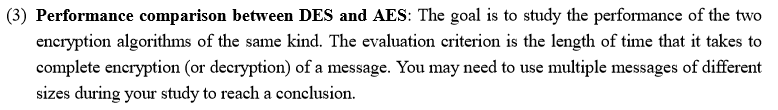
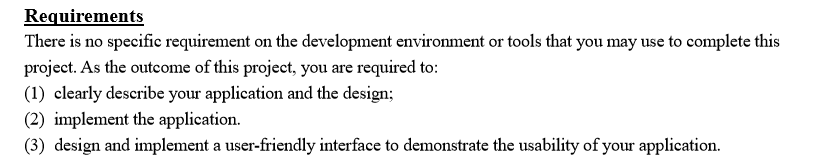
**Security and Privacy Project 2 AES and DES Comparison Study**

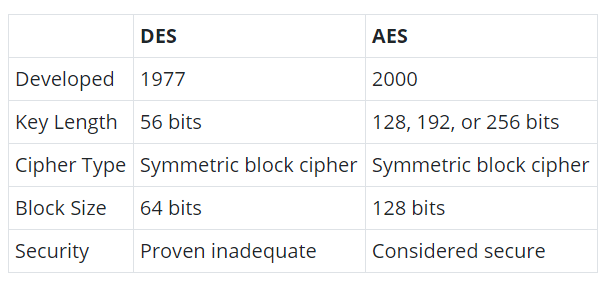
**James Crown 15534833 H20180909002**

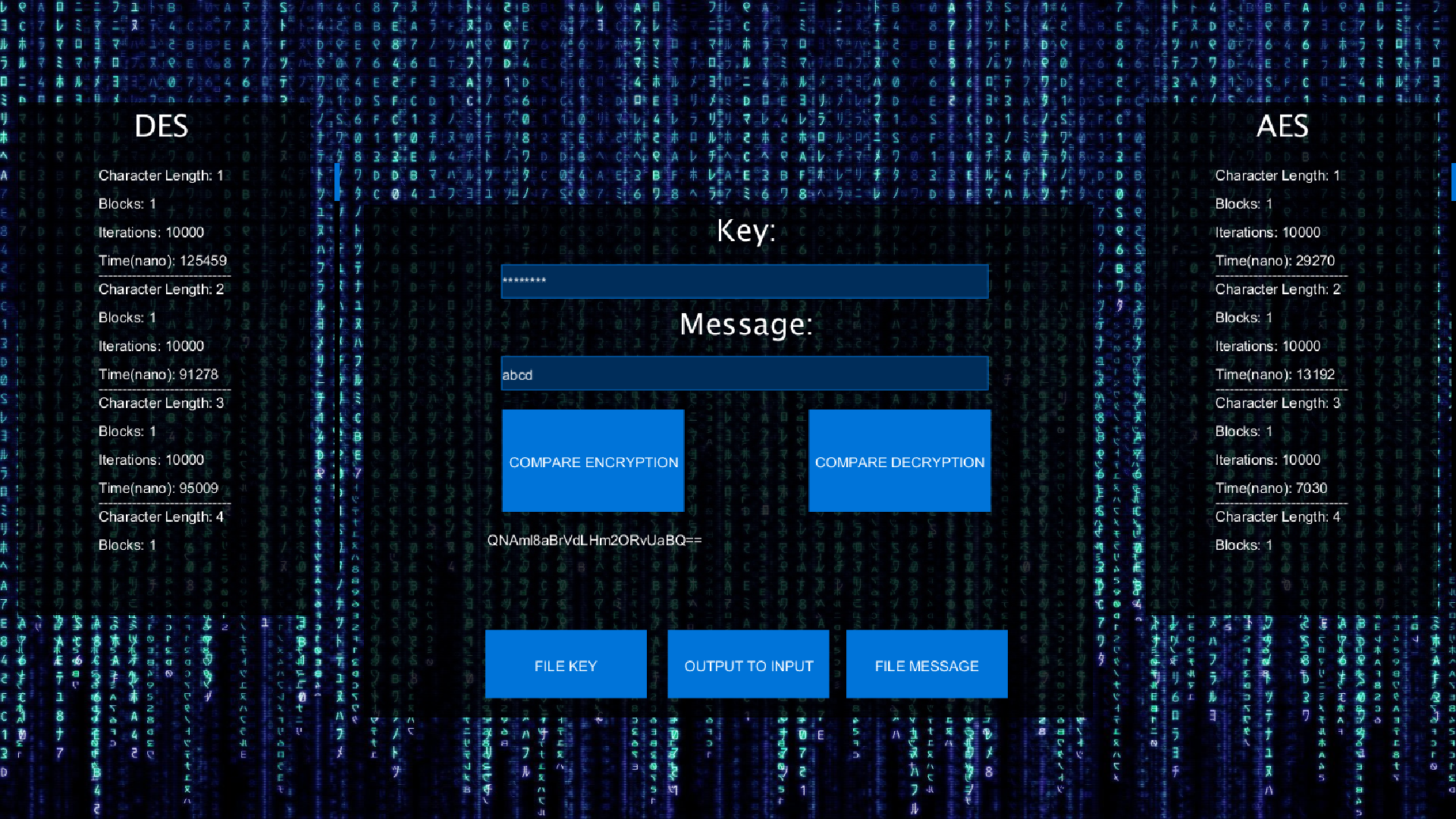




**Introduction**

* The aim of this assignment is to implement an Advanced Encryption Standard alongside an existing Data Encryption Standard and study the performance of both systems in regards to a multitude of input strings of varying lengths.





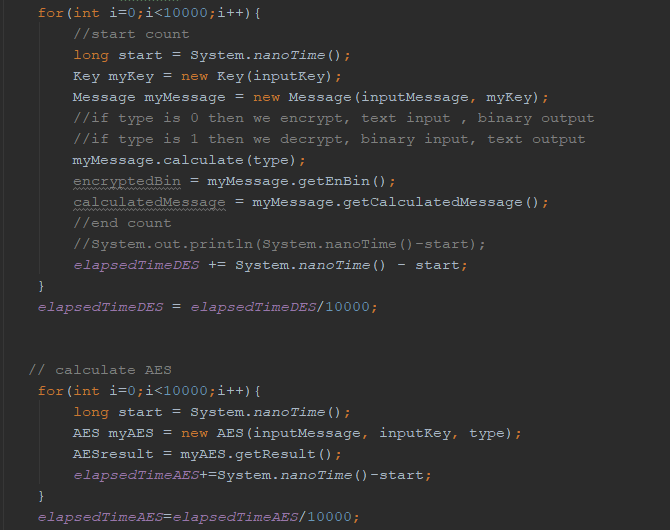
**User Interface**

* The UI above is an extended version of the UI submitted for the first assignment. The main differences consist of two “COMPARE ENCRYPTION”/”COMPARE DECRYPTION” buttons and two text area boxes on either side of the main interface that list the values associated with each algorithms performance. All original functionality of the UI remains the same. File input and output still exists. The statistics of each algorithm displayed on the edges are scrollable textboxes updating each time “COMPARE ENCRYPTION” or “COMPARE DECRYPTION” is pressed.

**Evaluation Criterion**

* The method used for evaluating the performance of both algorithms is solely time based. I will include in my code a method to evaluate the amount of time it takes to complete each task. Naturally this time will vary for each run of the algorithm so therefore I aim to run each algorithm 10,000 times and take the average time overall. This will give a mean evaluation and represent the true values more accurately. That being said, this evaluation method will also be affected by the hardware used so we will be altering the results into the form of fractions so that they will be constant between any system that runs the program.
* When evaluating the time performance of each algorithm there was an initial spike in latency which disappeared once the algorithm was run multiple times. Thus, I believe that the initial result acts as an outlier which I chose to ignore. I instead chose to run each input message 10 times ( each message was already being run 10000 times, so overall about 100k times) and I took a later value as its representative.
* I also do not intend to rely on the raw data, but on relative data between the two algorithms, that is to say that I will calculate a fraction representative of the relationship between the data which will hopefully apply to any system that runs the algorithms and not just my own.

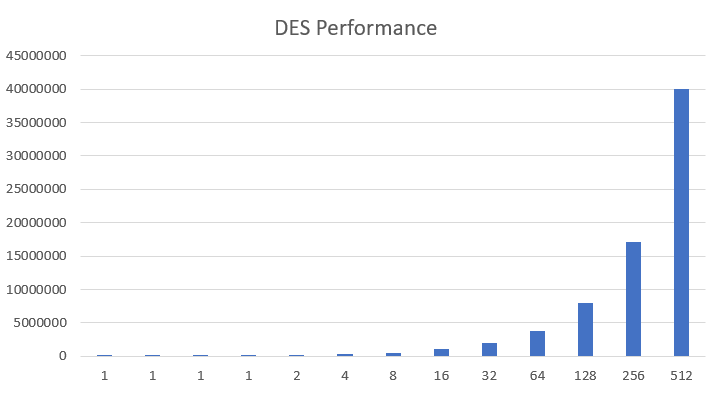
**Implementation**



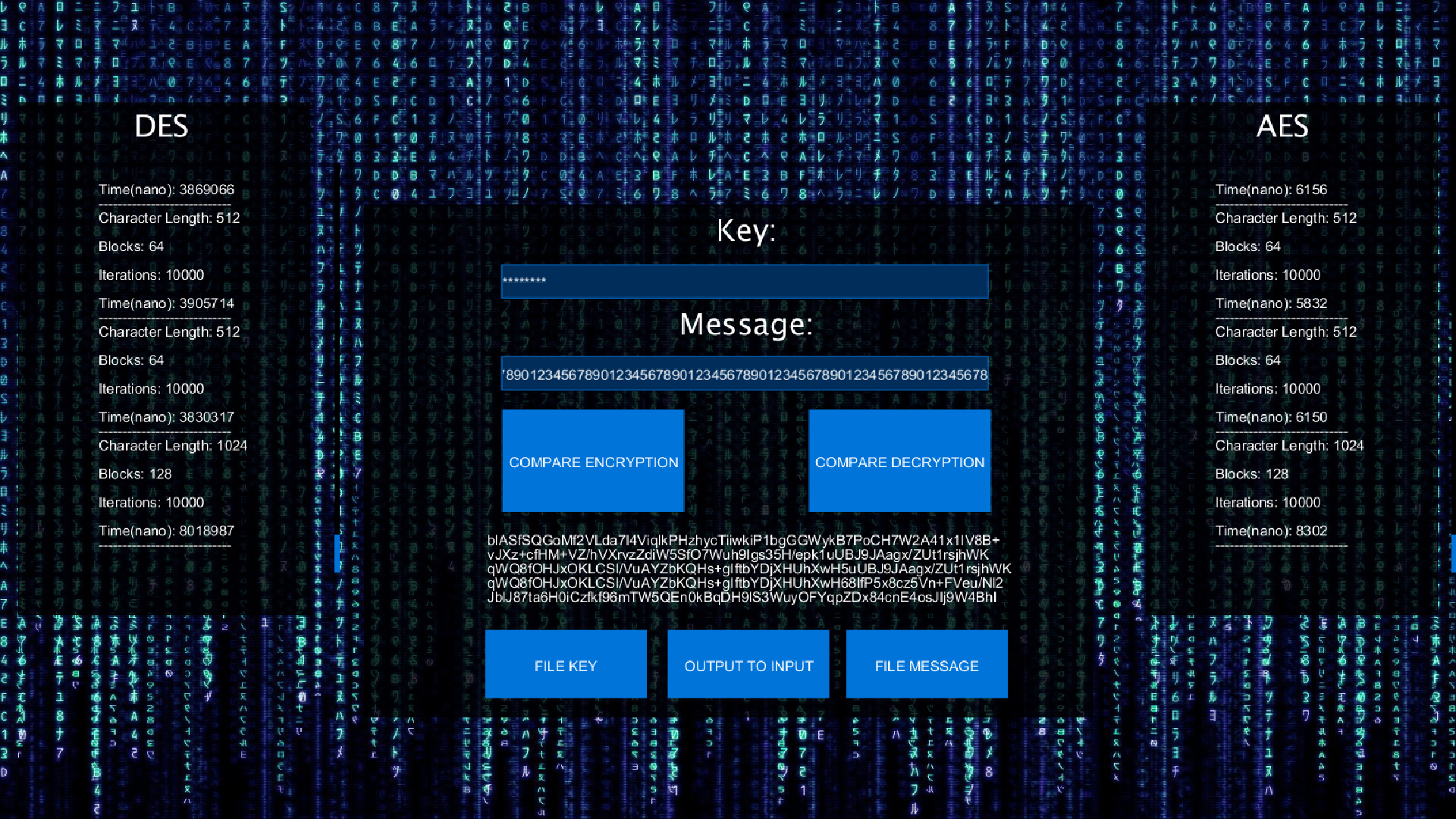
This screenshot portrays the main loops that runs both algorithms 10000 times and finds the averages using .nanoTime().

* This assignment involves the implementation of two algorithms.
* **DES**
  + The DES algorithm has been implemented in a previous assignment and no changes will be made to its implementation. This is due to the assignment specifications as we were asked to extend upon our previous assignment. I do understand that if a library were used it would potential be more efficient and therefore result in a more accurate final value.
* **AES**
  + The AES algorithm was implemented differently to DES as the assignment is based upon the study and comparison of these two algorithms and not the implementation. I used a few crucial libraries to implement this algorithm. These libraries may have caused an increased gap in performance between the two algorithms as it is very efficient.

**DES Performance**

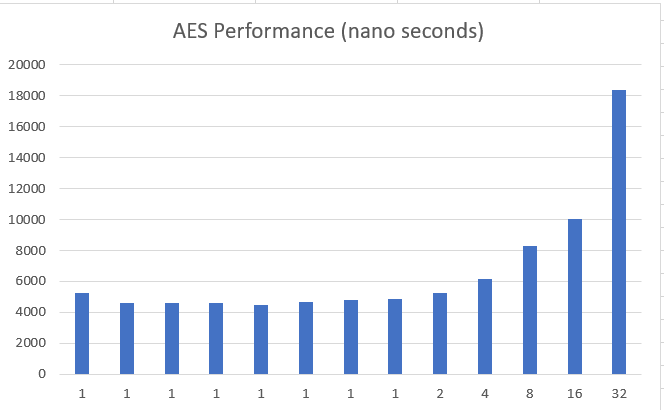


The above chart maps the nanoseconds taken for DES encryption to complete on the Y axis to the number of blocks inputted on the X. As you can see, there is a linear relationship between these two factors. We double the number of blocks and the time taken doubles.



* Overall DES performed far slower than its counterpart.
* On the left we can see the time taken to complete the process under DES and on the right under AES. DES does not outperform it under any circumstances

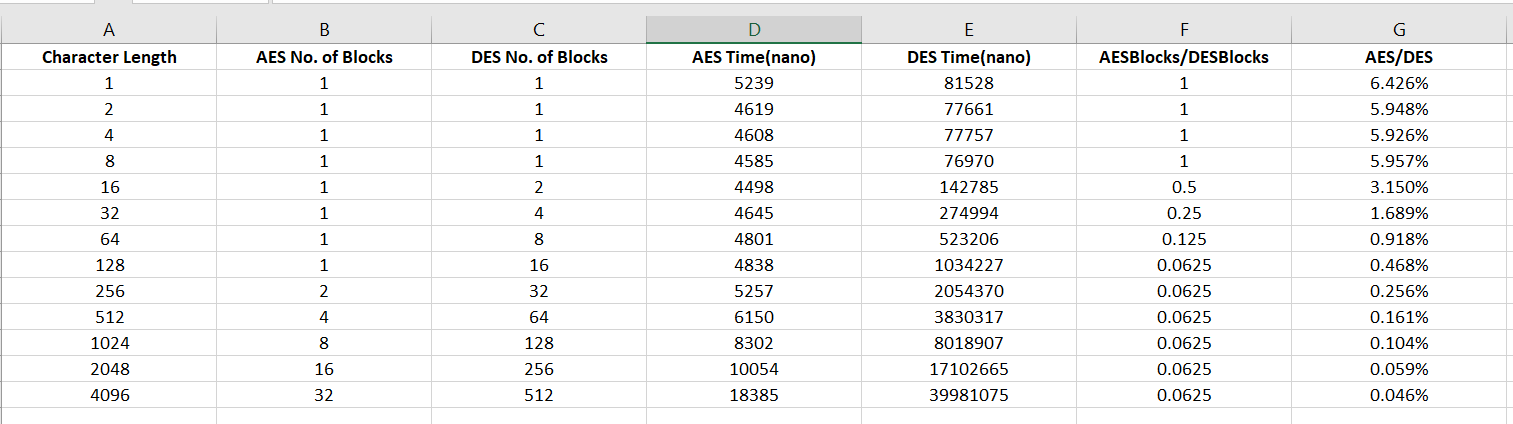
**AES Performance**



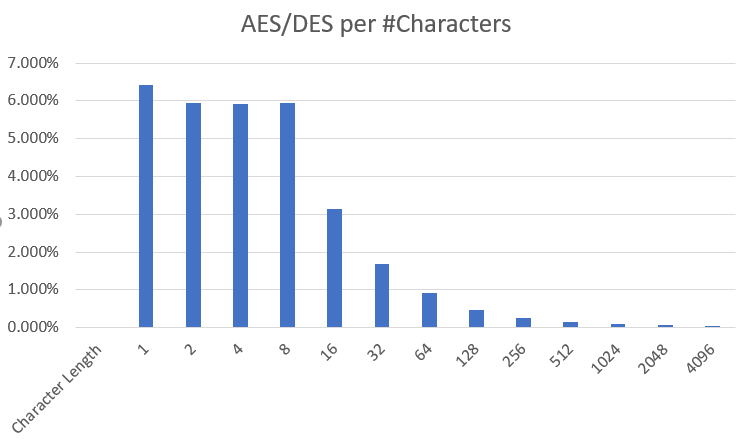
AES Performance on the same mapping shows a much more uniform distribution until the block amount increases significantly. It seems that DES had a linear relationship and AES does not. When the number of blocks increases from 1 to 2 we do not see a double in time taken. This points more towards a log relationship.

**Comparison**

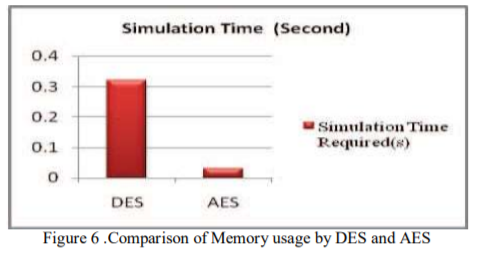
* AES was the clear winner of the two. With a block size of 128 bits it outperforms the DES algorithm by far. The block size plays a large role in the overall factor that distinguishes AES over DES, that factor is the number of iterations. AES is actually a more complex algorithm per block than DES but it makes up for it in the few iterations it does relative to DES. As seen in the Excel file provided, by character length 128, AES has 16 times less iterations to do than DES.
* AES’s key length is another factor to consider in the equation but when tested. The key length does not affect the performance of the algorithm in any relevant way.



After running multiple different strings through both algorithms 10,000 times and getting the means, I created this excel spreadsheet mapping the results. I then used these results to create visualisations of AES and DES’s performance in the form of graphs.



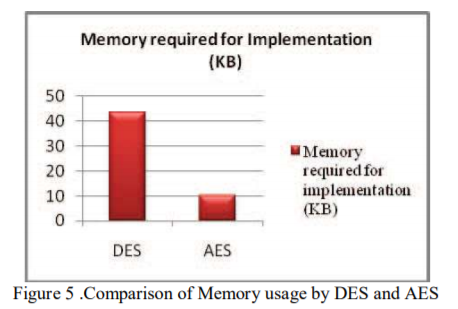
This graph portrays the AES/DES for each studied character lengths. As you can see, initially the AES outperforms DES as it only takes 7% of the time DES needs to complete the same task. Then as the number of characters increases this percentage only decreases down to 0.046%.



Comparing my results to that of an official publication, they seem to be very similar. AES is a viable option for encryption currently whereas DES was cracked within a day many years ago.

* I studied the performance of each algorithm at different string lengths. Starting at a length of 1 I increased the length exponentially from 1 to 2 to 4 to 8 up until 4096.
* This became increasingly difficult as each algorithm ran 10000 times and DES is very slow as shown above in the screenshot.
* I used a file input of a specific base 2 length and duplicated the string inside that file for each test to quickly create the new string length of twice what is was before.

**Memory Requirements**



Not only does AES perform well in performance time but also in memory requirement. Its advantage over DES in this category is not as extreme as under performance time but is still more than 4x more efficient.

**Conclusion**

* From the data collected, AES outperforms DES with ease and its advantage over DES only increases as you increase the input string size.
* We collected the data and normalized it in a way that other studies of DES/AES comparison should correlate with our results.
* The success of AES ultimately comes down to the larger block size and overall decrease in iterations necessary, the key sizes did not reveal any major changes in performance.

**Libraries**

* java.security
* javax.crypto

**References**

1. [**https://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption\_perf/**](https://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/)
2. [**https://aesencryption.net/**](https://aesencryption.net/)
3. [**http://blog.syncsort.com/2018/08/data-security/aes-vs-des-encryption-standard-3des-tdea/**](http://blog.syncsort.com/2018/08/data-security/aes-vs-des-encryption-standard-3des-tdea/)
4. [**https://pdfs.semanticscholar.org/cb26/d4781f93a7da9fd5664c8ff6cb2c06519491.pdf**](https://pdfs.semanticscholar.org/cb26/d4781f93a7da9fd5664c8ff6cb2c06519491.pdf)