# CS 5158/6058 Data Security and Privacy, Fall 2022
# Project 1: One-Time Pad


## Margi Amin

## M15219371


**Software:** Python 3.6.3 is required to run this if you're using Windows 10.

**Program Location**: ..\otp_m15219371\src

**Files and structure:**

Files included in program:

Data

-ciphertext.txt

-key.txt

-newkey.txt

-plaintext.txt

-result.txt

Src

-Decryption.py

-Encryption.py

-EncryptRunTime.py

-KeyFrequency.py

-main.py

-XOREncrypt.py

**Description**:

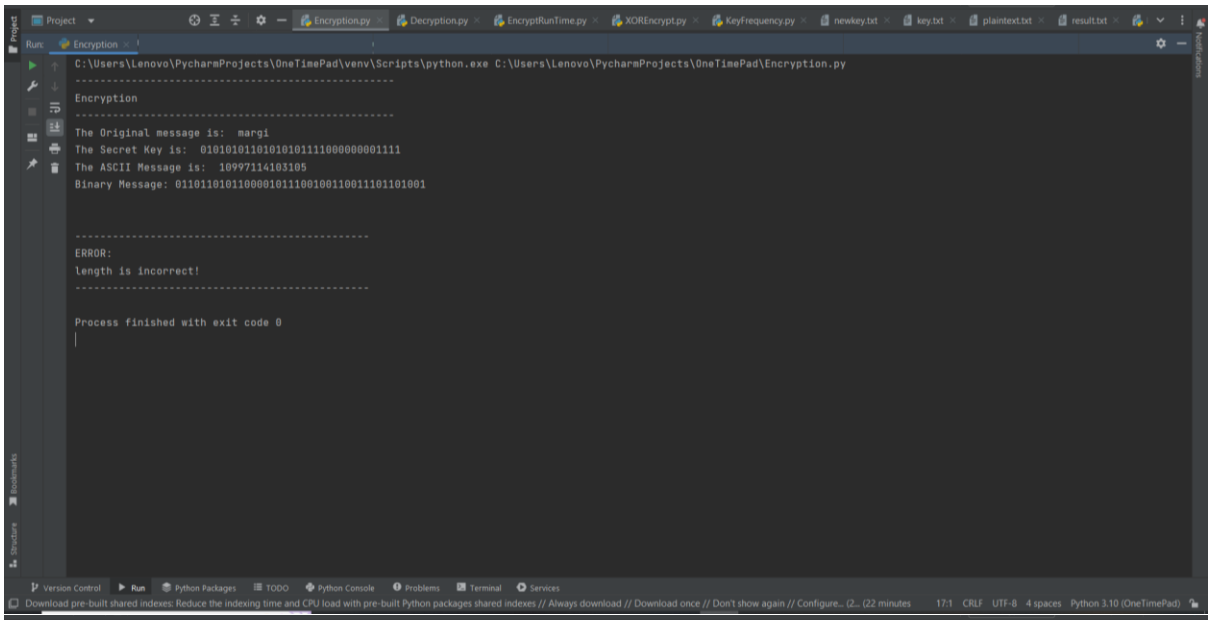
a) **Encryption Function**
This programs generates the ciphertext using message from the plaintext.txt file in the data folder based on the key in the key.txt file of the data folder and stores ciphertext in the ciphertext.txt file in the data folder.
To change the message open the plainytext.txt file and enter the message.
The key stored in the key.txt file is 32bit key given in the project details so if you pass a message longer than 32bits will generate error.

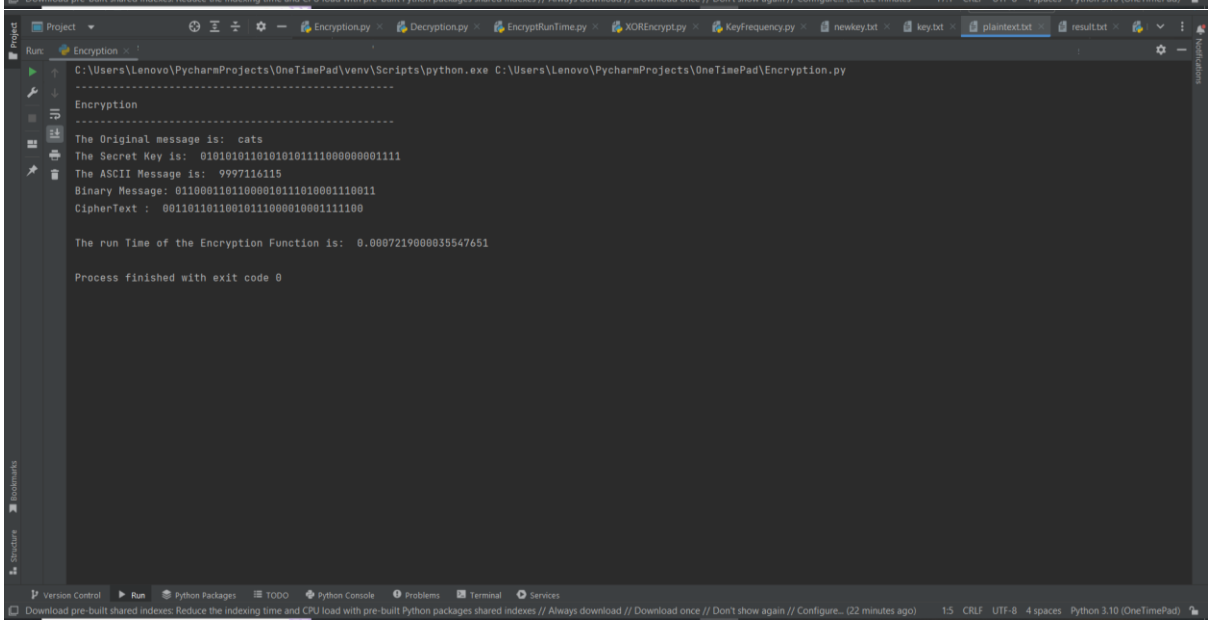
Command to run:
python src\Encryption.py

Output:

Here are screenshots of output of the encryption programs with different input i.e bear, eecs, cats, bike, rice which are 32bits and will generate the ciphertext.

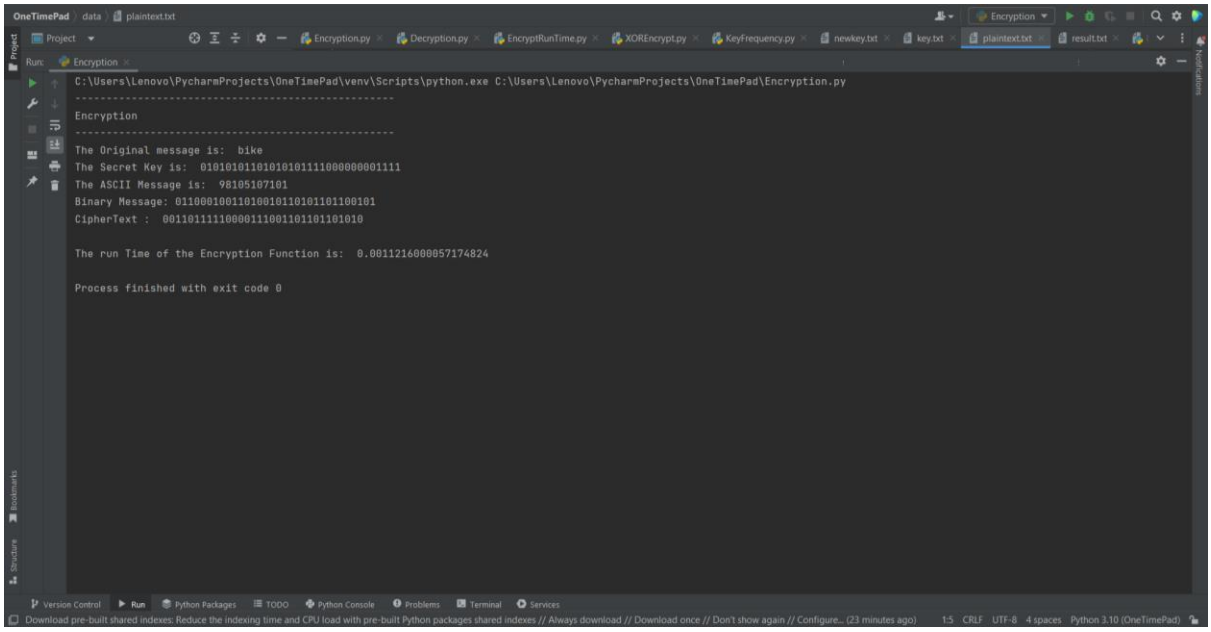
margi which is not 32bits will not generate ciphertext.

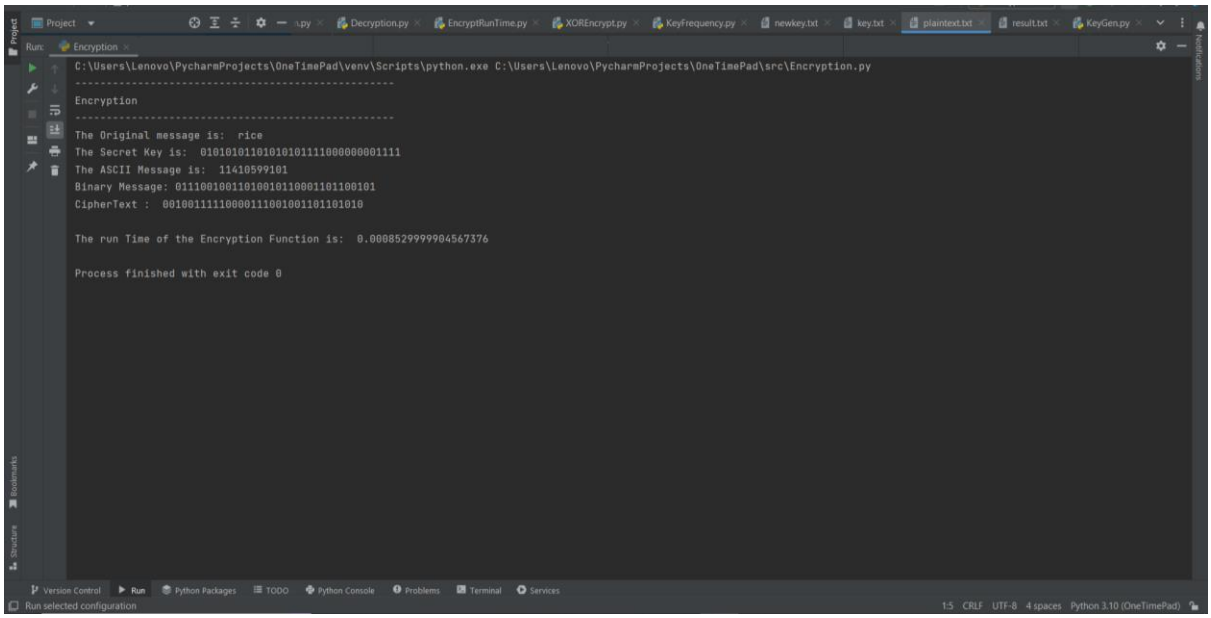


```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\Encryption.py
--------------------------------------------------
Encryption
--------------------------------------------------
The Original message is:  bear
The Secret Key is:  01010101101010101111000000001111
The ASCII Message is:  9810197114
Binary Message: 01100010011001010110000101110010
CipherText :  00110111110011111001000101111101

The run Time of the Encryption Function is:  0.0007301000005099922

Process finished with exit code 0
```

```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\Encryption.py
--------------------------------------------------
Encryption
--------------------------------------------------
The Original message is:  eecs
The Secret Key is:  01010101101010101111000000001111
The ASCII Message is:  10110199115
Binary Message: 01100101011001010110001101110011
CipherText :  00110001100111110010010111111100

The run Time of the Encryption Function is:  0.000826300005428493

Process finished with exit code 0
```




**b) Decryption Function**

This programs generates the original message back from the ciphertext in the ciphertext.txt file and save this message in the result.txt file.
If the key value does not match the length of the ciphertext then it will display error.

Commands to run:
Python src\Decryption.py

Output:
Here are screenshots of output of the decryption programs with different plaintext encrypted. i.e bear, eecs, cats, bike, rice which are 32bits and will encrypt and will generate plaintext back by decryption.
margi which is not 32bits will not encrypt and will not generate plaintext back by decryption.

```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\Encryption.py
-------------------------------------------------
Encryption
-------------------------------------------------
The Original message is:  rice
The Secret Key is:  01010101101010101111000000001111
The ASCII Message is:  11410599101
Binary Message: 01110010011010010110001101100101
CipherText :  00100111110000111001001101101010

The run Time of the Encryption Function is:  0.0007503000015276484

Process finished with exit code 0
```

```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\Decryption.py


-------------------------------------------------
Decryption
-------------------------------------------------
The Secret Key is:  01010101101010101111000000001111
The Ciphertext is:  00110111110011111001000101111101
The deciphered bitStream is:  01100010011001010110000101110010
Message in ASCII is:  1650811250
Here is your Message:  bear

Process finished with exit code 0
```

```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\Decryption.py


-------------------------------------------------
Decryption
-------------------------------------------------
The Secret Key is:  01010101101010101111000000001111
The Ciphertext is:  00110000110011111001001101111100
The deciphered bitStream is:  01100101011001010110001101110011
Message in ASCII is:  1701143411
Here is your Message:  eecs

Process finished with exit code 0
```

```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\Decryption.py


-----------------------------------------------
Decryption
-----------------------------------------------
The Secret Key is:  01010101101010101111000000001111
The Ciphertext is:  00110110110010111000010001111100
The deciphered bitStream is:  01100011011000010111010001110011
Message in ASCII is:  1667331187
Here is your Message:  cats

Process finished with exit code 0
```

```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\Decryption.py


-----------------------------------------------
Decryption
-----------------------------------------------
The Secret Key is:  01010101101010101111000000001111
The Ciphertext is:  00110111110000111001101101101010
The deciphered bitStream is:  01100010011010010110101101100101
Message in ASCII is:  1651075941
Here is your Message:  bike

Process finished with exit code 0
```

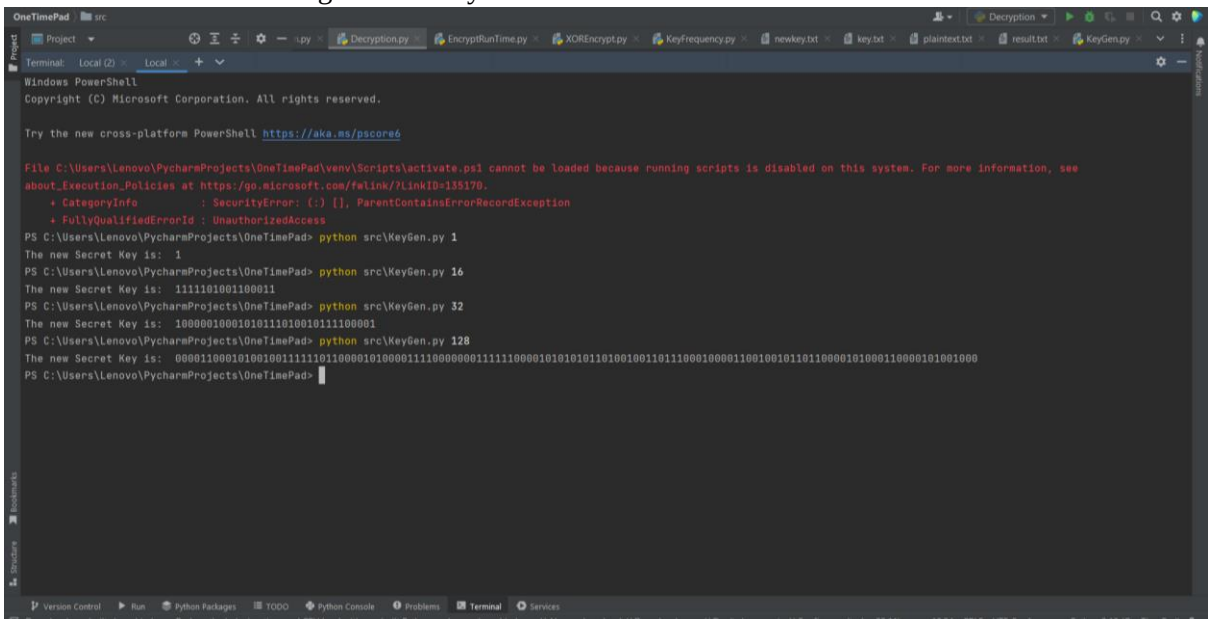
```
C:\Users\Lenovo\PycharmProjects\OneTimePad\venv\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\OneTimePad\src\Decryption.py


-----------------------------------------------
Decryption
-----------------------------------------------


---------------------------------------------------
ERROR:
length is incorrect!
---------------------------------------------------

Process finished with exit code 0
```

**c) Key Genration Function:**

This program generates a new key based on the number we input which must be between 1 and 128(given in the project details). The new generated key will be stored in the newkey.txt.

Command to generate key:
Python src\KeyGen.py (number between 1 and 128)
Example: Python src\KeyGen.py 16

Output:
Here is sceenshot of some generated key.




**d) Distribution of Keys:**

This program runs 5001 times to check how frequently the same newly generated 4-digit secret keys repeat themselves. It maintains a counter and provides a list of keys and a list of frequencies to show how many times the same keys were generated.

The list of Keys and Frequencies looks like this:

| Keys | Frequencies |
|---|---|
| 0000 | 351 |
| 0001 | 307 |
| 0010 | 343 |
| 0011 | 307 |
| 0100 | 275 |
| 0101 | 306 |
| 0110 | 330 |
| 0111 | 299 |
| 1000 | 325 |
| 1001 | 255 |
| 1010 | 295 |
| 1011 | 344 |
| 1100 | 320 |
| 1101 | 317 |
| 1110 | 314 |
| 1111 | 314 |

Command to run:
Python src\KeyFrequence.py

e) **Encryption runtime:**

This program is used to run the Enc() function 10 times and find the average running time that the function for Encryption takes.It will return the average time that the function runs for.

Command:

```
python EncryptRunTime.py
```