposed system. A prototype of the tool was developed to verify the technical feasibility. The prototype is working successfully and hence the project is feasible.

3. SYSTEM DESIGN

3.1 System Description

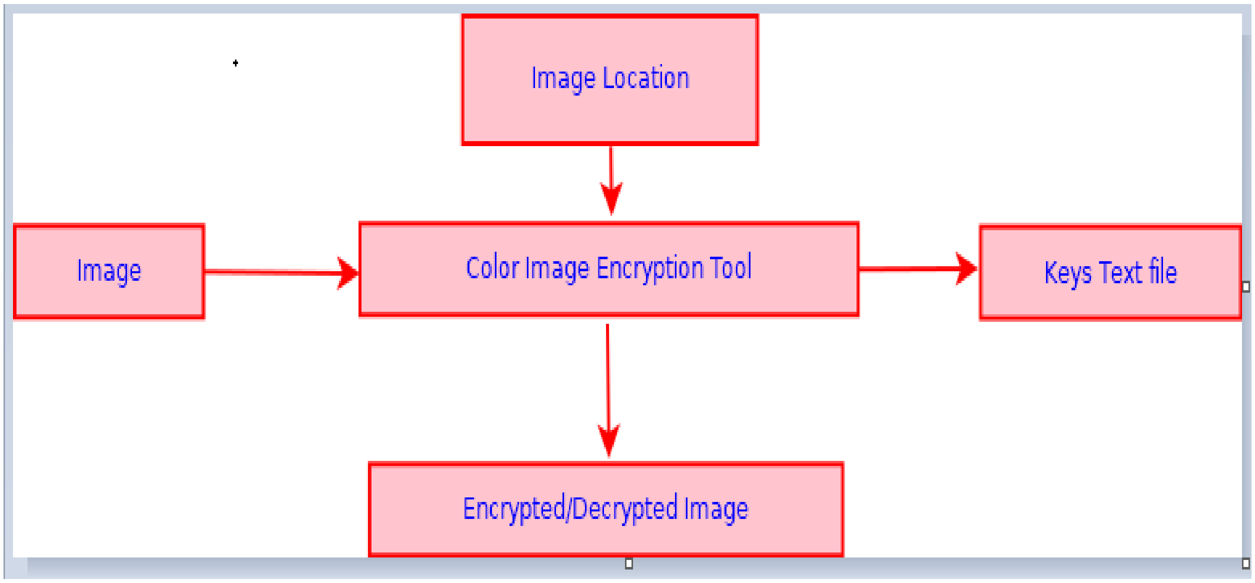
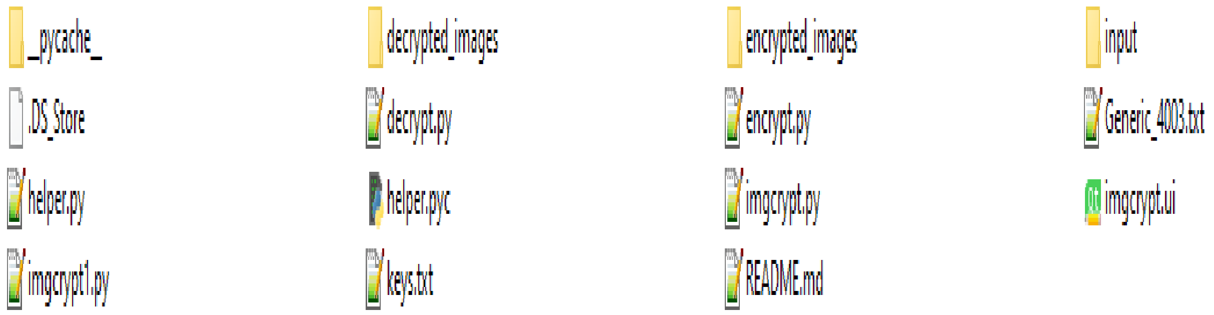


Fig: Data Flow Diagram

Details like Image Location & Image to be encrypted are to be provided as Input to the system, using the corresponding user interfaces. The following section describe the screen shots of these user interfaces.



Pic:1

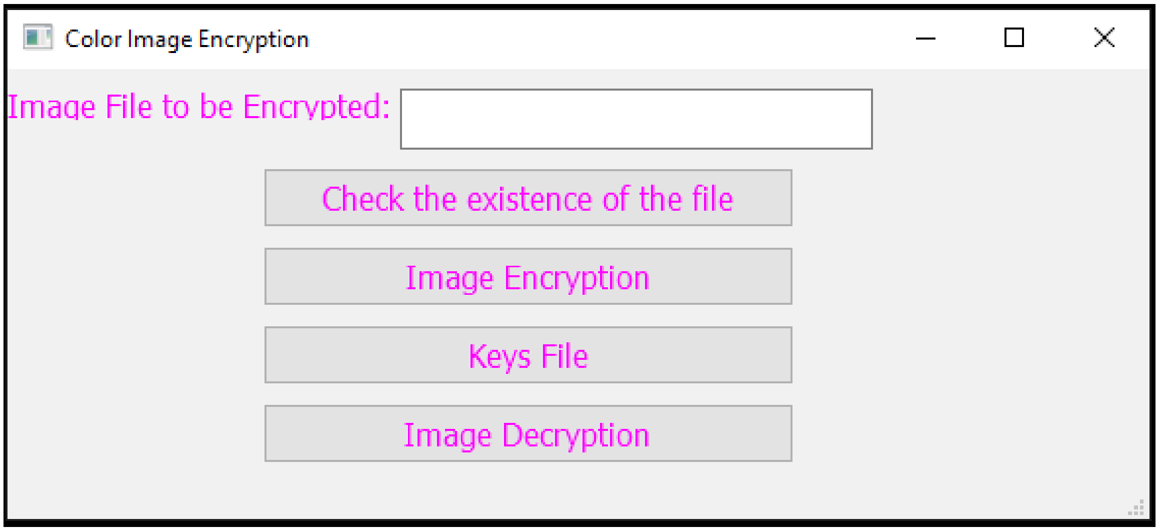
Pic1: Screen shot showing the files created during the project. The above screen shot shows different files created during this projects. There are three different types of files: (1) .py files (2) .ui files and (3).txt files

.ui files are the user interface files, created by using PyQt layout editor

.py files are python program files created either manually, or automatically. For instance, each .ui file has a corresponding .py file that is created automatically by using the PyUIC tool. .txt files contains the generic useful information about the project.

imgcrypt1.py, is the entry program for this project.

Execution of this python program leads to the entry screen as follows:



This screen consists of aline edit and four push buttons. The image file name, to be encrypted, is to be provided as the input in the line edit.

The first push button checks, whether the given image file is existing in the input directory, or not. The second button generate the encrypted image, the third button generates the keys, in a text file. The fourth button generates the decrypted image.

PYTHON: Python is a widely used high level programming language for general purpose programming, created by Guido Van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the operating system Amoeba. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL). About the origin of Python, Van Rossum wrote in 1996.

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release the development process was changed and became more transparent and community-backed. Python 3.0 (initially described as Python 3000 or py3k), is a major, backward-incompatible release that was released after a long period of testing on 3 December 2008.

Many of its major features have been back ported to the backwards-compatible Python 2.6.x and 2.7.x version series.

The End of Life date (EOL, sunset date) for Python 2.7 was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code cannot easily be forward ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go trans compiler, which The Register speculated was in response to Python 2.7's planned end-of life but Google cited performance under concurrent workloads as their only motivation.

3.2 Tools Used

Qt Designer:

Qt Designer is the Qt tool for designing and building graphical user interfaces (GUIs) with Qt Widgets. You can compose and customize your windows or dialogs in a what-you-see-is-what-you-get (WYSIWYG) manner, and test them using different styles and resolutions. Widgets and forms created with Qt Designer integrate seamlessly with programmed code, using Qt's signals and slots mechanism, so that you can easily assign behavior to graphical elements. All properties set in Qt Designer can be changed dynamically within the code. Furthermore, features like widget promotion and custom plugins allow you to use your own components with Qt Designer.

PIL/Pillow Tool:

PIL stands for Python Image Library and **Pillow** is the friendly PIL fork by Alex Clark and Contributors. It’s one of the powerful libraries. It supports a wide range of image formats like PPM, JPEG, TIFF, GIF, PNG, and BMP.

It can help you perform several operations on images like rotating, resizing, cropping, gray scaling etc. Let’s go through some of those operations

To carry out manipulation operations there is a module in this library called **Image.**

- To load an image use the **open ()** method.

- To display an image use **show ()** method.
- To know the file format use **format** attribute
- To know the size of the image use **size** attribute
- To know about the pixel format use **mode** attribute.
- To save the image file after desired processing, use **save ()** method. Pillow saves the image file in *png* format.
- To resize the image use **resize ()** method that takes two arguments as width and height.
- To crop the image, use **crop ()** method that takes one argument as a box tuple that defines position and size of the cropped region.
- To rotate the image use **rotate ()** method that takes one argument as an integer or float number representing the degree of rotation.
- To flip the image use **transform ()** method that take one argument among the following: Image.FLIP_LEFT_RIGHT, Image.FLIP_TOP_BOTTOM, Image.ROTATE_90, Image.ROTATE_180, and Image.ROTATE_270.

UML DIAGRAMS

a) Use case diagram:

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

As we can see the user is interacting with system by a UI through which the customer can perform above mentioned operations like proving the image and its location as inputs, generating the keys text file and encrypting/decrypting using the keys.

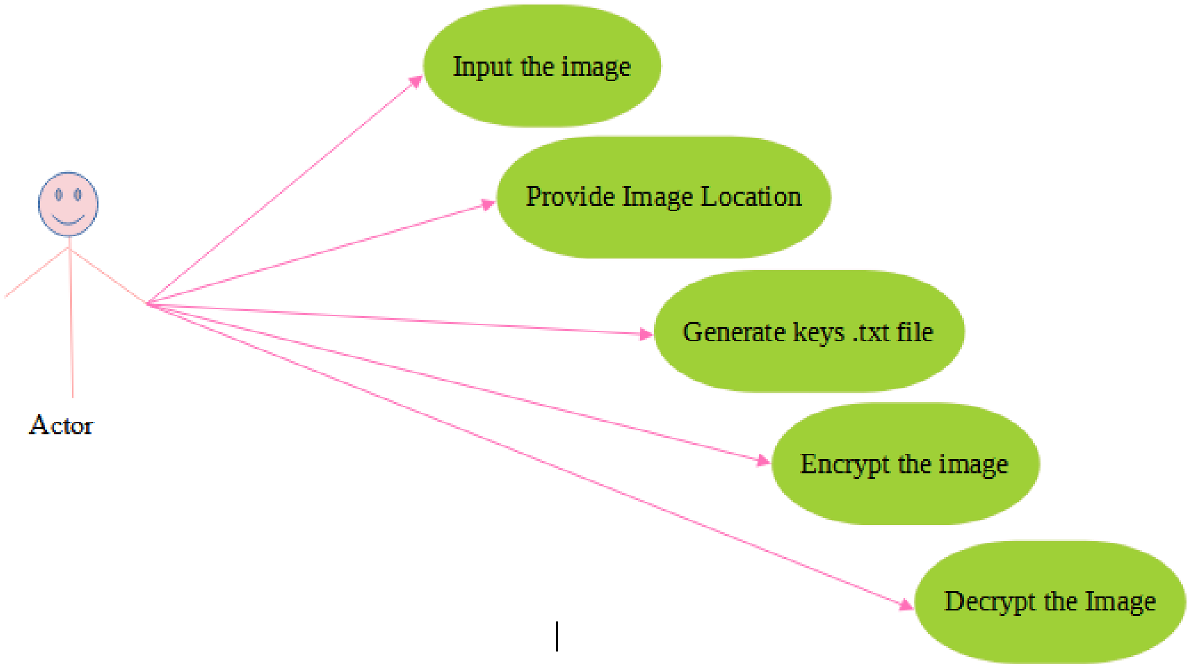


Fig: USE CASE DIAGRAM

b) <u>Sequence Diagram</u>:

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

From below mentioned sequence diagram we have to go in sequence: Provide the image to be encrypted, and it's location as Inputs, Generate the keys.txt file and perform encryption/decryption, as needed.

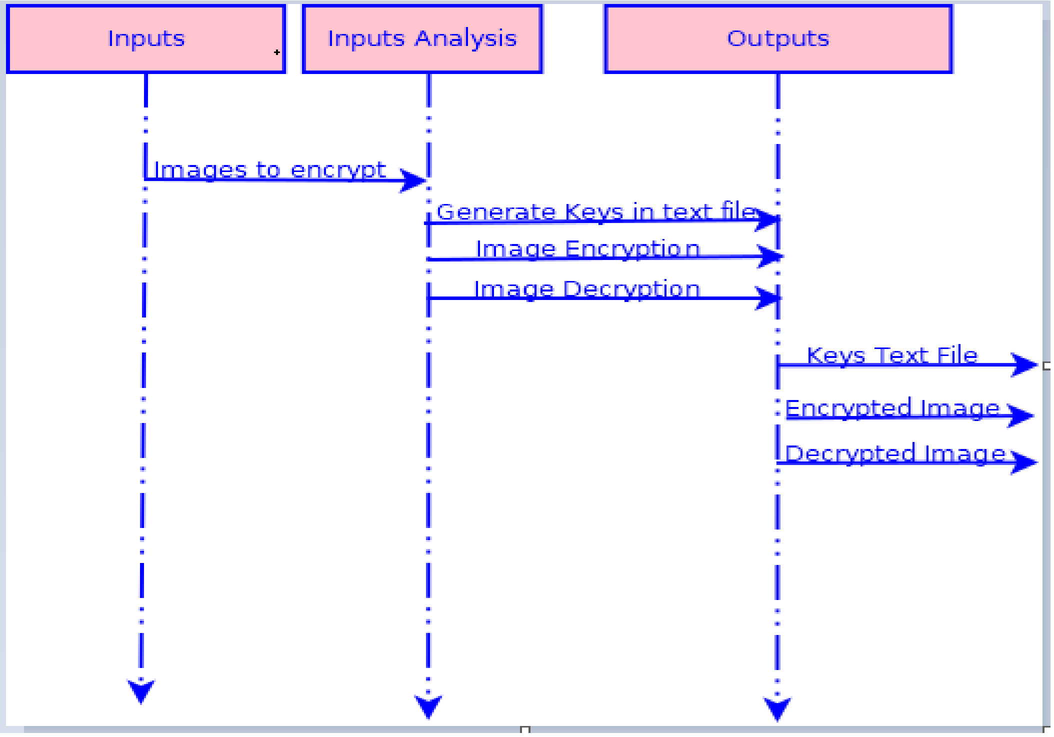
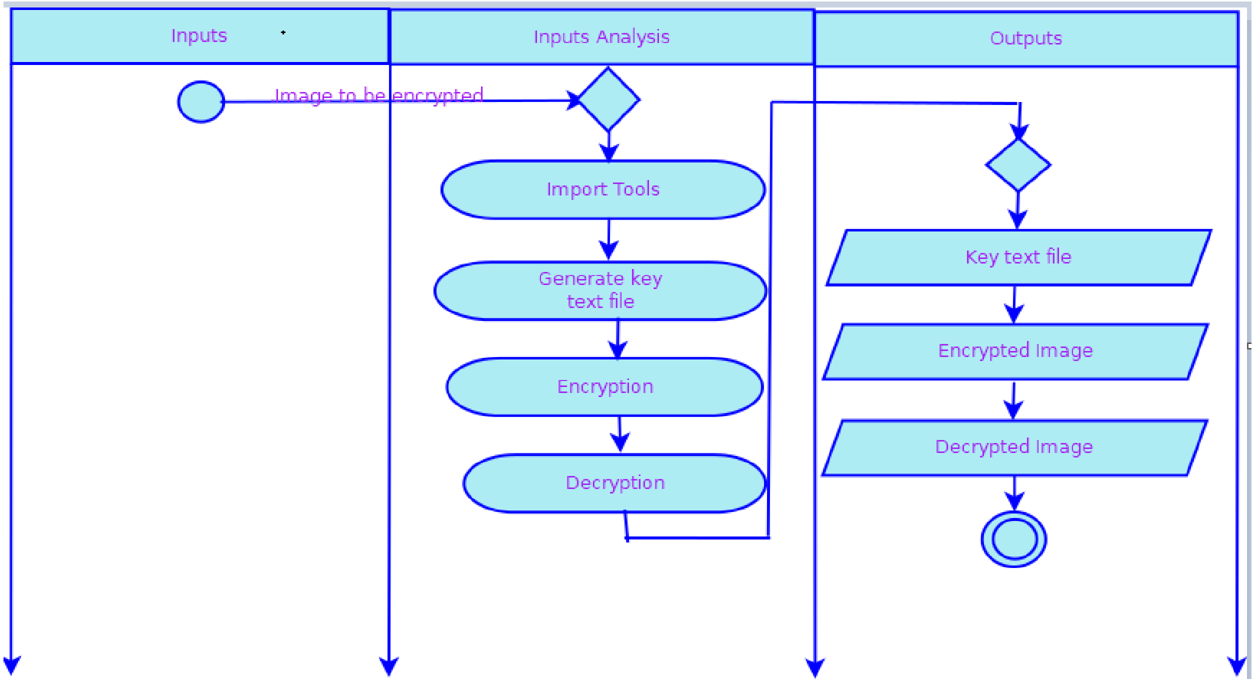


Fig: SEQUENCE DIAGRAM

c) ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. In activity diagram we can see that first we have to provide the image in the inputs folder, input its location to the system followed by the generation of Keys.txt file along with the encryption/decryption, as needed.



4. IMPLEMENTATION

4.1 Modules Imported in Python routines:

OS Module:

The OS module in Python provides a way of using operating system dependent functionality.

The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux.

OS functions:

Executing a shell command
os.system()

Get the users environment
os.environ ()

Returns the current working directory.
os.getcwd ()

Return the real group id of the current process.
os.getgid ()

Return the current process's user id.
os.getuid ()

Returns the real process ID of the current process.
os.getpid()

Set the current numeric umask and return the previous umask.
os.umask(mask)

Return information identifying the current operating system.
os.uname()

Change the root directory of the current process to path.

os.chroot(path)

Return a list of the entries in the directory given by path.
os.listdir(path)

Create a directory named path with numeric mode mode.
os.mkdir(path)

Recursive directory creation function.
os.makedirs(path)

Remove (delete) the file path.
os.remove(path)

Remove directories recursively.
os.removedirs(path)

Rename the file or directory src to dst.
os.rename(src, dst)

Remove (delete) the directory path.
os.rmdir(path)

Sys Module:

The sys module provides information about constants, functions and methods of the Python interpreter. dir(system) gives a summary of the available constants, functions and methods. Another possibility is the help () function. Using help(sys) provides valuable detail information.

The module sys informs e.g. about the maximal recursion depth (sys.getrecursionlimit ()) and provides the possibility to change (sys.setrecursionlimit ()).

The current version number of Python can be accessed as well by using this module.

Lots of scripts need access to the arguments passed to the script, when the script was started. argvargv (or to be precise sys.argv) is a list, which contains the command-line arguments passed to the script. The first item of this list contains the name of the script itself. The arguments follow the script name.

Every serious user of a UNIX or Linux operating system knows standard streams, i.e. input, standard output and standard error. They are known as pipes. They are commonly abbreviated as stdin, stdout, stderr.

The standard input (stdin) is normally connected to the keyboard, while the standard error and standard output go to the terminal (or window) in which you are working.

These data streams can be accessed from Python via the objects of the sys module with the same names, i.e. sys.stdin, sys.stdout and sys.stderr.

The standard output (stdout) can be redirected e.g. into a file, so that we can process this file later with another program. The same is possible with the standard error stream, we can redirect it into a file as well. We can redirect both stderr and stdout into the same file or into separate files.

Numpy:

NumPy's main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called axes.

NumPy is module for Python. The name is an acronym for "Numeric Python" or "Numerical Python". It is an extension module for Python, mostly written in C. This makes sure that the precompiled mathematical and numerical functions and functionalities of Numpy guarantee great execution speed.

Furthermore, NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays. The implementation is even aiming at huge matrices and arrays, better known under the heading of "big data". Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

SciPy (Scientific Python) is often mentioned in the same breath with NumPy. SciPy needs Numpy, as it is based on the data structures of Numpy and furthermore its basic creation and manipulation functions.

It extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier-transformation and many others.

Both NumPy and SciPy are not part of a basic Python installation. They have to be installed after the Python installation. NumPy has to be installed before installing SciPy.

NumPy is based on two earlier Python modules dealing with arrays. One of these is Numeric. Numeric is like NumPy a Python module for high-performance, numeric computing, but it is obsolete nowadays. Another predecessor of NumPy is Numarray, which is a complete rewrite of Numeric but is deprecated as well. NumPy is a merger of those two, i.e. it is built on the code of Numeric and the features of Numarray.NumPy's array class is called ndarray. It is also known by the alias array. Note that numpy.array is not the same as the Standard Python Library class array.array, which only handles one-dimensional arrays and offers less functionality. The more important attributes of an ndarray object are:

ndarray.ndim

The number of axes (dimensions) of the array.

ndarray.shape

The dimensions of the array. This is a tuple of integers indicating the size of the array in each dimension. For a matrix with n rows and m columns, shape will be (n,m). The length of the shape tuple is therefore the number of axes, ndim.

ndarray.size

The total number of elements of the array. This is equal to the product of the elements of shape.

ndarray.dtype

An object describing the type of the elements in the array. One can create or specify dtype's using standard Python types. Additionally NumPy provides types of its own. numpy.int32, numpy.int16, and numpy.float64 are some examples.

ndarray.itemsize

The size in bytes of each element of the array. For example, an array of elements of type float64 has itemsize 8 (=64/8), while one of type complex32 has itemsize 4 (=32/8). It is equivalent to ndarray.dtype.itemsize.

ndarray.data

The buffer containing the actual elements of the array. Normally, we won’t need to use this attribute because we will access the elements in an array using indexing facilities.

4.2 Languages

PYTHON

Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations.

FEATURES AND PHILOSOPHY

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python Code

```
import sys
import os
from imgcrypt import *
from PyQt5 import QtWidgets, QtGui, QtCore

class MyForm(QtWidgets.QMainWindow):
  def __init__(self,parent=None):
     QtWidgets.QWidget.__init__(self,parent)
     self.ui = Ui_MainWindow()
     self.ui.setupUi(self)
     self.ui.pushButton.clicked.connect(self.flexists)
     self.ui.pushButton_2.clicked.connect(self.encr)
     self.ui.pushButton_3.clicked.connect(self.kys)
     self.ui.pushButton_4.clicked.connect(self.decr)


  def flexists(self):
    fname = str(self.ui.lineEdit.text())
    fname = './input/'+fname
    if (os.path.isfile(fname)):
      print(' file exists')
    else:
      print(' file does not exists')

  def encr(self):
    fname = str(self.ui.lineEdit.text())
    str1 = "python"+" encrypt.py"+" "+fname
    os.system(str1)

  def kys(self):
    print("keys.txt file successfully generated")

  def decr(self):
    fname = str(self.ui.lineEdit.text())
    str1 = "python"+" decrypt.py"+" "+fname
    os.system(str1)



if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_()
```

5. TESTING

5.1 Software Testing

Testing is a procedure, which uncovers blunders in the program. Programming testing is a basic component of programming quality affirmation and speaks to a definitive audit of determination, outline and coding. The expanding perceivability of programming as a framework component and chaperon costs related with a product disappointment are propelling variables for we arranged, through testing. Testing is the way toward executing a program with the plan of finding a mistake. The plan of tests for programming and other built items can be as trying as the underlying outline of the item itself it is the significant quality measure utilized amid programming improvement. Amid testing, the program is executed with an arrangement of experiments and the yield of the program for the experiments is assessed to decide whether the program is executing as it is relied upon to perform.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

5.2 TESTING STRATEGIES

A technique for programming testing coordinates the outline of programming experiments into an all-around arranged arrangement of steps that outcome in fruitful improvement of the product. The procedure gives a guide that portrays the means to be taken, when, and how much exertion, time, and assets will be required.

The procedure joins test arranging, experiment configuration, test execution, and test outcome gathering and assessment. The procedure gives direction to the specialist and an arrangement of points of reference for the chief. Due to time weights, advance must be quantifiable and issues must surface as ahead of schedule as would be prudent Keeping in mind the end goal to ensure that the framework does not have blunders, the distinctive levels of testing techniques that are connected at varying periods of programming improvement are:

5.2.1 Unit Testing

It is done on singular modules as they are finished and turned out to be executable. It is restricted just to the planner's prerequisites. It centers testing around the capacity or programming module. It concentrates on the interior preparing rationale and information structures. It is rearranged when a module is composed with high union 36 • Reduces the quantity of experiments • Allows mistakes to be all the more effectively anticipated and revealed.

5.2.2 Black Box Testing

It is otherwise called Functional testing. A product testing strategy whereby the inward workings of the thing being tried are not known by the analyzer. For instance, in a discovery test on a product outline the analyzer just knows the information sources and what the normal results ought to be and not how the program touches base at those yields. The analyzer does not ever inspect the programming code and does not require any further learning of the program other than its determinations. In this system some experiments are produced as information conditions that completely execute every single practical prerequisite for the program. This testing has been utilizations to discover mistakes in the accompanying classifications:

- Incorrect or missing capacities
- Interface blunders
- Errors in information structure or outside database get to
- Performance blunders
- Initialization and end blunders.

5.2.3 White Box testing

It is otherwise called Glass box, Structural, Clear box and Open box testing. A product testing procedure whereby express learning of the inner workings of the thing being tried are utilized to choose the test information. Not at all like discovery has testing, white box testing utilized particular learning of programming code to inspect yields. The test is precise just if the analyzer comprehends what the program should do. He or she would then be able to check whether the program veers from its expected objective. White box testing does not represent blunders caused by oversight, and all obvious code should likewise be discernable. For an entire programming examination, both white box and discovery tests are required.

In this the experiments are produced on the rationale of every module by drawing stream diagrams of that module and sensible choices are tried on every one of the cases. It has been utilizations to produce the experiments in the accompanying cases:

- Guarantee that every single freeway have been Executed.
- Execute every single intelligent choice on their actual and false Sides.

5.2.4 Integration Testing

Coordination testing guarantees that product and subsystems cooperate an entirety.

It tests the interface of the considerable number of modules to ensure that the modules carry on legitimately when coordinated together. It is characterized as a deliberate procedure for developing the product engineering. In the meantime reconciliation is happening, lead tests to reveal blunders related with interfaces. Its Objective is to take unit tried modules and assemble a program structure in view of the recommended outline.

Two Approaches of Integration Testing

- Non-incremental Integration Testing
- Incremental Integration Testing

5.2.5 System Testing

Framework testing includes in-house testing of the whole framework before conveyance to the client. Its point is to fulfill the client the framework meets all necessities of the customer's determinations. This testing assesses working of framework from client perspective, with the assistance of particular report. It doesn't require any inward learning of framework like plan or structure of code. It contains utilitarian and non-useful zones of utilization/item. Framework Testing is known as a super arrangement of a wide range of testing as all the significant sorts of testing are shrouded in it. In spite of the fact that attention on sorts of testing may differ on the premise of item, association procedures, course of events and necessities. Framework Testing is the start of genuine testing where you test an item all in all and not a module/highlight.

5.2.6 Acceptance Testing

Acknowledgment testing, a testing method performed to decide if the product framework has met the prerequisite particulars.

The principle motivation behind this test is to assess the framework's consistence with the business necessities and check in the event that it is has met the required criteria for conveyance to end clients. It is a pre-conveyance testing in which whole framework is tried at customer's site on genuine information to discover blunders. The acknowledgment test bodies of evidence are executed against the test information or utilizing an acknowledgment test content and afterward the outcomes are contrasted and the normal ones. The acknowledgment test exercises are completed in stages. Right off the bat, the essential tests are executed, and if the test outcomes are palatable then the execution of more intricate situations are done.

5.3 TEST APPROACH

A Test approach is the test system usage of a venture, characterizes how testing would be done. The decision of test methodologies or test technique is a standout amongst the most intense factor in the achievement of the test exertion and the precision of the test designs and gauges.

Testing should be possible in two ways:

- Bottom up approach
- Top down approach

5.3.1 Bottom up Approach

Testing can be performed beginning from littlest and most reduced level modules and continuing each one in turn. In this approach testing is directed from sub module to primary module, if the fundamental module is not built up a transitory program called DRIVERS is utilized to recreate the principle module. At the point when base level modules are tried consideration swings to those on the following level that utilization the lower level ones they are tried exclusively and afterward connected with the already inspected bring down level modules.

5.3.2 Top down Approach

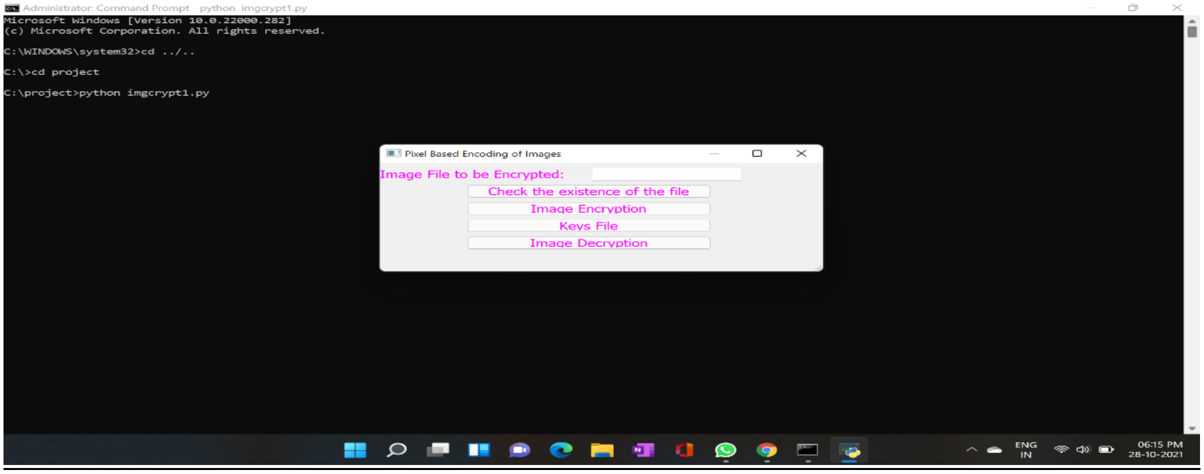
In this approach testing is directed from fundamental module to sub module. In the event that the sub module is not built up an impermanent program called STUB is utilized for mimic the sub module. This sort of testing begins from upper level modules. Since the nitty gritty exercises more often than not performed in the lower level schedules are not given stubs are composed. A stub is a module shell called by upper level module and that when achieved legitimately will restore a message to the calling module demonstrating that appropriate association happened.

5.4 Test Cases

Experiments include an arrangement of steps, conditions and sources of info that can be utilized while performing testing undertakings. The principle expectation of this action is to guarantee whether a product passes or bombs as far as usefulness and different perspectives. The way toward creating experiments can likewise help discover issues in the prerequisites or plan of an application. Experiment goes about as the beginning stage for the test execution, and in the wake of applying an arrangement of information esteems, the application has a conclusive result and leaves the framework at some end point or otherwise called execution post condition.

6. System Testing Results

Entry Screen



Administrator: Command Prompt - python imgcrypt1.py
Pixel Based Encoding of Images
Image File to be Encrypted:
Check the existence of the file
Image Encryption
Keys File
Image Decryption
ENG
IN
06:15 PM
28-10-2021

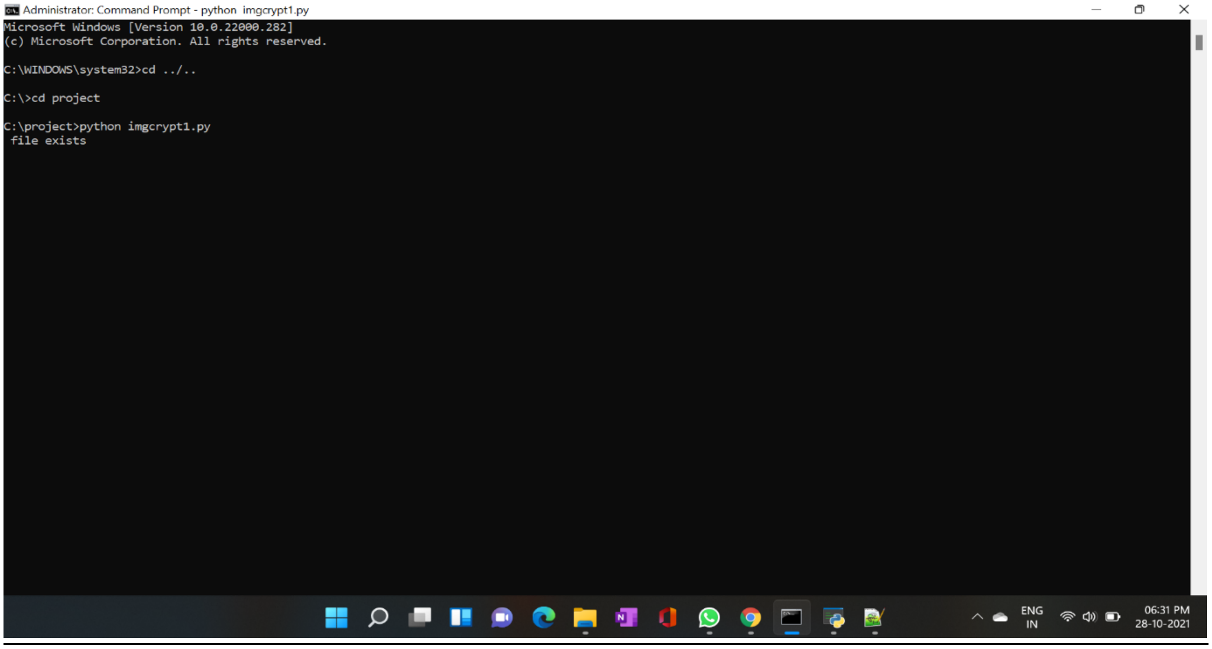
```
Microsoft Windows [Version 10.0.22000.282]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ../..

C:\>cd project

C:\project>python imgcrypt1.py
```

It will check the existence of the file



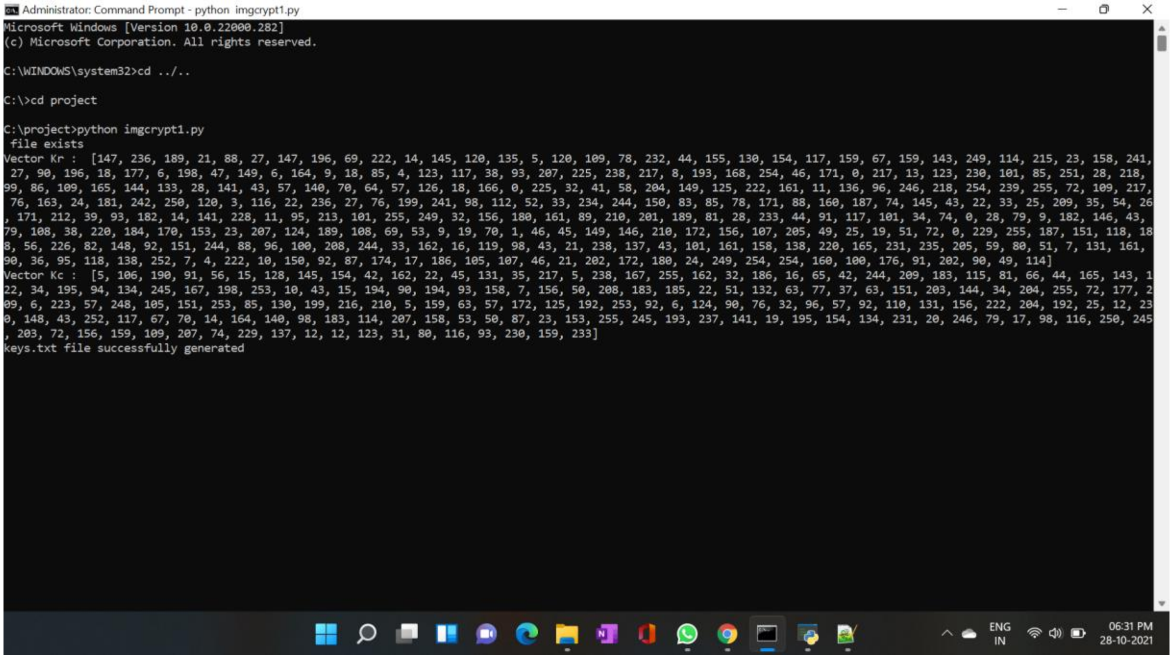
```
Microsoft Windows [Version 10.0.22000.282]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ../..

C:\>cd project

C:\project>python imgcrypt1.py
 file exists
```

Image Encryption is done with vector keys



Administrator: Command Prompt - python imgcrypt1.py

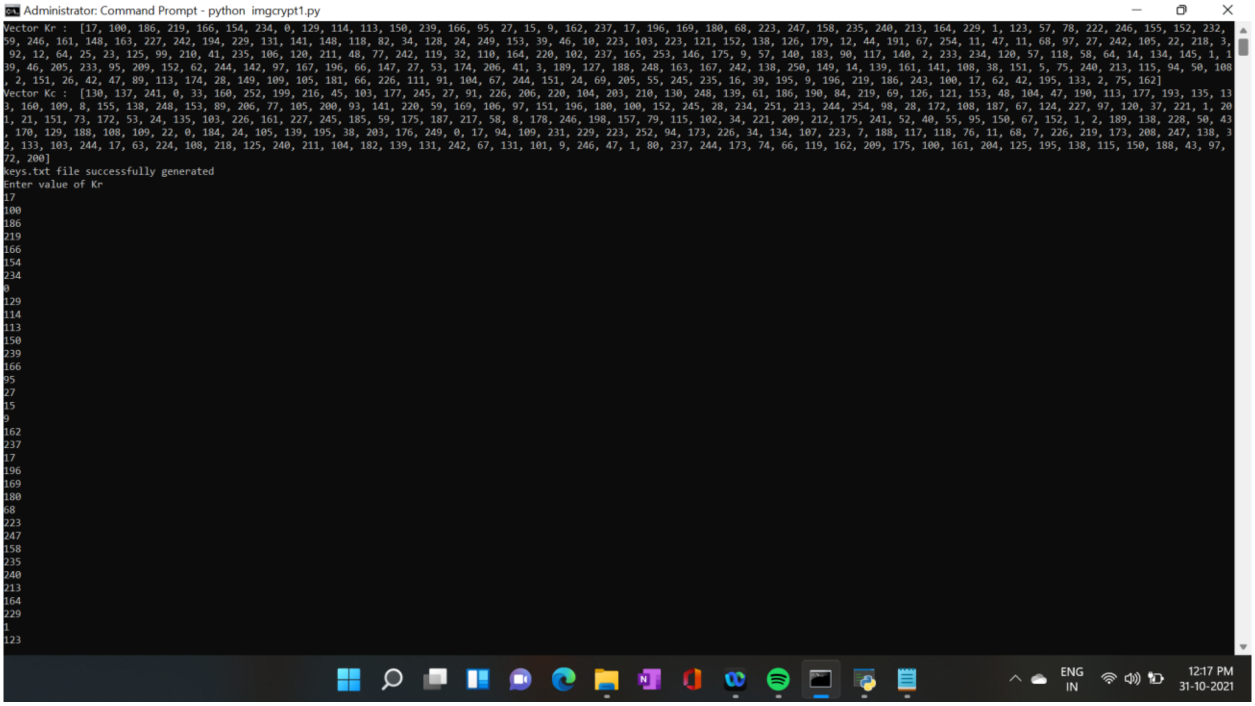
```
Microsoft Windows [Version 10.0.22000.282]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ../..

C:\>cd project

C:\project>python imgcrypt1.py
 file exists
Vector Kr :  [147, 236, 189, 21, 88, 27, 147, 196, 69, 222, 14, 145, 120, 135, 5, 120, 109, 78, 232, 44, 155, 130, 154, 117, 159, 67, 159, 143, 249, 114, 215, 23, 158, 241,
 27, 90, 196, 18, 177, 6, 198, 47, 149, 6, 164, 9, 18, 85, 4, 123, 117, 38, 93, 207, 225, 238, 217, 8, 193, 168, 254, 46, 171, 0, 217, 13, 123, 230, 101, 85, 251, 28, 218,
99, 86, 109, 165, 144, 133, 28, 141, 43, 57, 140, 70, 64, 57, 126, 18, 166, 0, 225, 32, 41, 58, 204, 149, 125, 222, 161, 11, 136, 96, 246, 218, 254, 239, 255, 72, 109, 217,
 76, 163, 24, 181, 242, 250, 120, 3, 116, 22, 236, 27, 76, 199, 241, 98, 112, 52, 33, 234, 244, 150, 83, 85, 78, 171, 88, 160, 187, 74, 145, 43, 22, 33, 25, 209, 35, 54, 26
, 171, 212, 39, 93, 182, 14, 141, 228, 11, 95, 213, 101, 255, 249, 32, 156, 180, 161, 89, 210, 201, 189, 81, 28, 233, 44, 91, 117, 101, 34, 74, 0, 28, 79, 9, 182, 146, 43,
79, 108, 38, 220, 184, 170, 153, 23, 207, 124, 189, 108, 69, 53, 9, 19, 70, 1, 46, 45, 149, 146, 210, 172, 156, 107, 205, 49, 25, 19, 51, 72, 0, 229, 255, 187, 151, 118, 18
8, 56, 226, 82, 148, 92, 151, 244, 88, 96, 100, 208, 244, 33, 162, 16, 119, 98, 43, 21, 238, 137, 43, 101, 161, 158, 138, 220, 165, 231, 235, 205, 59, 80, 51, 7, 131, 161,
90, 36, 95, 118, 138, 252, 7, 4, 222, 10, 150, 92, 87, 174, 17, 186, 105, 107, 46, 21, 202, 172, 180, 24, 249, 254, 254, 160, 100, 176, 91, 202, 90, 49, 114]
Vector Kc :  [5, 106, 190, 91, 56, 15, 128, 145, 154, 42, 162, 22, 45, 131, 35, 217, 5, 238, 167, 255, 162, 32, 186, 16, 65, 42, 244, 209, 183, 115, 81, 66, 44, 165, 143, 1
22, 34, 195, 94, 134, 245, 167, 198, 253, 10, 43, 15, 194, 90, 194, 93, 158, 7, 156, 50, 208, 183, 185, 22, 51, 132, 63, 77, 37, 63, 151, 203, 144, 34, 204, 255, 72, 177, 2
09, 6, 223, 57, 248, 105, 151, 253, 85, 130, 199, 216, 210, 5, 159, 63, 57, 172, 125, 192, 253, 92, 6, 124, 90, 76, 32, 96, 57, 92, 110, 131, 156, 222, 204, 192, 25, 12, 23
0, 148, 43, 252, 117, 67, 70, 14, 164, 140, 98, 183, 114, 207, 158, 53, 50, 87, 23, 153, 255, 245, 193, 237, 141, 19, 195, 154, 134, 231, 20, 246, 79, 17, 98, 116, 250, 245
, 203, 72, 156, 159, 109, 207, 74, 229, 137, 12, 12, 123, 31, 80, 116, 93, 230, 159, 233]
keys.txt file successfully generated
```

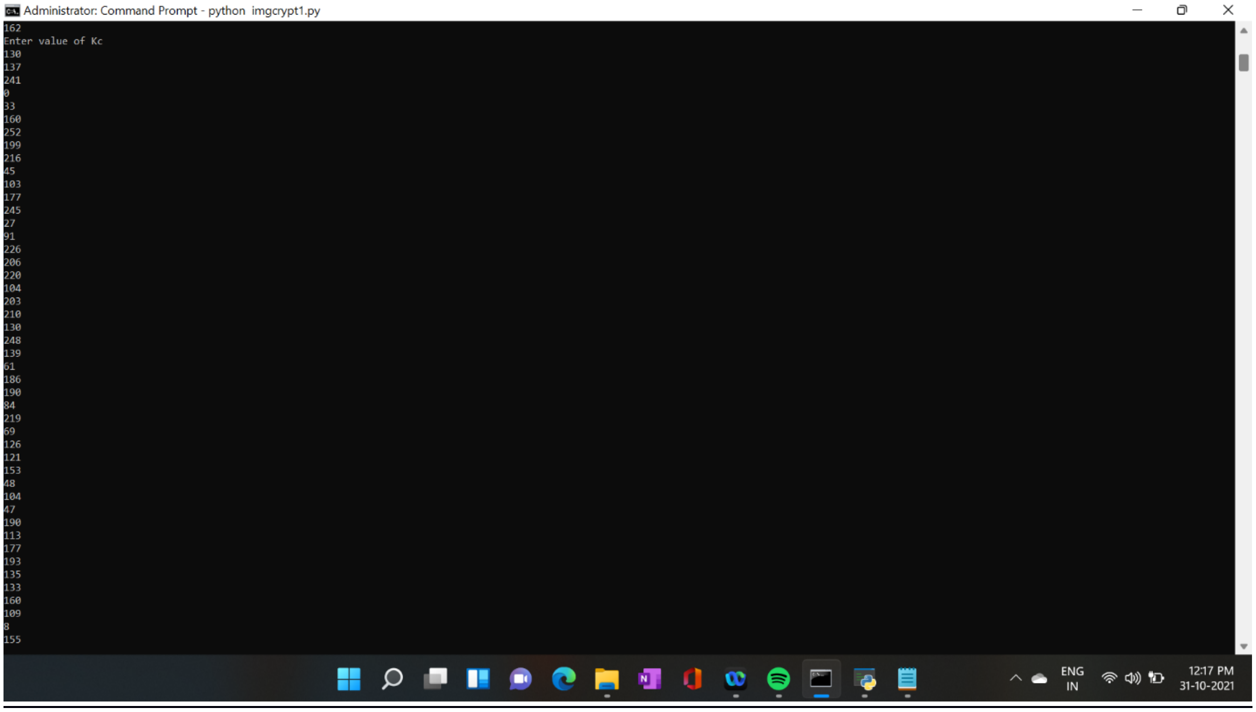
Keys Files will be generated



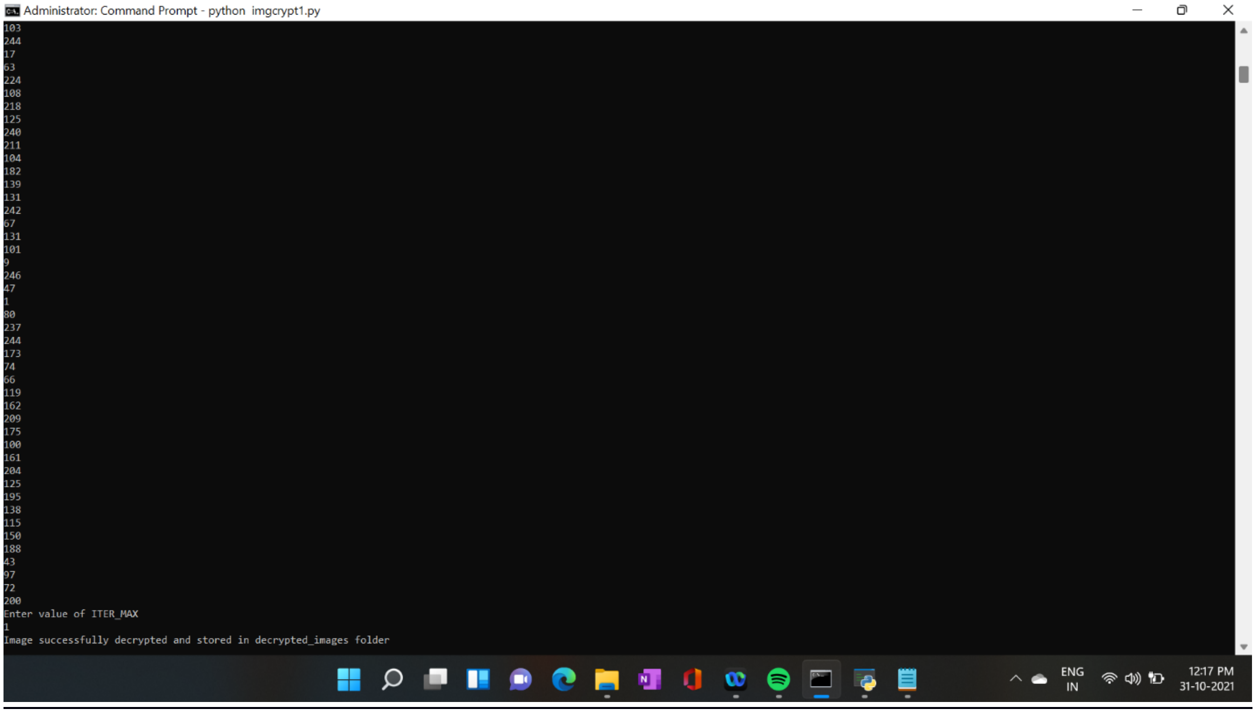
Administrator: Command Prompt - python imgcrypt1.py

```
Vector Kr :  [17, 100, 186, 219, 166, 154, 234, 0, 129, 114, 113, 150, 239, 166, 95, 27, 15, 9, 162, 237, 17, 196, 169, 180, 68, 223, 247, 158, 235, 240, 213, 164, 229, 1, 123, 57, 78, 222, 246, 155, 152, 232,
59, 246, 161, 148, 163, 227, 242, 194, 229, 131, 141, 148, 118, 82, 34, 128, 24, 249, 153, 39, 46, 10, 223, 103, 223, 121, 152, 138, 126, 179, 12, 44, 191, 67, 254, 11, 47, 11, 68, 97, 27, 242, 105, 22, 218, 3,
 92, 12, 64, 25, 23, 125, 99, 210, 41, 235, 106, 120, 211, 48, 77, 242, 119, 32, 110, 164, 220, 102, 237, 165, 253, 146, 175, 9, 57, 140, 183, 90, 117, 140, 2, 233, 234, 120, 57, 118, 58, 64, 14, 134, 145, 1, 1
39, 46, 205, 233, 95, 209, 152, 62, 244, 142, 97, 167, 196, 66, 147, 27, 53, 174, 206, 41, 3, 189, 127, 188, 248, 163, 167, 242, 138, 250, 149, 14, 139, 161, 141, 108, 38, 151, 5, 75, 240, 213, 115, 94, 50, 108
, 2, 151, 26, 42, 47, 89, 113, 174, 28, 149, 109, 105, 181, 66, 226, 111, 91, 104, 67, 244, 151, 24, 69, 205, 55, 245, 235, 16, 39, 195, 9, 196, 219, 186, 243, 100, 17, 62, 42, 195, 133, 2, 75, 162]
Vector Kc :  [130, 137, 241, 0, 33, 160, 252, 199, 216, 45, 103, 177, 245, 27, 91, 226, 206, 220, 104, 203, 210, 130, 248, 139, 61, 186, 190, 84, 219, 69, 126, 121, 153, 48, 104, 47, 190, 113, 177, 193, 135, 13
3, 160, 109, 8, 155, 138, 248, 153, 89, 206, 77, 105, 200, 93, 141, 220, 59, 169, 106, 97, 151, 196, 180, 100, 152, 245, 28, 234, 251, 213, 244, 254, 98, 28, 172, 108, 187, 67, 124, 227, 97, 120, 37, 221, 1, 20
1, 21, 151, 73, 172, 53, 24, 135, 103, 226, 161, 227, 245, 185, 59, 175, 187, 217, 58, 8, 178, 246, 198, 157, 79, 115, 102, 34, 221, 209, 212, 175, 241, 52, 40, 55, 95, 150, 67, 152, 1, 2, 189, 138, 228, 50, 43
, 170, 129, 188, 108, 109, 22, 0, 184, 24, 105, 139, 195, 38, 203, 176, 249, 0, 17, 94, 109, 231, 229, 223, 252, 94, 173, 226, 34, 134, 107, 223, 7, 188, 117, 118, 76, 11, 68, 7, 226, 219, 173, 208, 247, 138, 3
2, 133, 103, 244, 17, 63, 224, 108, 218, 125, 240, 211, 104, 182, 139, 131, 242, 67, 131, 101, 9, 246, 47, 1, 80, 237, 244, 173, 74, 66, 119, 162, 209, 175, 100, 161, 204, 125, 195, 138, 115, 150, 188, 43, 97,
72, 200]
keys.txt file successfully generated
Enter value of Kr
17
100
186
219
166
154
234
0
129
114
113
150
239
166
95
27
15
9
162
237
17
196
169
180
68
223
247
158
235
240
213
164
229
1
123
```

To decrypt the image vector keys should be entered



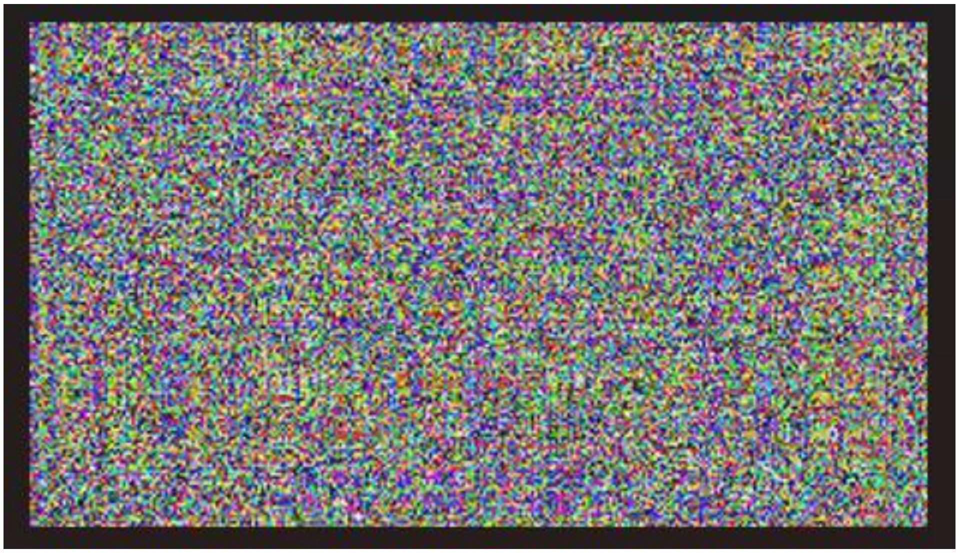
```
162
Enter value of Kc
130
137
241
0
33
160
252
199
216
45
103
177
245
27
91
226
206
220
104
203
210
130
248
139
61
186
190
84
219
69
126
121
153
48
104
47
190
113
177
193
135
133
160
109
8
155
```



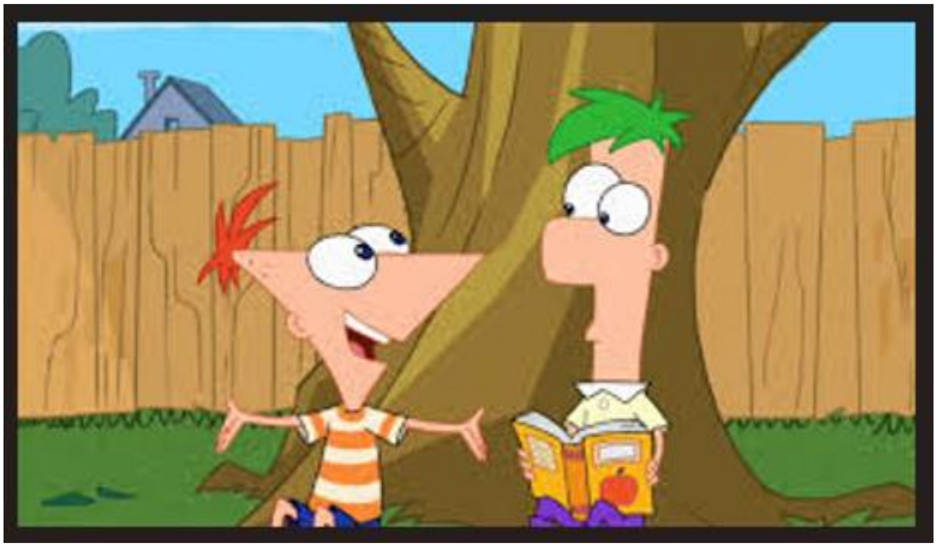
```
103
244
17
63
224
108
218
125
240
211
104
182
139
131
242
67
131
101
9
246
47
1
80
237
244
173
74
66
119
162
209
175
100
161
204
125
195
138
115
150
188
43
97
72
200
Enter value of ITER_MAX
1
Image successfully decrypted and stored in decrypted_images folder
```

OUTPUT

Encrypted Image



Decrypted Image



7. CONCLUSION AND FUTURE SCOPE

CONCLUSION

This project entitled **“Pixel Based Encoding of Images.”** is useful to the image galleries to protect their valuable images against digital cyber-attacks. The project is also useful to the military in protecting their digital secret map images. This project finally leads to the protection of valuable digital images.

FUTURE SCOPE

As of now, the keys are manually copied from the keys.txt file, while decrypting. It need to be further explored to automate this process, so that no manual intervention is needed for decryption.

8. BIBLIOGRAPHY

[1] AESHA N. ELGHANDOUR, AHMED M. SALAH, YASSER A. ELMASRY, AND ABDELRAHMAN A. KARAWIA (Member, IEEE): An Image Encryption Algorithm Based on Bisection Method and One-Dimensional Piecewise Chaotic Map, IEEE Access

[2] M.Kaur, D.Singh,n, K.Sun, U.Rawat,'Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5D chaotic map', Future Generation Computer Systems.

[3] https://www.python.org/

[4]https://github.com/baoboa/pyqt5/blob/master/pyuic/uic/pyuic.py

[5] https://www.numpy.org/

[6] https://riverbankcomputing.com/software/pyqt/intro