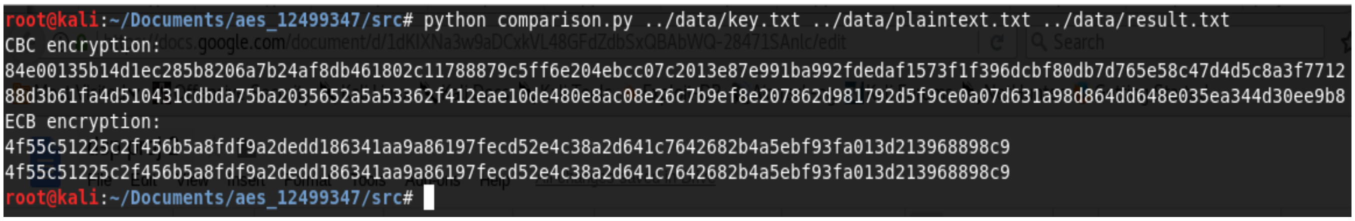
Data Security & Privacy
Project 2

Raghu Pusapati

pusaparv@mail.uc.edu

(a) A program has been written to run each encryption modes twice and print the cipher texts. The same program has been run and output is as shown below



```
root@kali:~/Documents/aes_12499347/src# python comparison.py ../data/key.txt ../data/plaintext.txt ../data/result.txt
CBC encryption:
84e00135b14d1ec285b8206a7b24af8db461802c11788879c5ff6e204ebcc07c2013e87e991ba992fdedaf1573f1f396dcbf80db7d765e58c47d4d5c8a3f7712
88d3b61fa4d510431cdbda75ba2035652a5a53362f412eae10de480e8ac08e26c7b9ef8e207862d981792d5f9ce0a07d631a98d864dd648e035ea344d30ee9b8
ECB encryption:
4f55c51225c2f456b5a8fdf9a2dedd186341aa9a86197fecd52e4c38a2d641c7642682b4a5ebf93fa013d213968898c9
4f55c51225c2f456b5a8fdf9a2dedd186341aa9a86197fecd52e4c38a2d641c7642682b4a5ebf93fa013d213968898c9
root@kali:~/Documents/aes_12499347/src# 
```

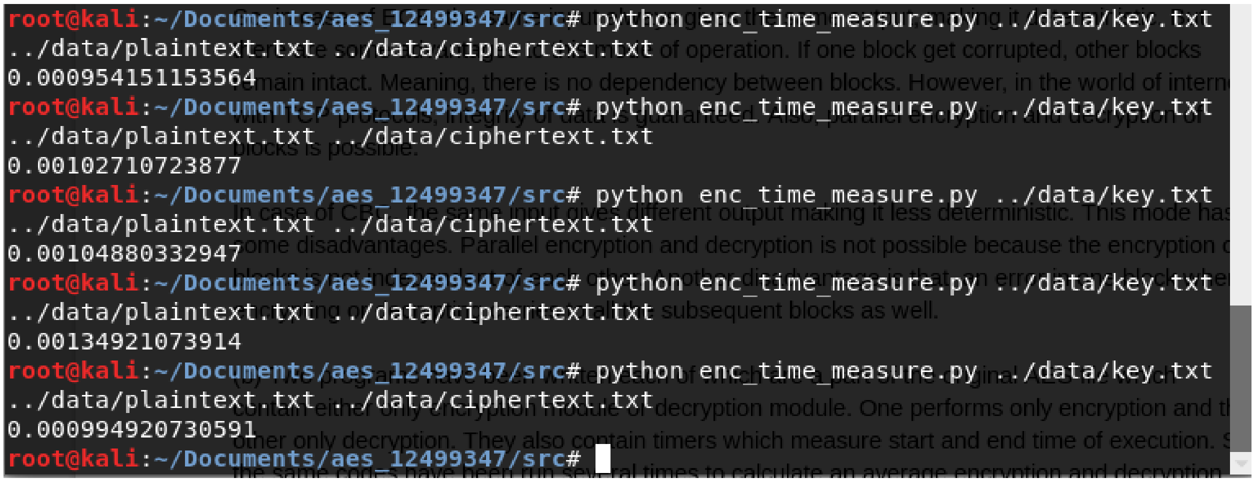
The cipher texts of ECB are same both the times whereas in the case of CBC, the ciphered texts differ because the IV is generated randomly every time. Also, the length of the ciphered texts appear to be different in case of CBC and ECB but they are actually same. CBC has IV prefixed to the ciphered text. Hence the length of cipher texts are actually same. The first 16 bytes (in the output, first 32 letters because each letter is 4 bits in length) is the IV. Clearly, CBC is more secure.

So, in case of ECB, the same input always gives the same output, making it deterministic. But there are some advantages to this mode of operation. If one block get corrupted, other blocks remain intact. Meaning, there is no dependency between blocks. However, in the world of internet with TCP protocols, integrity of data is guaranteed. Also, parallel encryption and decryption of blocks is possible.

In case of CBC, the same input gives different output making it less deterministic. This mode has some disadvantages. Parallel encryption and decryption is not possible because the encryption of blocks is not independent of each other. Another disadvantage is that, an error in one block when encrypting or decrypting carries to all the subsequent blocks as well.

(b) Two programs have been written each of which are a part of the original AES file which contain either only encryption module or decryption module. One performs only encryption and the other only decryption. They also contain timers which measure start and end time of execution. So the same codes have been run several times to calculate an average encryption and decryption times.

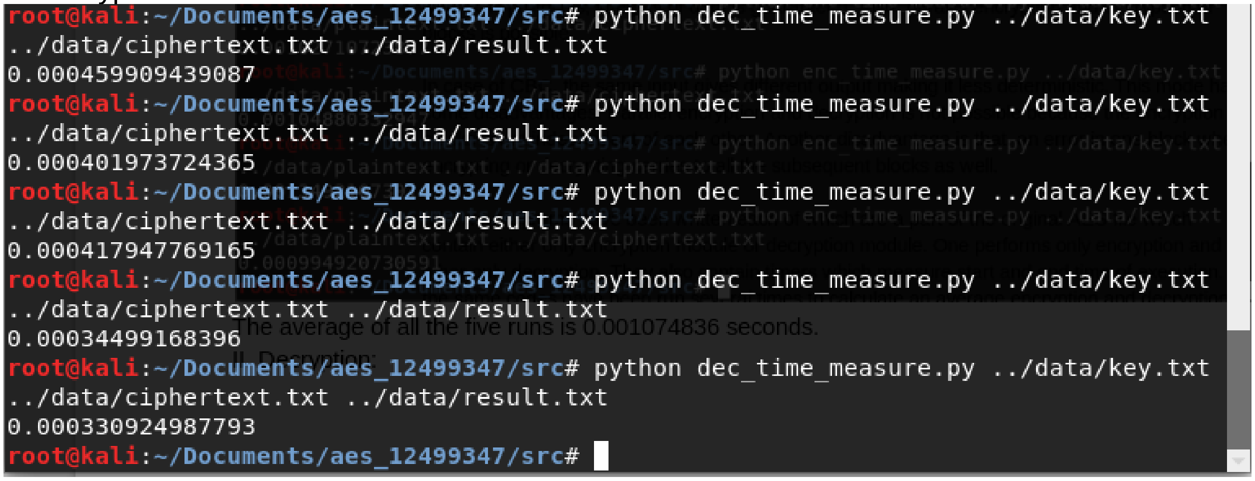
I. Encryption:



```
root@kali:~/Documents/aes_12499347/src# python enc_time_measure.py ../data/key.txt
../data/plaintext.txt ../data/ciphertext.txt
0.000954151153564
root@kali:~/Documents/aes_12499347/src# python enc_time_measure.py ../data/key.txt
../data/plaintext.txt ../data/ciphertext.txt
0.00102710723877
root@kali:~/Documents/aes_12499347/src# python enc_time_measure.py ../data/key.txt
../data/plaintext.txt ../data/ciphertext.txt
0.00104880332947
root@kali:~/Documents/aes_12499347/src# python enc_time_measure.py ../data/key.txt
../data/plaintext.txt ../data/ciphertext.txt
0.00134921073914
root@kali:~/Documents/aes_12499347/src# python enc_time_measure.py ../data/key.txt
../data/plaintext.txt ../data/ciphertext.txt
0.000994920730591
root@kali:~/Documents/aes_12499347/src# 
```

The average of all the five runs is 0.001074836 seconds.

II. Decryption:



```
root@kali:~/Documents/aes_12499347/src# python dec_time_measure.py ../data/key.txt
../data/ciphertext.txt ../data/result.txt
0.000459909439087
root@kali:~/Documents/aes_12499347/src# python dec_time_measure.py ../data/key.txt
../data/ciphertext.txt ../data/result.txt
0.000401973724365
root@kali:~/Documents/aes_12499347/src# python dec_time_measure.py ../data/key.txt
../data/ciphertext.txt ../data/result.txt
0.000417947769165
root@kali:~/Documents/aes_12499347/src# python dec_time_measure.py ../data/key.txt
../data/ciphertext.txt ../data/result.txt
0.00034499168396
root@kali:~/Documents/aes_12499347/src# python dec_time_measure.py ../data/key.txt
../data/ciphertext.txt ../data/result.txt
0.000330924987793
root@kali:~/Documents/aes_12499347/src# 
```

The average of the five runs is 0.000391147 seconds.

The details of the OS used, language, packages and execution guides are mentioned in the readme.txt file.