Anthony Inlavong
CIT 496
October 12, 2017

Assignment #1

You intercepted a single ciphertext. Decipher it as much as you can. To receive full or partial credit you must show all your work. Attach any code you have implemented (you can use any programming language) and you can use any libraries you find online that is publicly available. You may not use existing code available on online repositories such as GitHub. You must include citations of all sources you used in the report.

HUKUUEUUYREUYYKGRRGNZZKXNGNOLKXTSAQNZYEGNZTRGTOYALSSEH SYGRVSOOLEGIKVKNZOEJYXT

1 Introduction

In order to approach this assignment a large amount of research had to be done in order to understand the complexity of ciphers and how they are deciphered. Hints were given off throughout the class sessions along with the documentation of the project so that was the beginning.

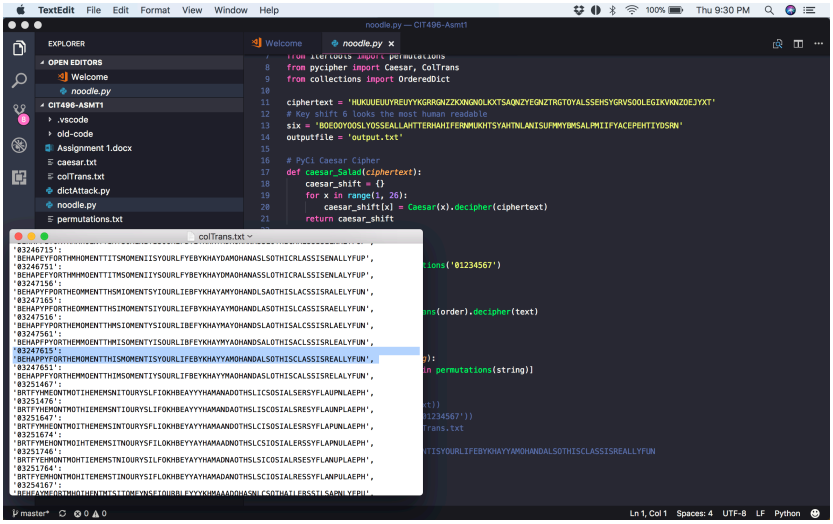
Due to the open nature of the this project, researching Caesar ciphers, Columnar transposition, and much more topics made the assignment intimidating from the start. Site such as learn cryptography[1] and dcode[3] plus many random blogs increased my understanding. After hours of reach I felt confident about the assignment and began cracking down on the cipher.

2 Solving the Cipher

At first, I tried solving by hand which lead me to a dead end. Initially I performed a Caesar Cipher by hand using all available keys which was too time consuming and redundant. Python became my new best friend during this assignment because of my previous experience using it and the many libraries that it came with. At first I wrote a script that would perform a Caesar Cipher just so I can have idea of how to manipulate the string(Cipher). After that I looked at each shift and saw that the sixth shift really stood out because it had a combination of letters that just made sense. I broke it down and pull words from the shifted key by hand and saw some basic words like be, happy, fun, your, ect. It was only until later that this way of analyzing letters is called “Frequency Analysis”. I had trouble writing a script for it, instead I ran it through dcode which spat out some number that didn’t help me much. I was stuck at a brick wall until PyCipher came in.

Along my research I stumbled upon PyCipher[2] which made solving this Cipher extremely less stressful. Built in were tools such as Caesar Cipher and Columnar Transposition which were given in the hints. PyCipher’s method of implementing a Caesar cipher became a drop in replacement from my original method. In order to use PyCipher Column Transposition tool I had to find a key, which created another roadblock because the key was not given. Up until the

professor mentioned the use of itertools[7], a python library, to create permutations of generated keys. I created the Columnar Transposition using a key of 01234567 because the key had to be between 1-10 and there are 77 letters, so 7 fit perfectly. I used the six shifted cipher because I was able to pull some words from it by hand. I ran the python script and appended the results in colTrans.txt. I loaded up the text file in my edited and by pure luck used ctrl+f and typed happy, one of my guessed words which resulted in: **03247615: BEHAPPYFORTHEMOMENTTHISMOMENTISYOURLIFEBYKHAYYAMOHANDALSOTHISCLASSISREALLYFUN**



3 Conclusion

Solving the Cipher was a stroke of luck because my last option was to brute for the colTrans.txt with a dictionary. If I did not luckily pick out the random word “happy” by hand my next python function would have to compare the dictionary values in colTrans to a random wordlist, such as rock4you or /usr/share/dict/words found in unix system. There was also the option to create a python script to automate the whole process to spit out every Caesar Cipher and permutation to use for columnar transposition, then brute forcing those results.

Sources

---

[1] https://learncryptography.com/classical-encryption/caesar-cipher
[2] http://pycipher.readthedocs.io/en/master
[3] http://www.dcode.fr/caesar-cipher
[4] http://www.dcode.fr/frequency-analysis
[5] http://practicalcryptography.com/ciphers/columnar-transposition-cipher/
[6] https://inventwithpython.com/hacking/chapter20.html
[7] https://docs.python.org/3/library/itertools.html#itertools.permutations