Cryptography and Network Security Lab

Assignment 2  
Performing Cryptanalysis on Caesar Cipher

2019BTECS00058  
Devang K

Batch: B2

Title: Performing Cryptanalysis on Caesar Cipher.

Aim: To perform cryptanalysis on Caesar Cipher and guess the shift-key value from an encryption.

Theory:

Caesar Cipher is named after ‘Julius Caesar’ who used to use this cipher algorithm in his private correspondence. It is a form of Substitution Cipher wherein we replace each letter in the plaintext by a letter some fixed number of positions ahead/behind. Let us called these number of positions as ‘shift key’.

We studied the Caesar Cipher in Assignment 1 and implemented the logic to encrypt and decrypt it. The shift-key of the Caesar Cipher is cyclic. Meaning, a shift-key of 0 and a Shift-Key of 26 would give the same result. Therefore, the Cipher can have only 26 alternatives – one of which would be the right Shift-Key.

We exploit this vulnerability of the Cipher algorithm by performing a dictionary attack. For a given word of Ciphertext, we make a list of all the possible words and then check if they exist in our dictionary, the word among the list which does exist, would give us the corresponding key.

Code:

Python has a library called ‘enchant’ which has a large dictionary of words and has a function to check whether a word exists in that dictionary. We shall make use of the same.

For the given Ciphertext word, we generate all the combinations for all shift-keys and then one-by-one compare if the generated word exists in the enchant dictionary.

Here is the code implementation of breaking the Caesar Cipher.

*import* enchant

d = enchant.Dict("en\_US")

*def* shiftCharWithKeyDecrypt(c, step):

    amt = ord(c)

    amt -= 65

    amt -= step

    amt %= 26

    amt += 65

*return* chr(amt)

print("Enter the Cipherword: ",end='')

cipherword = input()

*# Get all shift combos to guess the shift amt*

*# From 0 to 26*

possibleShiftedCombos = []

*for* shift *in* range(26):

    theString = ""

*for* i *in* cipherword:

        theString += shiftCharWithKeyDecrypt(i, shift)

    possibleShiftedCombos.append(theString)

shiftValue = 0

*for* i *in* range(len(possibleShiftedCombos)):

*if* d.check(possibleShiftedCombos[i]):

        shiftValue = i

*break*

decryptedText = ""

*for* i *in* cipherword:

    decryptedText += shiftCharWithKeyDecrypt(i, shiftValue)

print("Possible Values of Plaintext:")

print(possibleShiftedCombos)

print()

print("Ciphertext: "+cipherword)

print("Key:", shiftValue)

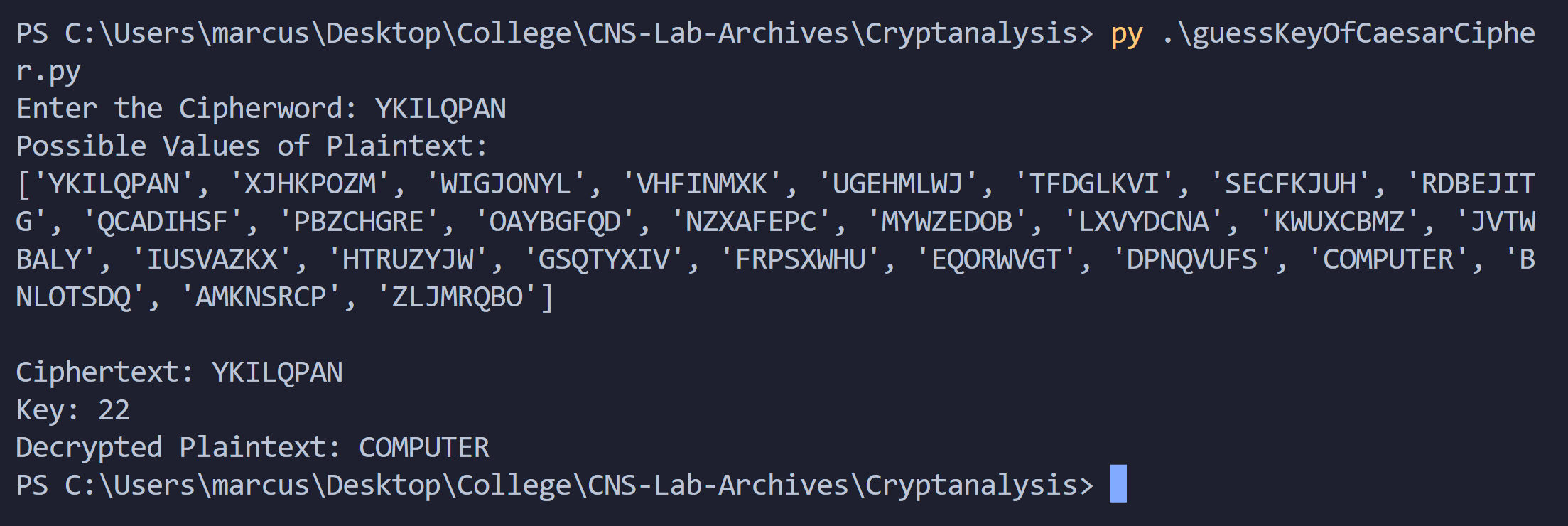
print("Decrypted Plaintext: "+decryptedText)

We now solve some examples with the code.  
Say we have an encrypted word ‘OVSPKHF’. Let’s put it through our code.



We thus generated 26 possible values and checked to find that word at index 7 is valid – ‘HOLIDAY’. Therefore, the shift-key was 7.

We take another example:  
Say we have an encrypted word ‘YKILQPAN’. Let’s put it through our code.



We thus generated 26 possible values and checked to find that word at index 22 is valid – ‘COMPUTER’. Therefore, the shift-key was 22.

Conclusion:

Thus, we cryptanalyzed the Caesar Cipher and broke the algorithm using dictionary attack. Thus, Caesar Cipher is not secure in modern cryptography. Also, this can only be done on a Cipher word and not on any sentence.

A potential drawback of this approach is also that, we may receive ambiguous answer, if 2 or more words are in the dictionary among the possible combinations.