

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	
Implementa-tion						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 understand 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 a 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 identified those risks, analyze their occurrences and effects on our project and must prevent them from occurring.

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 a 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.

◆ FEASIBILITY 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**

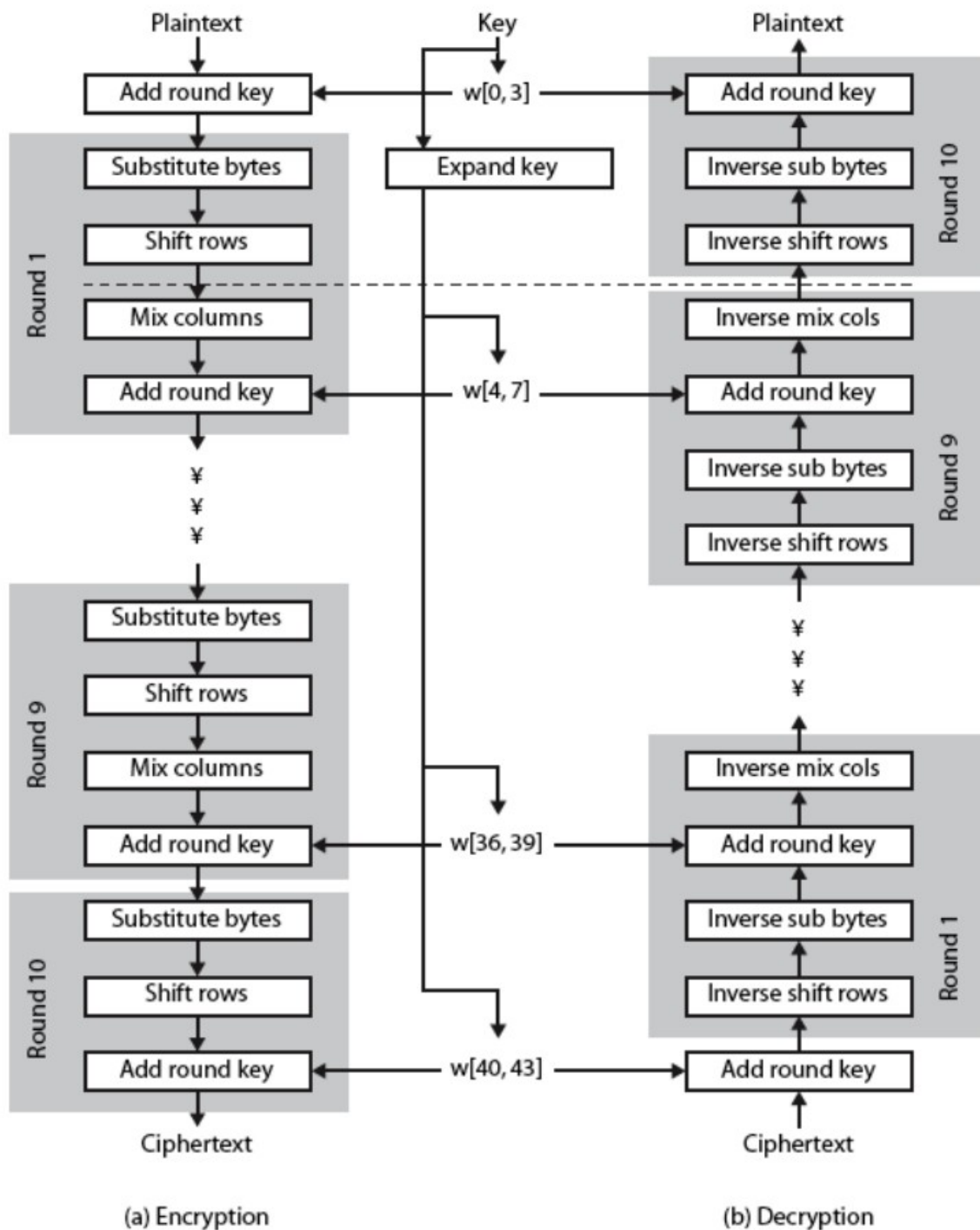
The requirements are given in verifiable manner (e.g.: Using quantifiable measures) to reduce disputes between client and developer.

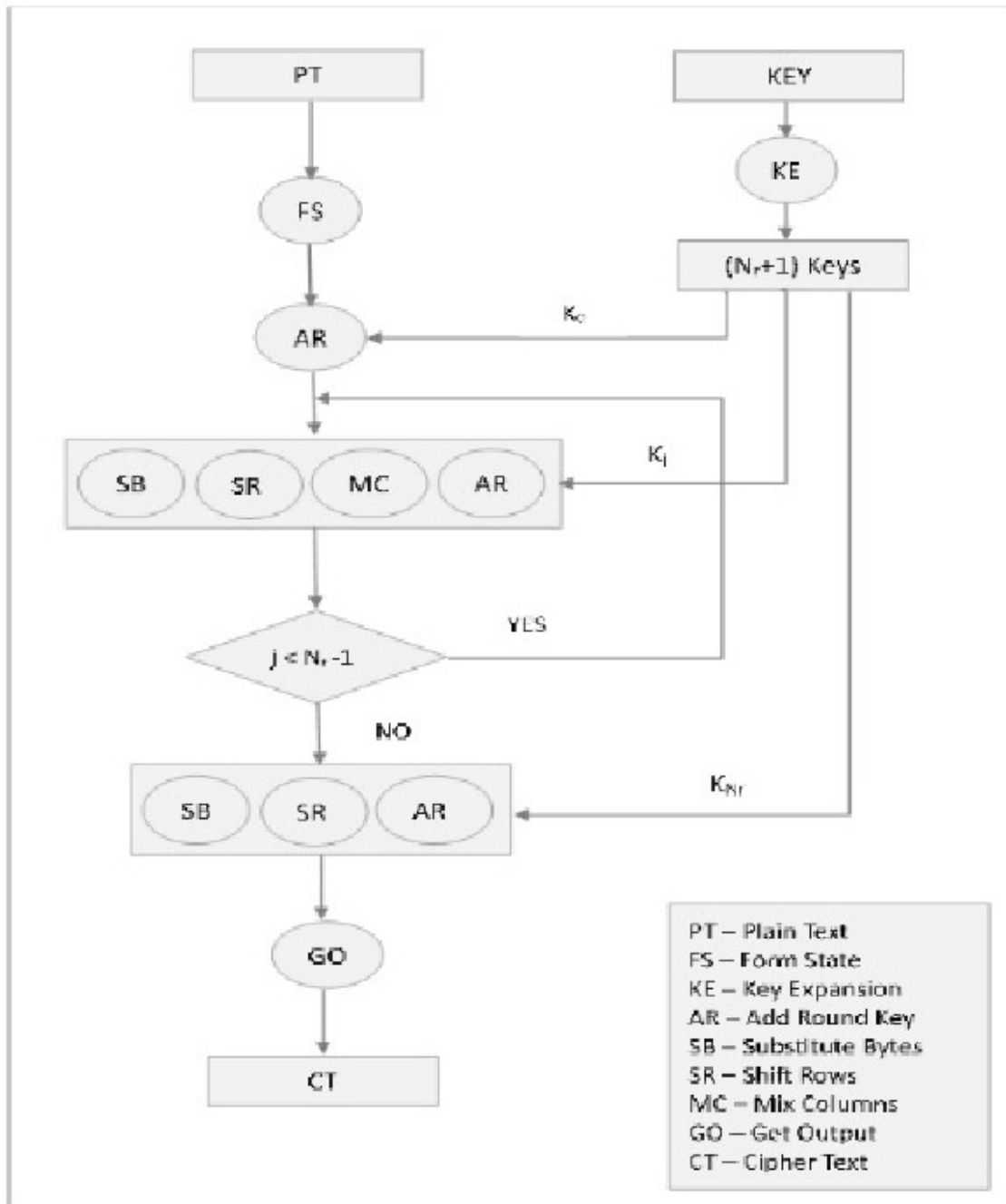
6. FEATURES OF NEW SYSTEM

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:



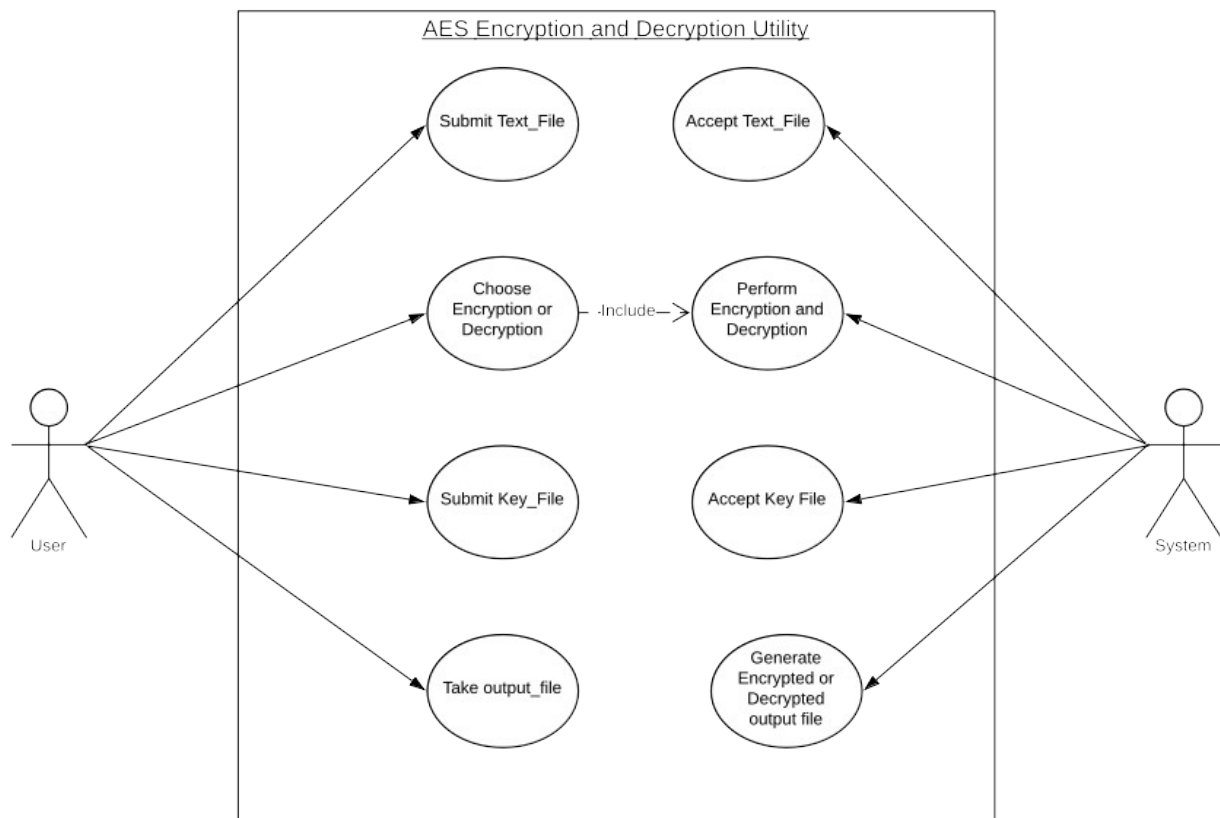


Data Flow Diagram

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

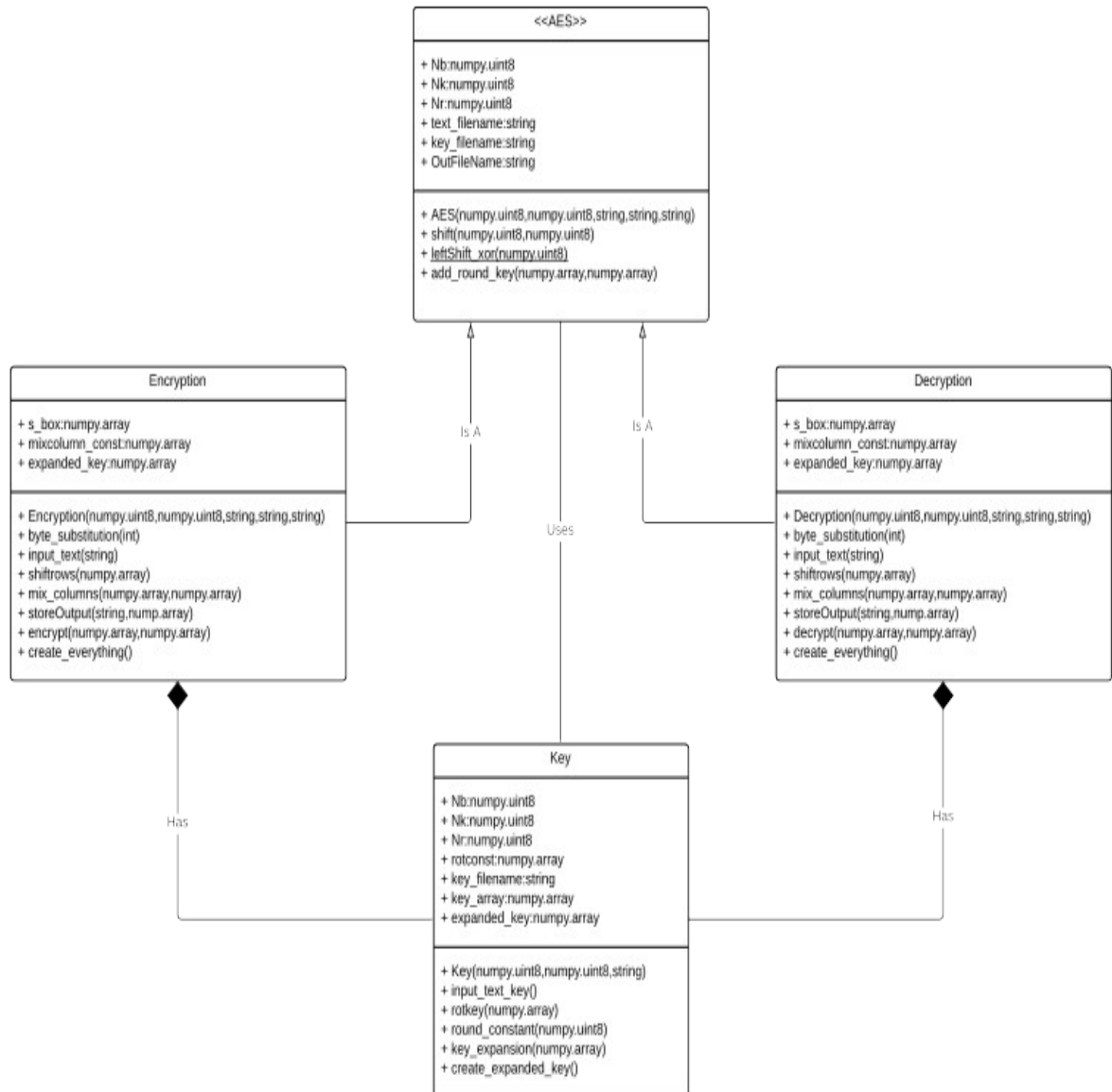


Use Case Diagram 1

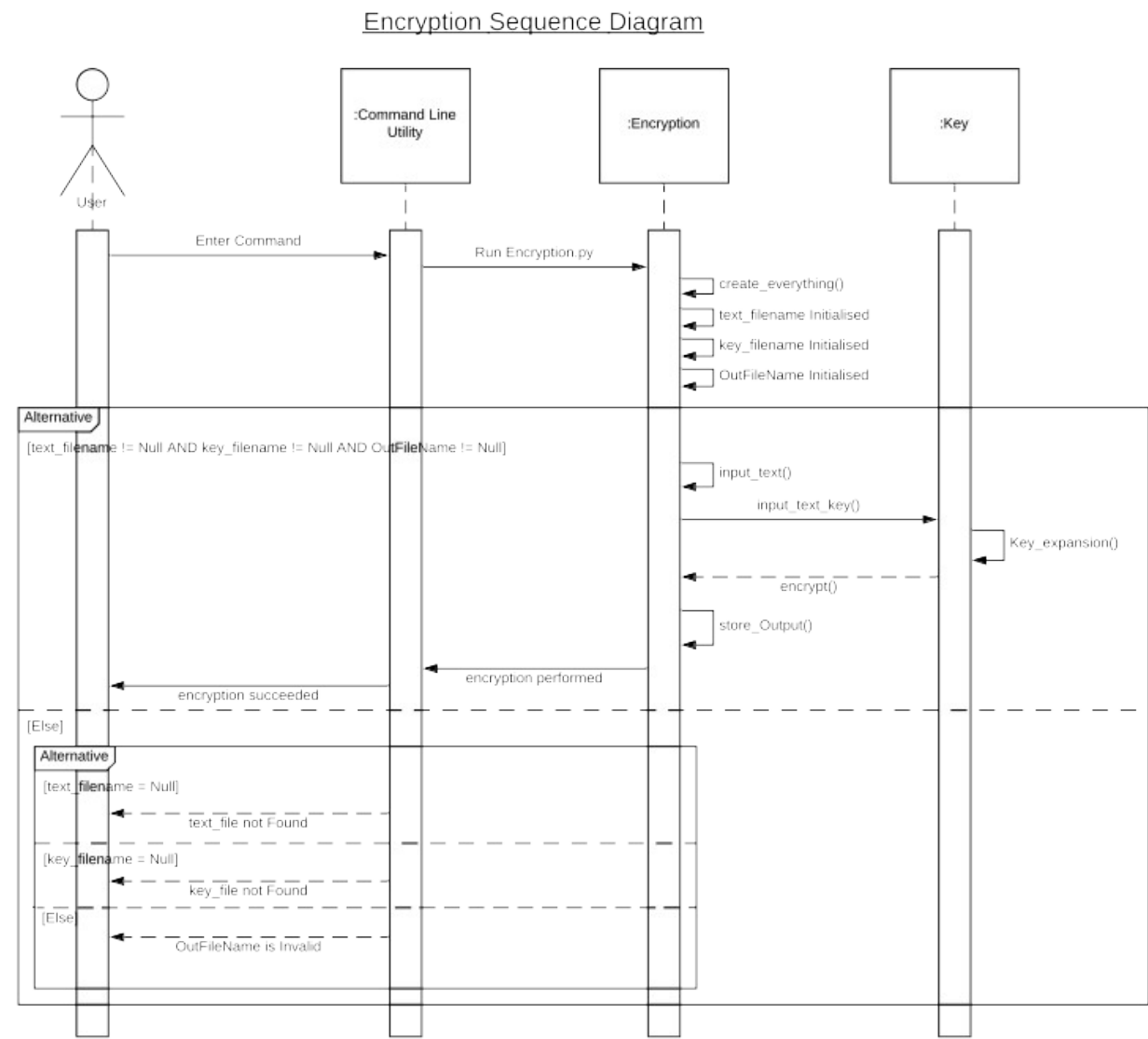
This use case diagram tell the various work that should be done by the user and the software admin, and it tells that how these two entities are related with each other in the software.

9. Class Diagram

Class Diagram

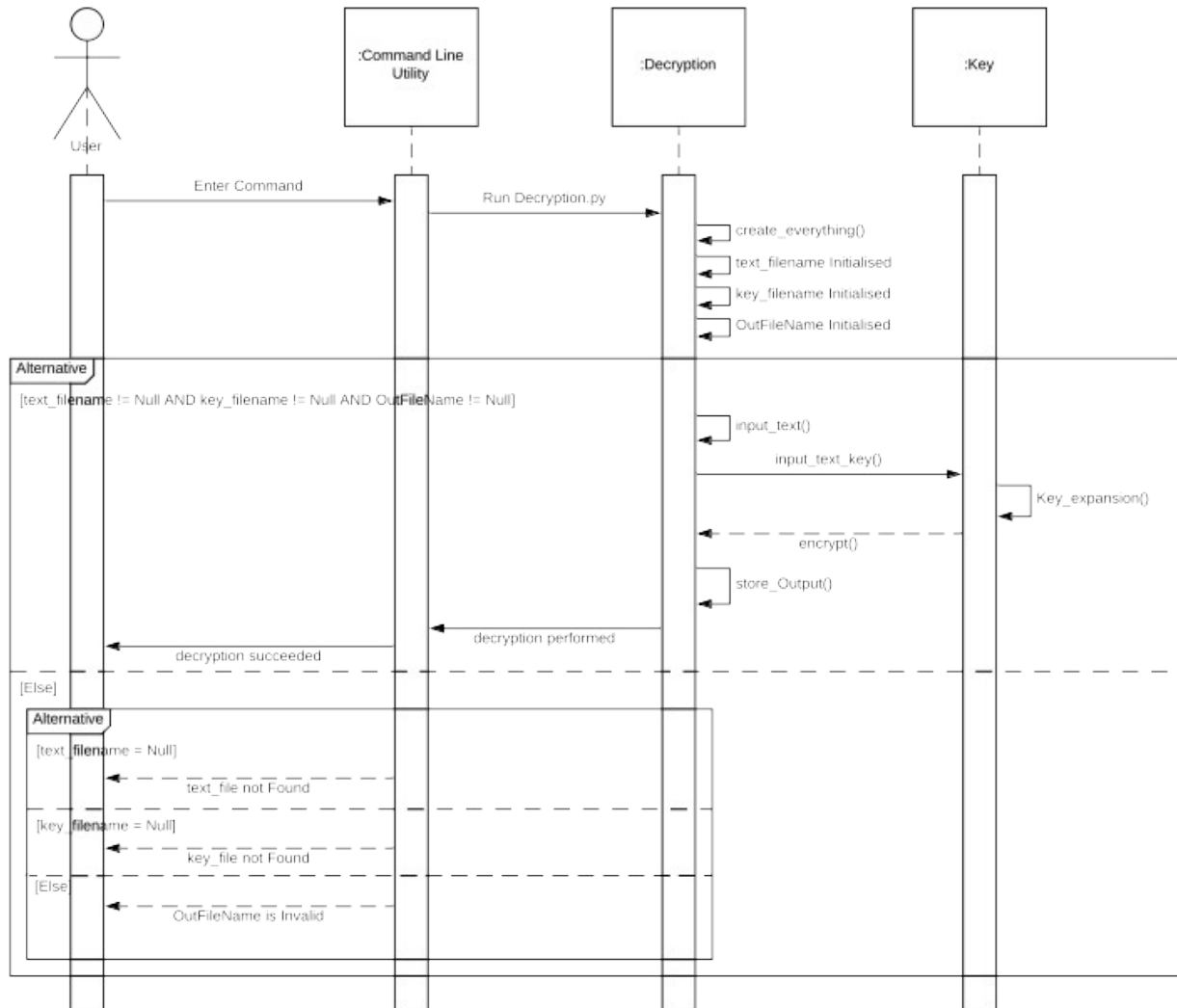


10. Sequence Diagram for Encryption



11. Sequence Diagram for Decryption

Decryption Sequence Diagram



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 its uses, here the 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 the 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 wants the decrypted or encrypted file output respectively.
3. Once the process is completed then the user can take its output file as the encrypted or the decrypted file whatever be the case.

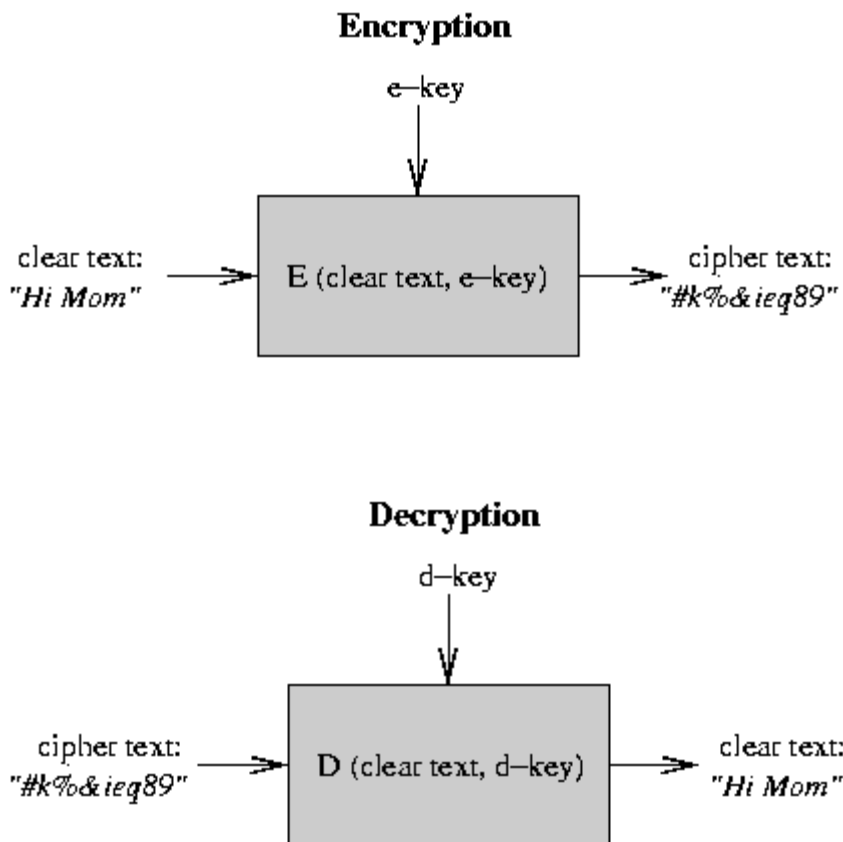


Figure Layered Architecture

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 generated.

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.

```
~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts/Aes_encryption.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Aes_encryption.py x
1 import numpy as np
2 import math
3 import Aes_decryption as decrypt
4 # To use other file variable use the way mentioned below rather than anyother way else
5 # import file1
6
7
8 Nb=Nk=np.uint8(4) # here Nb is the number of columns(32 bit words) in the state array and Nk is the number of columns
9 # (32 bit words) in key array, Nk could be 4,6,8 but for this case it is 4
10
11 Nr=np.uint8(10) # Nr is the number of rounds which is a functon of Nk and Nb (which is fixed). for this standard Nr = 10
12
13 mixcolumn_const=np.array([2,3,1,1,1,2,3,1,1,1,2,3,3,1,1,2],dtype=np.uint8)
14 mixcolumn_const=np.reshape(mixcolumn_const,(4,4))
15
16 rotconst=np.zeros((1,4,4),dtype=np.uint8)
17 for i in range(4): # this is the loop to conver the 'rotconst' into required rotation matrix
18     rotconst[0,i,(i+1)%4]=1
19     # as the pattern is always 1+i but when it(i+1) reaches the value of 4 it turns to 0 therefore
20     # use modulus by 4
21
22 # Sbox creation for the Gf(2^8)
23
24 s_box = np.array([0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67,
25 0x2b, 0xfe, 0xd7, 0xab, 0x76, 0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59,
26 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0, 0xb7,
27 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1,
28 0x71, 0xd8, 0x31, 0x15, 0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05,
29 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75, 0x09, 0x83,
30 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29,
31 0xe3, 0x2f, 0x84, 0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b,
32 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf, 0xd0, 0xef, 0xaa,
33 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c,
34 0x9f, 0xa8, 0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc,
35 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2, 0xcd, 0x0c, 0x13, 0xec,
36 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19,
37 0x73, 0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee,
38 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb, 0xe0, 0x32, 0x3a, 0x0a, 0x49,
39 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79,
40 0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4,
41 0xea, 0x65, 0x7a, 0xae, 0x88, 0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6,
42 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a, 0x70,
43 0x3e, 0xb5, 0x66, 0x48, 0x83, 0xf6, 0xbe, 0x61, 0x35, 0x57, 0xb9,
44 0x86, 0xc1, 0x1d, 0x9e, 0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e,
45 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf, 0x8c, 0xa1,
46 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x8f, 0xb8,
47 0x54, 0xbb, 0x16],dtype=np.uint8) # this representation is in hexadecimal format so while printing it, we wil get corresponding integers
48
49 # most of the lambdas and one liner functions are here
50
51
```

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 an 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 a command line encryptor and decryptor based on AES algorithm was a challenging problem due to the 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.

```
133 #-----
134 # Naive Bayes classifier for Multinomial model
135 #-----
136
137 clf = MultinomialNB()
138
139 clf.fit(tfidf_train, y_train)           # Fit Naive Bayes classifier according to X, y
140
141 pred = clf.predict(tfidf_test)          # Perform classification on an array of test vectors X.
142 score = metrics.accuracy_score(y_test, pred)
143 print("accuracy:  %0.3f" % score)
144 cm = metrics.confusion_matrix(y_test, pred, labels=['FAKE', 'REAL'])
145 plot_confusion_matrix(cm, classes=['FAKE', 'REAL'])
146 print(cm)
147
148
149 clf = MultinomialNB()
150
151 clf.fit(count_train, y_train)
152
153 pred = clf.predict(count_test)
154 score = metrics.accuracy_score(y_test, pred)
155 print("accuracy:  %0.3f" % score)
156 cm = metrics.confusion_matrix(y_test, pred, labels=['FAKE', 'REAL'])
157 plot_confusion_matrix(cm, classes=['FAKE', 'REAL'])
158 print(cm)
159
```

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 array 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 will 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 substitution 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()))

```

so store it result in a variable
`state_array= np.array(result, dtype=np.uint8)`
 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))`
 therefore the values of the
 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`

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

`def input_text_key(filename):`

`with open(filename,'r',encoding='utf-8') as f:`
`result=list(map(ord,f.read()))`
 so store it result in a variable
`state_array= np.array(result, dtype=np.uint8)`
 encoding in a variable of 8-bit integer.

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


```

# 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 because 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 absoulutely 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 array 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 function
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 will get corresponding integers

most of the lambdas and short methods are here

```

character_conversion=np.vectorize(chr)    # this is to convert 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 substitution 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 absolutely 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:

```
~/repositories/AES_library_utility/keytxtfile.txt - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

keytxtfile.txt x
1 |2345 this is the practice test key 1234 and can be names as a trial key also i would say that i have to write something in it.
2
```

Encrypted Text File after encryption process:

```
~/repositories/AES_library_utility/mid.txt - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

mid.txt x
1 |132 122 25 76 99 229 192 60 88 72 232 225 210 250 216 50 57 251 17 45 65 216 169 211 24 23 129 126 175 83 54 243 235 187 211 95 46 92 145 218 95 251 140 238 1 191 157 191 91 239 56 16 70 249 42 239 134 76 220
167 139 250 92 198 243 127 151 196 156 111 69 68 171 16 100 209 198 103 94 248 190 112 178 101 114 74 32 169 65 119 25 56 187 111 247 233 199 139 145 217 19 94 245 243 2 162 24 63 136 64 113 204 227 173 209 10
102 174 76 15 39 232 32 21 34 206 72 139 241 2 109 41 67 17 173 174 87 105 68 71 173 199 207 226 245 41 36 35 178 180 56 158 160 88 65 249 152 48 164 44 131 164 170 238 109 206 94 100 103 68 28 217 12 214 162
226 246 165 57 236 33 123 104 237 166 99 229 202 134 216 82 223 21 149 82 142 160 12 235 227 202 128 215 56 124 241 113 236 116 36 102 87 55 45 3 231 25 12 109 28 92 150 237 84 215 219 79 251 237 179 123 1 175
151 84 161 188 193 182 190 163 154 170 165 138 195 2 116 205 16 206 33 118 51 15 196 90 154 51 89 122 88 97 57 142 96 234 79 196 87 208 89 173 108 252 238 36 199 29 21 160 22 221 251 163 193 92 206 221 87 39 117
142 6 252 170 236 196 37 111 102 173 135 107 82 200 238 12 16 175 110 178 234 67 241 144 45 203 187 127 96 227 134 222 115 9 134 48 114 90 234 18 87 61 134 101 250 47 115 58 55 48 123 19 75 37 154 205 189 44 32
3 91 115 174 10 232 236 27 141 173 120 137 170 86 251 218 34 148 62 35 198 81 161 196 186 220 192 95 134 12 17 24 227 251 38 84 21 110 1 95 138 37 223 176 198 199 138 215 62 41 220 191 160 44 14 176 218 142 202
33 82 255 140 155 146 176 10 87 135 162 46 214 51 157 191 133 248 229 174 191 99 27 253 100 15 118 165 244 196 8 203 27 123 78 17 226 21 49 6 36 79 56 66 105 225 78 142 251 242 121 52 245 72 56 84 39 92 77 226
93 250 10 78 215 223 1 116 73 68 154 143 209 201 169 22 192 25 30 215 248 110 3 12 107 61 6 245 208 168 159 101 153 121 88 190 195 230 5 77 128 176 107 62 172 103 195 164 210 160 27 151 105 206 101 58 225 211 6
172 81 19 143 14 67 47 35 45 182 80 108 183 208 106 26 78 249 223 236 121 178 232 81 56 96 238 104 252 245 136 92 195 77 41 226 5 31 196 227 129 98 200 216 96 111 96 70 104 3 89 250 26 82 181 177 54 241 136 133
152 52 209 82 65 65 19 47 166 222 106 150 54 97 236 34 176 253 109 26 32 33 222 138 35 22 35 51 74 7 43 234 255 235 242 2 237 205 212 162 47 194 110 221 9 97 11 154 210 27 110 190 108 177 75 145 75 182 238 23 19
165 43 66 175 6 202 120 107 85 183 5 229 62 228 111 6 43 42 23 185 141 63 94 214 224 17 252 56 146 221 100 207 14 137 231 153 58 200 139 140 110 68 36 99 198 173 75 68 122 240 84 166 31 220 97 94 62 115 44 111
28 182 19 168 228 96 49 53 212 18 104 144 81 172 188 66 251 105 191 227 224 70 104 242 233 110 249 36 127 98 61 26 146 209 205 230 150 206 109 206 146 21 88 59 41 105 1 99 234 144 99 218 115 207 149 211 64 181
254 166 173 150 236 13 83 28 212 243 71 224 79 103 0 105 52 160 72 67 153 8 150 248 107 138 26 162 166 125 184 3 241 165 65 194 29 54 58 172 69 17 165 11 97 109 45 106 124 179 142 28 207 102 244 172 44 64 198 13
128 162 1 222 53 46 56 58 18 153 114 26 92 43 252 52 29 176 109 211 65 253 49 121 84 152 117 114 24 25 30 220 78 206 42 213 10 244 222 149 251 133 28 138 70 74 147 163 75 200 119 29 221 70 159 67 92 47 15 157
212 190 130 142 89 57 76 229 115 122 184 2 129 92 224 214 177 111 119 140 117 221 69 210 97 171 67 59 222 75 24 22 22 243 213 101 51 208 67 253 79 186 65 189 18 161 90 183 65 32 155 186 35 85 70 237 223 180 30
35 22 4 184 228 187 109 228 98 54 144 211 154 106 115 117 188 127 74 223 78 205 53 168 20 169 54 102 227 184 124 116 145 158 243 65 142 208 173 209 95 124 220 41 69 159 26 79 14 220 109 113 78 39 88 173 117
131 201 215 70 54 198 254 165 33 233 103 280 221 109 54 46 209 45 208 52 147 180 189 19 150 140 74 125 170 135 211 65 157 182 141 206 127 54 19 142 10 153 80 38 46 161 102 89 57 98 4 47 193 217 177 159 49 65
13 207 23 65 253 72 14 143 204 127 209 216 229 178 204 29 1 199 217 243 227 119 56 101 216 60 8 97 103 110 96 71 26 30 152 252 203 96 206 28 202 150 228 142 118 180 100 66 38 148 141 148 116 169 107 172 124
247 174 98 13 112 143 240 142 67 231 48 245 151 207 57 19 70 8 17 162 206 26 62 73 242 9 126 219 216 165 204 227 224 229 216 229 178 204 29 1 199 217 243 227 119 56 25 220 65 152 97 6 252 82 21 215 191 177 105
205 188 2 85 238 75 193 65 163 249 208 97 64 52 79 23 224 246 28 17 236 166 226 96 59 68 45 129 145 26 151 164 122 11 249 170 247 237 238 75 187 106 70 223 250 54 222 247 178 217 238 52 28 133 110 4 172 206
131 23 10 93 77 24 105 52 185 39 227 57 96 195 65 232 181 150 94 105 217 37 93 157 125 19 219 108 151 151 41 109 137 242 128 176 201 66 85 3 109 140 228 8 192 84 5 63 14 209 181 199 6 216 248 221 154 233 127 95
123 220 180 173 190 135 100 44 205 236 36 26 85 29 54 173 249 217 251 104 240 2 172 184 71 177 243 141 47 232 169 218 91 7 176 164 221 34 120 149 254 31 132 46 181 187 132 224 230 211 74 199 19 59 232 52 178
124 83 251 130 197 79 252 30 195 246 2 157 130 186 202 44 217 221 13 240 7 71 93 250 124 172 110 128 45 184 135 36 10 151 138 130 140 5 60 219 203 20 119 210 3 5 237 214 224 137 31 254 165 171 208 102 176 18 232
225 11 251 18 93 202 123 105 25 160 8 166 125 244 79 176 153 190 129 98 11 211 49 70 127 221 104 24 36 24 171 130 95 195 205 24 248 122 96 81 125 180 36 159 138 68 246 77 186 218 94 177 92 240 212 180 1 180
100 180 149 88 232 210 184 143 179 229 135 255 216 113 149 214 202 199 164 142 109 86 97 129 184 139 112 171 230 126 196 46 223 182 87 222 204 70 132 117 117 213 50 53 103 13 45 164 210 218 200 1 0 212
127 235 183 103 209 101 43 237 2 204 84 186 160 35 194 116 61 76 64 57 249 189 169 122 78 71 209 32 117 77 228 149 195 83 93 29 17 251 200 87 123 34 43 205 35 107 100 206 84 161 132 248 157 231 21 240 140 159
225 192 226 61 239 181 237 134 14 47 85 80 102 122 62 58 165 197 171 222 110 60 11 56 209 112 12 167 144 44 224 145 24 193 36 69 243 67 26 59 0 16 199 92 113 6 215 120 65 198 212 145 205 149 105 46 143 19 233 38
71 69 150 8 81 173 192 22 130 73 105 9 116 78 89 194 62 94 175 145 224 146 20 76 113 89 195 241 69 194 31 22 47 49 161 85 25 175 76 228 32 45 241 150 186 192 215 255 178 98 205 181 34 102 107 136 194 250 246 50
134 210 5 166 235 63 99 207 253 67 202 149 212 146 195 46 214 40 5 1 191 154 54 232 71 47 192 142 49 145 172 83 57 30 252 257 72 26 99 123 29 200 154 189 176 104 77 225 112 210 230 207 87 211 195 96 203 131
178 91 91 74 37 215 127 147 27 79 161 236 35 144 67 113 85 47 241 102 64 200 63 3 25 44 154 225 39 170 147 226 7 23 136 137 99 14 89 138 255 76 223 234 18 157 180 161 68 56 187 92 7 20 150 202 0 16 183 127 191
128 102 92 62 22 246 213 215 218 248 78 210 23 47 220 110 141 5 167 66 160 185 25 9 165 65 223 69 85 179 144 28 94 158 48 105 182 37 254 237 61 230 167 98 72 93 42 163 135 98 225 51 170 18 108 21 16 174 162
121 47 189 161 108 5 46 207 232 231 238 245 117 244 90 5 42 2 90 237 76 80 254 33 67 40 129 104 204 35 104 148 187 180 146 107 243 0 133 216 197 157 185 125 60 135 71 87 99 100 124 249 13 72 181 232 2 11 24 253
128 108 0 191 116 229 194 214 58 44 197 56 187 77 201 105 201 21 88 219 96 85 83 179 144 28 94 158 48 105 182 37 254 237 61 230 167 98 72 93 42 163 135 98 225 51 170 18 108 21 16 174 162
71 31 108 132 125 111 154 92 80 35 41 48 238 210 22 252 92 84 137 113 140 228 245 143 243 243 27 160 6 195 244 209 241 16 200 129 61 235 216 3 159 84 234 179 18 4 29 142 176 4 71 109 194 44 8 192 70 211 129 50
160 94 30 139 14 171 82 195 184 14 30 228 169 202 55 148 167 241 218 96 142 235 233 76 228 249 231 139 52 249 91 210 182 204 82 106 214 241 167 111 80 244 254 29 96 145 62 207 35 108 150 177 208 122 36 130 68 6
252 175 185 129 80 131 168 193 239 232 115 31 69 130 70 46 112 99 86 175 119 102 115 204 30 190 74 79 252 39 38 54 217 250 104 09 238 08 242 186 139 158 7 73 109 11 60 122 197 184 61 129 145 149 163 3 64 136 150
16 235 123 140 207 74 54 238 201 169 93 144 70 180 59 92 220 180 203 227 42 134 203 65 41 16 52 55 74 150 154 9 252 71 0 235 120 37 13 106 217 45 253 114 73 66 206 205 226 170 230 73 229 45 112 106 222 75 202 28
201 114 182 162 154 88 115 166 174 76 11 195 20 52 66 83 226 81 62 53 195 30 5 113 180 87 191 81 120 76 71 90 127 202 210 10 197 186 29 14 180 251 30 190 72 150 96 138 32 4 102 249 109 143 92 129 142 46 27 82
193 119 189 38 158 67 208 251 130 165 153 121 65 161 157 121 101 90 247 242 161 63 248 234 19 194 177 189 105 246 83 253 106 36 252 87 69 114 241 62 91 16 137 177 21 135 230 103 42 112 194 141 4 70 2 35 79 67 36
27 202 236 42 136 40 65 170 135 75 173 222 91 122 234 172 215 43 126 0 70 226 225 1 193 218 26 242 80 50 154 217 220 138 198 165 31 228 85 26 195 225 78 129 190 171 254 67 222 161 60 240 217 60 214 200 240 45
136 121 255 81 175 231 202 112 201 137 171 36 146 210 216 71 71 3 70 17 224 66 223 110 55 56 42 90 115 100 36 111 186 110 150 41 142 20 59 24 147 216 153 122 13 79 147 168 152 170 207 184 16 34 104 100 202 233
48 223 246 130 56 18 42 153 34 213 145 117 97 50 237 57 87 231 43 20 80 78 19 246 154 97 122 6 150 105 198 170 5 200 41 184 241 120 197 73 247 146 4 210 250 102 2 176 125 229 202 13 107 173 233 187 54 230 184 159
72 14 50 81 0 71 220 101 243 223 91 122 250 160 161 166 193 33 206 101 74 184 237 163 201 221 243 106 29 156 1 237 54 217 80 152 150 17 19 33 181 134 82 26 30 104 211 13 107 75 192 98 73 240 147 132 224 63 252
237 30 100 119 127 26 67 68 63 59 207 94 183 232 237 63 94 35 253 235 61 18 172 118 166 204 151 73 218 11 186 209 93 3 173 150 112 174 169 63 132 42 137 130 111 106 169 158 96 246 6 39 97 5 123 218 56 83 97 129
56 19 193 33 207 132 153 152 20 39 67 5 5 191 123 81 102 254 182 134 37 7 195 114 149 109 143 202 171 28 60 89 91 230 214 140 77 119 6 241 237 186 202 73 225 180 186 9 54 37 5 114 227 196 30 129 90 58 125 93 224
53 125 33 162 42 110 184 167 49 47 251 53 255 62 75 63 145 242 69 223 157 139 11 156 160 243 106 50 241 63 225 255 194 215 122 138 125 240 81 106 253 195 221 209 95 220 99 146 113 74 2 190 164 103 145 143 241
146 143 93 156 244 0 110 149 245 63 207 39 139 164 212 59 8 47 31 163 31 251 21 50 173 174 8 19 69 150 64 159 76 190 280 53 66 82 43 25 90 54 28 122 214 222 202 226 24 84 35 226 216 156 81 241 161 10 230 203 92
200 28 148 210 150 10 146 44 75 99 127 242 41 63 153 98 37 42 11 123 111 160 147 216 74 248 89 50 230 11 163 218 250 59 192 139 134 169 73 2 202 214 206 100 154 102 58 74 191 220 178 39 102 246 206 60 183
200 218 190 188 33 160 146 134 142 123 139 340 60 126 80 111 59 213 74 71 135 149 185 106 193 217 125 90 220 118 173 63 133 92 100 245 132 36 43 134 195 171 24 158 205 65 81 73 86 66 119 90 122 102 182 98 13
153 10 212 155 120 214 224 5 125 46 221 7 255 209 88 231 147 248 159 187 109 68 93 12 124 182 132 196 78 172 223 121 53 235 52 115 123 82 170 152 8 114 215 216 113 106 38 75 246 62 140 110 154 35 46 32
3 246 8 07 176 164 148 85 91 234 81 93 189 213 101 152 183 221 234 159 80 200 142 141 100 242 65 110 16 139 62 83 196 228 3 15 240 143 244 226 249 255 70 245 132 198 162 247 199 88 72 72 12 112 69 170 250 216 90 79 65
255 243 12 0 107 11 228 90 69 186 79 246 218 123 23 241 86 46 120 156 71 183 22 181 128 254 145 53 133 72 198 87 249 33 175 52 170 64 103 30 58 149 84 61 104 255 88 253 182 177 136 136 18 131 249 150 147 90
136 122 224 104 228 182 222 89 74 71 238 167 98 185 27 45 197 17 157 43 197 254 27 109 59 71 180 86 24 141 176 141 25 14 214 45 237 250 44 58 82 196 75 64 202 68 177 250 252 198 15 158 166 67 84 250 198 251 66
151 178 90 143 2 2 57 209 114 173 113 32 12 143 94 248 177 158 7 189 218 232 206 129 112 254 41 3 237 127 26 102 115 189 127 114 61 188 236 99 238 22 43 80 125 211 61 187 187 137 173 136 18 131 249 150 147 90
53 197 110 239 249 7 20 34 102 163 126 163 27 174 202 30 124 112 250 45 185 250 169 111 246 199 91 174 238 70 16 131 61 5 155 26 164 231 14 218 149 219 70 137 10 16 209 254 130 151 115 92 97 147 18 120 96 21
160 21 240 200 136 64 231 238 103 11 136 242 216 135 31 5 232 83 67 52 124 102 159 52 98 111 66 179 191 243 8 188 117 153 91 123 82 242 180 178 176 248 92 191 181 242 200 214 123 61 173 178 181 60 215 211 77
108 214 110 108 90 239 163 199 76 51 250 228 241 178 142 152 196 213 54 131 67 75 130 191 70 110 145 22 237 182 157 45 145 207 225 55 206 145 122 31 37 177 209 184 120 81 35 108 224 129 46 
```

Decrypted Text File after Decryption process:

```
~/repositories/AES_library_utility/out.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
out.txt x
1 |
2 Overview
3
4 'pyenv' has the feature to set the 'local' and the 'global' version,
5
6 where 'local' is the version that we set for a particular directory and it automatically gets activated if you are in that directory or any of its subdirectory(if another 'local' is not set for that
7 subdirectory) in a hierarchical fashion.
8
9 whereas 'global' is the version that we set for all over the system and it is available to any directory(if the local is not set for that directory otherwise the local version would be available in that
10 directory)
11
12 How to set 'global' and 'local'
13
14 now to set 'global' or the 'local' version of the python, that python version should be installed in your system, be it by the 'os' you used or by the 'pyenv'. And there is no need of virtual environment
15 creation at all.
16
17 To Install any version via 'pyenv'
18
19 see the output of
20 pyenv install --list
21 choose one of the name(eg. 3.6.0) and use command
22 'pyenv install <version-name>'
23 and to uninstall use
24 pyenv uninstall <version-name>
25
26 To update pyenv
27 pyenv update
28
29 Note that the version of python that is installed by the os is called 'system' by the 'pyenv' and the versions that are installed by the 'pyenv' would be represented by the version number of that python version.
30 to see all the versions installed by the pyenv use 'pyenv versions'
31
32 Now coming to the question of how to set the global version and the local version, use
33
34 pyenv global <version-name-as used-by-pyenv> (the version has to be installed) to set the global version
35
36 and for setting the local python, first, move to the directory in which you want to set the local version, and then
37
38 pyenv local <version-name-as used-by-pyenv>
39
40 and to unset the local use the command
41 pyenv local --unset
42
43 Issue that I faced in ubuntu 20.04, and in Linux mint 20
44
45 Now sometimes the 'system' python is not accessible due to its unreachability by 'pyenv' and the reason is well explained by @ivan_pozdeev, but I would like to address the wired issue that I face in ubuntu 20.04
46 and Linux mint 20(as it's based on ubuntu 20.04).
47
48 Here I am not able to access the system python, even though the system python binaries are well in the path of pyenv.
49
50 the error was pyenv: system version not found in PATH
51
52 reason: the 'system' is not found by the pyenv, because pyenv was looking for binaries with name 'python' and not 'python3' in the path(which is '/usr/bin/'), and in ubuntu 20.04 the python binaries are addressed
53 with name 'python3', and not with 'python'
54
55 solution: the solution is to create the symlink for 'python3' named as 'python' in '/usr/bin/' and the command used is sudo ln -s /usr/bin/python3 /usr/bin/python
56
57 from here a combo of pipenv and pyenv
58
59 once we set the local version of python by pyenv then after that the command
60 'pipenv shell'
61
62 Line 1, Column 1 Tab
```

```
~/repositories/AES_library_utility/out.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
out.txt x
56 'pipenv shell'
57 will create and activate the python virtual environment in that folder, though the pipenv stores all the files of every virtual environment in a different folder.
58
59 to install any package in that environment use the command
60 pipenv install package-name
61
62 to display all the packet install in the virtual environment use
63 pipenv lock -r
64
65 to display the dependency graph use the command
66 pipenv graph
67
68 to exit the virtual environment use the command
69 exit
70
71 to uninstall any package in that environment use the command
72 pipenv uninstall package-name
73
74 to delete the virtual environment in the folder, move to the folder and use the command
75 pipenv --rm (no need to activate the environment)
76
77 see 'pipenv --help' command output
78
79
```

Process for Encryption :

Activating the python virtual environment:

```
bhanu@bhanu-laptop: ~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts
File Edit View Search Terminal Help
bhanu@bhanu-laptop:~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts$ pipenv shell
Launching subshell in virtual environment.
. /home/bhanu/.local/share/virtualenvs/jupyter_notebooks_and_python_scripts-atQ7fmKD/bin/activate
bhanu@bhanu-laptop:~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts$ . /home/bhanu/.local/share/virtualenvs/jupyter_notebooks_and_python_scripts-atQ7fmKD/bin/activate
(jupyter_notebooks_and_python_scripts) bhanu@bhanu-laptop:~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts$
```

Performing Encryption:

```
bhanu@bhanu-laptop: ~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts
File Edit View Search Terminal Help
(jupyter_notebooks_and_python_scripts) bhanu@bhanu-laptop:~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts$ Encryption
enter the name of the file with path that need to be encrypted /home/bhanu/repositories/AES_library_utility/text.txt
enter the name of the key file with path that need to encrypt the file, max length of file is 16 characters /home/bhanu/repositories/AES_library_utility/keytxtfile.txt
enter the name of the output file with path that is used to store the encrypted output /home/bhanu/repositories/AES_library_utility/mid.txt
True 3504
(jupyter_notebooks_and_python_scripts) bhanu@bhanu-laptop:~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts$
```

Performing Decryption:

```
bhanu@bhanu-laptop: ~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts
File Edit View Search Terminal Help
(jupyter_notebooks_and_python_scripts) bhanu@bhanu-laptop:~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts$ Decryption
enter the name of the file with path that need to be decrypted /home/bhanu/repositories/AES_library_utility/mid.txt
enter the name of the key file with path that need to decrypt the file, max length of file is 16 characters /home/bhanu/repositories/AES_library_utility/keytxtfile.txt
enter the name of the output file with path that is used to store the decrypted output /home/bhanu/repositories/AES_library_utility/out.txt
total characters wrote is 3504
(jupyter_notebooks_and_python_scripts) bhanu@bhanu-laptop:~/repositories/AES_library_utility/jupyter_notebooks_and_python_scripts$
```

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>