Part 3 - Report

Tested with plaintext sizes ranging from 100kb to 1gb. The plaintext size impacted encryption/decryption times. I can notice through observing results.csv and results_average.txt that if plaintext size increases so does the average time for encrypting and decrypting. This can be seen in the analysis below.

Analysis of ECB Mode
For encryption, the smaller file sizes (97-488Mb), the encryption times for key sizes on average were fairly close, with the 256-bit key being slightly faster for 97Mb and 192 faster for 488Mb. For larger file sizes(4847Mb +), the encryption time increases with key size as we can see with 256-bit key performing the slowest (1Gb) and 128 the best across. Decryption for smaller files (97-488Mb) performed better with the 192-bit key. For larger files, the decryption time increased with key size, with 128 and 192 performing the best. I would recommend that using a 192-bit key for ECB mode may be preferable and suitable for an efficient encryption/decryption algorithm and also offers more security.

Analysis of CBC Mode
For encryption, with the smaller and larger file sizes, the average encryption time increased with key size with a 256-bit key being the slowest across, and 128- and 192-bit keys performing best for smaller files (97-488mb). Can make the pattern that with key size increases the longer the encryption. For decryption of the smaller file sizes (97 MB), the decryption times show that the 256-bit key performs better than the 128 and 192-bit keys. However, for larger files(488251+) the pattern is less consistent, with 256 being the slowest, and 192-bit keys being faster in more than 128. My recommendation is if CBC mode is being used on small files, then the key size has little impact on performance due to the speed -> 256 slow encryption but faster for decryption so using a 256-bit key would be a better implementation. For larger files, having an efficient time is more significant, with larger keys leading to slower encryption times, showing that a lower key is more suitable for an efficient encryption algorithm. I would recommend 192-bit key for files larger than 4847 based on decryption times.

Analysis of CTR Mode
For encryption of smaller files and larger files, 128-bit keys performed best for 97Mb, 48806Mb, and 1Gb files, with 256-bit keys performing better on the other three. For larger files, a 256-bit key often had better encryption and decryption performance compared to the other key sizes. The ability of CTR mode to process data in parallel helps maintain efficient performance even as file sizes increase. My recommendation for the use of CTR with dealing with smaller files, while 128-bit keys provide faster performance for both encryption and decryption in some cases, the differences are minor and the choice of key size might be more influenced for security reasons rather than performance on smaller files, so using a 256-bit key would be best. For larger files as well, I would recommend 256-bit keys based on the performance observed.

Analysis of OFB Mode
For encryption across small and large files showed that the fastest average encryption time was the 128-bit key (with the exception of 192 for 488Mb), with the 256-bit key taking the longest. The increase in encryption time with key size is fairly noticeable. For decryption, across small and large files, 128-bit keys are the fastest (with the exception of 192 for 488 Mb) compared to 192-bit and 256-bit keys. As file size increases, encryption and decryption times increase as expected and

performance difference is more noticeable. I would recommend 128-bit keys as they offer the best performance in terms of both encryption and decryption times. As OFB is more of a confidentiality mode having a key size higher than 128 would be beneficial however but based on the results shown unless you're willing to give up performance for security, then a 128-bit key is the better choice.

Analysis of CFB Mode

Similar to OFB, encryption, and decryption of smaller files, 128-bit keys generally offer the fastest encryption and decryption times compared to 192-bit and 256-bit keys. For larger files encryption and decryption using 128-bit key consistently provided the fastest encryption times across increased file sizes, with the exception of 256-bit key at 1gb for decryption. For smaller files use 128-bit keys if you want fast performance, however since encryption based on larger keys is still fast in consideration that the data is nanoseconds, using a higher key size for security as well would be a good implementation. For larger files where encryption and decryption take a longer time, 128-bit keys are typically the fastest so I would recommend that for performance, as there is a clear pattern of larger keys taking longer.

Analysis of GCM Mode

For encryption for smaller files (97-488Mb), encryption times are fastest for 128-bit keys. For larger files encryption times vary with 128-bit keys being faster for 48806 and 488251 files and 192 being faster for 4847Mb and 1Gb, although the differences are not very noticeable as they produce very similar average times. For decryption for smaller files(97-48806 MB), decryption times are faster with 128-bit keys and highest with 256-bit keys. Decryption for larger files (488251Mb-1Gb), 192-bit keys, and 256-bit keys perform better. Due to no identifiable patterns for encryption and decryption across file sizes based on the key, I would recommend a 192-bit or 256-bit key based on performance across encryption/decryption times for larger files while also being a more secure option compared to a 128-bit key.

From analysis, we can see that using different AES modes has diverse performance when tested with the same parameters. From my observations, I can state that ECB mode showed faster encryption/decryption for large plaintext files in comparison to other modes. We can also see that OFB and CFB modes showed similar performance as well for smaller and larger plaintext files.