Chapter 1

Introduction

- Purpose
- Scope
- Objective
- Technology and Tool

INTRODUCTION

◆ PURPOSE:

In today's world most of the communication is done using electronic media. Data security plays vital role in such communication. Hence, there is a need to protect data from malicious attacks. This can be achieved by cryptography. The earlier encryption algorithm is Data Encryption Standard (DES) which has several loopholes such as small key size and sensible to brute force attack etc. These loopholes overcome by a new algorithm called as Advanced Encryption Standard Algorithm.

◆ SCOPE:

The scope of our project is presently specific. Both the sender and the receiver must have this software installed on their systems to encrypt or decrypt and compress or decompress the files transmitted between them. This includes all the users who want to interact electronically, whether it is through emails, sending a files etc.through local area network in order to keep their private information confidential.

- Each step is clearly stated and user will not face any ambiguity in using the software.

- The software provides clarity in its functionality even to naïve users.
- No complexity is involved.
- The various scope which cryptographic algorithms guarantees certain level of security, confidentiality and integrity of data.

◆ OBJECTIVE:

The main objective of our project is to encrypt or decrypt the any files for personal and professional security. Encryption and Decryption protects privacy of our documents and sensitive files by encrypting them using Advanced Encryption Standard (AES) algorithm to provide high protection against unauthorized data access.

In today's world the networking plays a very important role in our life. Most of the activities occur through the network. For the safe and secured exchange of information, we need to have security. The encryption has very

wide applications for securing data.Encryption refers to set of algorithms, which are used to convert the documents and any files to code or the unreadable form of files, and provides privacy. To decrypt the file to receiver uses the “key” for the encrypted files.

If you want to send sensitive information via email, simply paste the encrypted text or any files into your email or attach the encrypted file.

All the recipient has to do is to decrypt your text or any file. Encryption and Decryption works with text information and any files. Just select what you want to encrypt, and Encryption and Decryption software helps you keep documents, private information and files in a confidential way.

The project has the following objectives

1) Storing important information in encrypted form ensuring security.

2) We can prevent information loss when system crashes occurred.

3) The information will be recovered from the backup data.

4) Enhancing efficiency of data retrieval.

5) File Sending.

6) Better accuracy and improved consistency.

7) Help facility will be provided.

8) To understand and improve the computer data security through encryption of data.

9) To enhance the integrity of data.

10) To develop a platform to complement physical security.

◆ TECHNOLOGY AND TOOLS:

1) Jupyter notebook:

It is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning etc. The Notebook is a server-client application that allows editing and running notebook documents via a web browser. It can be executed on a local desktop requiring no internet access or can be installed on a remote server and accessed through the internet.

In addition to displaying/editing/running notebook documents. It has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating, executing, and visualizing Jupyter notebooks.

It is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s.It supports execution environments (aka kernels) in dozens of languages. By default Jupyter Notebook ships with the IPython kernel but there are over 100 Jupyter kernels as of May 2018.

2). Python:

Python is an interpreted, object-oriented, high level programming with dynamic semantics.

Its high level built in data structures, combined with dynamic typing and binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python’s simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. It supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Debugging Python program is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it causes an exception. When the program doesn’t catch the exception, the

interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on.

3) Pipenv and Pyenv:

pipenv lets you easily switch between multiple versions of Python. It's simple, unobtrusive, and follows the UNIX tradition of single-purpose tools that do one thing well.

This project was forked from rbenv and ruby-build, and modified for Python.

Pyenv does...

- Let you **change the global Python version** on a per-user basis.
- Provide support for **per-project Python versions**.
- Allow you to **override the Python version** with an environment variable.
- Search commands from **multiple versions of Python at a time**. This may be helpful to test across Python versions with tox.

In contrast with pythonbrew and pythonz, pyenv does not...

- **Depend on Python itself.** pyenv was made from pure shell scripts. There is no bootstrap problem of Python.
- **Need to be loaded into your shell.** Instead, pyenv's shim approach works by adding a directory to your $PATH.
- **Manage virtualenv.** Of course, you can create virtualenv yourself, or pyenv-virtualenv to automate the process.

Pipenv is a tool that aims to bring the best of all packaging worlds (bundler, composer, npm, cargo, yarn, etc.) to the Python world. Windows is a first-class citizen, in our world.

It automatically creates and manages a virtualenv for your projects, as well as adds/removes packages from your Pipfile as you install/uninstall packages. It also generates the ever-important Pipfile.lock, which is used to produce deterministic builds.

Pipenv is primarily meant to provide users and developers of applications with an easy method to setup a working environment. For the distinction between libraries and applications and the usage of setup.py vs Pipfile to define dependencies.

The problems that Pipenv seeks to solve are multi-faceted:

- You no longer need to use pip and virtualenv separately. They work together.
- Managing a requirements.txt file can be problematic, so Pipenv uses Pipfile and Pipfile.lock to separate abstract dependency declarations from the last tested combination.
- Hashes are used everywhere, always. Security. Automatically expose security vulnerabilities.
- Strongly encourage the use of the latest versions of dependencies to minimize security risks arising from outdated components.
- Give you insight into your dependency graph (e.g. $ pipenv graph).
- Streamline development workflow by loading .env files.

4). Numpy:

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

Chapter 2

Project Management

- Project Planning
- Project Scheduling
- Risk Management

2.0. PROJECT MANAGEMENT

◆ PROJECT PLANNING

Project Planning is concerned with identifying and measuring the activities, milestones and deliverables produced by the project. Project planning is undertaken and completed sometimes even before any development activity starts. Project planning consists of following essential activities:

◆ Scheduling manpower and other resources needed to develop the system.

◆ Staff organization and staffing plans.

◆ Risk identification, analysis, and accurate planning.

◆ Estimating some of the basic attributes of the project like cost, duration and efforts the effectiveness of the subsequent planning activities is based on the accuracy of these estimations.

◆ Miscellaneous plans like quality assurance plan, configuration management plan, etc.

Project management involves planning, monitoring and control of the process, and the events that occurs as the software evolves from a preliminary concept to an operational implementation. Cost estimation is a relative activity that is concerned with the resources required to accomplish the project plan.

1.1) Project Development Approach And Justification:

A Software process model is a simplified abstract representation of a software process, which is presented from a particular perspective. A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required. All software development can be characterized as a problem-solving loop which in four distinct stages is encountered:

- Requirement analysis
- Coding
- Testing
- Deployment

1.2) Milestones and Deliverables:

As software is tangible, this information can only be provided as documents that describe the state of the software being developed without this information it is impossible to judge progress at different phases and therefore schedules cannot be determined or updated.

Milestone is an end point of the software process activity. At each milestone there should be formal output such as report that can be represented to the guide. Milestones are the completion of the outputs for each activity. Deliverables are the requirements definition and the requirements specification.

Milestone represents the end of the distinct, logical stage in the project. Milestone may be internal project results that are used by the project manager to check progress. Deliverables are usually Milestones but reverse need not be true. We have divided the software process into activities for the following milestone that should be achieved.

Software Process Activity	Milestone
Project Plan	Project schedule
Requirement Collection	User requirements, System Requirements
Analysis of Dataset	Choosing of appropriate dataset.
Implementation	Algorithm implementation.

Table Milestones and Deliverables

1.3) Roles and Responsibilities:

This phase defines the role and responsibilities of each and every member involved in developing the system. To develop this system there is only one person involved in working on the whole application. The same was responsible for each and every part of developing the system. Our team structure is of single control team organization as it consist of me and my guide as chief programmer organization.

1.4) Group Dependencies:

The structure chosen for the system is the chief programmer structure .In this system, Chief Programmer team structure is used because in the organization, a senior engineer provides the technical leadership and is designated as the chief programmer. The chief programmer partitions the task into small activities and assigns them to me on time deadline basis. He also verifies and integrates the products developed by me and i work under the constant supervision of the chief programmer. For this system reporting entity represents myself and the role of chief programmer is played by my internal guide.

◆ PROJECT SCHEDULING

The scheduling is the peak of a planning activity, a primary component of software project management. When combined with estimation methods and risk analysis, scheduling establishes a roadmap for project management. The characteristics of the project are used to adapt an appropriate task set for doing work.

Task	1Dec-25Dec	31Jan-10Feb	10Feb-20Feb	20Feb-30Feb	30Feb-5March	5March-10March
Develop project proposal	25 days					
Analysis		11 days				
Designing			10 days			
Coding				10days		
Unit Testing					5 days	
Implementation						5 days

Fig. shows Gant chart of this Project

◆ RISK MANAGEMENT

Risk management consists of a series of steps that help a software development team to understood and manage uncertain problems that may arise during the course of software development and can plague a software project.

Risks are the dangerous conditions or potential problems for the system which may damage the system functionalities to very high level which would not be acceptable at any cost. so in order to make our system stable and give its 100% performance we must have identify those risks, analyze their occurrences and effects on our project and must prevent them to occur.

3.1) Risk Identification

Risk identification is a first systematic attempt to specify risks to project plan, scheduling resources, project development. It may be carried out as a team process using brainstorming approach.

Technology risk: Technical risks concern implementation and testing problems.

- Dataset Enlargement
- Algorithm Output.

People Risks: These risks are concerns with the team and its members who are taking part in developing the system.

- Lack of knowledge
- Lack of clear vision.
- Poor communication between people.

Tools Risks:

These are more concerned with tools used to develop the project.

- Tools containing virus.

General Risks:

General Risks are the risks, which are concerned with the mentality and resources.

- Rapidly changing Datasets.
- Lack of resources can cause great harm to efficiency and timelines of project.
- Changes in dataset can cause a great harm to implementation and schedule of developing the system.
- Insufficient planning and task identification.
- Decision making conflicts.

3.2) Risk Analysis

“Risk analysis = risk assessment + risk management + risk communication. ” Risk analysis is employed in its broadest sense to include:

Risk assessment

Involves identifying sources of potential harm, assessing the likelihood that harm will occur and the consequences if harm does occur.

For this project It might be :- Software(Tool) Crashing.

Risk management

Evaluates which risks identified in the risk assessment process require management and selects and implements the plans or actions that are required to ensure that those risks are controlled.

Precautions taken to make risks minimal are as under:-

Keeping the software tool up to date by updating the software periodically.

Risk communication

Involves an interactive dialogue between guide and us, which actively informs the other processes.

Steps taken for risk communication is as under: -

◆ All the possible risks are listed out during communication and project is developed taking care of that risks.

Chapter 3

System Requirements Study

- User Characteristics
- Hardware and Software Requirements
- Constraints Assumptions and Dependencies

◆ SYSTEM REQUIREMENT STUDY

◆ USER CHARACTERISTICS

Admin:-

- ◆ Mange project
- ◆ Add Features

User:-

- ◆ Encrypt Text Files.
- ◆ Insert Key Text File.
- ◆ Decrypt the Encrypted Text Files.

◆ HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATION

This shows minimum requirements to carry on to run this system efficiently.

1.2.1) Hardware Requirements Server side Hardware Requirement:

Devices	Description
Processor	Intel Core Duo 2.0 GHz or more
RAM	512 MB or more
Hard Disk	10 GB or more

Table Server side Hardware Requirement

1.2.2) Software Requirements

For which	Software
Operating System	Windows XP/2003/vista/7/8/10,Linux, Mac OS x
Front End	Jupyter notebook
Back End	Numpy
Scripting Language	Python

Table Software Requirements

1.2.3) Client side Requirements

For which	Requirement
Terminal	Any command line supported OS.

Table client-side Requirements

- **CONSTRAINTS**

1.3.1) Hardware Limitations

The major hardware limitations faced by the system are as follows:

If the appropriate hardware is not there like processor, RAM, hard disks

-the problem in processing requests of client

-if appropriate storage is not there our whole database will crash due to less storage because our main requirement is large storage.

1.3.2) Interfacing with other systems

There should be the compatible terminal to perfectly detect operate with the script. The functionality of the system should be such that it can be used as sub module of some larger applications.

1.3.3) Reliability Constraints

The major reliability constraints are as follows:

- The software should be efficiently designed so as to give reliable recognition of fake news and so that it can be used for more pragmatic purpose.

- The design should be versatile and user friendly.

- The application should be fast, reliable and time saving.

- The system should have universal adaptations.

- The system be compatible with future upgradation.

- **DEPENDENCIES**

The entire project depends on various libraries of python. The libraries are as follows:

NumPy: NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Python: This module implements a number of iterator building blocks inspired by constructs from APL, Haskell and SML. Each has been recast in a form suitable for Python.

Chapter 4

System Analysis

- Study of Current System
- Problem and Weaknesses of Current System
- Requirements of New System
- Feasibility Study
- Requirements Validation
- Features of New System
- Data Flow Diagram
- ER Diagram
- UML Diagrams
- Selection of Hardware and Software and Justification

◆ STUDY OF CURRENT SYSTEM

There are various encryption system available which can be used to perform encryption and decryption based AES Encryption algorithm.

They uses various approaches to perform the encryption and decryption in the various possible ways.

◆ PROBLEMS AND WEAKNESS OF CURRENT SYSTEM

The current system is undoubtedly well-designed for performing encryption and decryption but it has some following limitations:

- Lack of an awareness of this system.
- Implementation is difficult and complex
- Some security related issues may be created.
- Cost Effectiveness

◆ REQUIREMENTS SPECIFICATION

Requirements specification adds further information to the requirements definition.

3.1) Algorithm Requirements

- Dataset
- Input
- Appropriate functions
- Efficiency
- Output

3.2) System Requirements

- **Usability:**

The system should be easily able to encrypt and decrypt the text files.

- **Efficiency:**

The system should provide easy and fast response.

◆ FEASIBILTIY STUDY

An important outcome of the preliminary investigation is the determination that the system is feasible or not. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop a project.

The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different text files which would be input to the system, the processing required to be carried out on these text files, the output required to be produced by the system as well as the various constraints on the behaviors of the system.

4.1) Does the system contribute to the overall objectives of the organization?

The main aim of behind development of this system is to provide free encryption and decryption of text files that can prevent the social bullying of the persons which need it and also for the people who doesn't want to waste their time on bothering about security of their documents while transferring files over the internet.

4.2) Can the system be implemented using the current technology and within the given cost and schedule constraints?

◆ The system can be easily implemented using existing technology. The technology used is numpy and python which is user friendly and freeware. After seeing the functionality that system provides the cost of developing the application does not matter.

◆ Taking the schedule constraints in consideration the time available is approximately 1 months. The time period is enough to develop the system.

5. REQUIREMENT VALIDATION

A requirements validation is concerned to check whether the requirements actually define the system, which the customer wants? Requirements validation is important because errors in requirements document can lead to extensive rework costs when they are subsequently discovered. We have performed the following validation checks

- **Validity checks**

Check whether the information entered is in valid format

- **Consistency checks**

A requirement in a document is not conflicting.

- **Completeness checks**

The requirements document includes requirement, which define all functions, and constraints intended by the system user.

- **Realism checks**

Using knowledge of existing technology, the requirements are checked to ensure that they could actually be implemented.

- **Verifiability**

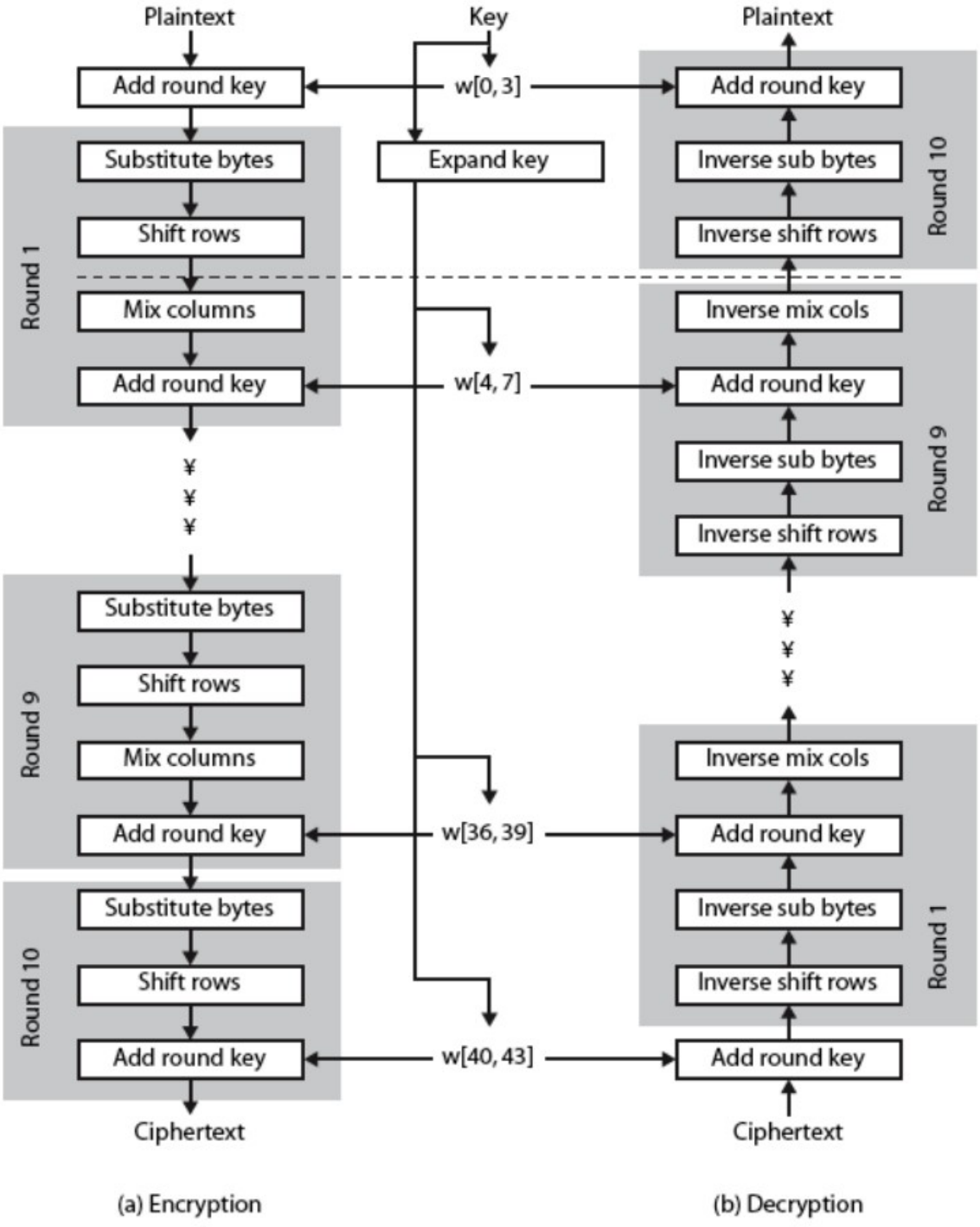
The requirements are given in verifiable manner (e.g.: Using quantifiable measures) to reduce disputes between client and developer.

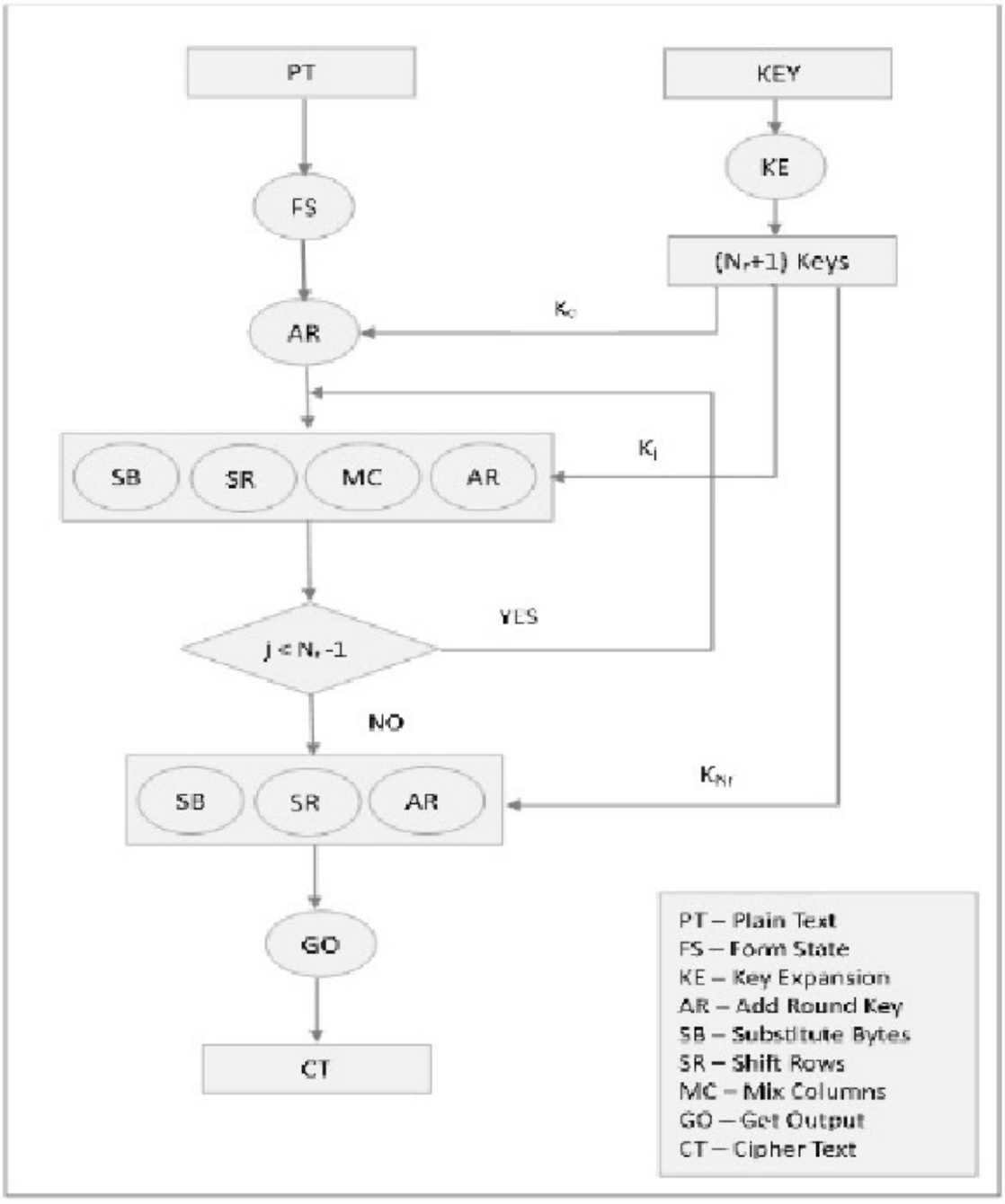
6. FEATURES OF NEW SYSTEM

We will try to develop application as follows:

- The system being available in regional languages.
- Provide the more awareness in our country India about this concept.
- User can upload his/her idea through description, team information, videos of his/her work, and the form of reward and main for which purpose he/she needed the money.
- One can pledge the money if one like the idea.
- Communication provided between innovators and investors.
- Safety for money transfer and surety of security of ideas.

7. FLOWCHART OF NEW SYSTEM:



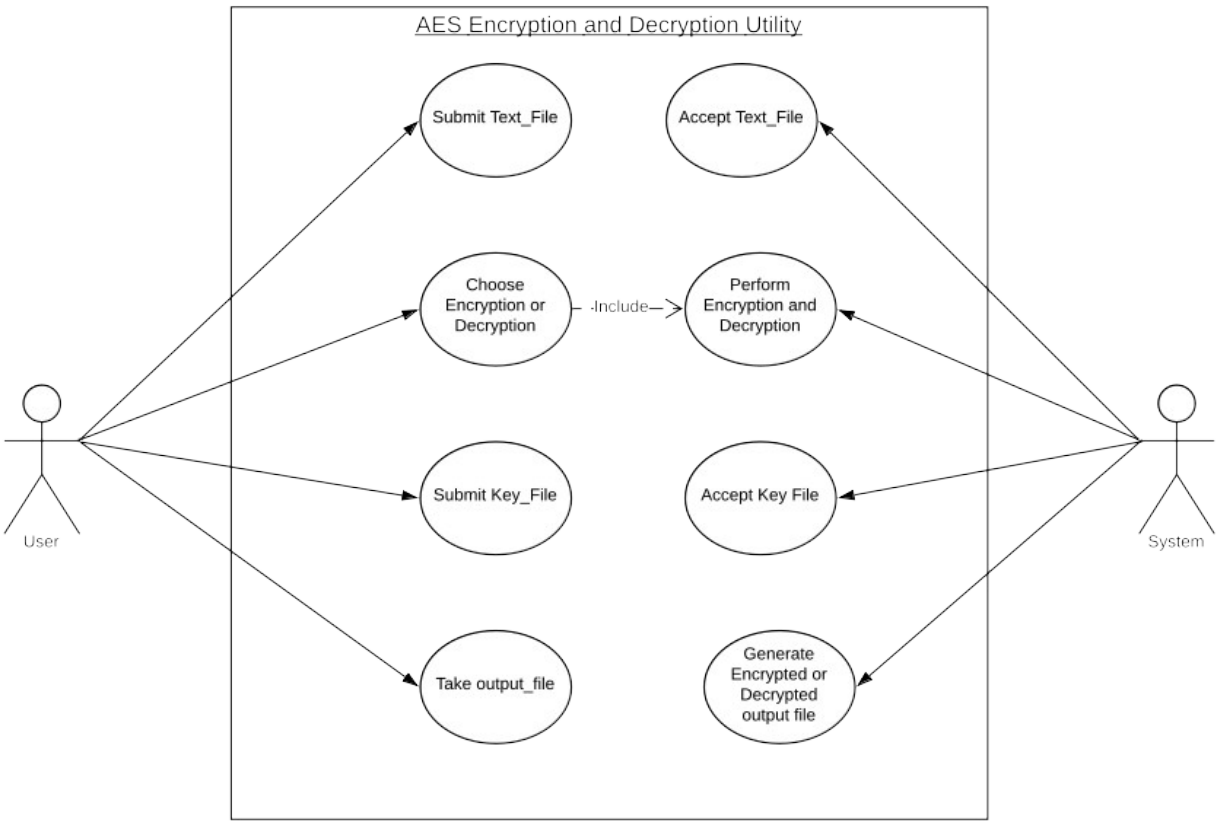


Data Flow Diagram

8. Use Case Diagrams

Following are the use case diagrams for our system that describe a set of actions (use cases) that the system should or can perform in collaboration with one or more external users of the system (actors).

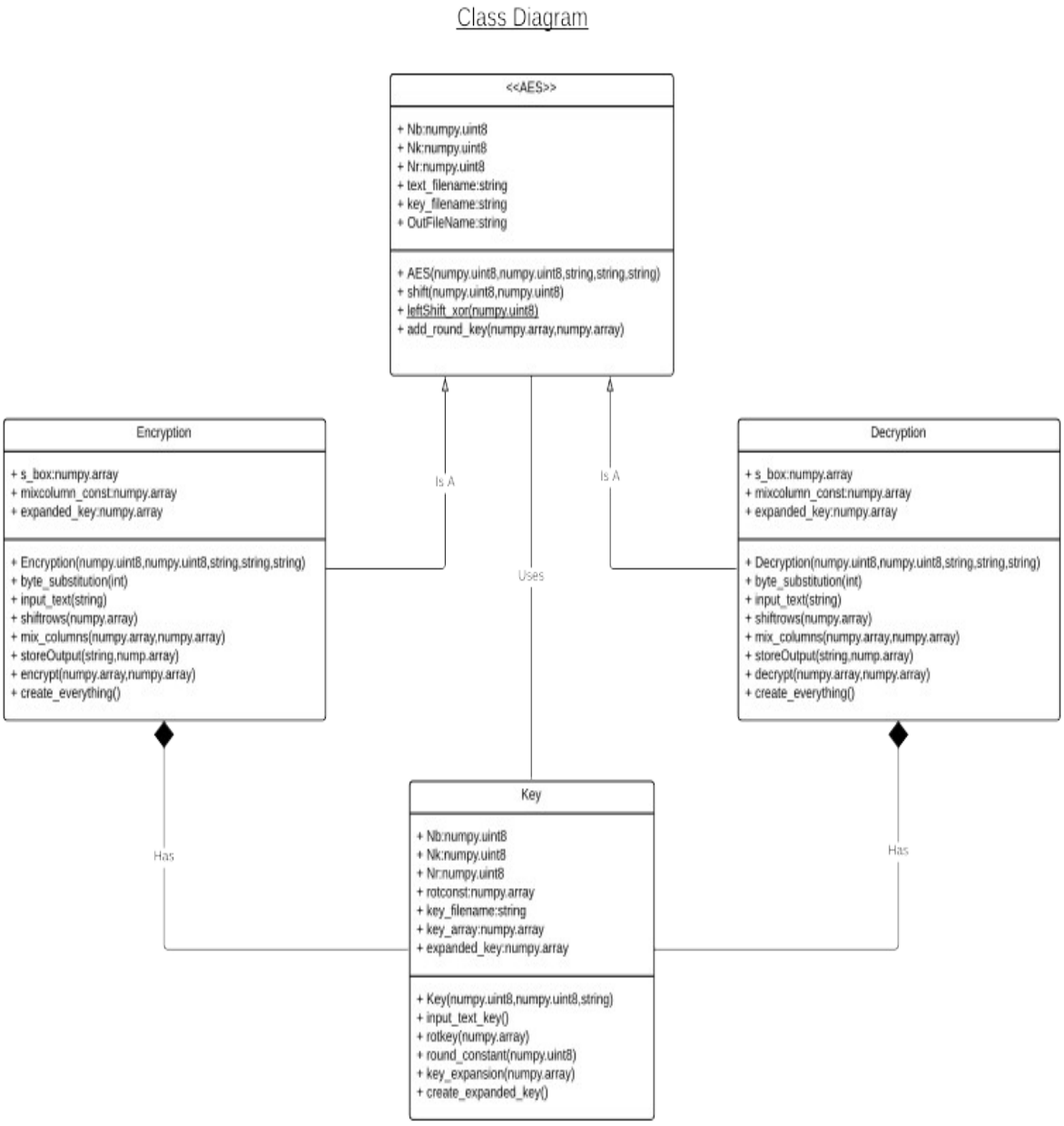
8.1 Use Case Diagram 1



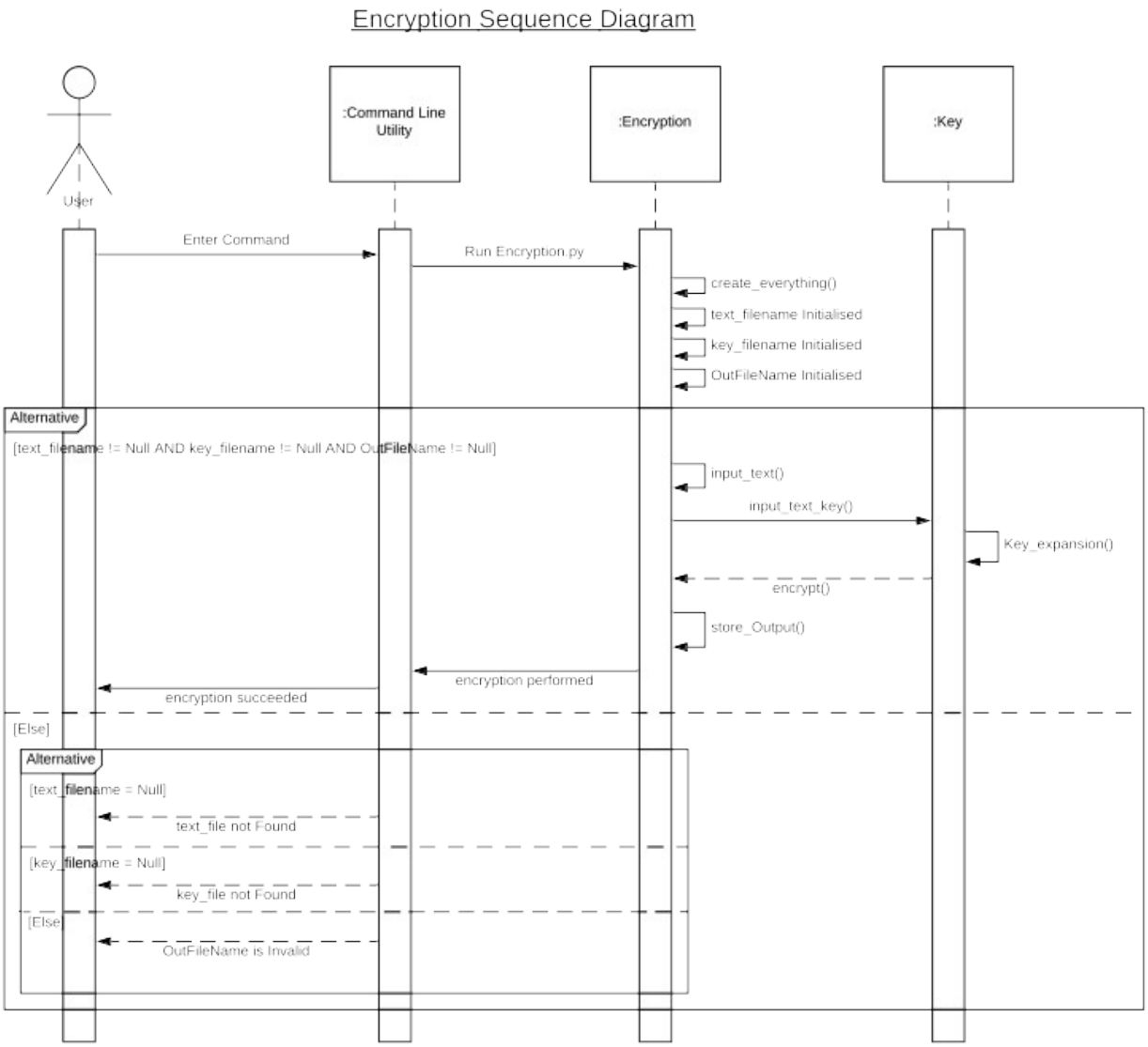
Use Case Diagram 1

This use case diagram tell the various work that should be done by the user and the software admin, and it tells that how these two entities are related with each other in the software.

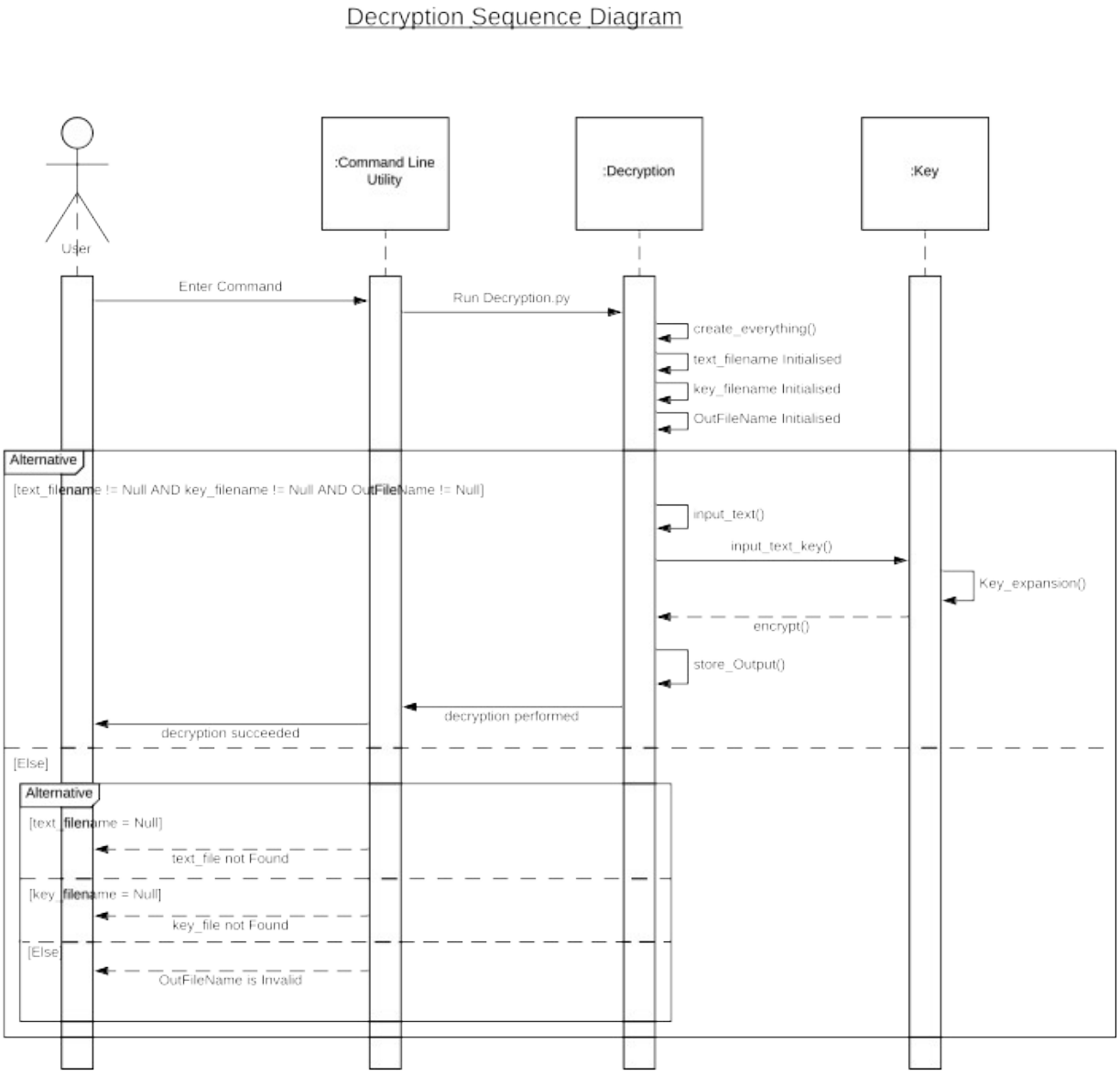
9. Class Diagram



10. Sequence Diagram for Encryption



11. Sequence Diagram for Decryption



12 SELECTION OF HARDWARE AND SOFTWARE

The Tables below give idea of the hardware and software required for the system and client side requirements.

- **Hardware Selection**

Devices	Description
Processor	Intel Core Duo 2.0 GHz or more
RAM	512 MB or more
Hard Disk	10 GB or more

Table Hardware Requirements

- **Software Selection**

For which	Software
Operating System	Windows XP/2003/vista/7/8/10,Linux, Mac os x
Front End	Jupyter Notebook
Back End	Numpy
Scripting Language	Python

Table Software Requirements

- **Client side requirements:**

For which	Requirement
Terminal	Any Compatible terminal or command line os device

Table Client Side Requirements

Chapter 5

System Design

- Overview
- Product Function
- User Characteristics
- Constraints
- User Requirements
- Performance Requirements
- Code Snippet

1. Overview

This software is fairly simple in terms of it uses, here user has been provided with two commands one for encryption and one for decryption and executing which he can perform the encryption or the decryption. The summary of overall procedure is as follows.

1. User will enter COMMAND in the command prompt or the terminal(ENCRYPTION FOR encrypting the text file and DECRYPTION FOR decrypting the already encrypted text files).

2. After entering the command the user has to enter the filename of the file to be encrypted or decrypted and the keytextfile and the file in which the user want the decrypted or encrypted file output respectively.

3. Once the process completed then user can take its output file as the encrypted or the decrypted file whatever be the case.

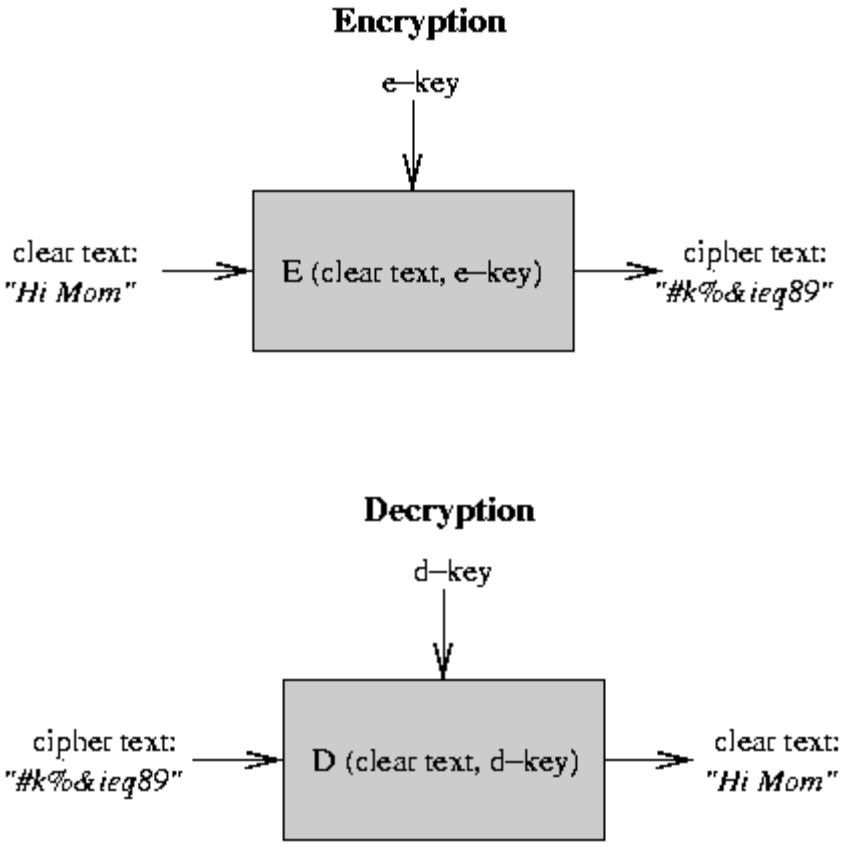


Figure Layered Architecture

2 Product Functions

1. The command for Encryption or Decryption must be entered.

2. The file need to be encrypted or decrypted must be supplied to the terminal with the respective keytxtfile.

3. AES algorithm steps are performed on the file supplied, based on the Encryption and Decryption needs.

4. Once encryption or decryption is completed the resultant data would now be written on the output file name supplied by the user.

3 User Characteristics

Administrator: Will add new features and restrictions on the software or the command line tool genrerated.

User: The main actor would be performing encryption or decryption based on its needs.

4 Constraints

1 Its is a command line tool and hence it requires user to be bit familiar with terminal or command prompt.

2 Our software will only be available in English language.

3 It can be use to encrypt and decrypt the text files only.

4 To share the file with other person user has to use the some network means on his own.

5 It can take time to encrypt or decrypt the text file based on the length of the file.

5 User Requirements

Following are the user requirements that describe what the user expects from the software to do.

5.1 External Interface Requirements

The user interface will be web based provided to user through a web browser. The screen will consist of a log in form. Upon logging in the user will presented with a dashboard. The dashboard will consist of a header, sidebar menu and body. On the top right the menu for managing user preferences

will be provided. The body will be consisting of dialogue box which will be used to get the input from user. There will be a button to submit the query entered by user in the dialogue box. Below the dialogue and button, a list of previously processed URLs with their rating from user will be displayed. Against each list item the user will be able to rate that corresponding processed URL result either good or bad.

1. Numpy: a scientific computing package generating N-dimensional array objects. As for this project, several machine learning models use Numpy as the data container; the implementation of our random tree and random forest also depends on this.

5.2 Functional Requirements

1. Take a valid file (by this we mean text file) from the user.

2. Take a valid key file(a text file) from the user.

3. Expand the key properly to be used in the encryption process or decryption process.

4. The we will convert the relevant file(that is both the text and key text file) to its corresponding UTF-8 coded file.

5. Properly encode or decode the file using the AES algorithm.

6. Properly writing the decoded or the encoded file to the output file as supplied by the user.

7. Proper execution of the commands created to execute the respective encoding or the decoding script.

6. Performance Requirements

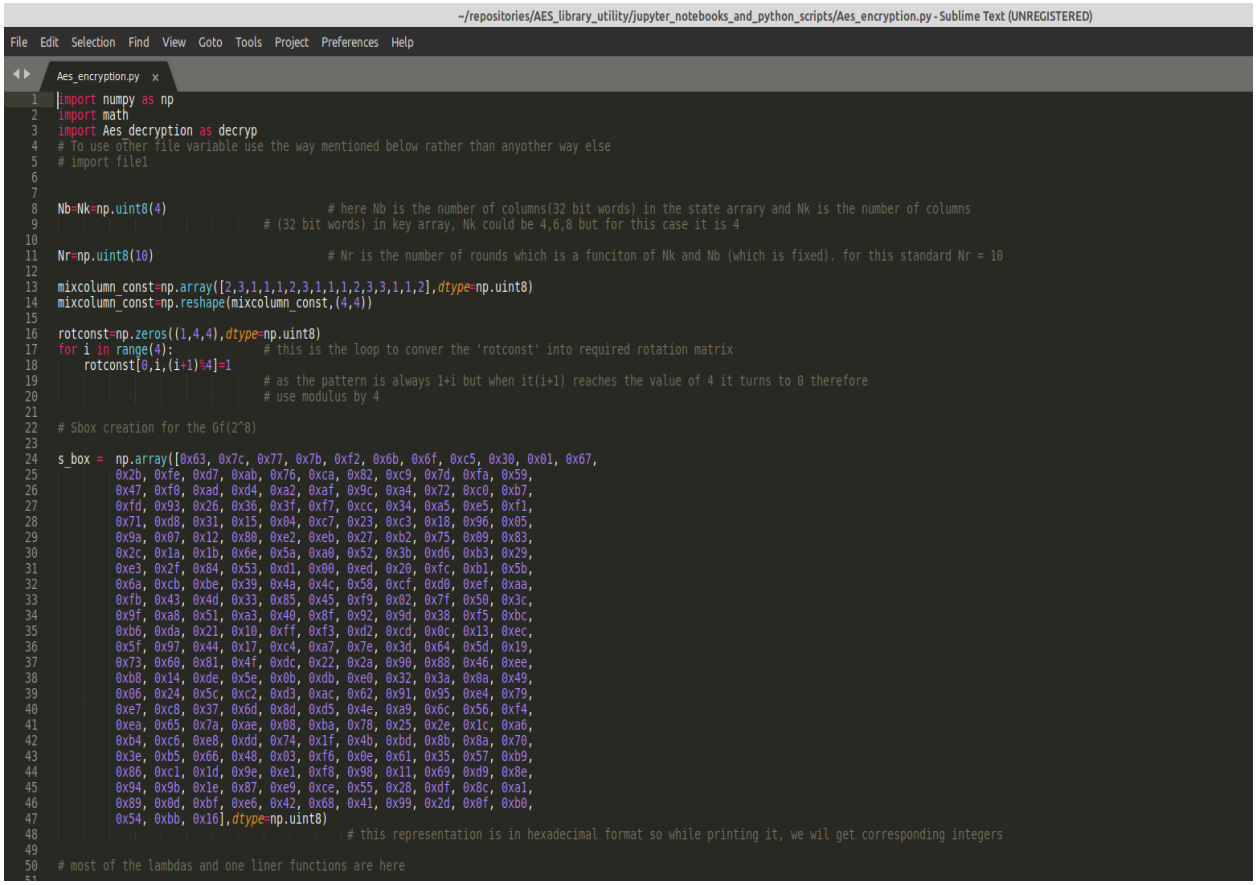
Table Performance Requirements

ID	Performance Requirement
1	Commands should called the script within one or two clock cycle.
2	Time taken by AES algorithms should be in milliseconds for average length text file.
3	System should be able to handle multiple simultaneous requests.

7. CODESNIPPET:

The Jupyter notebook will be used for implementing AES algorithm and it has many files including test text files and python notebooks which has following extensions i.e. “.tsv” “.pynb” .

We also tried to use python libraries like numpy. A small level implementation of our project is shown below.



Chapter 6

Proposed Solution and Code Implementation

- Proposed Solution
- Implementation Environment
- Program/Module Specification
- Coding Standards
- Coding

1. Proposed Solution

The solution to the problem defined in the earlier section was to create a open source utility which is reasonably fast and accurate to encode and decode the text files based on the key text files provided.

Since our code is free from the GUI implementations so it is reasonable fast as compared to the other implementations available.

1.1 Methodology

Developing an command line encryptor and decryptor based on AES algorithm was a challenging problem due to math involved in it. To make sure, that we

accomplished this task efficiently, without facing major problems, which would have caused major

redesigns and re-engineering of the software architecture, in a time and cost constrained project

environment, we started off with developing SRS (Software Requirement Specifications) and detailed

design of the system. Gantt chart and work break down structure were created in that phase to monitor the project and when a phase should start or end.

After that we started to gather research papers for proper understanding of the AES algorithm. After that we started our research on which language to choose and this results in choosing python as the language.

Further the implementation of AES using the numpy library which helps us to develop the project easily.

2. IMPLEMENTATION ENVIRONMENT

As our project is study based project and the best tool which is used at the undergraduate level is “Anaconda” . It consists of different modules in which we can code but for our project we have used Jupyter Notebook, which is used for high level python programming. Jupyter Notebook provides browser environment as it opens up in the *browser. It* can also connect to kernel and terminal.

Moreover the IDE Sublime Text and Visual Code are the tremendous helpful tool for such kind of development.

3. PROGRAM/MODULE SPECIFICATION

The naive bayes classifier algorithm is the most applicable algorithm to implement fake news detection as it works on conditional probability and other major concepts of Data mining that are used in this project and we have also studied it in 4th semester which made the understanding of code quite easy.

```
#-------------------------------------------------------------
# Naive Bayes classifier for Multinomial model
#-------------------------------------------------------------

clf = MultinomialNB()

clf.fit(tfidf_train, y_train)                        # Fit Naive Bayes classifier according to X, y

pred = clf.predict(tfidf_test)                      # Perform classification on an array of test vectors X.
score = metrics.accuracy_score(y_test, pred)
print("accuracy:   %0.3f" % score)
cm = metrics.confusion_matrix(y_test, pred, labels=['FAKE', 'REAL'])
plot_confusion_matrix(cm, classes=['FAKE', 'REAL'])
print(cm)


clf = MultinomialNB()

clf.fit(count_train, y_train)

pred = clf.predict(count_test)
score = metrics.accuracy_score(y_test, pred)
print("accuracy:   %0.3f" % score)
cm = metrics.confusion_matrix(y_test, pred, labels=['FAKE', 'REAL'])
plot_confusion_matrix(cm, classes=['FAKE', 'REAL'])
print(cm)

```

The final output is generated with the help of command line terminal and linux operating system basically linux mint 20 and sublime text for the editing in the text files.

4. CODING STANDARDS

Normally, good software development organization requires their programmers to adhere to some well-defined and standard style of coding called coding standard.

4.1 Variable Standards:

Our project implementation uses apt variable names that makes the understanding of the domain quite easy.

4.2 Comment Standards:

Comments increases readability of our code and makes it easy for the third party to understand it. We have used comments everywhere needed and also used the references of the online codes.

Every code block and the different modules start with the comments, describing in brief about the code and the details.

Comments may also be used in between and along with the lines of code to explain one specific line or lines.

In python we can use. '#' to for single comment and for multiple lines we can use delimiters that is," ' ' ' " . We have used both during programming.

5. Coding

AES_Encryption.py:

```
import numpy as np
import math
import Aes_decryption as decryp
# To use other file variable use the way mentioned below rather than anyother way
else
# import file1


Nb=Nk=np.uint8(4)                    # here Nb is the number of columns(32 bit
words) in the state arrary and Nk is the number of columns
                    # (32 bit words) in key array, Nk could be 4,6,8 but for this
case it is 4

Nr=np.uint8(10)                    # Nr is the number of rounds which is a funciton
of Nk and Nb (which is fixed). for this standard Nr = 10

mixcolumn_const=np.array([2,3,1,1,1,2,3,1,1,1,2,3,3,1,1,2],dtype=np.uint8)
mixcolumn_const=np.reshape(mixcolumn_const,(4,4))

rotconst=np.zeros((1,4,4),dtype=np.uint8)
for i in range(4):          # this is the loop to conver the 'rotconst' into required
rotation matrix
   rotconst[0,i,(i+1)%4]=1
                    # as the pattern is always 1+i but when it(i+1) reaches the
value of 4 it turns to 0 therefore
                    # use modulus by 4

# Sbox creation for the Gf(2^8)

s_box =  np.array([0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01,
0x67,
       0x2b, 0xfe, 0xd7, 0xab, 0x76, 0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59,
       0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0, 0xb7,
       0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1,
       0x71, 0xd8, 0x31, 0x15, 0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05,
       0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75, 0x09, 0x83,
       0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29,
       0xe3, 0x2f, 0x84, 0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b,
       0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf, 0xd0, 0xef, 0xaa,
```

```
        0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c,
        0x9f, 0xa8, 0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc,
        0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2, 0xcd, 0x0c, 0x13, 0xec,
        0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19,
        0x73, 0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee,
        0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb, 0xe0, 0x32, 0x3a, 0x0a, 0x49,
        0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79,
        0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4,
        0xea, 0x65, 0x7a, 0xae, 0x08, 0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6,
        0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a, 0x70,
        0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9,
        0x86, 0xc1, 0x1d, 0x9e, 0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e,
        0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf, 0x8c, 0xa1,
        0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0,
        0x54, 0xbb, 0x16],dtype=np.uint8)
                                # this representation is in hexadecimal format so while
printing it, we wil get corresponding integers

# most of the lambdas and one liner functions are here

shift = lambda r,Nb: (1 if r== 1 else (2 if r == 2 else (3 if r== 3 else (0 if r==0 else
None)))) if Nb == 4 else None

byte_substitution= np.vectorize(lambda index: s_box[index])
                                # a copy of state_array or array for which subtitution is
required should be passed in it.

# calculating the value of x^i for where x = 2 in decimal and the multiplication is
abiding the rules of galois field 2^8
lefShift_xor= lambda x : np.uint8(x<<1) if (x<128) else np.uint8(x<<1)^27
                                # this is equivalent to multiply a number here 'x' with 2
in galois field 2^8
                                # here i use np.uint8 because python integer is not of
8-bit and bit shift wouldn't work
                                # correctly and that is it would drop the bit shifted after
8 bits positions

# methods are all here
# comment for any line of code is listed just below that line at a suitable distance

def input_text(filename):

    with open(filename,'r',encoding='utf-8') as f:
        result=list(map(ord,f.read()))
```

```
                                                    # note that 'map' can work only once,
so store it result in a varaiable
    state_array= np.array(result, dtype=np.uint8)
                                                    # store them into utf-8 integer
encoding in a variable of 8-bit integer.

    length= len(state_array)
    padding= length%16
    state_array= np.append(state_array,np.zeros((16-padding)
%16).astype(np.uint8))
                                                    # since we don't adding axis here
therefore the values of the
                                                    # added array will be flattened before
use and would then be merged.

    state_array= np.array([np.reshape(i,(4,4)).transpose() for i in
np.split(state_array,len(state_array)/16)])
    return state_array

                                                    # here each element of the
'state_array' will represent a state array
                                                    # for the given text.



def input_text_key(filename):

    with open(filename,'r',encoding='utf-8') as f:
        result=list(map(ord,f.read()))
                                                    # note that 'map' can work only once,
so store it result in a varaiable
    state_array= np.array(result, dtype=np.uint8)
                                                    # store them into utf-8 integer
encoding in a variable of 8-bit integer.

    length= len(state_array)
    if length >= 16:
        state_array= state_array[:16]
                                                # here neglecting the rest of the file txt if
more than 16 characters
    else:
        padding= length%16
        state_array= np.append(state_array,np.zeros(16-padding).astype(np.uint8))
```

```
                                        # here adding the rest of characters as 0 if
have less than 16
                                        # character in the input text key file.

    state_array= np.array([np.reshape(state_array,(4,4)).transpose()])
    return state_array
                                        # reshaping the array as a 4x4 matrix and
using list to encapsulate it
                                        # into another np.array to have the
structure similar to the
                                        # state_array

def rotkey(roundkey):           # this is the method for applying the left rotation in
the given roundkey in total, column wise
    return rotconst.dot(roundkey).transpose()
                        # since the rotconst can be used to left rotate a given matrix
element column wise hence we apply
                        # multiplication of the 'roundkey' with rotconst(specifically
rotconst with roundkey)


def round_constant(i):
    roundconstant= np.uint8(1)
    for j in range(2,i+1):
                                # use i because while calling the round_constant
function, index of the array is used
                                # and which is equal to the round number itself and we
want one less than round number as
                                # as per page 19 and 20 article and sudo code given in
nist notes of AES
        roundconstant= lefShift_xor(roundconstant)

    return np.array([roundconstant,0,0,0]).reshape(4,1)


def keyexpansion(key):

    for index in range(1,11):
        key= np.append(key,np.zeros((1,4,4),dtype=np.uint8),axis=0)
        key[index,:,0]= (key[index-
1,:,0].reshape(4,1)^(byte_substitution(rotkey(key[index-
1,:,3]))^round_constant(index))).reshape(4)
        key[index,:,1]= key[index,:,0]^key[index-1,:,1]
        key[index,:,2]= key[index,:,1]^key[index-1,:,2]
```

```
        key[index,:,3]= key[index,:,2]^key[index-1,:,3]

    return key
                # returning the value here is kind of necessity because we can't use
np.append inside a function as it wouldn't
                # change the actual passed array beacause we are assigning a new
array reference to another object which is "key"
                # in this case whereas if we apply np.insert then it do make the
changes because it change the actual passed array
                # reference.

def shiftrows(element):
    temp = np.zeros(element.shape,dtype=np.uint8)
    for r in range(4):
        for j in range(Nb):
            temp[r,(Nb-shift(r,Nb)+j)%Nb]= element[r,j]
    element[:,:]=np.copy(temp)
                #  don't panic about assignment to value of 'element' reference as it is
working absloutely fine, moreover remember that
                # list are immutable and that they are called by reference in functions
and not called by value


def multiplication_for_matrix(X,Y):
    # iterate through rows of X
    rough= np.zeros((len(X),len(Y[0])),dtype= np.uint8)
    for i in range(len(X)):
        # iterate through columns of Y
        for j in range(len(Y[0])):
            # iterate through rows of Y
            for k in range(len(Y)):
                if X[i,k]==2:
                    rough[i,j] = rough[i,j]^lefShift_xor(Y[k,j])
                if X[i,k]== 3:
                    rough[i,j] = rough[i,j]^(lefShift_xor(Y[k,j])^Y[k,j])
                if X[i,k]== 1:
                    rough[i,j] = rough[i,j]^Y[k,j]

    Y[:,:]=rough[:,:]


def add_round_key(cipher,expanded_key):
    cipher[:,:]= cipher[:,:]^expanded_key[:,:]
```

```
def storeOutput(filename,state_array_out):
    char_written_length=0
    rowwritten=''
    with open(filename,'w',encoding='utf-8') as f:
        for i in state_array_out:
            rowwritten=' '.join(map(str, np.ravel(i)))
            char_written_length=char_written_length+len(rowwritten.split())
            f.write(rowwritten+' ')

    return char_written_length

# only the block of cipher should be passed that need to be ecrypted and not the
whole cipher,containing all the blocks of the cipher
def final_encryption(cipher_input,expanded_key):
    round_number= 0
    add_round_key(cipher_input,expanded_key[round_number,:,:])
                        # this has been done because 0 round_key should be added
before any processing of the input cipher
    for round_number in range(1,10):
        cipher_input= byte_substitution(cipher_input)
        shiftrows(cipher_input)
        multiplication_for_matrix(mixcolumn_const,cipher_input)
        add_round_key(cipher_input,expanded_key[round_number,:,:])
                        # one more thing that has been done here is that passing only
the necessary part of the
                        # expanded_key with no passing of round number is needed
then(also round_number variable
                        # should also be used in passing the expanded_key as follows
expanded_key[round_number,:,:])

                        # above is the process for the round 1 to round 9

    cipher_input= byte_substitution(cipher_input)
    shiftrows(cipher_input)
    round_number= 10
    add_round_key(cipher_input,expanded_key[round_number,:,:])

                        # above is the process for the round 10 only

    return cipher_input

# code for creation of state array and performing encryption on all blocks of the
cipher created
def creation_everything():
```

```
    filename= input("enter the name of the file with path that need to be encrypted
")
    encryption_key= input("enter the name of the key file with path that need to
encrypt the file, max length of file is 16 characters ")
    OutFileName= input("enter the name of the output file with path that is used to
store the encrypted output ;")
    state_array= input_text(filename)
    original_key= input_text_key(encryption_key)
    expanded_key= keyexpansion(np.copy(original_key))
    state_array_out= np.zeros(state_array.shape,dtype=np.uint8)


    for index,block in enumerate(state_array):
        state_array_out[index]= final_encryption(np.copy(block),expanded_key)

    total_char_wrote= storeOutput(OutFileName,np.copy(state_array_out))

    return
np.array_equal(decryp.encrypted_text_read(OutFileName),state_array_out),
total_char_wrote
    # below code should only be used for debugging purpose (and before return) as
it tells that whether array written to the output file
    # is same as the array read again from the output file
    #print(decryp.encrypted_text_read(OutFileName),"   \n",state_array_out)



if __name__=='__main__':

    status,total_char_wrote= creation_everything()
    print(status," ",total_char_wrote)
```

AES_Decryption.py

```
import numpy as np
import math
import Aes_encryption as encryp



Nb=Nk=np.uint8(4)                    # here Nb is the number of columns(32 bit words) in the state arrary and Nk is the number of columns
                                     # (32 bit words) in key array, Nk could be 4,6,8 but for this case it is 4


Nr=np.uint8(10)                      # Nr is the number of rounds which is a funciton of Nk and Nb (which is fixed). for this standard Nr = 10
inverse_mixcolumn_const=np.array([14,11,13,9,9,14,11,13,13,9,14,11,11,13,9,14],dtype=np.uint8)
inverse_mixcolumn_const=np.reshape(inverse_mixcolumn_const,(4,4))



# RSbox creation for the Gf(2^8)


r_s_box = np.array([0x52, 0x09, 0x6a, 0xd5, 0x30, 0x36, 0xa5, 0x38, 0xbf, 0x40, 0xa3,
                    0x9e, 0x81, 0xf3, 0xd7, 0xfb , 0x7c, 0xe3, 0x39, 0x82, 0x9b, 0x2f,
                    0xff, 0x87, 0x34, 0x8e, 0x43, 0x44, 0xc4, 0xde, 0xe9, 0xcb , 0x54,
                    0x7b, 0x94, 0x32, 0xa6, 0xc2, 0x23, 0x3d, 0xee, 0x4c, 0x95, 0x0b,
                    0x42, 0xfa, 0xc3, 0x4e , 0x08, 0x2e, 0xa1, 0x66, 0x28, 0xd9, 0x24,
                    0xb2, 0x76, 0x5b, 0xa2, 0x49, 0x6d, 0x8b, 0xd1, 0x25 , 0x72, 0xf8,
                    0xf6, 0x64, 0x86, 0x68, 0x98, 0x16, 0xd4, 0xa4, 0x5c, 0xcc, 0x5d,
                    0x65, 0xb6, 0x92 , 0x6c, 0x70, 0x48, 0x50, 0xfd, 0xed, 0xb9, 0xda,
                    0x5e, 0x15, 0x46, 0x57, 0xa7, 0x8d, 0x9d, 0x84 , 0x90, 0xd8, 0xab,
                    0x00, 0x8c, 0xbc, 0xd3, 0x0a, 0xf7, 0xe4, 0x58, 0x05, 0xb8, 0xb3,
                    0x45, 0x06 , 0xd0, 0x2c, 0x1e, 0x8f, 0xca, 0x3f, 0x0f, 0x02, 0xc1,
                    0xaf, 0xbd, 0x03, 0x01, 0x13, 0x8a, 0x6b , 0x3a, 0x91, 0x11, 0x41,
                    0x4f, 0x67, 0xdc, 0xea, 0x97, 0xf2, 0xcf, 0xce, 0xf0, 0xb4, 0xe6,
                    0x73 , 0x96, 0xac, 0x74, 0x22, 0xe7, 0xad, 0x35, 0x85, 0xe2, 0xf9,
                    0x37, 0xe8, 0x1c, 0x75, 0xdf, 0x6e , 0x47, 0xf1, 0x1a, 0x71, 0x1d,
                    0x29, 0xc5, 0x89, 0x6f, 0xb7, 0x62, 0x0e, 0xaa, 0x18, 0xbe, 0x1b ,
```

```
                0xfc, 0x56, 0x3e, 0x4b, 0xc6, 0xd2, 0x79, 0x20, 0x9a, 0xdb, 0xc0,
                0xfe, 0x78, 0xcd, 0x5a, 0xf4 , 0x1f, 0xdd, 0xa8, 0x33, 0x88, 0x07,
                0xc7, 0x31, 0xb1, 0x12, 0x10, 0x59, 0x27, 0x80, 0xec, 0x5f , 0x60,
                0x51, 0x7f, 0xa9, 0x19, 0xb5, 0x4a, 0x0d, 0x2d, 0xe5, 0x7a, 0x9f,
                0x93, 0xc9, 0x9c, 0xef , 0xa0, 0xe0, 0x3b, 0x4d, 0xae, 0x2a, 0xf5,
                0xb0, 0xc8, 0xeb, 0xbb, 0x3c, 0x83, 0x53, 0x99, 0x61 , 0x17, 0x2b,
                0x04, 0x7e, 0xba, 0x77, 0xd6, 0x26, 0xe1, 0x69, 0x14, 0x63, 0x55,
                0x21, 0x0c, 0x7d],dtype=np.uint8)
                                # this representation is in hexadecimal format so while
printing it, we wil get corresponding integers



# most of the lambdas and short methods are here


character_conversion=np.vectorize(chr)      # this is to conver a numpy uint8 array
to its unicode containing numpy array


inverse_byte_substitution= np.vectorize(lambda index: r_s_box[index])
                                # a copy of state_array or array for which subtitution is
required should be passed in it.


# all the definitions/methods are here


def encrypted_text_read(filename):


    state_array= np.loadtxt(filename,dtype=str,delimiter=" ",encoding="utf-8")[:-
1].astype(np.uint8)
                                                # store them into utf-8 integer
encoding in a variable of 8-bit integer.
                                                # here is we have no need of padding
zeroes at the end because the
                                                # filtered input from loadtxt line of
code is already in multiple of 16.
    state_array= np.array([np.reshape(i,(4,4)) for i in
np.split(state_array,len(state_array)/16)])
    return state_array
```

```
def invshiftrows(element):
    temp = np.zeros(element.shape,dtype=np.uint8)
    for r in range(4):
        for j in range(Nb):
            temp[r,(encryp.shift(r,Nb)+j)%Nb]= element[r,j]
    element[:,:]=np.copy(temp)
                #  don't panic about assignment to value of 'element' reference as it is
working absloutely fine, moreover remember that
                # list are immutable and that they are called by reference in functions
and not called by value



#inverse_multiplication_for_matrix(inverse_mixcolumn_const,cipher_input) is the
prototype for this
def inverse_multiplication_for_matrix(X,Y):
    # iterate through rows of X
    rough= np.zeros((len(X),len(Y[0])),dtype= np.uint8)
    for i in range(len(X)):
        # iterate through columns of Y
        for j in range(len(Y[0])):
            # iterate through rows of Y
            for k in range(len(Y)):
                if X[i,k]==14:
                    a=b=c=Y[k,j]
                    a= encryp.lefShift_xor(a);a= encryp.lefShift_xor(a);a=
encryp.lefShift_xor(a)
                    b= encryp.lefShift_xor(b);b= encryp.lefShift_xor(b)
                    c= encryp.lefShift_xor(c)
                    rough[i,j] = rough[i,j]^(a^b^c)


                if X[i,k]== 9:
                    a=Y[k,j]
                    a= encryp.lefShift_xor(a);a= encryp.lefShift_xor(a);a=
encryp.lefShift_xor(a)
                    rough[i,j] = rough[i,j]^(a^Y[k,j])


                if X[i,k]== 13:
                    a=b=Y[k,j]
                    a= encryp.lefShift_xor(a);a= encryp.lefShift_xor(a);a=
encryp.lefShift_xor(a)
```

```
                b= encryp.lefShift_xor(b);b= encryp.lefShift_xor(b)
                rough[i,j] = rough[i,j]^(a^b^Y[k,j])


            if X[i,k]==11:
                a=b=Y[k,j]
                a= encryp.lefShift_xor(a);a= encryp.lefShift_xor(a);a=
encryp.lefShift_xor(a)
                b= encryp.lefShift_xor(b)
                rough[i,j] = rough[i,j]^(a^b^Y[k,j])


    Y[:,:]=rough[:,:]


# the final method to store the output of the decrypted array into a passed filename
def storeOutput(filename,state_array_out):
    char_written_length= 0
    with open(filename, 'w') as f:
        for i in state_array_out:
            rowwritten= ''.join(map(chr, np.ravel(i,order='F')))
            char_written_length=char_written_length+len(rowwritten)
            f.write(rowwritten)
            #print(rowwritten)  # only for debugging purpouse


    return char_written_length




# only the block of cipher should be passed that need to be decrypted and not the
whole cipher,containing all the blocks of the cipher


def final_decryption(cipher_input,expanded_key):
    round_number= 10
    encryp.add_round_key(cipher_input,expanded_key[round_number,:,:])
                        # this has been done because 0 round_key should be added
before any processing of the input cipher
    for round_number in range(9,0,-1):
        invshiftrows(cipher_input)
        cipher_input= inverse_byte_substitution(cipher_input)
        encryp.add_round_key(cipher_input,expanded_key[round_number,:,:])
        inverse_multiplication_for_matrix(inverse_mixcolumn_const,cipher_input)
```

```
                    # one more thing that has been done here is that passing only
the necessary part of the
                    # expanded_key with no passing of round number is needed
then(also round_number variable
                    # should also be used in passing the expanded_key as follows
expanded_key[round_number,:,:])


                    # above is the process for the round 9 to round 1


    round_number= 0
    invshiftrows(cipher_input)
    cipher_input= inverse_byte_substitution(cipher_input)
    encryp.add_round_key(cipher_input,expanded_key[round_number,:,:])
                    # above is the process for the round 10 only


    return cipher_input




# code for creation of state array and performing encryption on all blocks of the
cipher created
def creation_everything():
    filename= input("enter the name of the file with path that need to be decrypted
")
    encryption_key= input("enter the name of the key file with path that need to
decrypt the file, max length of file is 16 characters ")
    OutFileName= input("enter the name of the output file with path that is used to
store the decrypted output ")
    state_array= encrypted_text_read(filename)
    original_key= encryp.input_text_key(encryption_key)
    expanded_key= encryp.keyexpansion(np.copy(original_key))
    state_array_out= np.zeros(state_array.shape,dtype=np.uint8)



    for index,block in enumerate(state_array):
        state_array_out[index]= final_decryption(np.copy(block),expanded_key)


    total_char_wrote= storeOutput(OutFileName,np.copy(state_array_out))
```

```
    # print(state_array_out)  # for debugging purouse only
    # print(encryp.input_text("Text_File.txt"))  # for debugging purouse only


    #return np.array_equal(encryp.input_text("Text_File.txt"),state_array_out),
total_char_wrote
    return total_char_wrote




if __name__=='__main__':


    total_char_wrote= creation_everything()
    print("total characters wrote is ",total_char_wrote)
```

Encryption:

python /home/bhanu/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts/AES_Encryption.py.

Decryption:

python /home/bhanu/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts/AES_Decryption.py.

Chapter 7

Results and Discussion

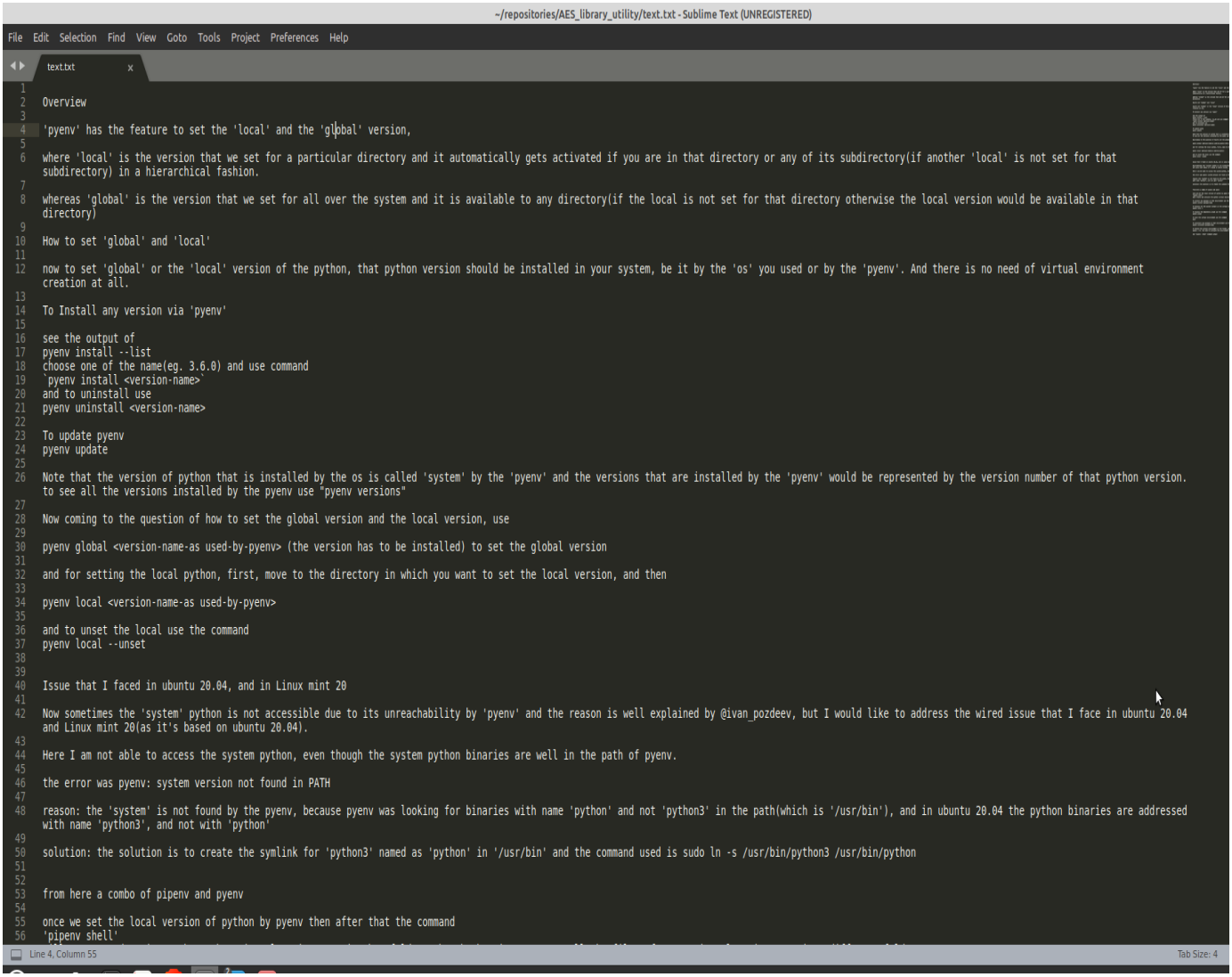
- Output and ScreenShots

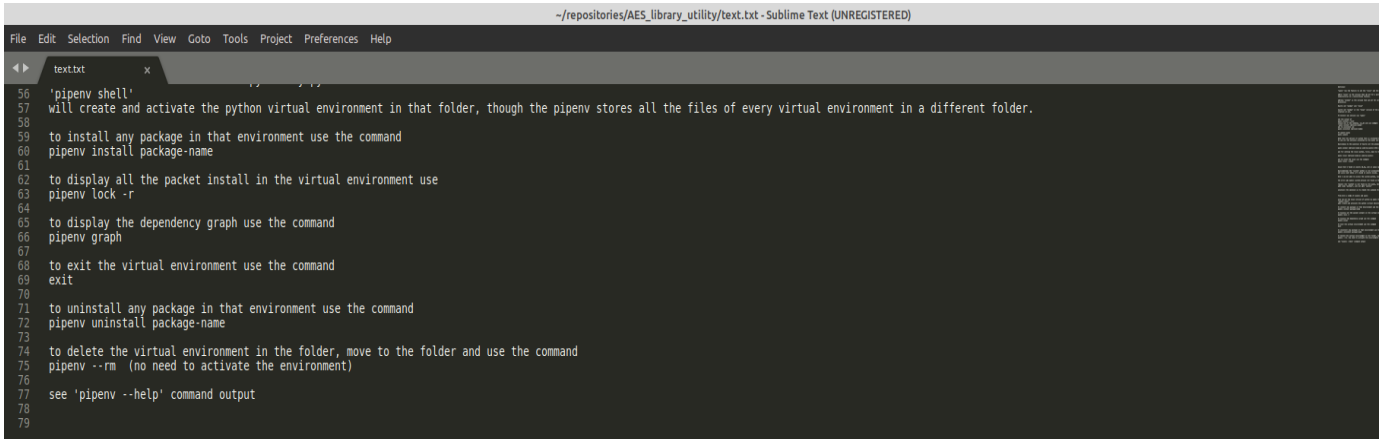
Results and Discussion

Outputs and Screenshots:

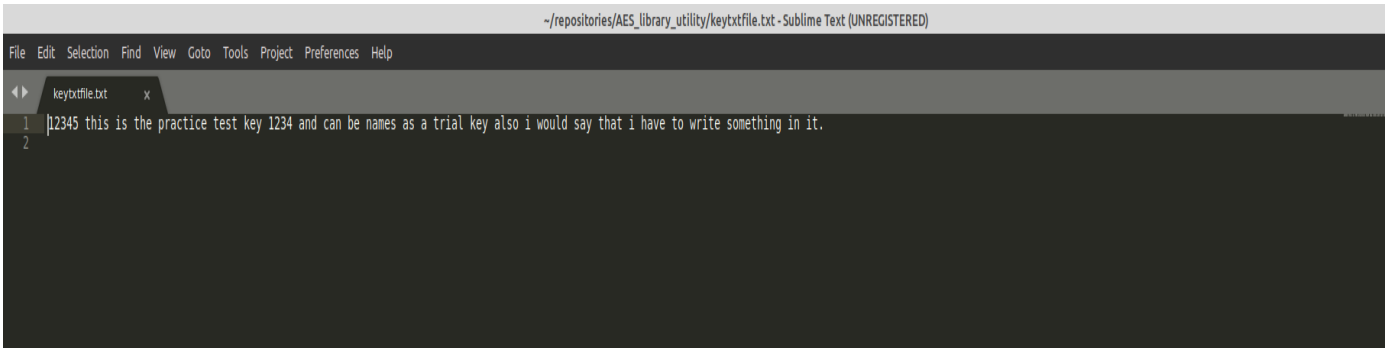
below are the outputs and the screenshots of the project:

Text File use for encryption process:

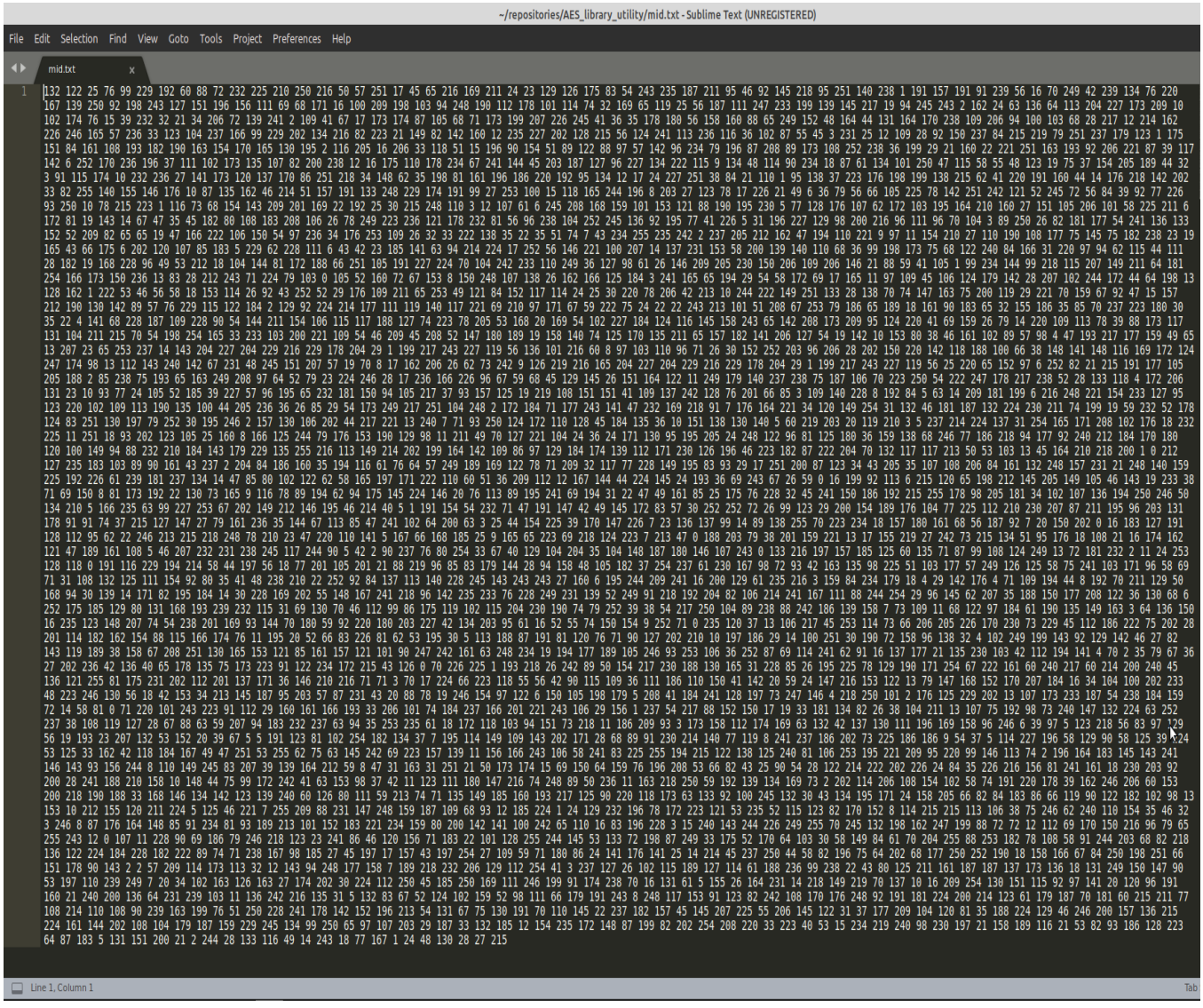




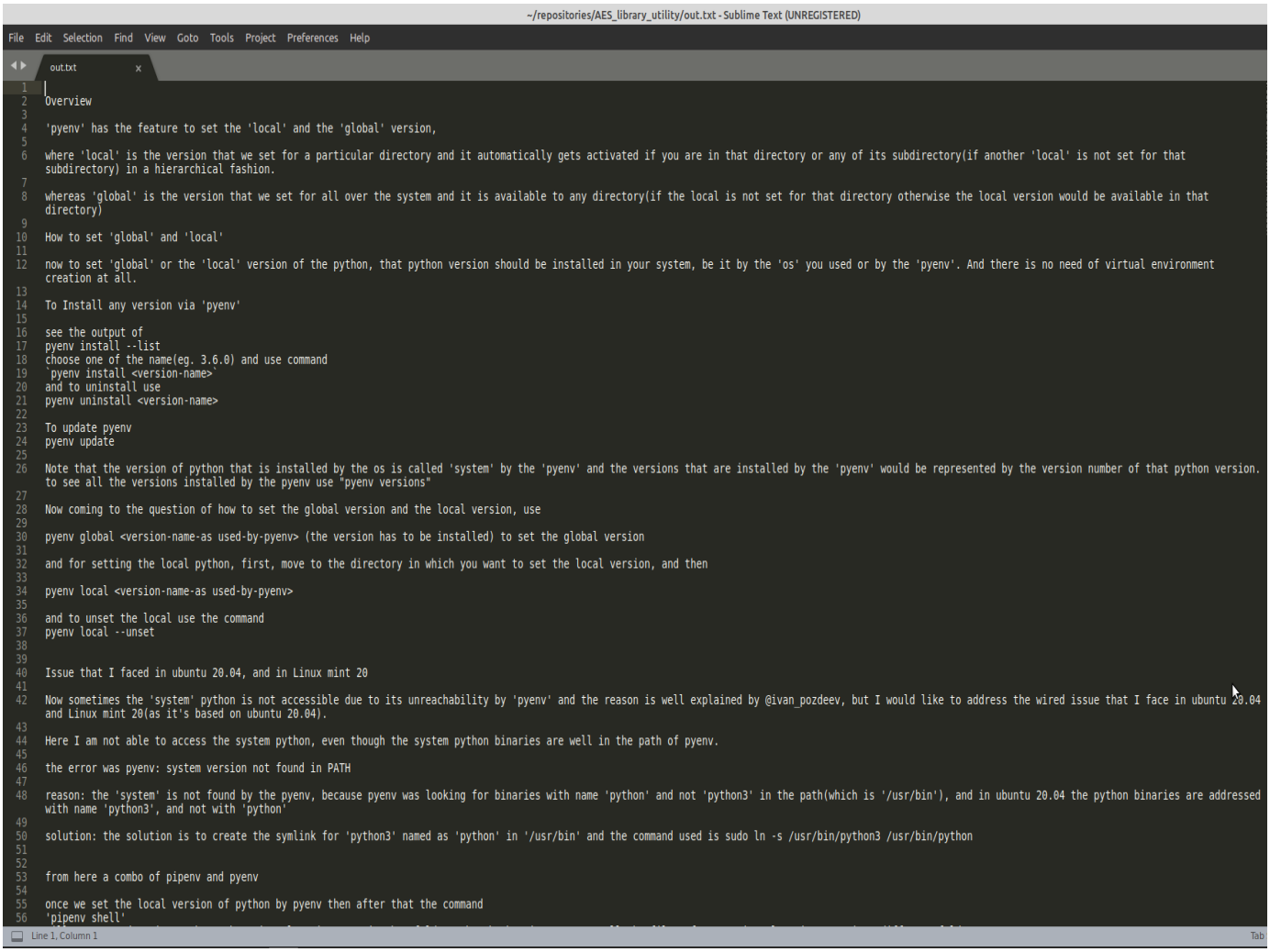
Key File use for encryption and Decryption process:

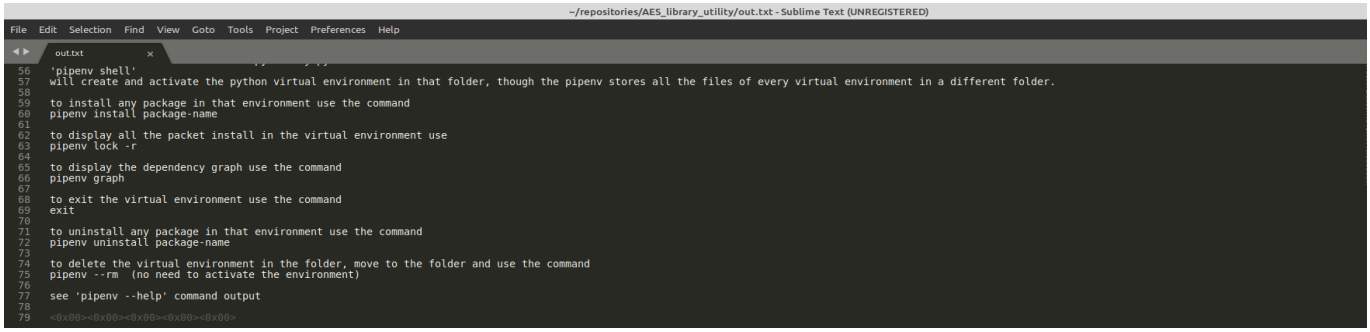


Encrypted Text File after encryption process:



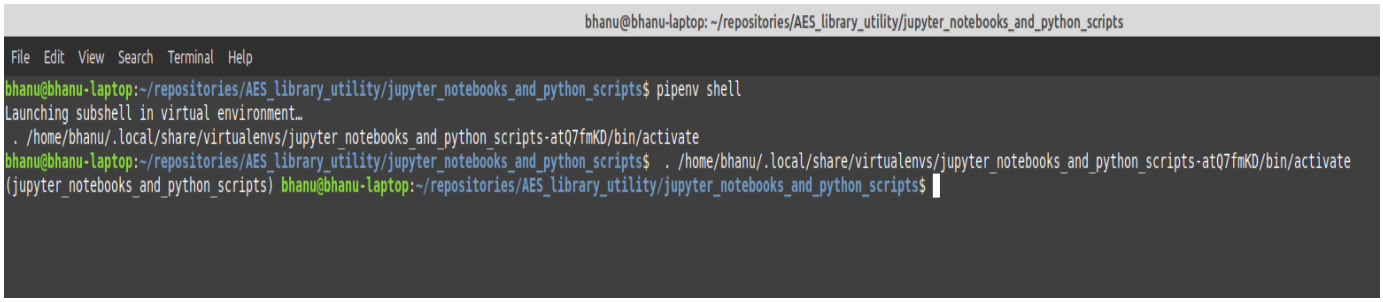
Decrypted Text File after Decryption process:



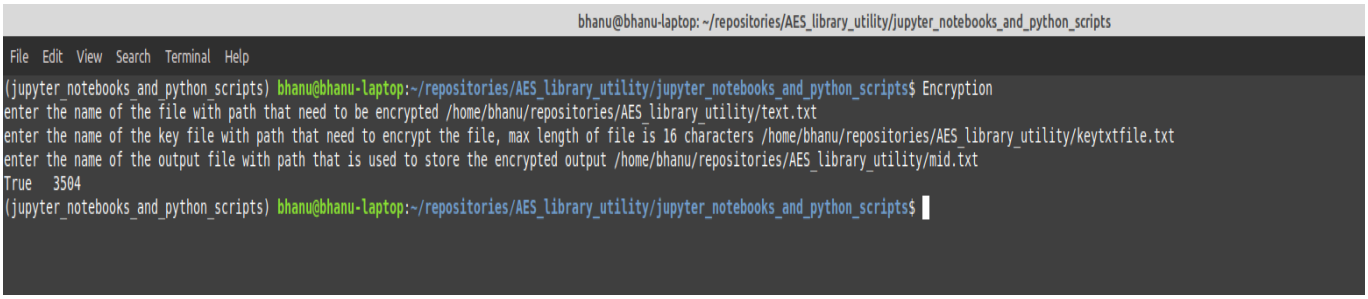


Process for Encryption :

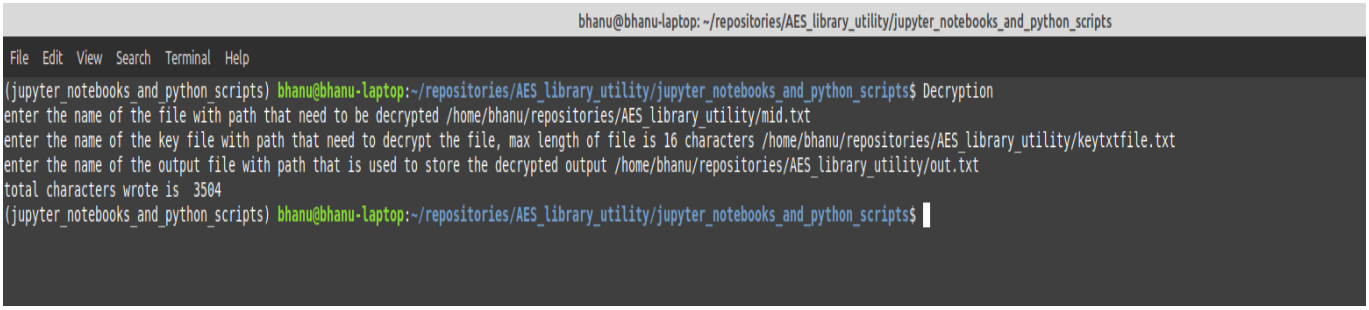
Activating the python virtual environment:



Performing Encryption:



Performing Decryption:



Chapter 8

Testing

- Testing Plan
- Testing Strategy
- Testing Methods
- Test Cases

Testing

Various parameters like implementation environment, program modules and coding standards are explained in previous chapter while this chapter is aimed to provide brief account of testing the software.

There are two principal motives of testing the software

- To rectify the error in execution
- To check the viability of software

The testing ensures that the software is according to the required specification standards and performs the task meant for it. The testing is done by our in house employee that act as novice user and test the application with all possible way to find the bugs and error as well as check validation.

1. TESTING PLAN

Testing is carried out at the following three stages :

- Design
- Implementation
- Coding

1.1 Design Testing:

The design errors are to be rectified at the initial stage. Such errors are very difficult to repair after the execution of software.

1.2 Implementation Testing:

The errors occurred at this stage can't be overlooked because such errors do not allow the further process.

1.3 Coding Testing:

The coding procedure plays significant role in software designing. The improper coding of any software can generate inconsistent results. Such errors may occur due to incorrect syntax or false logic. If the errors at coding stage remain unnoticed may give rise to grave failure of the system.

2. TESTING STRATEGY

A strategy for software testing integrates software test case design method into a well-planned series of steps that result in the successful construction of the software.

The strategy provides the roadmap that describes the steps to be conducted as a part of testing, then these steps are planned and then undertaken, and how much effort, time and resource will be required.

◆ We have tested our whole system using bottom up testing strategy.

◆ Bottom up testing involves integrating and testing the modules to the lower levels in the hierarchy, and then working up hierarchy of modules until the final module is tested.

◆ Bottom up testing strategy shows how actual testing is to be done with whole system but it does not show any detail about each module testing.

◆ When all modules are tested successfully then I will move to one step up and continue with white box testing strategy.

◆ When all modules will be tested successfully then I will integrate those modules and try to test integrated system using black box testing strategy.

Why Black Box Testing in my Project?

In my project whatever I have implemented was going to be tested by guide Mr. Rajesh Davda so there was a black box testing involve directly.

3. TESTING METHOD

3.1 Unit Testing

The unit testing is meant for testing smallest unit of software. There are two approaches namely bottom-up and top-down.

In bottom up approach the last module is tested and then moving towards the first module while top down approach reverses the action. In present work we opt for the first one.

The bottom up approach for the current project is carried out as shown in.

3.2 Integration Testing

The integration testing is meant to test all the modules simultaneously because it is possible that all the modules may function correctly when tested individually. But they may not work altogether and may lead to unexpected outcome.

3.3 Validation Testing

After the integration testing software is completely assembled as a package, interfacing error have been uncovered and corrected, and then validation testing may begin. Validation can be defined in many ways but a simple definition is what a validation succeeds when software functions in a manner.

3.4 Storage Testing

The dataset of the system has to be stored on the hard disk. So the storage capacity of the hard disk should be enough to store all the data required for the efficient running of the software.

4. TEST CASES

4.1 Purpose

The purpose of this project is to use to generate a fast utility for encryption and decryption process of the text file which is easy to used and contain only the necessary features to make the utmost important work that is encryption and decryption faster.

Chapter 9

Limitations and Future Enhancement

- Limitations and Future Enhancement

1.1 LIMITATIONS:

This project has an assumption that is both the sender and receiver must have shared some secret information before imprisonment. Pure cryptography means that there is none prior information shared by two communication parties.

Technology constraint:

The problem encountered here is searching information about computer security through Data Encryption and Key Algorithm and another problem is since the secret key has to be sending to the receiver of the encrypted data, it is hard to securely pass the key over the network to the receiver.

Time constraint:

The time giving for the submission of this project work was not really enough for the researcher to extensively carry out more research on this work.

Financial constraint:

There was not enough money to extensively carry out this work

1.2 FUTURE ENHANCEMENT:

The project “Advanced Encryption system” is designed for many future additions so that any user requirements can be made easy. Though the system is working on various assumptions it can be modified easily to a kind of requirements.

Future enhancements are possible even in specific modules as entire systems are computerized and modifiable approach. The system is flexible enough to incorporate new database to existing one. Since the entire system is developed in a modular approach, modification if necessary can be done on specific module without distributing the system.

Existing system used 128-bit scheme. It can be further improved by increases **128 bits to 192 and 256-bit scheme**. System performance can be further increased by applying pipe lines stages in between modules.

Chapter 10

Conclusion and Discussion

- Self analysis and Project viabilities
- Problem encountered and possible solutions
- Summary of project

1. SELF ANALYSIS AND PROJECT VIABILITIES

This shows a simple approach for fake news detection using naive Bayes classifier. This approach was implemented as a software system and tested against a data set of Facebook news posts. We achieved classification accuracy of approximately 74% on the test set which is a decent result considering the relative simplicity of the model. These results may be improved in several ways, that are described in the article as well. Received results suggest, that fake news detection problem can be addressed with artificial intelligence methods.

2. PROBLEM ENCOUNTERED AND POSSIBLE SOLUTIONS:

2.1 Resource Availability:

An important part of checking the veracity of a specific claim is to evaluate the stance different news sources take towards the assertion. Automatic stance evaluation, i.e. stance detection, would arguably facilitate the process of fact checking.

2.2 Requirement Understanding:

Automatic fake news detection is a challenging problem in deception detection, and it has tremendous real-world political and social impacts. However, statistical approaches to combating fake news has been dramatically limited by the lack of labeled benchmark datasets.

2.3 Problem Encountered and Possible Solutions:

Problem:

Encryption and Decryption system based on AES 128 bit algorithm.

Solution:

To Encrypt and Decrypt the text file using the AES 128 bit algorithm.

3. SUMMARY OF PROJECT

The scourge of cyberbullying has assumed alarming proportions with an ever-increasing number of adolescents admitting to having dealt with it either as a victim or as a bystander.

Anonymity and the lack of meaningful supervision in the electronic medium are two factors that have exacerbated this social menace.

Digital insecurity is a phenomenon which is having a significant impact on our social life, in particular in the political world. Encryption system is an emerging research area which is gaining interest but involved some challenges due to the limited amount of resources available.

We propose in this paper, an Encryption and Decryption system that use AES 128 bit algorithm technique. We investigate and compare other GUI based systems for encryption with our command line based Encryption system.

References:

Abdullah, A. M., & Aziz, R. H. H. (2016, June). New Approaches to Encrypt and Decrypt Data

in Image using Cryptography and Steganography Algorithm., International Journal of Computer

Applications, Vol. 143, No.4 (pp. 11-17).

Singh, G. (2013). A study of encryption algorithms (RSA, DES, 3DES and AES) for information

security. International Journal of Computer Applications, 67(19).

Gaj, K., & Chodowiec, P. (2001, April). Fast implementation and fair comparison of the final

candidates for Advanced Encryption Standard using Field Programmable Gate Arrays. In

Cryptographers' Track at the RSA Conference (pp. 84-99). Springer Berlin Heidelberg.

Stallings, W. (2006). Cryptography and network security: principles and practices. Pearson

Education India.

Yenuguvanilanka, J., & Elkeelany, O. (2008, April). Performance evaluation of hardware models

of Advanced Encryption Standard (AES) algorithm. In Southeastcon, 2008. IEEE (pp. 222-225).

Lu, C. C., & Tseng, S. Y. (2002). Integrated design of AES (Advanced Encryption Standard)

encrypter and decrypter. In Application-Specific Systems, Architectures and Processors, 2002.

Proceedings. The IEEE International Conference on (pp. 277-285).

Mohamed, A. A., & Madian, A. H. (2010, December). A Modified Rijndael Algorithm and it's

Implementation using FPGA. In Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE

International Conference on (pp. 335-338).

Deshpande, H. S., Karande, K. J., & Mulani, A. O. (2014, April). Efficient implementation of

AES algorithm on FPGA. In Communications and Signal Processing (ICCSP), 2014 IEEE

International Conference on (pp. 1895-1899).

https://rushter.com/blog/python-strings-and-memory/

https://en.wikipedia.org/wiki/Finite_field_arithmetic#Multiplication

https://www.slideshare.net/hisunilkumarr/advanced-encryption-st