

Challenge: Friends

(Cryptography)

Description.txt:

My friends along with me found the flag, but the bully encrypted all the files on the school computer system, he also encrypted the reference for spell test. Please look for things that can help us.

flag.txt:

brcfxba_vfr_mid_hosbrm_iprc_exa_hoav_vwcrm

encrypted.py:

```
import random
import os

files = [
    os.path.join(path, file)
    for path, dirs, files in os.walk('.')
    for file in files
    if file.split('.')[-1] == 'txt'
]
```

```
alphabet = list('abcdefghijklmnopqrstuvwxyz')
random.shuffle(shuffled := alphabet[:])
dictionary = dict(zip(alphabet, shuffled))
```

```
for filename in files:
    text = open(filename, 'r').read()
    encrypted = ''.join([
        dictionary[c]
        if c in dictionary else c
        for c in text
    ])
    open(filename, 'w').write(encrypted)
```

From the above lines of code the following 3 lines seem to be of interest :

```
alphabet = list('abcdefghijklmnopqrstuvwxyz')
random.shuffle(shuffled := alphabet[:])
dictionary = dict(zip(alphabet, shuffled))
```

A list 'alphabet' is defined with all lowercase letters, which is then being shuffled followed by a dictionary creation of the alphabet and the shuffled characters, which means that each alphabet letter maps to one of the shuffled letters.

So what does that remind us of? A **Substitution Cipher**.


Now we can dig into the rest of this to see if it's basically doing the same thing, i.e., replacing characters of the alphabet list with the characters corresponding to them from the dictionary. It's looping through the flag file and replacing each of the plaintext characters with the shuffled character corresponding to the letter from the alphabet list.

(Sorry **Reference.txt** was just a distraction although those words can also be decrypted using the same procedure.)

Next we will be analyzing the frequency of each letter in the flag for that we can use any frequency analyzer tool:

Quick note: In English language it is noticed that the most repeating alphabet in a sentence is usually 'e'.

Here we notice that 'r' is being used multiple times, therefore we can assume that 'e' is substituted by 'r' in the encrypted text



Search for a tool

★ SEARCH A TOOL ON dCode BY KEYWORDS:

★ BROWSE THE FULL dCode TOOLS LIST

Results

Occurrence and Frequency Analysis
1-grams

	11	11	11	11
R		5×	14.29%	
B		3×	8.57%	
C		3×	8.57%	
A		3×	8.57%	
V		3×	8.57%	
M		3×	8.57%	
F		2×	5.71%	
X		2×	5.71%	
I		2×	5.71%	
H		2×	5.71%	
O		2×	5.71%	
D		1×	2.86%	
S		1×	2.86%	
P		1×	2.86%	
E		1×	2.86%	
W		1×	2.86%	

#N : 16 Σ = 35.000 Σ = 99.990 #N : 16

FREQUENCY ANALYSIS

Cryptography > Cryptanalysis > Frequency Analysis

FREQUENCY ANALYSIS (ADVANCED)

★ TEXT TO ANALYZE

★ PLAINTEXT EXPECTED LANGUAGE English

TARGET CHARACTERS FOR FREQUENCY ANALYSIS

☒ LETTERS (A-Z) ONLY
☐ LETTERS (A-Z) AND DIGITS (0-9) ONLY
☐ DIGITS (0-9) ONLY
☐ ONLY THESE CHARACTERS:
☐ ALL EXCEPT SPACES
☐ ALL (INCLUDING SPACES, PUNCTUATION AND SYMBOLS)
★ STANDARDIZE LETTERS (IGNORE UPPER-LOWER CASE AND DIACRITICS) ☒

ITEMS TO ANALYZE

☒ EACH CHARACTER SEPARATELY
☐ BIGRAMS (COUPLES OF 2 CHARACTERS)
☐ TRIGRAMS (SET OF 3 CHARACTERS)
☐ N-GRAMS N=
★ (FOR NGRAMS)
☒ BLOCKS ANALYSIS (ABCDEF => AB,CD,EF)
☐ SLIDING WINDOW/OVERLAPPING (ABCDEF => AB,BC,CD,DE,EF)

Further we may use any cryptogram tool (such as QuipQuip)

Clue : R=E

Solve: Statistics

The highest likelihood score was received by :

0 -1.665 perhaps_the_dog_jumped_over_was_just_tired

Therefore we found out the flag to be:

iotctf{perhaps_the_dog_jumped_over_was_just_tired}

quipqiup beta3

quipqiup is a fast and automated cryptogram solver by [Edwin Olson](#). It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (inwhi chwor dboun darie saren t).

Puzzle:

brcfxba_vfr_mid_hosbrm_iprc_exa_hoav_vwcrm

Clues: For example G=R QVW=THE

r=e

Ads by Google

Send feedback

Why this ad? ⓘ

solve (statistics)
solve (patristocrat)
solve (dictionary)
just a clue
scramble
vigenere
group in 5s
lowercase
uppercase
undo

0	-1.665	perhaps_the_dog_jumped_over_was_just_tired
1	-2.186	perhaps_the_dog_limped_over_was_list_tyred
2	-2.346	perhaps_the_dun_comped_uyer_was_cost_tired