SIL765 - Net. & Sys. Security
CSE, IIT Delhi (SEM2, 2020-21)

2018CS10388
Sharique Shamim

Assignment 1

1 Basic Cryptanalysis

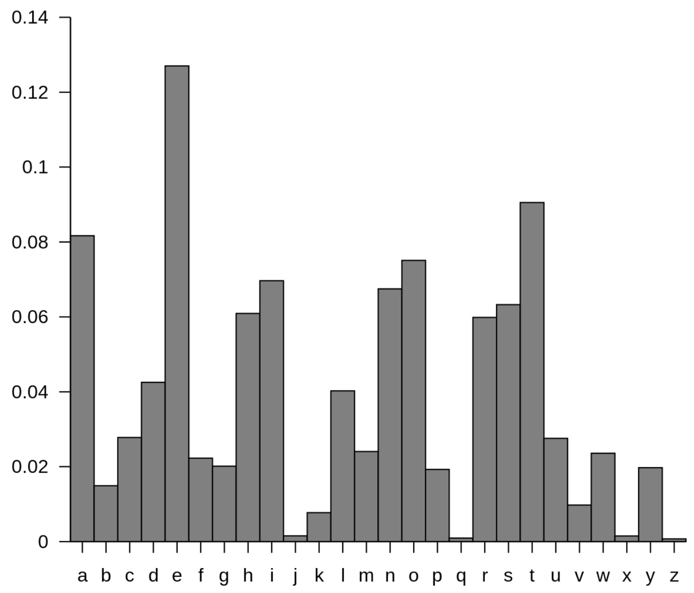
To run the script do:

```
cd problem-1
python3 substitutioncrack.py cipher1.txt
python3 substitutioncrack.py cipher2.txt
```

We have seen that there are too many possible keys to try in a brute force attack in the Mixed Alphabet Cipher, and given that we could also use symbols in our substitution, there are infinitely many different keys for a Monoalphabetic Substitution Cipher. Despite this, however, every single example of this type of cipher is easily broken, using a single method that works on all of them: Frequency Analysis.

The methodology behind frequency analysis relies on the fact that in any language, each letter has its own personality. The most obvious trait that letters have is the frequency with which they appear in a language. Clearly in English the letter "Z" appears far less frequently than, say, "A". In times gone by, if you wanted to find out the frequencies of letters within a language, you had to find a large piece of text and count each frequency. Now, however, we have computers that can do the hard work for us. But in fact, we don't even need to do this step, as for most languages there are databases of the letter frequencies, which have been calculated by looking at millions of texts, and are thus very highly accurate.

From these databases we find that "E" is the most common letter in English, appearing about 12% of the time (that is just over one in ten letters is an "E"). The next most common letter is "T" at 9%. The full frequency list is given by the graph below.



This procedure is repeated in the same way using frequecny anaylysis for digrams, trigrams, ngrams($n > 4$) and so on. And the cipher text is substituted until a desired plain text is obtained. The frequency table for digrams and higher order can be found here:

https://en.wikipedia.org/wiki/Frequency_analysis

http://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html

Cipher1.txt

a disadvantage of the general monoalphabetic cipher is that both sender and receiver must commit the permuted cipher sequence to memory. a common technique for avoiding this is to use a keyword from which the cipher sequence can be generated. for example, using the keyword cipher, write out the keyword followed by unused letters in normal order and match this against the plaintext letters. make reasonable assumptions about how to treat redundant letters and excess letters in the memory words and how to treat spaces and punctuation. indicate what your assumptions are. note, the message is from the sherlock holmes novel, the sign of four.

Cipher1 Mapping:

cipher = ['v', 'o', 'q', '9', '$', 'p', '1', '@', '8', '#', '5', 'y', 's', 't', '2', '6', '7', 'u', '0', '3', '4', 'x', 'w', 'r']

plain = ['a', 't', 'h', 'e', 'o', 'f', 'b', 'n', 'd', 'w', 'i', 's', 'u', 'c', 'r', 'v', 'g', 'l', 'm', 'p', 'q', 'y', 'k', 'x']

Each character of cipher was mapped from the corresponding character of plaintext.

Cipher2.txt

defeated and leaving his dinner untouched, he went to bed. that night he did not sleep well, having feverish dreams, having no rest. he was unsure whether he was asleep or dreaming. conscious, unconscious, all was a blur. he remembered crying, wishing, hoping, begging, even laughing. he floated through the universe, seeing stars, planets, seeing earth, all but himself. when he looked down, trying to see his body, there was nothing. it was just that he was there, but he could not feel anything for just his presence.

Cipher2 Mapping:

cipher = ['1', 'n', 'q', '4', 'v', 'r', '8', '@', 't', 'z', 'y', '5', 'p', '$', '9', '7', '#', '3', 's', '2', 'w', '6', 'x']

plain = ['a', 'h', 'e', 'l', 's', 'i', 'p', 't', 'r', 'd', 'n', 'w', 'g', 'o', 'v', 'b', 'u', 'c', 'm', 'y', 'f', 'k', 'j']

Each character of cipher was mapped from the corresponding character of plaintext.

Conclusion: Better encryption techniques should be used, as susbtitution cipher can be easily broken.