Class Activity: Caesar Cipher

Objective

The objective of this class activity was to understand and implement the Caesar Cipher. Additionally, the activity also aimed to demonstrate the vulnerability of the Caesar Cipher to brute force attacks by systematically attempting all possible shifts to decrypt a given ciphertext.

Introduction

The Caesar Cipher is one of the simplest and most widely known encryption techniques named after Julius Caesar. It is a type of substitution cipher where each letter in the plaintext is shifted a certain number of places down or up the alphabet.

How the Caesar Cipher Works

1. **Choose a Shift Value:** Decide on the number of positions each letter in the plaintext will be shifted. This value is called the key. For example, if the key is 3, each letter in the plaintext is moved three places to the right.

2. **Encrypting:** For each letter in the plaintext, replace it with the letter that is a fixed number of positions down the alphabet. If the end of the alphabet is reached, the cipher wraps around to the beginning.

   Example: With a shift of 3, 'A' becomes 'D', 'B' becomes 'E', 'C' becomes 'F', and so on.

3. **Decrypting:** To decrypt the message, shift the letters back by the same number of positions.

   Example: With a shift of 3, 'D' becomes 'A', 'E' becomes 'B', 'F' becomes 'C', and so on.

Example:

Replace each letter with 3rd letter on in the alphabet

Meet me after the toga party

PHHW PH DIWHU WKH WRJD SDUWB

Transformation shows how each letter of the alphabet is shifted by a fixed number of positions. For example:

a b c d e f g h i j k l m n o p q r s t u v w x y z

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

The above example implies a shift of 3 positions to the right:

- 'a' becomes 'D'
- 'b' becomes 'E'
- 'c' becomes 'F'
- and so on.

Each letter is assigned a corresponding number:

a b c d e f g h i j k l m n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

The encryption (E) and decryption (D) processes are given by the following formulas:

1. **Encryption:**

   $c=E(p)=(p+k) \bmod 26$

   c is the ciphertext letter.

   p is the plaintext letter.

   k is the shift value (in the example above k=3).

   mod 26 ensures that the result wraps around the alphabet if it exceeds 25.

2. **Decryption:**

   $p=D(c)=(c-k) \bmod 26$

   p is the plaintext letter.

   c is the ciphertext letter.

   k is the shift value (again, k=3).

   mod26 ensures that the result wraps around the alphabet if it goes below 0.

Cryptanalysis of Caesar Cipher

Since the Caesar Cipher involves shifting the alphabet by a fixed number of positions, there are only 26 possible shifts.

Vulnerable to brute force attack. This means trying each of the 26 possible shifts on the ciphertext

Break the Caesar Cipher ciphertext "GCUA VQ DTGCM" using a brute force approach in Python

Solution

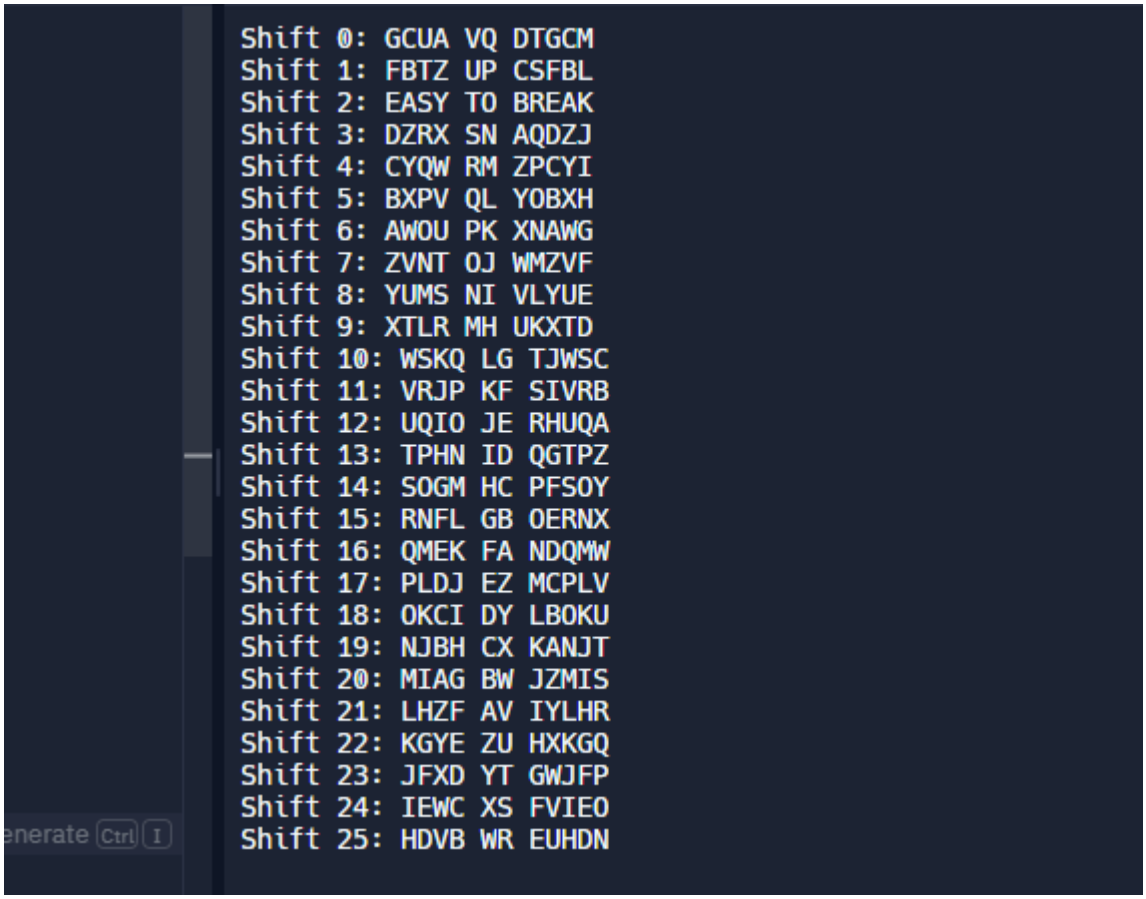
Since the shift value is not known, I will use Brute force

```
def caesar_cipher(text, shift, direction='encrypt'):
    result = ""
    for char in text:
        if char.isalpha():
            shift_amount = shift if direction == 'encrypt' else -shift
            if char.isupper():
                result += chr((ord(char) + shift_amount - 65) % 26 + 65)
            else:
                result += chr((ord(char) + shift_amount - 97) % 26 + 97)
        else:
            result += char
    return result

ciphertext = "GCUA VQ DTGCM"

# Try all possible shifts
for shift in range(26):
    decrypted_text = caesar_cipher(ciphertext, shift, direction='decrypt')
    print(f"Shift {shift}: {decrypted_text}")
```

OUTPUT



By examining the output, I can see that when the shift is 2, the plaintext reads "EASY TO BREAK". Therefore, the correct shift is 2.

Conclusion

This activity effectively demonstrated the Caesar Cipher's encryption and decryption process. By shifting each letter in the plaintext by a fixed number of positions, the text can be transformed into a seemingly unintelligible string. However, due to its simplicity, the Caesar Cipher is easily broken using brute force methods. The exercise of breaking the ciphertext "GCUA VQ DTGCM" by trying all 26 possible shifts illustrates this vulnerability. The correct shift of 2, which revealed the plaintext as "EASY TO BREAK," highlights the need for more secure encryption techniques in practical applications.