Chelsea Gantt

10.7.20

AES Report

OS – Mac

Language - Python 3.8.5

I used VSCode to build my project. In order to have it run, go into the src folder and select

the app.py file. If you are in VSCode, you will need to select run at the top of your screen and

click the dropdown either run w/ or w/o debugging. From there the program should run as

desired. It will automatically run each function, with the last one being decryption. All of my txt files are in the data folder.

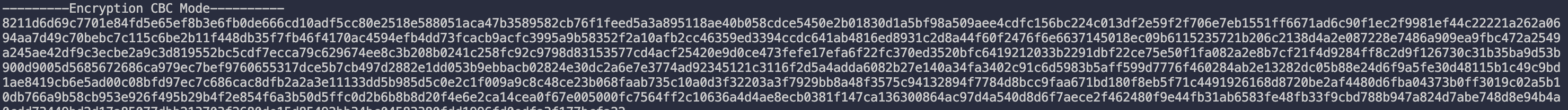
**KeyGen Function**

The keyGen() uses the *binascii* import to create a random hex variable, using urandom(), then decoding it to UTF-8. From there the key is written to key.txt file and printed to the console



**Encryption Function**

The encryption () reads from the key.txt and plaintext.txt to get the values in the AES encryption. From there we generate a random initialization vector, which is generated in the same fashion the secret key was created. In order to do the encryption, I leverage the pycrypto library. During the encryption I multiply the plaintext by 16 to ensure we have 16 byte sized boxes. This is then converted to hex and written to the ciphertext.txt file and printed in the console.



**Decryption Function**

The decryption() reads from the key.txt, iv.txt, and ciphertext.txt to get the values for the AES decryption. Prior to decryption I unhexlify the ciphertext, once the decryption is completed, I revert the length back to its original, then decode it so it does not have the leading ‘b’ prefix. Thereafter the final result is printed to the console and written to the result.txt.

A close up of a sign

Description automatically generated

**Different Encryption Modes**

Upon reviewing each of the ciphertexts for CBC & EBC Mode I wasn’t able to find any obvious differences other than the fact that it was two different ciphertexts generated.

I did notice on decryption I did not have to decode the results to remove the leading ‘b’ prefix. It was already done in the encryption.

**Text

Description automatically generated**

**Average time for Encryption to run: 0.0012 Seconds**

**Average time for Decryption to run: 0.0006 Seconds**

**Calculated using the time function. Available but commented out in code**