**Performance Analysis of Python AES module using 2k factorial design**

**Project By:**

Shuvo Saha Roy- SAU/CS(M)/2020/22

Mahen Mondol-SAU/CS(M)/2020/09



Submitted To:

[**Shachi Sharma**](http://www.sau.int/faculty/faculty-profile.html?staff_id=87)

Assistant Professor

South Asian University, New Delhi

**Submission Date:**

12-07-2021

**Abstract**

Nowadays information security has become an important issue for exchanging information in data communication. Encryption algorithm plays a vital role in information security system. Many algorithm technique are used to provide data confidentiality and privacy by making the information indecipherable which can be only be decoded or decrypted by party those possesses the associated key. But at the same time, for implementing these algorithm technique that consume a significant amount of computing resources such as CPU time, memory, and battery power. [1] So it is important to find out the effect of algorithm upon these resources. It will help us to determine significant factors that affecting the performance of the algorithm. This project provides evaluation of Python AES module from pycryptodome[2] package, a cryptographic algorithms by taking different types of files like mp3, text, image, py scripts, pdf, and video files. A comparison has been conducted among different resources using evaluation parameters such as encryption time, decryption time and memory throughput. Simulation results are given to demonstrate the effectiveness of each.

**Keywords:** **Algorithm, Encryption, Decryption, AES, Python**

pip install pycryptodome

**Introduction**

Data is the most important things in this era. The amount of data is increasing day by day with the use of internet. So it need to ensure the safety of data as efficient as possible. Different encryption technique is used to ensure the safety of data or secure the data. Now Advanced Encryption Standard AES encryption is the most popular encryption technique that is used all over the world. [3]

Here, in this project pycrypto, which is a python package used to analysis the performance of AES 256bit module. The implementation code is available here ([github](https://github.com/ssroy548/Performance-Modeling-/blob/main/script.py)).

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, 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.

**Advantage of python**

reduces the cost of program maintenance

supports modules and packages

availability of interpreter and the extensive standard library.

**AES**

AES (Advanced Encryption Standard) is a symmetric block cipher standardized by NIST . It has a fixed data block size of 16 bytes. Its keys can be 128, 192, or 256 bits long. AES is very fast and secure, and it is the de facto standard for symmetric encryption.

**Python Crpyto**

Python crypto mudule provides a simple interface to symmetric Gnu Privacy Guard encryption and decryption for one or more files on UNIX and Linux platforms. It runs on top of gpg and requires a gpg install on your system. Encryption is performed with the AES256 cipher algorithm. Benchmarks relative to default gpg settings are available for text and binary file mime types.

**Encryption**

Encryption is a process which transforms the original information into an unrecognizable form. This new form of the message is entirely different from the original message. That's why a hacker is not able to read the data as senders use an encryption algorithm. Encryption is usually done using key algorithms.

**Decryption**

Decryption is a process of converting encoded/encrypted data in a form that is readable and understood by a human or a computer. This method is performed by un-encrypting the text manually or by using keys used to encrypt the original data.

***1. Setting of Goal***

*Effect of performance of the program (total CPU utilization, Memory Utilization, time taken to complete operation, etc.) due to workload (type of file, size of file) while encryption and decryption*

***2. Factors affecting our program***

***Main factors that may affect my programs are:***

*⮚ Type of file used for encryption or the type of file which we are decrypted.*

*⮚ Size of file- Higher the file size, there is a higher chance of time and CPU utilization for operations or executions.*

*⮚ Nature of operations: Here, In this program, we can perform only two operations, i.e., encryption and decryption of file.*

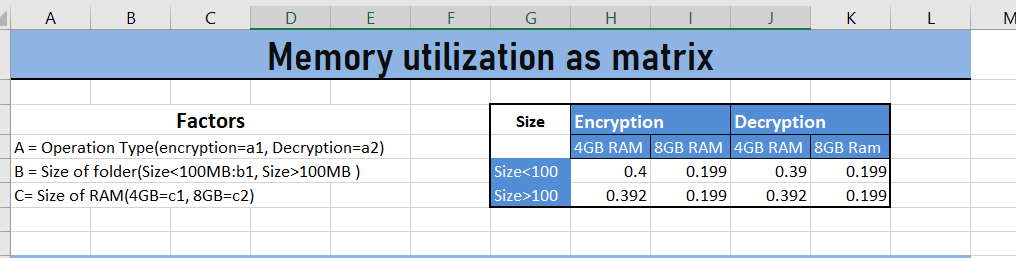
***3. Designing workload***

*Although we can use readymade image files and video files for workload, we have tried to create a word document of different sizes and directories of various sizes for the experiment. Here we assumed a word file and directory larger than 100 MB as a more significant workload and directory or file smaller than 100 MB.*

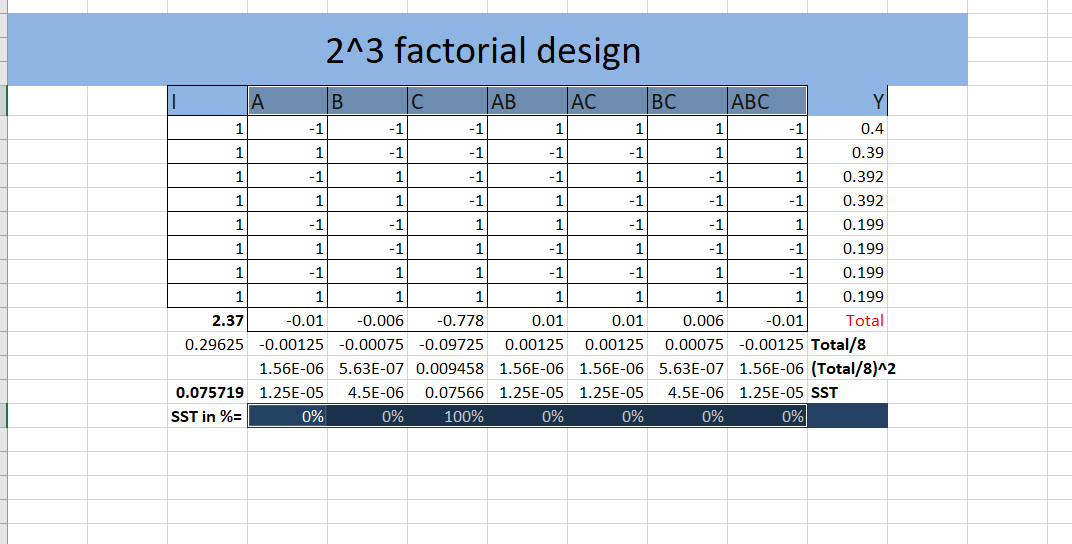
**Experimental Design of our program**

(Here for the experimental design, we choose two types of files (viz. Word Files and Folder) of two different sizes (such that one is more significant than 100 MB and another is smaller than 100 MB) for main two operation (*encryption and decryption* ) on both. Hence, we can use 23 factorial designs for experimental analysis)

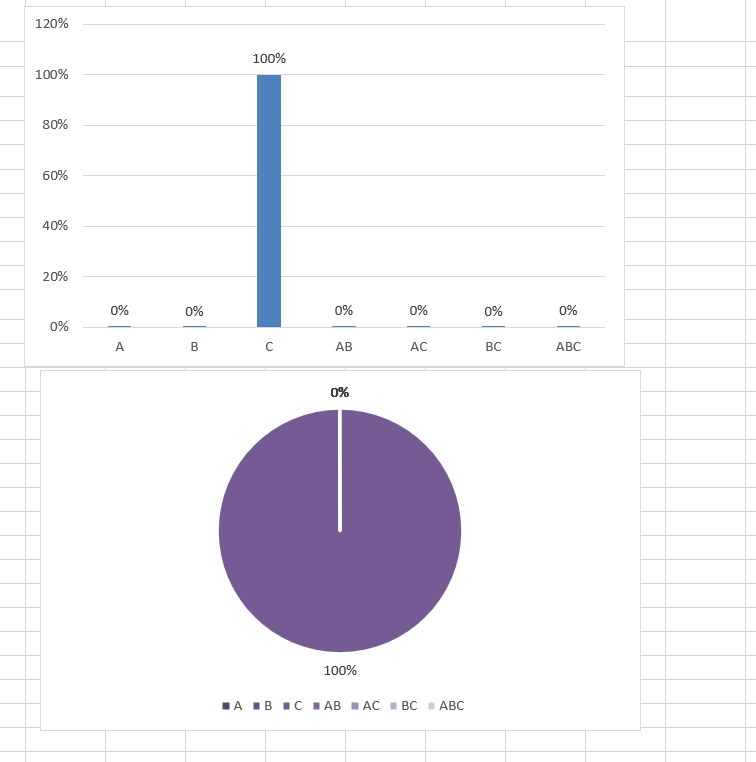
**Implementation in excel sheet (for memory utilization)**

****

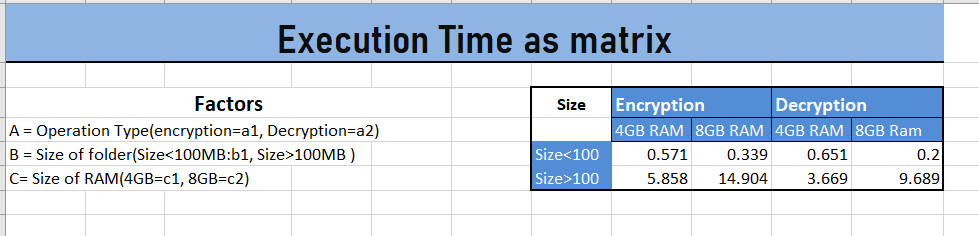
**Calculation using excel sheet**

****

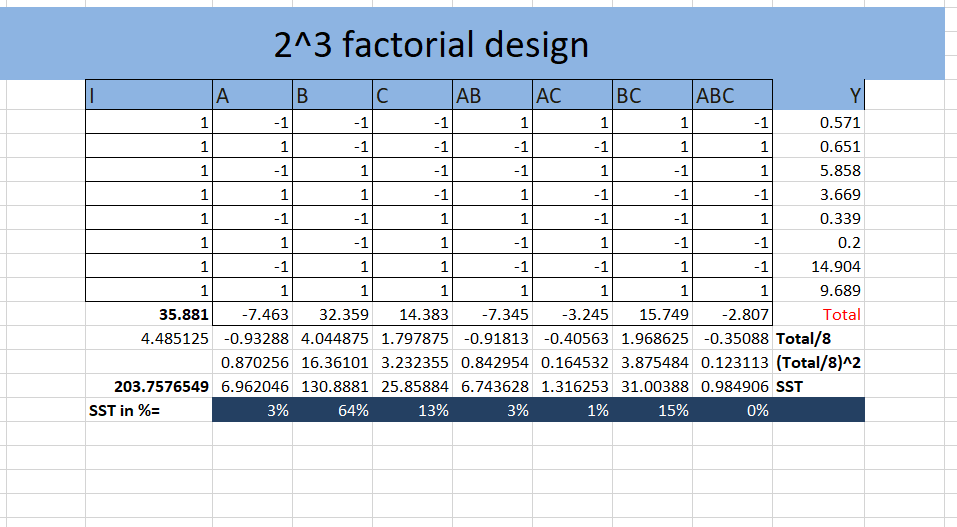
**Pie chart and line graph showing the effect of interaction:**

****

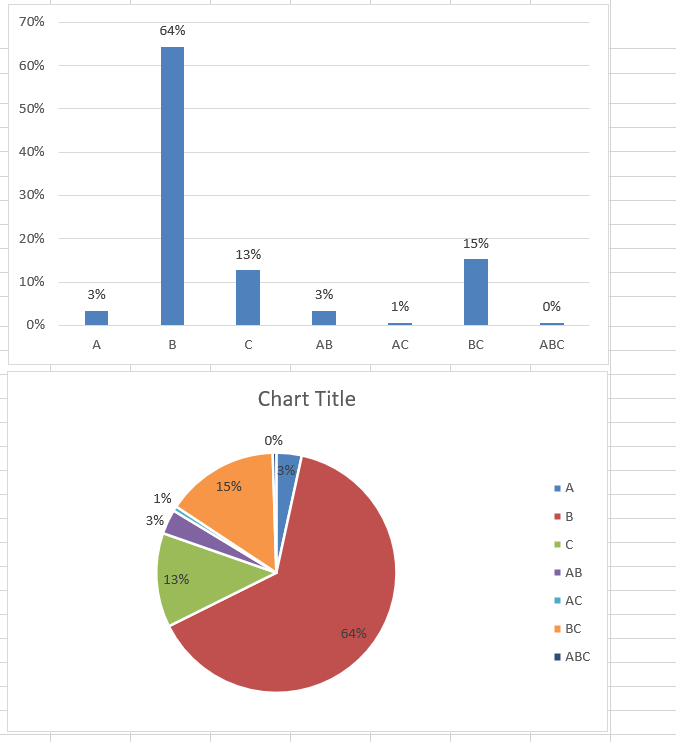
**2.Implementation in excel sheet (for time taken)**

****

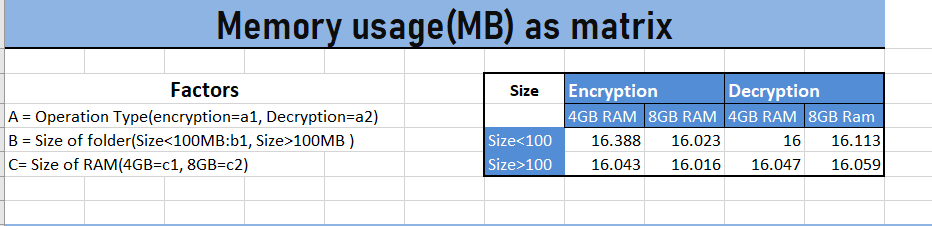
**Calculation**

****

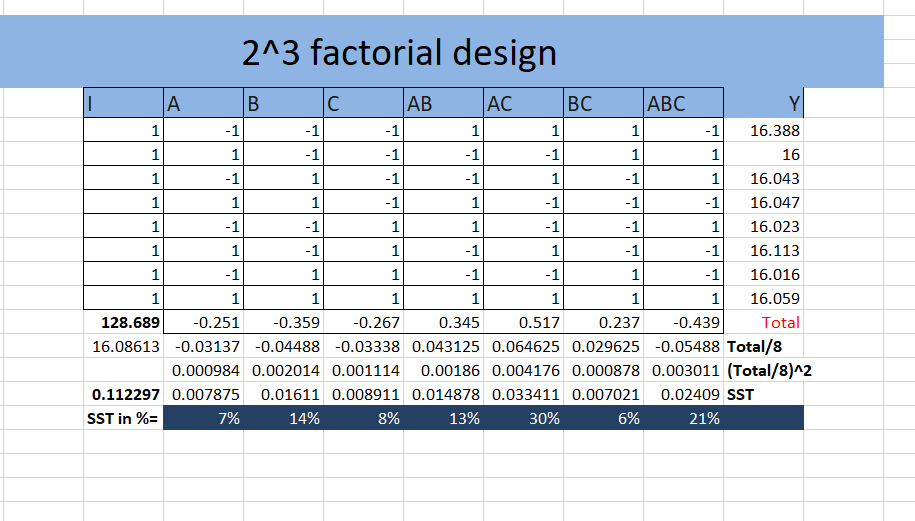
**Pie-chart and graph showing interaction:**

****

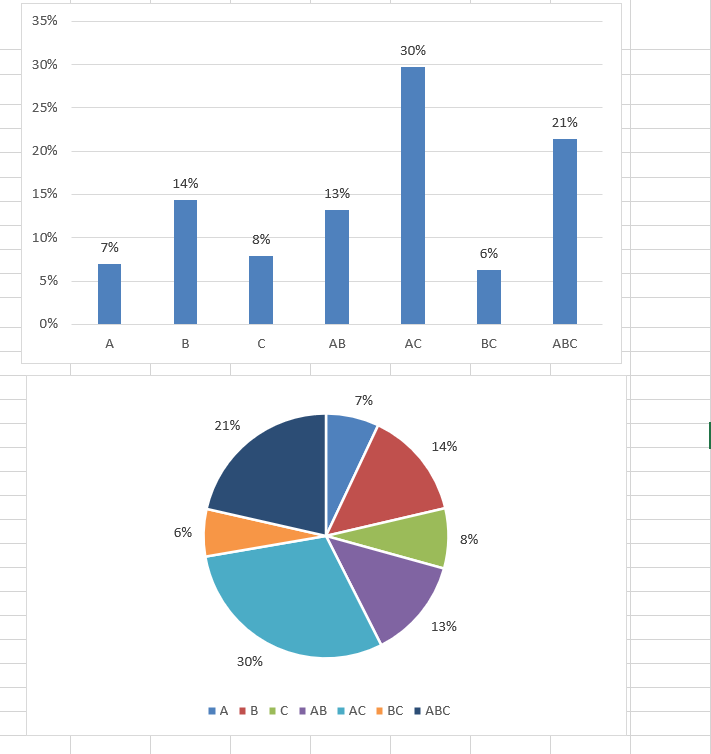
**c. Implementation in excel sheet (for Memory usage)**

****

**Calculation:**

****

**Pie chart and graph:(showing the effect of interaction.)**

****

**1.** **Conclusion**

**⮚**  When we used memory utilization as a metric for measuring our program’s performance, we found that interaction of A and B is the primary factor affecting the performance with about 13%.

⮚ When we used Total CPU usage as a metric for measuring our program’s performance, we found that interaction of A and B is the primary factor affecting the concert with about 3%.

⮚ But, when we used time taken in second as a metric for measuring our program’s performance, we found that factor C, i.e., Size of file, is the crucial factor affecting the performance.

**2.** **Key findings**

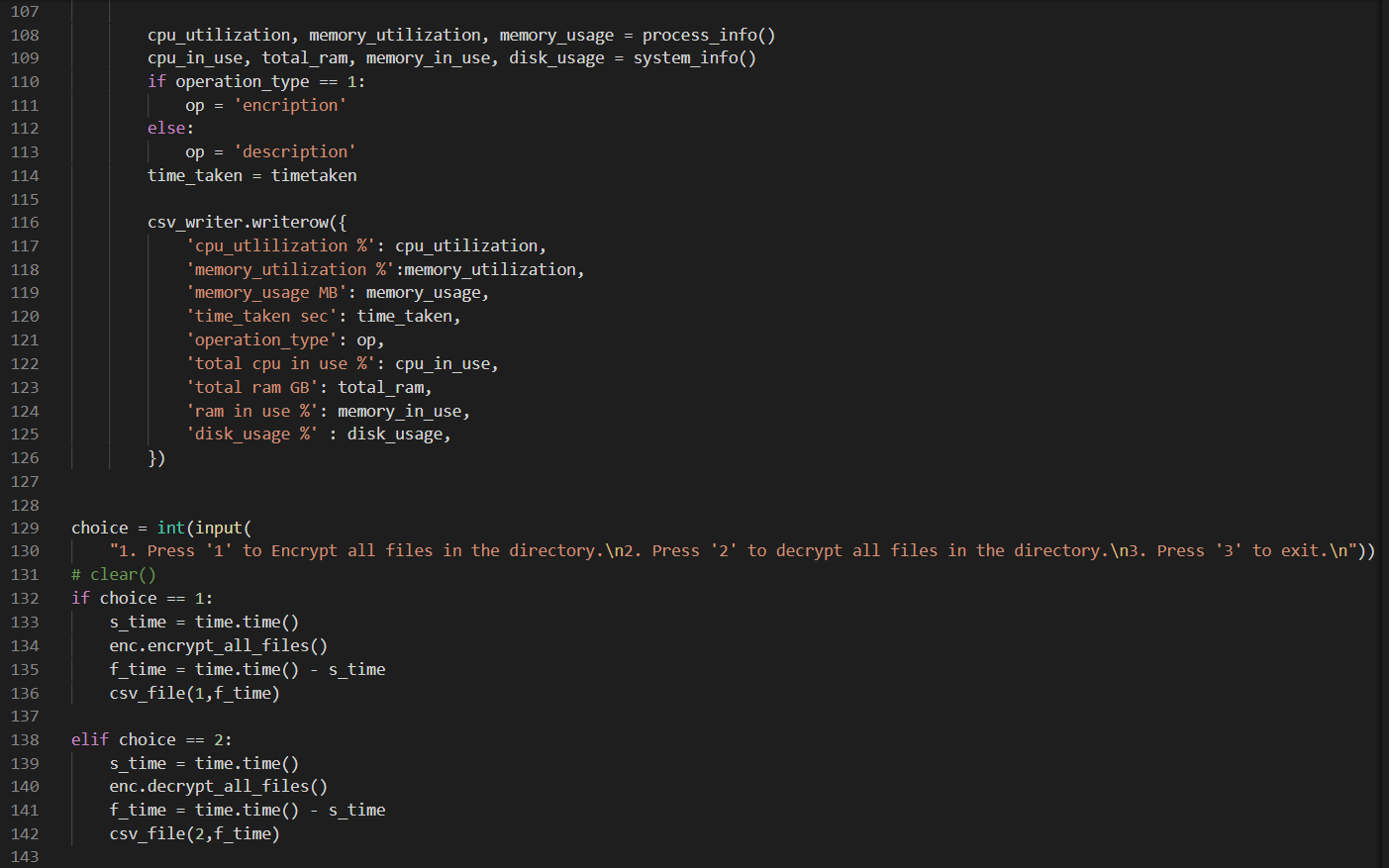
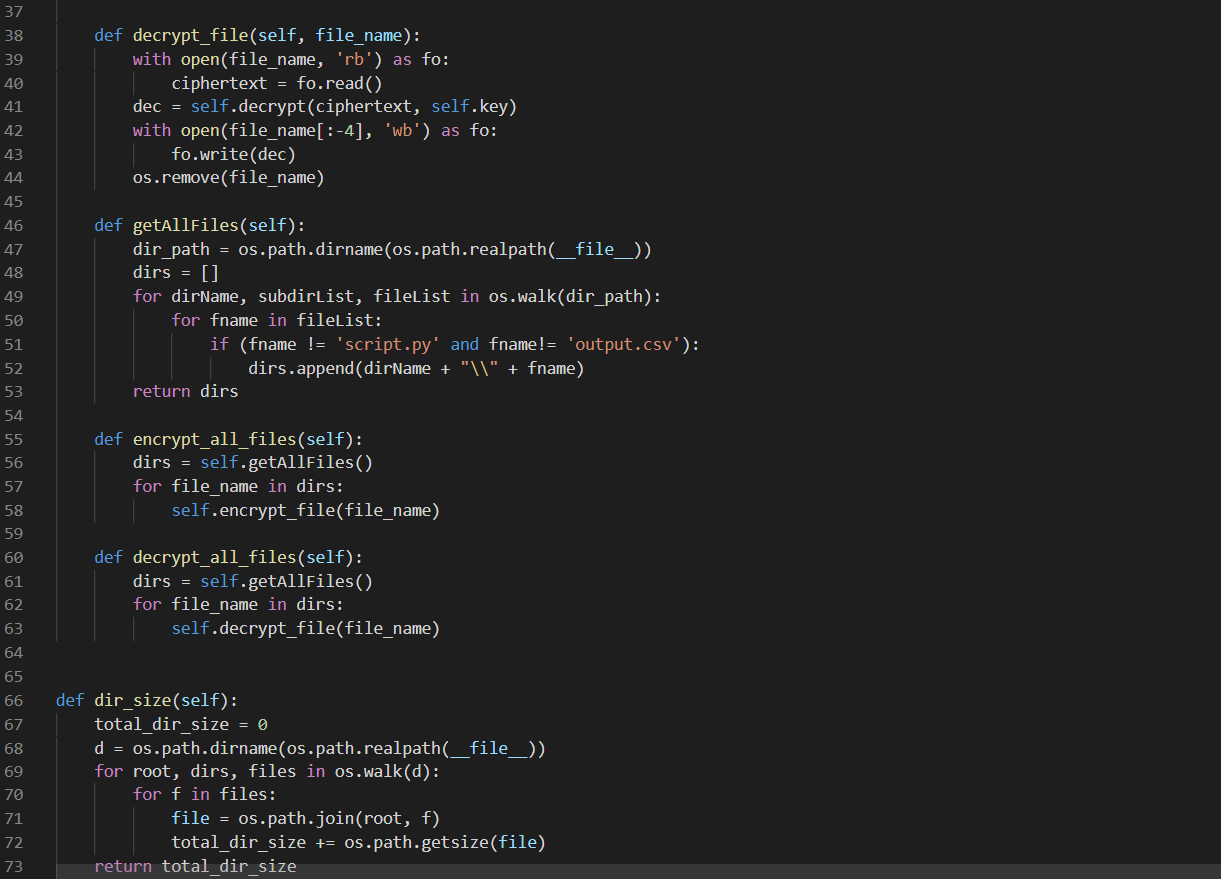
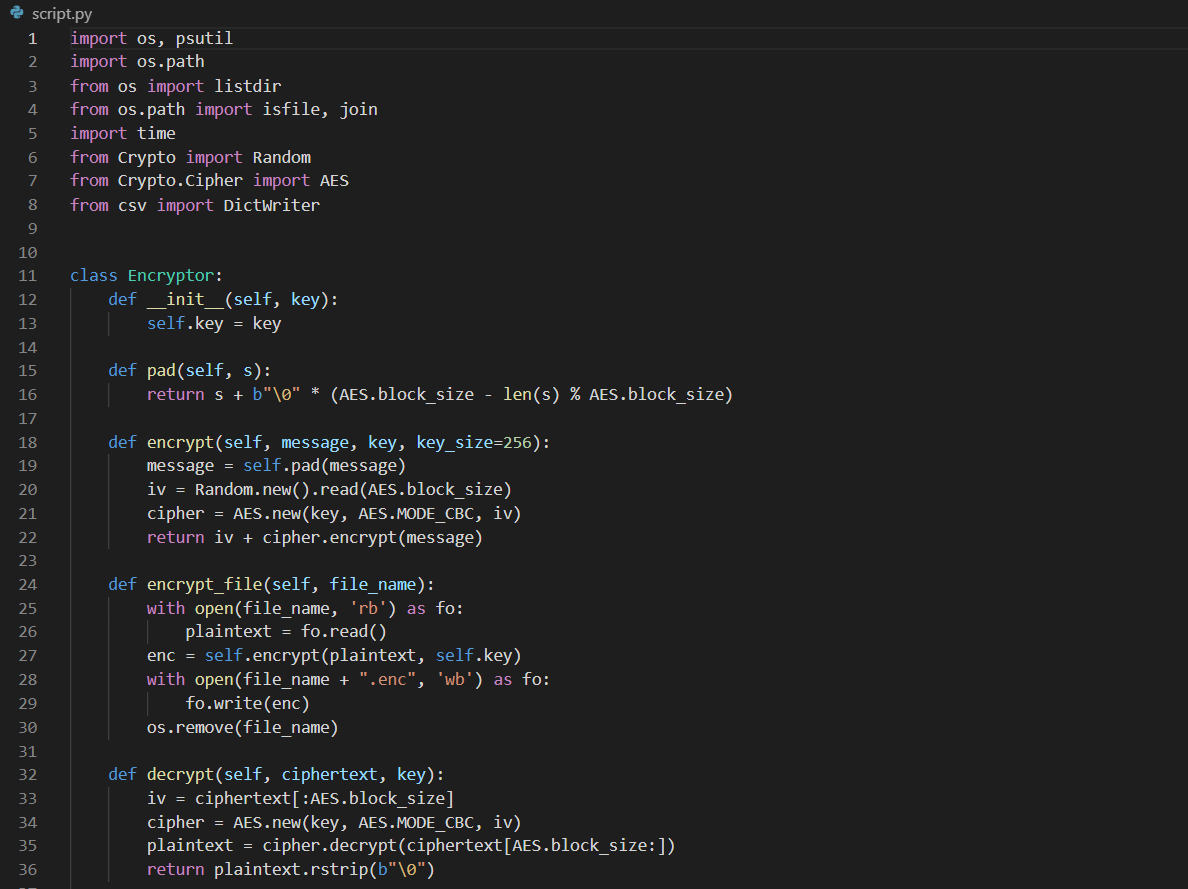
⮚ Interaction of type of file and operation are responsible factors for effecting the memory utilization and CPU utilization

⮚ Size of file is responsible for effecting the time metric as metrics. As the analysis showed that larger the file larger will be the time taken.

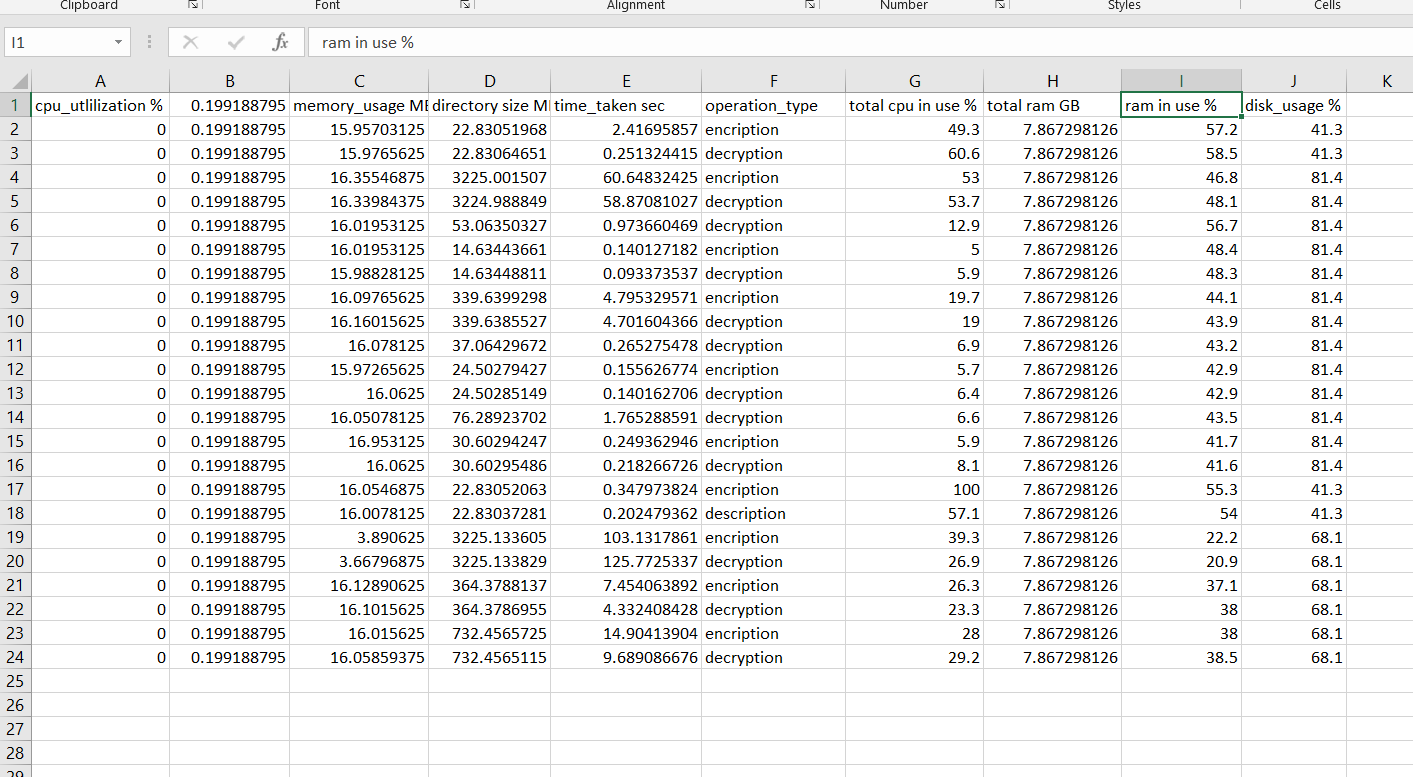
⮚ No information gets lost during the decrytion of file or restoration of original file.

**1.** **Screenshots and some graphical representation**

**a)** **Screenshot of main program:**

****

**Output:**

****

**Reference**

1. M. Panda, "Performance analysis of encryption algorithms for security," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 2016, pp. 278-284, doi: 10.1109/SCOPES.2016.7955835.
2. *pycrypto*. PyPI. (n.d.). https://pypi.org/project/pycrypto/.
3. Kumar P dan Rana S B 2016 Development of modified AES algorithm for data security *Opt. - Int. J. Light Electron Opt.* **127** 2341–5