**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 pycrypto[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**

**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/master/script.py)). 2^k factorial designed is used for the performance analysis. 2^k factorial design is used to find out if a variable is significant to affect a process or not. If k number of variables/factors are studied to determine/screen the important ones, the total number of treatment combinations for a k number of factors can be calculate. Therefore, this screening technique is known as the 2K design of experiments. [4] Python programming language used to monitor the resource uses for the AES encryption algorithm. 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.

**Methodology**

Python programming language with 2^k factorial design is used to identify the significant factors for AES algorithm implementation.

**Encryption and Decryption (AES):** 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.The Advanced Encryption Standard (AES) is a Federal Information Processing Standard (FIPS) which was declared after the competition for encryption algorithms held by National Standards and Technology in 2011. AES is a very high security algorithm [1]. Some algorithms were selected as candidates in the top 5 in a row after Rijndael are Serpent, Twofish, RC6, and MARS algorithms [5]. AES is proven immune to conventional attacks (linear and differential are: resistant to known password analysis, flexible to use in various hardware and software, good for hash functions, suitable for devices that require fast key agility, and suitable for stream ciphers.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.

**Python**

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**

* Presence of third-party modules
* Extensive support libraries
* Open source and community development
* Versatile, Easy to read, learn and write
* User-friendly data structures
* High-level language
* Portable and Interactive
* Ideal for prototypes – provide more functionality with less coding

**PyCrypto**

Python crypto module 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.

**Data Selection**

To collect the data two steps are taken in consideration.

1. Installing essential program and module.

First step is to install python into the machine. To run AES encryption and decryption pyCrypto module is needed and it has installed using following command:

pip install pyCrypto

Another module used to compute the system information which is psutil, available by default.

1. Writing script to compute the effect of total CPU utilization, main memory (ram utilization, execution time of the AES encryption algorithm.

**Factor identification**

To apply 2^k factorial design it’s important to identify factors, parameters. In the data set, cpu utilization, ram utilization , ram usage, execution time, total ram, cpu usage of whole system, disk usage etc are shown. Among them some of system information, which have not any relation with program. Also cpu utilization in 0 percent for the program. That’s mean this program is not cpu sensitive. So if we look deeper, among different parameters there are 3 factors that are considered for the performance analysis of the AES algorithm.

* **Size of folder**
* **Size of RAM**
* **Operation type (Encryption, Decryption)**

As encryption and decryption are performed in bytes level, so format of file does not matter and has no impact as factor. On the other-hand, for the simplification of calculation size folders are consider rather than each file size.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cpu\_utlilization % | memory\_utilization % | memory\_usage MB | directory size MB | time\_taken sec | operation\_type | total cpu in use % | total ram GB | ram in use % | disk\_usage % |
| 0 | 0.198752403 | 16.0234375 | 22.83258343 | 0.338886261 | encription | 19.1 | 7.867298126 | 40.3 | 65.9 |
| 0 | 0.198655427 | 16.015625 | 732.4565725 | 14.90413904 | encription | 28 | 7.867298126 | 38 | 68.1 |
| 0 | 0.400014879 | 16.38671875 | 27.05152798 | 0.571216106 | encription | 7.5 | 3.99956131 | 40.1 | 61.2 |
| 0 | 0.391526247 | 16.04296875 | 175.2835598 | 5.858032465 | encription | 29.5 | 3.99956131 | 39.8 | 61.2 |
| 0 | 0.199867628 | 16.11328125 | 22.83254623 | 0.200044632 | decryption | 39.2 | 7.867298126 | 38.1 | 65.7 |
| 0 | 0.199188795 | 16.05859375 | 732.4565115 | 9.689086676 | decryption | 29.2 | 7.867298126 | 38.5 | 68.1 |
| 0 | 0.39047709 | 16 | 27.05177689 | 0.651234627 | decryption | 13.4 | 3.99956131 | 39.8 | 61.2 |
| 0 | 0.391621624 | 16.046875 | 175.2836447 | 3.669473648 | decryption | 33.2 | 3.99956131 | 39.8 | 61.2 |

**Table 1: Output and Selected dataset for 2^k factorial design ([github](https://github.com/ssroy548/Performance-Modeling-/blob/master/test%20dataset.xlsx)). Some properties are belong to total system. So these are ignored during 2^k factorial design**

**Designing Workload**

For the analysis, existing system used as real workload. Several request has been made by running the script.py file. We categorized different folders with different size to run the experiment. There are two type operation encryption, two different size of folders, and two different system with two different RAM size (8GB and 4GB).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Folder Size** | **Encryption** |  | **Decryption** |  |
|  | 4GB RAM | 8GB RAM | 4GB RAM | 8GB Ram |
| Size<100 | Selected matrix | Selected matrix | Selected matrix | Selected matrix |
| Size>100 | Selected matrix | Selected matrix | Selected matrix | Selected matrix |

**Table 2: Key idea for 2^k factorial design**

**Result & Discussion**

To run 2^k factorial design, as there are 3 factors, 2^3 = 8 experiments are conducted. Four encryption and four decryption are performed in two different system. Three matrix are taken in consideration as output variable from the dataset in this project work. These are Execution time- Time taken to execute the program, Memory usage, Memory utilization.

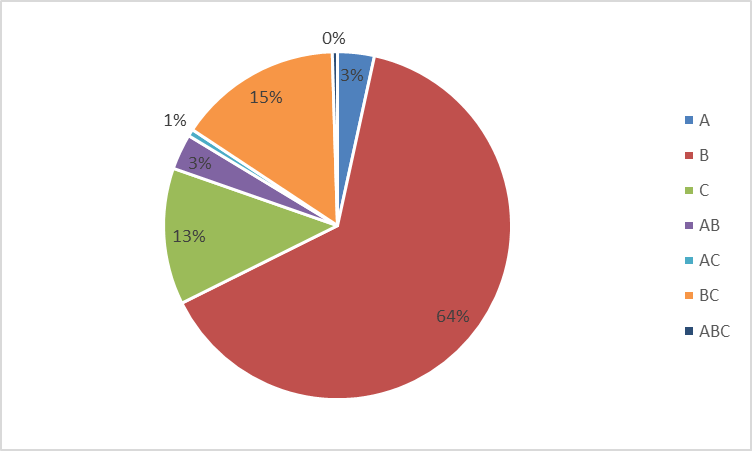
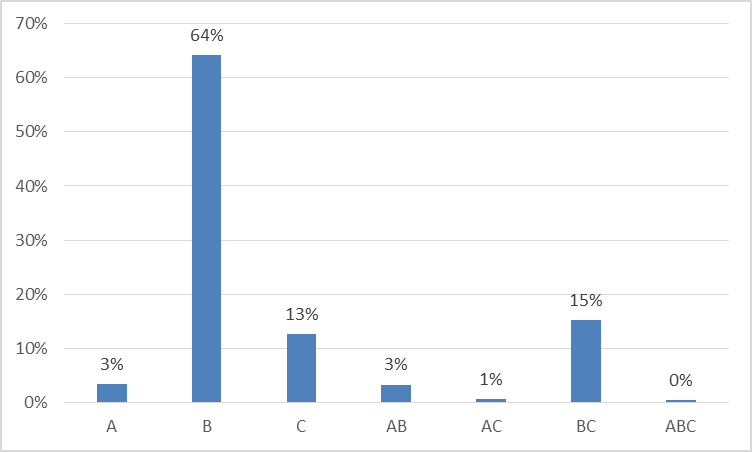


**(a)**



**(b)**

**Table-3(a,b): Shows the factorial design for execution time**

****

**(a) (b)**

**Figure 1(a,b): Graphical representation of 2^k factorial design for execution time**

Table 3 and figure 1([github](https://github.com/ssroy548/Performance-Modeling-/blob/master/execution%20timetime%20as%20matrix.xlsx)) shows us B means folder size is the most important factor, which effect the execution time for encryption and decryption. And BC together is the next significant factor then C means RAM size is the most significant factor and so on.

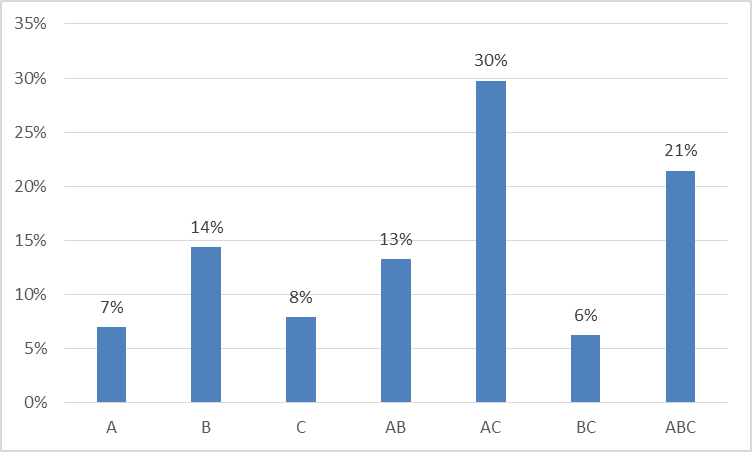
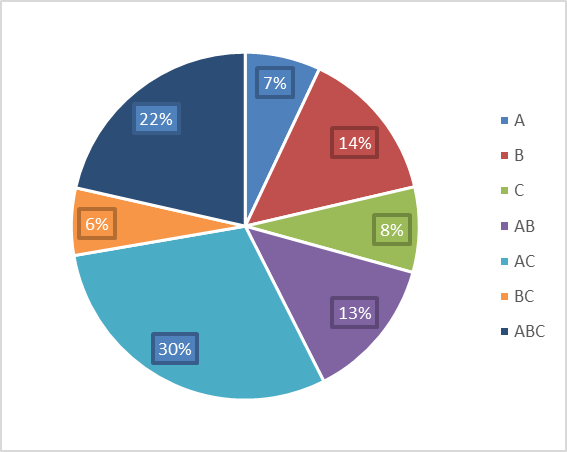


**(a)**



**(b)**

**Table 4(a,b): Shows the factorial design for RAM usage**

****

**(a) (b)**

**Figure 2(a,b): Graphical representation of 2^k factorial design for memory usage**

Table 4 and figure 2([github](https://github.com/ssroy548/Performance-Modeling-/blob/master/memory%20usage%20as%20matrix.xlsx)) shows us the analytical result 2^k factorial design for RAM usage. When RAM usage is used the combination of A(Operation type) and C(RAM size) shows the highest impact on performance.

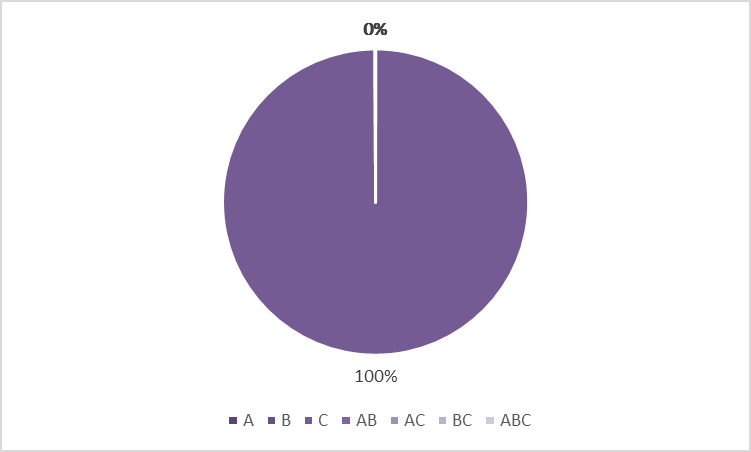


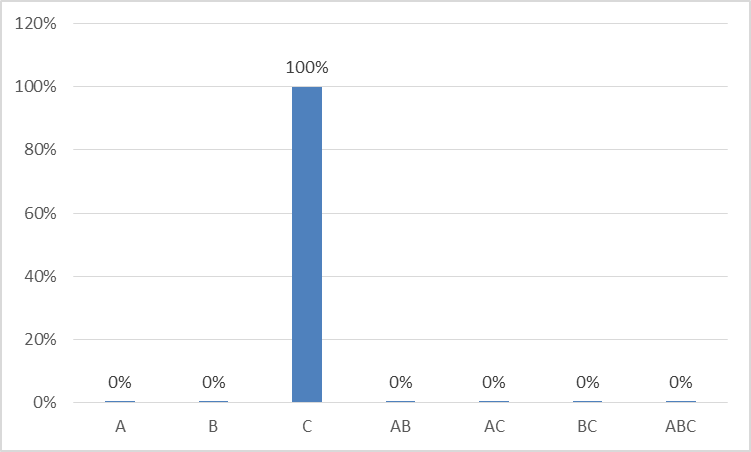
**(a)**



**(b)**

**Table 4(a,b): Shows the factorial design for RAM usage**

****

**  
(a) (b)**

**Figure 3(a,b): Graphical representation of 2^k factorial design for memory utilization**

Table 4 and figure 4 shows ([github](https://github.com/ssroy548/Performance-Modeling-/blob/master/memory%20utilization%20as%20matrix.xlsx)), when we take memory utilization as matric we found that it totally depends on RAM size. Size of folder, operation type have no impact on the performance of the program. The program is totally RAM sensitive.

**Conclusion & Future Work**

This project shows us that for the execution time folder size is the most important factor for the. In 2nd analysis it’s shows that ram usage is depended on operation type and ram mostly.

And in the 3rd analysis it’s shows memory utilization totally depends on the memory size. So depending on these three result we can say that the performance of the pyCrypto package AES (module) algorithm depends on folder size and ram mostly. And, this experiment shows us, it can be extended by using different system or machine. Because here only 2 type system us used. We can continue it for different ram size system and for different CPU configuration. Also, it can be done on other programming language with appropriate packages. This will help us to understand the impact of algorithm implementation of different language on different machine. After that we can pick up the best one for our working purpose.

**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. ((2021, June 30).). 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
4. The Open Educator - 1. What is 2K Design. (n.d.). https://www.theopeneducator.com/doe/2K-Factorial-Design-of-Experiments/What-is-2K-Factorial-Design-of-Experiments.
5. Karale S N, Pendke K dan Dahiwale P 2015 The survey of various techniques & algorithms for SMS security *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (IEEE) hal 1–6
6. *Python Language advantages and applications*. GeeksforGeeks. (2021, June 30). https://www.geeksforgeeks.org/python-language-advantages-applications/.