CE235: Computer Security

Decrypting a Caesar Cipher using a Brute Force strategy

## The Caesar Cipher

In cryptography, a Caesar cipher is one of the simplest and oldest encryption techniques. It’s a type of substitution algorithm which each letter in the plaintext is replaced by a letter for some fixed number of positions, called shift number, on the alphabet. This encryption method was named after Julius Caesar, who used it in its private correspondence.

For example, with a right shift of 3 positions, the plaintext hello would result in the ciphertext khoor. The encryption is done by shifting each letter of the plaintext 3 shift to the right in the given alphabet [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,w,x,y,z]

The decryption process is done by shifting the ciphertext some number of fixed positions on the alphabet in the opposite direction of the encryption. Following the above example, the decryption of the ciphertext khoor is determined by applying a left shift of 3 positions.

Therefore, the decryption of the ciphertext khoor will result in the plaintext hello only if the decryption is carried out with the same shift number as the encryption.

## Breaking the Cipher with a Brute Force Attack

A Caesar cipher can be easily broken by applying a brute force attack. A brute force attack will try all possible combinations to decrypt the ciphertext and then choose the combination that is more likely to be the original plaintext.

To break the Caesar cipher by using Brute Force, we only need to compute all the shift combinations. As the English alphabet has 26 letters, we must compute 26 different combinations, each one with a different shift number. Therefore, if we want to decrypt the ciphertext khoor, we will compute all possible shifts for the entire ciphertext as:

Shift 1 position to the left: jgnnq

Shift 2 positions to the left: ifmmp

Shift 3 positions to the left: **hello**

...

Shift 25 positions to the left: kipps

The 3 positions shift results in the original plaintext since it’s the only one that makes sense in English language.

To determine which shift combination is the original plaintext you’re provided with a Natural Language Tool that you don’t need to worry about.

Note: You do not need to understand the code of the Natural Language Tool (though if you are interested in computers and programming it is well worth looking).

## Provided Files

You’re provided with two python files and two text files; DecryptMessage.py, ScoresDict.py, ciphertext.txt, ciphertext.txt and the\_hunger\_games.txt

* The DecryptMessage.py module contains the code for decrypting a Caesar cipher that **you need to complete.**
* The ScoresDict.py module contains a Natural Language Toolkit used to determine which of the 26 shift combinations is more likely to be the original plaintext by analysing the language. **This code is provided to you and you don’t need to modify it.**
* The ciphertext.txt file contains the ciphertext you have to decrypt.
* The the\_hunger\_games.txt file contains the book The Hunger Games used to train the Natural language Tool.

## Your Work

You have intercepted a ciphertext and you know it’s been encrypted with a Caesar method, but you ignore the shift number used to encrypt. Your work is to complete the code given at the file *DecryptMessage.py* so that it decrypts the ciphertext using a Brute Force strategy.

These is the ciphertext intercepted:

te td fyvyzhy szh pqqpnetgp esp nlpdlc ntaspc hld le esp etxp, mfe te td wtvpwj ez slgp mppy cpldzylmwj dpnfcp, yze wplde mpnlfdp xzde zq nlpdlc'd pypxtpd hzfwo slgp mppy twwtepclep lyo zespcd hzfwo slgp lddfxpo esle esp xpddlrpd hpcp hcteepy ty ly fyvyzhy qzcptry wlyrflrp.

You have to complete the following tasks :

* Complete the decrypt(ciphertext) method. These method has to compute all the 26 shift combinations and determine which one is the plaintext by using the provided Natural Language Tool, calling the scores\_dict.argmax(all\_shifts)method.
* Complete the shift(ciphertext,shift) method. These method receives as argument a ciphertext and a fixed shift number. It has to **perform the shift for every letter in the ciphertext.**
* Complete the shift\_char(char,shift) method. These method receives as argument a character and a fixed shift number. It has to **perform the shift for the given character.**

## Hints

* The shift\_char(char,shift)method must return the new shifted character. Therefore, if the method arguments are shift\_char(char=”c”,shift=3), it should shift the given character c, 3 times to the right and return the shifted character f.If the index position within the alphabet of the character a is and , then the index within the alphabet of the shifted character would be .Note that you will need to use modulus operations to make the alphabet cyclic, so that if you try to shift the letter z 3 positions , you obtain the index 2 that corresponds to the letter c as result of the shifting.
* The shift(ciphertext,shift)method must return the ciphertext shifted. For every letter in the ciphertext. It will shift it by calling shift\_char(char,shift) and store it into a new variable.
* To obtain the shifted letter index given letter and fixed shift number , you must use modulus operations to compute the new letter index in the alphabet as

When your code is run, you should see that the decryption of the ciphertext is written to the plaintext.txt file.

## Execute your code

Your code have to be executed from the command terminal as

python3 DecryptMessage.py