Chelsea Gantt
9.28.2020
One Time Pad Report

OS – Mac
Language - Python 3.8.5

I used VSCode to build me project. In order to have it run, go into the srcc 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 keygen where you will have to end a number as input. All of my txt files are in the data folder.

Encryption Function

In the encryption function we first use a conditional statement to check to ensure the plaintext and key are the same length. We get this information by grabbing the key and the plaintext, converting the plaintext to binary and sending both to the function. Then after create a for loop to run the length of the plaintext. Then we xor the plaintext & key. Once that is done, we format the resulting output to ensure it is only 32 bits. Once that is completed, we write the output to the ciphertext file and display in the terminal.

```
0.9.111407/pythonFiles/lib/python/debugpy/launcher 64566
-------- Encryption --------
Key: 01010101101010101111000000001111
Encrypted string: 00110111110011111001000101111101
```

Decryption Function

In the decryption function we first use a conditional statement to check to ensure the key and ciphertext are the same length. We get this information by grabbing the key and the ciphertext, encoding the ciphertext to utf-8 and sending both to the function. Then after create a for loop to run the length of the ciphertext. Then we xor the ciphertext & key. Once that is done, we format the resulting output to ensure it is only 32 bits. We then we convert the decryption from binary to plaintext. Lastly we write the resulting output to the result.txt file and display in the terminal.

```
Encrypted string: 00110111110011111001
-------- Decryption --------
Decrypted string: bear
```

KeyGen Function

In the keygen function we take in security param user input that is verified with and if else statement that it is between 1 and 128. After it is verified, we move into a for loop that will loop for how much was given by user. We then use that to create a random key that is

composed of a random 1 or 0. We then iterate over this temporary key for the duration of the loop creating the new security key. From there we write the new key to the newkey.txt file and display it in the terminal.