Text	Mode	Key	EorD	Time
plaintext100words.txt	AES/CBC/PKCS5PADDING	128	encryption	0.108458ms
100wordsciphertext.enc	AES/CBC/PKCS5PADDING	128	decryption	0.104292ms
plaintext100words.txt	AES/ECB/PKCS5PADDING	128	encryption	0.261708ms
100wordsciphertext.enc	AES/ECB/PKCS5PADDING	128	decryption	0.106292ms
plaintext100words.txt	AES/CTR/PKCS5PADDING	128	encryption	0.113083ms
100wordsciphertext.enc	AES/CTR/PKCS5PADDING	128	decryption	0.096958ms
plaintext100words.txt	AES/OFB/PKCS5PADDING	128	encryption	0.115166ms
100wordsciphertext.enc	AES/OFB/PKCS5PADDING	128	decryption	0.098708ms
plaintext100words.txt	AES/CFB/PKCS5PADDING	128	encryption	0.101625ms
100wordsciphertext.enc	AES/CFB/PKCS5PADDING	128	decryption	0.094125ms
plaintext100words.txt	AES/GCM/PKCS5PADDING	128	encryption	0.112833ms
100wordsciphertext.enc	AES/GCM/PKCS5PADDING	128	decryption	0.095167ms
plaintext100words.txt	AES/CBC/PKCS5PADDING	192	encryption	0.121167ms
100wordsciphertext.enc	AES/CBC/PKCS5PADDING	192	decryption	0.102208ms
plaintext100words.txt	AES/ECB/PKCS5PADDING	192	encryption	0.111ms
100wordsciphertext.enc	AES/ECB/PKCS5PADDING	192	decryption	0.101875ms
plaintext100words.txt	AES/CTR/PKCS5PADDING	192	encryption	0.106834ms
100wordsciphertext.enc	AES/CTR/PKCS5PADDING	192	decryption	0.1025ms
plaintext100words.txt	AES/OFB/PKCS5PADDING	192	encryption	0.112709ms
100wordsciphertext.enc	AES/OFB/PKCS5PADDING	192	decryption	0.104ms
plaintext100words.txt	AES/CFB/PKCS5PADDING	192	encryption	0.11825ms
100wordsciphertext.enc	AES/CFB/PKCS5PADDING	192	decryption	0.241208ms
plaintext100words.txt	AES/GCM/PKCS5PADDING	192	encryption	0.106ms
100wordsciphertext.enc	AES/GCM/PKCS5PADDING	192	decryption	0.100542ms
plaintext100words.txt	AES/CBC/PKCS5PADDING	256	encryption	0.189208ms
100wordsciphertext.enc	AES/CBC/PKCS5PADDING	256	decryption	0.113833ms
plaintext100words.txt	AES/ECB/PKCS5PADDING	256	encryption	0.121333ms
100wordsciphertext.enc	AES/ECB/PKCS5PADDING	256	decryption	0.311458ms
plaintext100words.txt	AES/CTR/PKCS5PADDING	256	encryption	0.120167ms
100wordsciphertext.enc	AES/CTR/PKCS5PADDING	256	decryption	0.110167ms
plaintext100words.txt	AES/OFB/PKCS5PADDING	256	encryption	0.120041ms
100wordsciphertext.enc	AES/OFB/PKCS5PADDING	256	decryption	0.109083ms
plaintext100words.txt	AES/CFB/PKCS5PADDING	256	encryption	0.120667ms
100wordsciphertext.enc	AES/CFB/PKCS5PADDING	256	decryption	0.11ms
plaintext100words.txt	AES/GCM/PKCS5PADDING	256	encryption	0.128083ms
100wordsciphertext.enc	AES/GCM/PKCS5PADDING	256	decryption	0.111625ms
3000words.txt	AES/CBC/PKCS5PADDING	128	encryption	0.971834ms
3000wordsciphertext.enc	AES/CBC/PKCS5PADDING	128	decryption	1.43275ms
3000words.txt	AES/ECB/PKCS5PADDING	128	encryption	0.956542ms
3000wordsciphertext.enc	AES/ECB/PKCS5PADDING	128	decryption	1.429167ms
3000words.txt	AES/CTR/PKCS5PADDING	128	encryption	0.939708ms

3000wordsciphertext.enc	AES/CTR/PKCS5PADDING	128	decryption	1.373583ms
3000words.txt	AES/OFB/PKCS5PADDING	128	encryption	0.930834ms
3000wordsciphertext.enc	AES/OFB/PKCS5PADDING	128	decryption	1.443458ms
3000words.txt	AES/CFB/PKCS5PADDING	128	encryption	0.964125ms
3000wordsciphertext.enc	AES/CFB/PKCS5PADDING	128	decryption	1.427417ms
3000words.txt	AES/GCM/PKCS5PADDING	128	encryption	1.034084ms
3000wordsciphertext.enc	AES/GCM/PKCS5PADDING	128	decryption	1.467083ms
3000words.txt	AES/CBC/PKCS5PADDING	192	encryption	0.941916ms
3000wordsciphertext.enc	AES/CBC/PKCS5PADDING	192	decryption	1.524875ms
3000words.txt	AES/ECB/PKCS5PADDING	192	encryption	0.945917ms
3000wordsciphertext.enc	AES/ECB/PKCS5PADDING	192	decryption	1.522084ms
3000words.txt	AES/CTR/PKCS5PADDING	192	encryption	0.942ms
3000wordsciphertext.enc	AES/CTR/PKCS5PADDING	192	decryption	1.5565ms
3000words.txt	AES/OFB/PKCS5PADDING	192	encryption	1.049709ms
3000wordsciphertext.enc	AES/OFB/PKCS5PADDING	192	decryption	1.830209ms
3000words.txt	AES/CFB/PKCS5PADDING	192	encryption	1.19725ms
3000wordsciphertext.enc	AES/CFB/PKCS5PADDING	192	decryption	1.613292ms
3000words.txt	AES/GCM/PKCS5PADDING	192	encryption	0.992667ms
3000wordsciphertext.enc	AES/GCM/PKCS5PADDING	192	decryption	1.524042ms
3000words.txt	AES/CBC/PKCS5PADDING	256	encryption	0.953125ms
3000wordsciphertext.enc	AES/CBC/PKCS5PADDING	256	decryption	1.596584ms
3000words.txt	AES/ECB/PKCS5PADDING	256	encryption	0.959125ms
3000wordsciphertext.enc	AES/ECB/PKCS5PADDING	256	decryption	1.615333ms
3000words.txt	AES/CTR/PKCS5PADDING	256	encryption	0.945ms
3000wordsciphertext.enc	AES/CTR/PKCS5PADDING	256	decryption	1.878042ms
3000words.txt	AES/OFB/PKCS5PADDING	256	encryption	0.945834ms
3000wordsciphertext.enc	AES/OFB/PKCS5PADDING	256	decryption	1.788125ms
3000words.txt	AES/CFB/PKCS5PADDING	256	encryption	0.947834ms
3000wordsciphertext.enc	AES/CFB/PKCS5PADDING	256	decryption	1.600333ms
3000words.txt	AES/GCM/PKCS5PADDING	256	encryption	0.955875ms
3000wordsciphertext.enc	AES/GCM/PKCS5PADDING	256	decryption	1.657375ms

I decided to do my tests on two different plaintexts, one has 100 words and the other has 3000 words. I wanted to see if having a much larger text file document would affect the encryption and decryption time by much and what cipher mode was most efficient.

When encrypting the 100-word plaintext file with a key size of 128bits, I found that CFB mode was the most efficient with a time of 0.101625ms. But this is only 0.007ms quicker than CBC. All the times for 128bit key with the 100-word plaintext were similar except for ECB mode which was 0.16ms slower than CFB mode. This means ECB is not very efficient, so I expect to see similar results with the other key sizes. With the same plaintext file but with the 192-bit key size the quickest mode for encryption was GCM but by only 0.008ms. ECB however was quicker than the 128bit key. Next with 256bit key size, the quickest

encryption was OFB. The fastest time to encrypt the 3000-word plaintext with 128bit key length was OFB, and the slowest was GCM. Next with the 192bit key length was CBC but only by 0.0001ms. Next with 256bit key was CTR.

Now decrypting the 100-word plaintext file with 128bit key the quickest was CFB. Next with 192bit key was ECB. Next with 256-bit key was OFB. Now with the 3000-word plaintext and 128bit key was CTR, next with 192bit key was ECB and finally with 256-bit key was CBC.

What I can conclude from my results are that basically all these methods of encrypting and decrypting are very similar in how long they take to complete the process. One thing I found out was that decrypting proved to be more time consuming which means it is more complex to do. This makes sense as encryption we just generate random bytes as the ciphertext which is easy to do, but to decipher it, there is many more things that must be done. But overall OFB was a cipher mode that came up the most in this for test for being the quickest. OFB is very efficient and works the same was as CTR but does not require a nonce for each message, making it slightly more efficient. OFB can individually process blocks of plaintext which CFB cannot making OFB quicker than it. But OFB is not parallelizable like GCM and CTR are.