Cryptography & Network Security

PRN - 2019BTECS00026
Name - Niraja Vasudev Kulkarni
Batch - B1

Assignment - 2

- **Title - Cryptanalysis (Decryption of Caeser Cipher)**

- **Objective -**

Decrypting the cipher text encrypted using Caesar Cipher

- **Theory** -

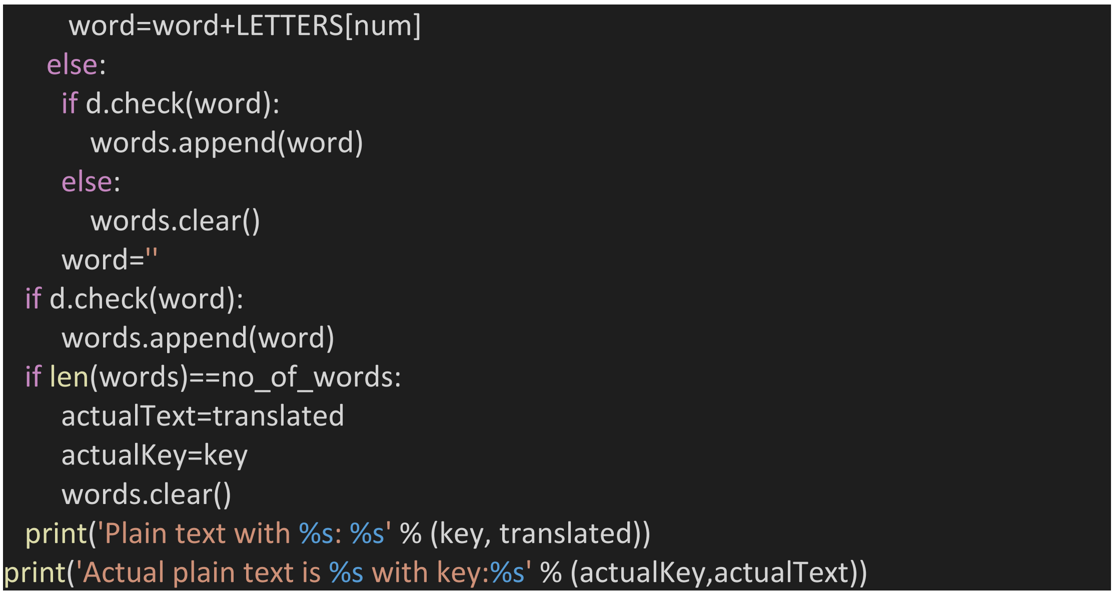
Cryptanalysis is the decryption and analysis of codes, ciphers or encrypted text. Here , the task is to perform cryptanalysis that is , to decrypt the cipher text which is actually encrypted by the Caeser Cipher . We can write another function decrypt that'll apply the shift in the opposite direction to decrypt the original text. The shift is not known , so we will have to try all possible combinations and find out which one gives meaningful output.

Procedure -

1. Take the cipher text as an input from the user
2. Considering all possible 26 shifts , decrypt the given text by applying shift in opposite direction
3. Find the meaningful output using PyEnchant library that returns whether the translated word is present in the dictionary or not

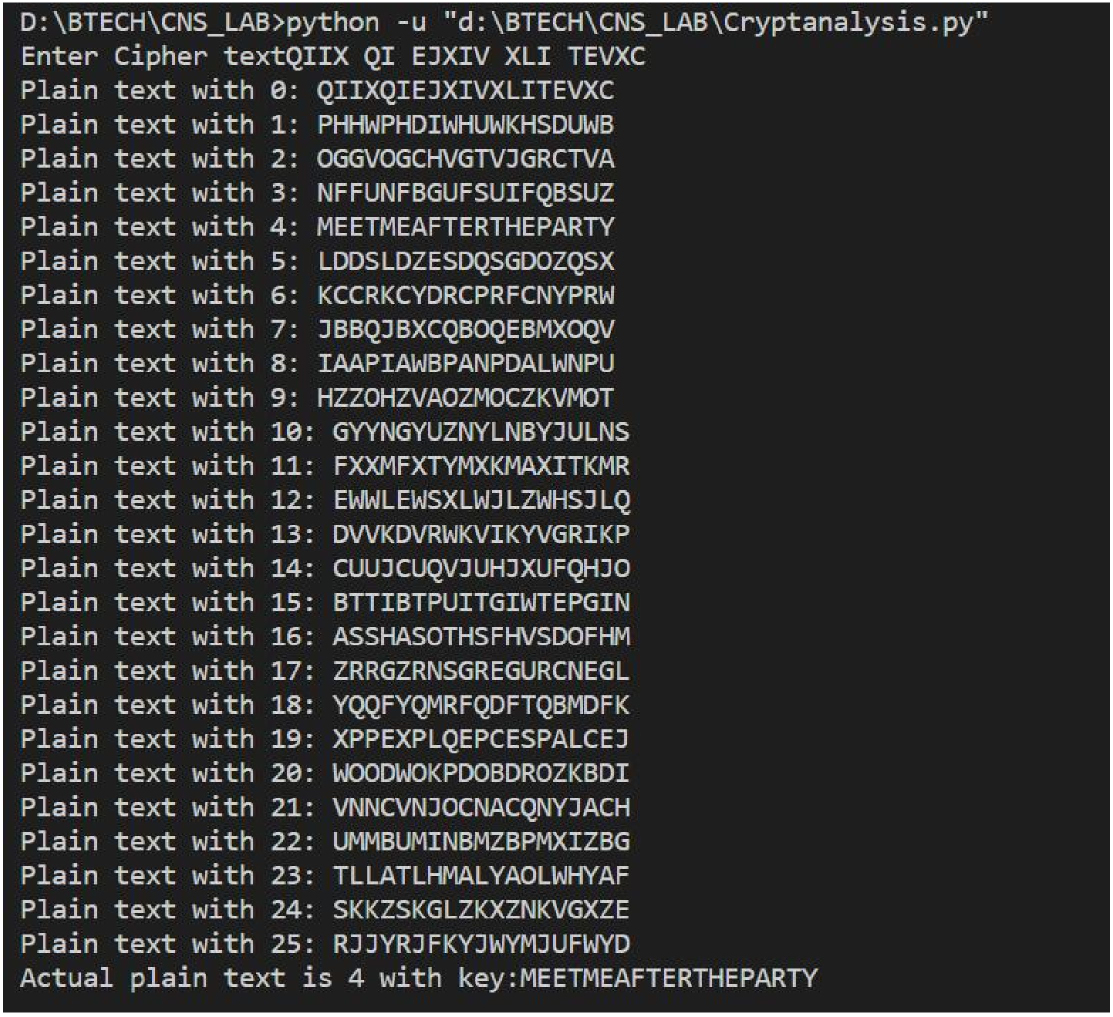
- **Code snapshots -**

```
import enchant
d = enchant.Dict("en_US")
message = input('Enter Cipher text')
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
actualText=''
actualKey=0
no_of_words=len(message.split())
words=[]
for key in range(len(LETTERS)):
  translated = ''
  word=''
  for symbol in message:
    if symbol in LETTERS:
      num = LETTERS.find(symbol)
      num = num - key
      if num < 0:
        num = num + len(LETTERS)
      translated = translated + LETTERS[num]
```



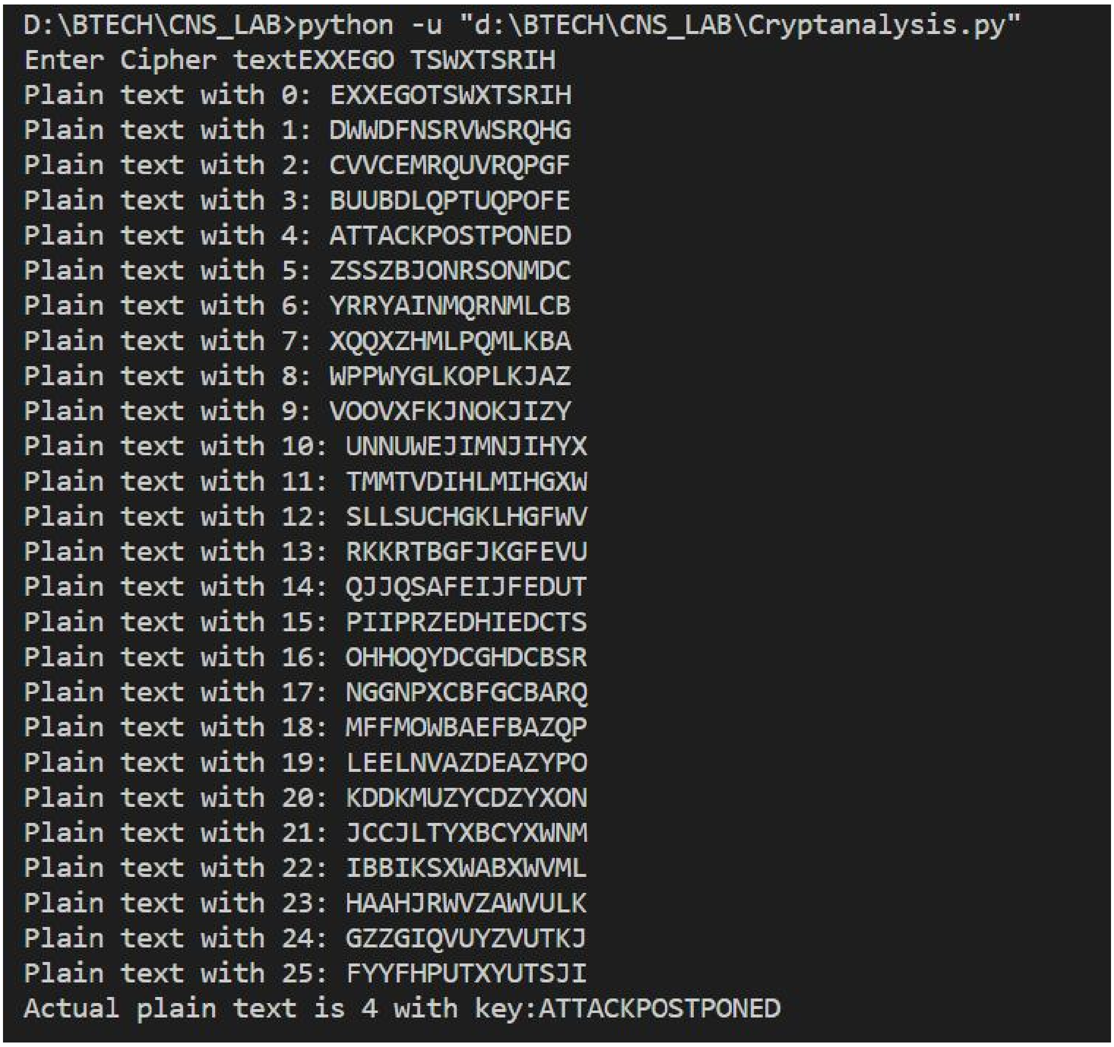
```
        word=word+LETTERS[num]
    else:
      if d.check(word):
          words.append(word)
      else:
          words.clear()
      word=''
  if d.check(word):
      words.append(word)
  if len(words)==no_of_words:
      actualText=translated
      actualKey=key
      words.clear()
  print('Plain text with %s: %s' % (key, translated))
print('Actual plain text is %s with key:%s' % (actualKey,actualText))
```

- **Outputs -**
- **Sample Output 1 -**



```
D:\BTECH\CNS_LAB>python -u "d:\BTECH\CNS_LAB\Cryptanalysis.py"
Enter Cipher textQIIX QI EJXIV XLI TEVXC
Plain text with 0: QIIXQIEJXIVXLITEVXC
Plain text with 1: PHHWPHDIWHUWKHSDUWB
Plain text with 2: OGGVOGCHVGTVJGRCTVA
Plain text with 3: NFFUNFBGUFSUIFQBSUZ
Plain text with 4: MEETMEAFTERTHEPARTY
Plain text with 5: LDDSLDZESDQSGDOZQSX
Plain text with 6: KCCRKCYDRCPRFCNYPRW
Plain text with 7: JBBQJBXCQBOQEBMXOQV
Plain text with 8: IAAPIAWBPANPDALWNPU
Plain text with 9: HZZOHZVAOZMOCZKVMOT
Plain text with 10: GYYNGYUZNYLNBYJULNS
Plain text with 11: FXXMFXTYMXKMAXITKMR
Plain text with 12: EWWLEWSXLWJLZWHSJLQ
Plain text with 13: DVVKDVRWKVIKYVGRIKP
Plain text with 14: CUUJCUQVJUHJXUFQHJO
Plain text with 15: BTTIBTPUITGIWTEPGIN
Plain text with 16: ASSHASOTHSFHVSDOFHM
Plain text with 17: ZRRGZRNSGREGURCNEGL
Plain text with 18: YQQFYQMRFQDFTQBMDFK
Plain text with 19: XPPEXPLQEPCESPALCEJ
Plain text with 20: WOODWOKPDOBDROZKBDI
Plain text with 21: VNNCVNJOCNACQNYJACH
Plain text with 22: UMMBUMINBMZBPMXIZBG
Plain text with 23: TLLATLHMALYAOLWHYAF
Plain text with 24: SKKZSKGLZKXZNKVGXZE
Plain text with 25: RJJYRJFKYJWYMJUFWYD
Actual plain text is 4 with key:MEETMEAFTERTHEPARTY
```

- Sample Output 2 -



```
D:\BTECH\CNS_LAB>python -u "d:\BTECH\CNS_LAB\Cryptanalysis.py"
Enter Cipher textEXXEGO TSWXTSRIH
Plain text with 0: EXXEGOTSWXTSRIH
Plain text with 1: DWWDFNSRVWSRQHG
Plain text with 2: CVVCEMRQUVRQPGF
Plain text with 3: BUUBDLQPTUQPOFE
Plain text with 4: ATTACKPOSTPONED
Plain text with 5: ZSSZBJONRSONMDC
Plain text with 6: YRRYAINMQRNMLCB
Plain text with 7: XQQXZHMLPQMLKBA
Plain text with 8: WPPWYGLKOPLKJAZ
Plain text with 9: VOOVXFKJNOKJIZY
Plain text with 10: UNNUWEJIMNJIHYX
Plain text with 11: TMMTVDIHLMIHGXW
Plain text with 12: SLLSUCHGKLHGFWV
Plain text with 13: RKKRTBGFJKGFEVU
Plain text with 14: QJJQSAFEIJFEDUT
Plain text with 15: PIIPRZEDHIEDCTS
Plain text with 16: OHHOQYDCGHDCBSR
Plain text with 17: NGGNPXCBFGCBARQ
Plain text with 18: MFFMOWBAEFBAZQP
Plain text with 19: LEELNVAZDEAZYPO
Plain text with 20: KDDKMUZYCDZYXON
Plain text with 21: JCCJLTYXBCYXWNM
Plain text with 22: IBBIKSXWABXWVML
Plain text with 23: HAAHJRWVZAWVULK
Plain text with 24: GZZGIQVUYZVUTKJ
Plain text with 25: FYYFHPUTXYUTSJI
Actual plain text is 4 with key:ATTACKPOSTPONED
```

- **Conclusion** -

Caeser Cipher is a monoalphabetic classical cipher which can be easily decrptyed. It is a naive way of encrypting. Here , PyEnchant lilbrary is used for finding the meaningful output from suggested set of sentences as the shift is unknown.