Spring 2023 CSCI 350 – PE1 Computer Security Basics

Group 4 – Long Huang, Andrew Lee, John Granell, Mert Dogan

Assignment 4

Report

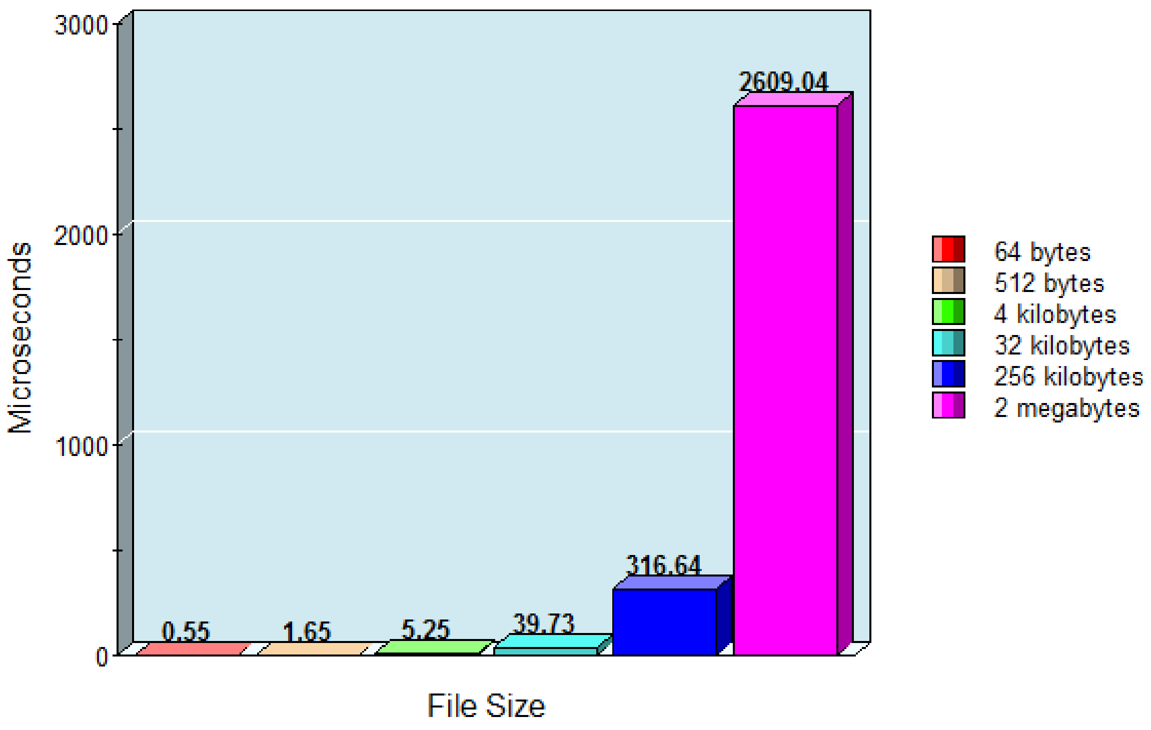
Of the two symmetric encryptions with block ciphers (AES and Camellia), AES is significantly faster. This is especially noticeable with encrypting larger plaintext files. This is to be expected and can be explained by AES being more efficient of an encryption algorithm. It performs less rounds/operations than its Camellia counterpart.

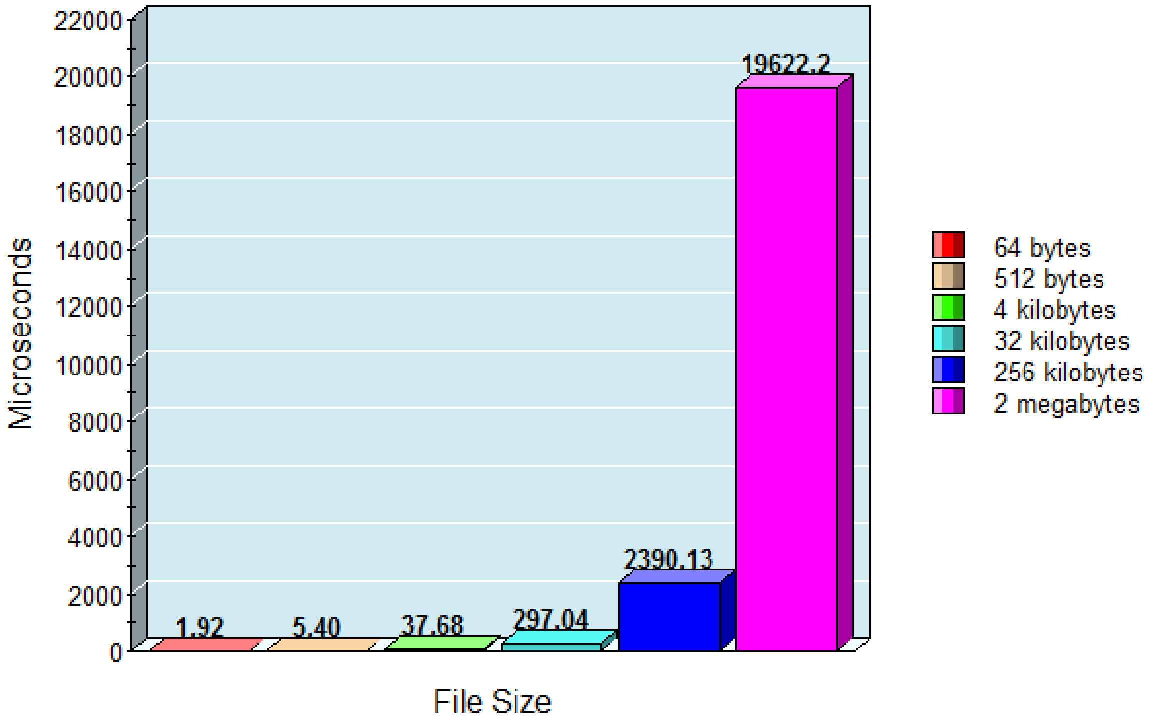
Falling right in the middle of the two symmetric encryption algorithms in terms of speed, RC4 sets itself apart with its stream cipher. While block ciphers encrypt a group of plaintext characters as one block, stream ciphers convert one character of plaintext directly into a character of cipher text. Because of the nature of stream ciphers, RC4 is faster than most block ciphers like Camellia; however, it leaves itself vulnerable to attacks. AES is recommended over RC4 because it is faster and less vulnerable.

Hashing, a different type of algorithm that uses digest generation, appears to be ever so slightly faster than the aforementioned RC4 stream cipher. Out of the four techniques used here, it tests second to AES in terms of speed. Hashing generally has different uses than encryption. However, as it is irreversible – one cannot reverse the output of the function into the original input.

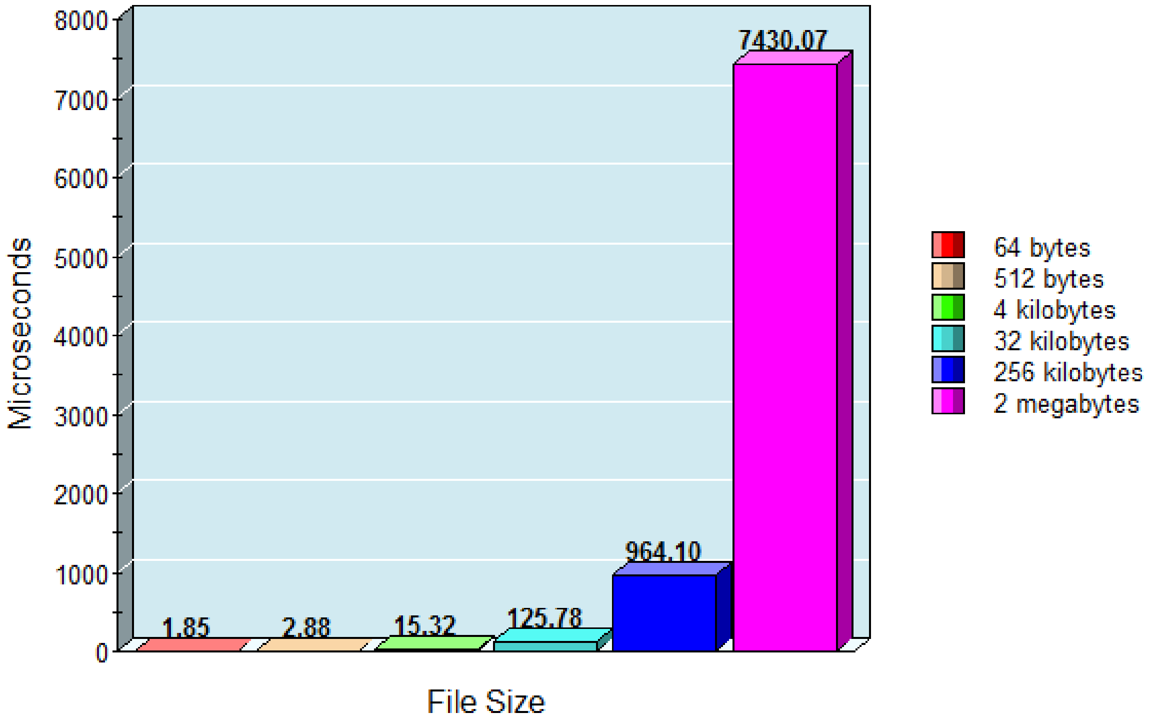
The trend of all of the methods is an exponential curve relating to encryption time and how large the file is that is being ecrypted. The larger the file the longer the ecryption process takes. At 256 kilobytes it starts to take a little longer than the others, at 2 megabytes the time it takes to encrypt a file increases by a magnitude. Using these methods to encrypt larger files would take a while.

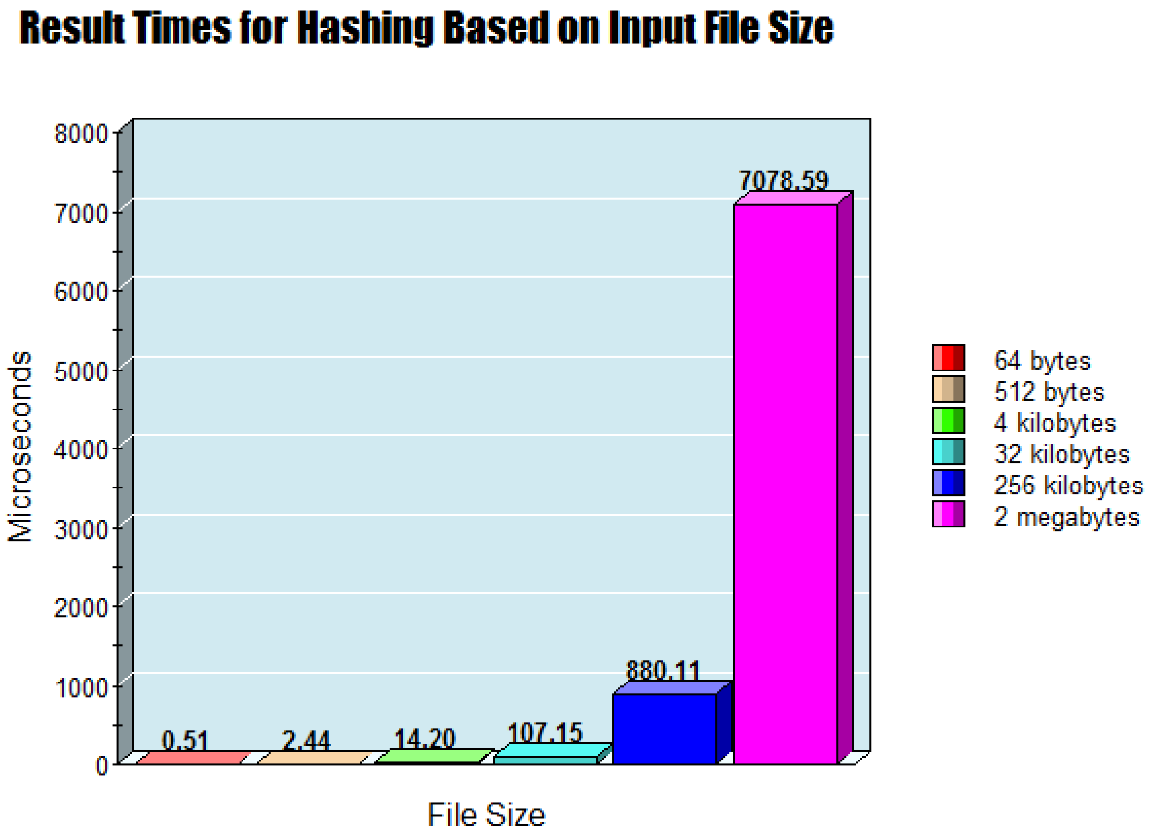
Result Times for AES Based on Input File Size




Result Times for Camellia Based on Input File Size

Result Times for RC4 Based on Input File Size




Result Times for Hashing Based on Input File Size
Microseconds
8000
7000
6000
5000
4000
3000
2000
1000
0
0.51
2.44
14.20
107.15
880.11
7078.59
File Size
64 bytes
512 bytes
4 kilobytes
32 kilobytes
256 kilobytes
2 megabytes