

Project 2: Playfair Cipher and Vigenère Cipher

Design Document

Francis Jose Patron Fidalgo

Student Number: 802180833

CIIC 5018 -060

9/4/2022

Table of Contents

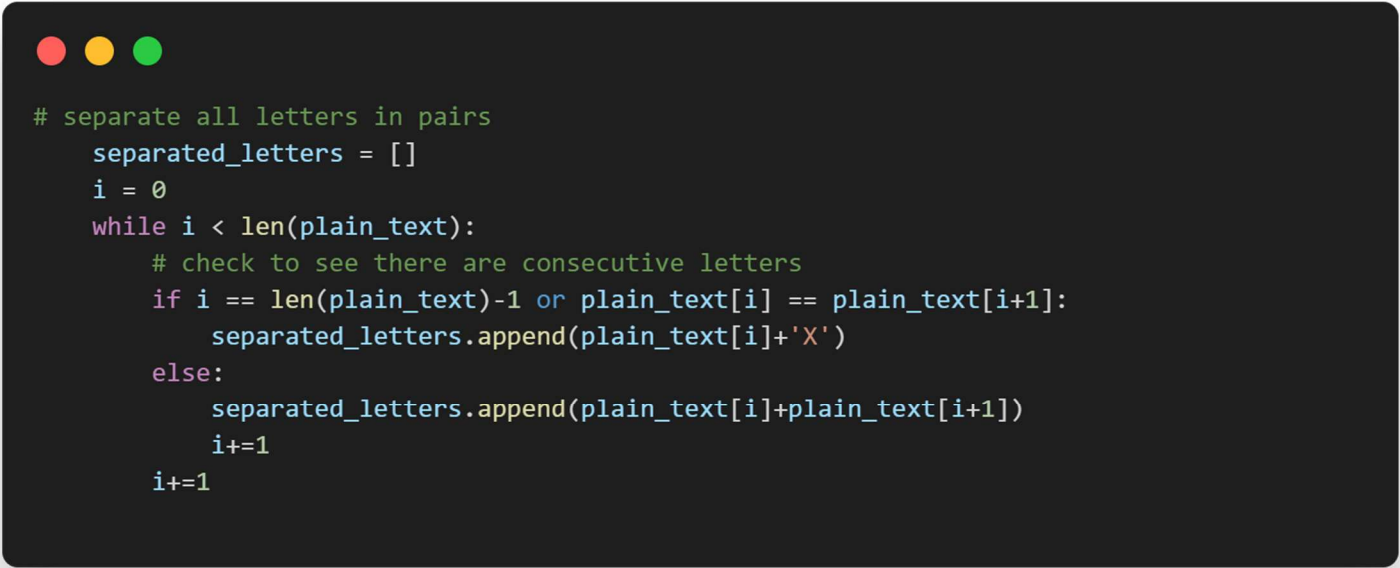
Section 1: The Playfair Cypher	3
Section 2: The Vigenère Cypher	4
Section 3: The Relative Frequencies of a Letter in a String	5
References	6

Section 1: The Playfair Cypher

The first thing we need to do in the Playfair cypher is generate the cypher matrix. This is achieved by replacing all “J” letters with “I” and then filling out each row in the 5 x 5 matrix from top to bottom & left to right. Once we have this matrix we can proceed with the encryption. To prepare the plain text, we use the function from project 1 to convert everything to only uppercase English letters. Then, we iterate over this plain text and create pairs of letters. Note that there are two special cases in this step of the cypher (implementation in Figure 1):

- 1) If two letters that are next to each other are the same, add an “X” between them and proceed with creating the pairs.
- 2) If the plain text has an odd number of characters, add an “X” to the end.

After the plain text is separated into letter pairs, we proceed with encrypting each pair using the three rules of this encryption. The decryption of this algorithm follows the exact same logic as the encryption, but in reverse.



```
# separate all letters in pairs
separated_letters = []
i = 0
while i < len(plain_text):
    # check to see there are consecutive letters
    if i == len(plain_text)-1 or plain_text[i] == plain_text[i+1]:
        separated_letters.append(plain_text[i]+'X')
    else:
        separated_letters.append(plain_text[i]+plain_text[i+1])
        i+=1
    i+=1
```

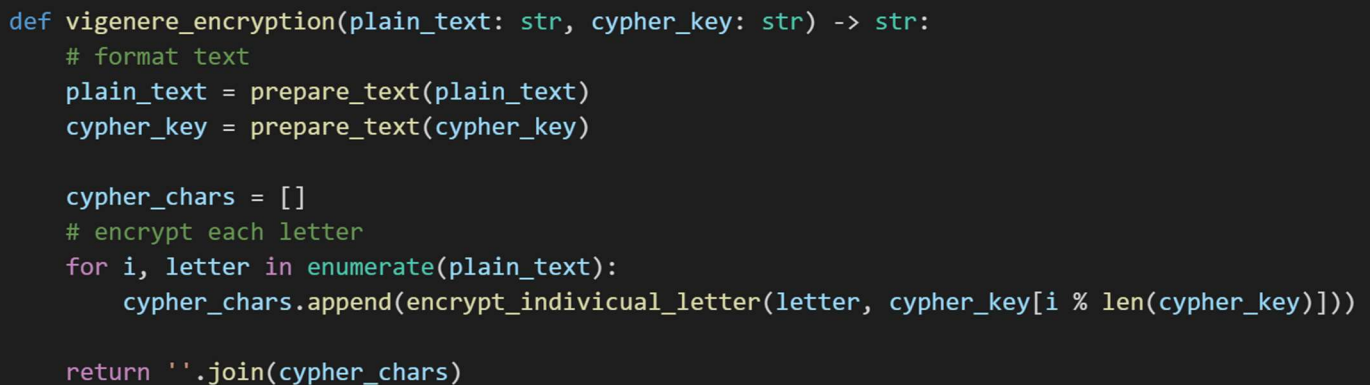
Figure 1: Code snippet of encryption algorithm showing the handling of the two special cases

Section 2: The Vigenère Cypher

This encryption method borrows the logic from project 1's Caesar cypher, having a similar encryption equation of:

$$C_i = (p_i + k_{i \bmod m}) \bmod 26$$

This means this cypher uses a different Caesar cypher key for each character in the plain text. To encrypt a message, we simply iterate over the plaintext and apply the Caesar encryption to each letter. The corresponding shift number for a letter in the plain text is defined by its equivalent position in the key. We use the modulus operation when retrieving said key to ensure the cypher key string can be repeating. Figure 2 demonstrates the cypher implementation. The decryption follow the exact same logic but in reverse.

A code block with a dark background and light-colored text, featuring three colored circles (red, yellow, green) in the top-left corner. The code defines a function for Vigenère encryption.

```
def vigenere_encryption(plain_text: str, cypher_key: str) -> str:
    # format text
    plain_text = prepare_text(plain_text)
    cypher_key = prepare_text(cypher_key)

    cypher_chars = []
    # encrypt each letter
    for i, letter in enumerate(plain_text):
        cypher_chars.append(encrypt_individual_letter(letter, cypher_key[i % len(cypher_key)]))

    return ''.join(cypher_chars)
```

Figure 2: Vigenère encryption implementation in python.

Section 3: The Relative Frequencies of letters in a string

This section uses three texts:

- 1) the project description word document text contents
- 2) the same text as above but encrypted with the Playfair cypher
- 3) the same text as above but encrypted with the Vigenère cypher

To analyze and compare these three texts, we first get the relative letter frequencies of each one (as we did in project 1), normalize the value by dividing each frequency with the highest value (E in this case with 12.702%), and the plotting all three in the same graph ordered from the highest to lowest frequencies. Figure 3 shows the output of this procedure.

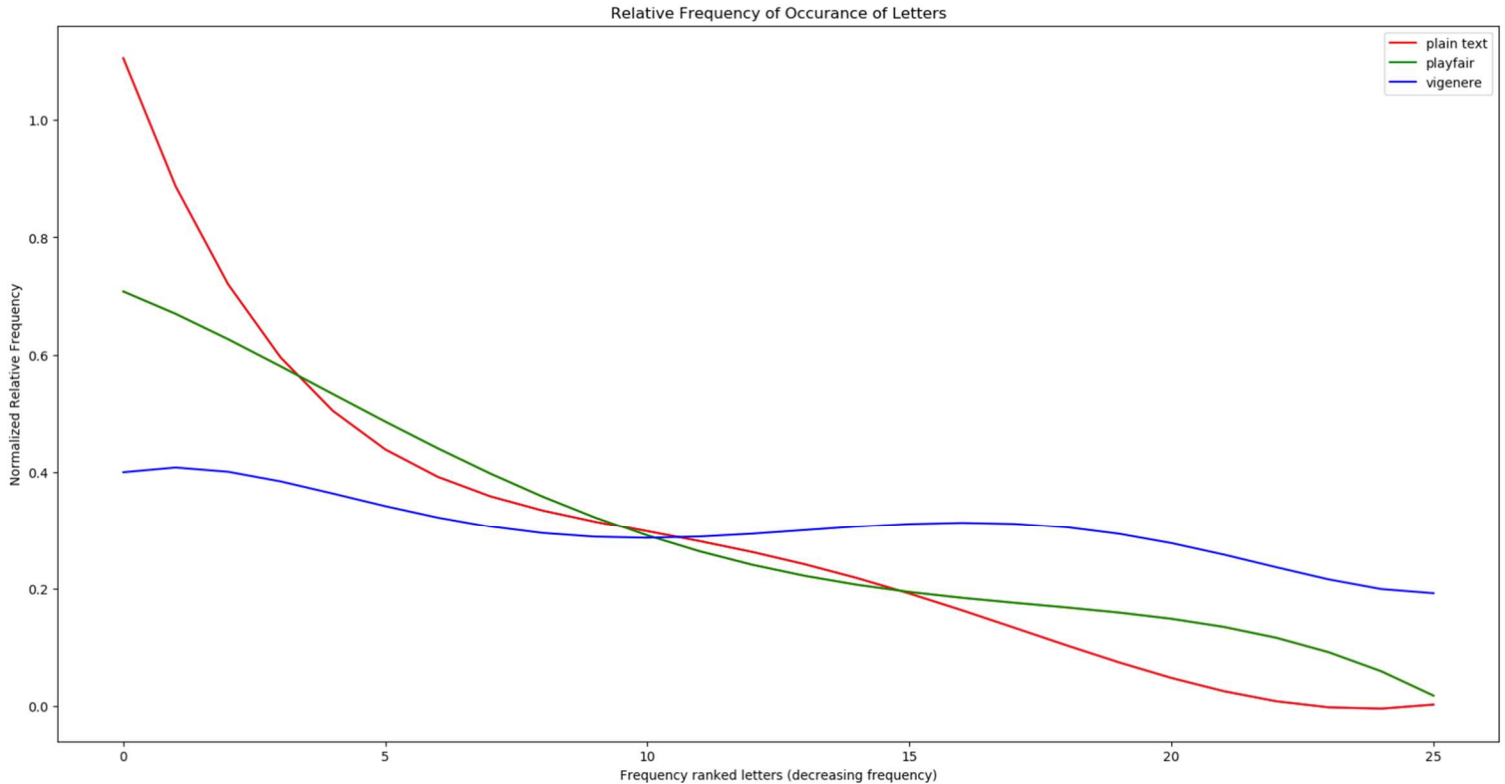


Figure 3

Looking at this plot, it is evident that the Vigenère cypher is the least affected by a letter frequency attack because its letter frequency distribution is more even than the others.

References

- 1) Stallings, William. "Cryptography and Network Security 6th edition", Pearson Education
- 2) "Vigenère_cipher", Wikipedia, 05/30/2022,
https://en.wikipedia.org/wiki/Vigenère_cipher
- 3) "Playfair cypher", Wikipedia, 05/30/2022,
https://en.wikipedia.org/wiki/Playfair_cipher
- 4) Shakil, Tahmid & Islam, Md. (2014). An Efficient Modification to Playfair Cipher. ULAB Journal of Science and Engineering. 5. 26.
https://www.researchgate.net/publication/274709511_An_Efficient_Modification_to_Playfair_Cipher
- 5) Data Security Using Vigenere Cipher and Goldbach Codes Algorithm,
<https://www.ijert.org/research/data-security-using-vigenere-cipher-and-goldbach-codes-algorithm-IJERTV6IS010245.pdf>

YouTube video link: <https://youtu.be/GUFbZpWLjv8>