Name: **Shrey Pendurkar**
Class: **D15C**
Batch: **C**
Roll No: **64**

CNS - Experiment 3

Aim: Write a program in Java or Python to perform Cryptanalysis or decoding of Vigenere Cipher

Theory:

Key Concepts:

● Polyalphabetic Substitution Cipher: Unlike monoalphabetic substitution ciphers (like Caesar cipher) that use a fixed substitution pattern for each letter, Vigenère cipher uses multiple substitution patterns based on a keyword. This makes it more secure compared to simple substitution ciphers.

● Keyword: The keyword is a secret sequence of characters (usually letters) that determines the shift value for each letter in the plaintext. Its the core of the Vigenère cipher and dictates the encryption and decryption process.

● Encryption: To encrypt a message using the Vigenère cipher, each letter of the plaintext is shifted based on the corresponding letter in the keyword. The shift value for a letter is determined by its position in the alphabet (A=0, B=1, ..., Z=25).

● Decryption: To decrypt an encrypted message, the reverse process is applied. Each letter of the ciphertext is shifted back based on the corresponding letter in the keyword.

Advantages: Vigenère cipher is stronger than simple Caesar cipher due to its use of multiple substitution patterns. It’s relatively easy to understand and implement.

Disadvantages: The security of Vigenère cipher depends on the length and secrecy of the keyword. If the keyword is short or not truly random, it’s susceptible to attacks such as frequency analysis. It’s also vulnerable to Kasiski examination and Friedman test if the keyword length is too short.

Review Questions

1. **Discuss the strengths and weaknesses of the Vigenère cipher.**
   The Vigenère cipher is a classical polyalphabetic substitution cipher that uses a keyword to encrypt a message by shifting each letter based on the corresponding letter of the keyword.

Strengths:

- The cipher is stronger than simple monoalphabetic ciphers like Caesar cipher because it uses multiple shifting alphabets, making frequency analysis more difficult.
- It is easy to understand and implement by hand, making it suitable for manual encryption and decryption.
- The use of a keyword introduces variability, so the same letter in the plaintext can be encrypted to different letters in the ciphertext.
- It was historically considered secure and was known as "le chiffre indéchiffrable" (the indecipherable cipher).

Weaknesses:

- If the keyword is short or repeated, patterns may appear in the ciphertext, which can be exploited using attacks like the Kasiski examination or the Friedman test.
- It is vulnerable to known-plaintext and chosen-plaintext attacks if part of the plaintext is known or guessed.
- The cipher does not provide any authentication or data integrity, so it cannot detect if the message has been altered.
- With modern computing power and cryptanalysis techniques, it is no longer considered secure and can be broken relatively easily.

Conclusion: While the Vigenère cipher was a major advancement in its time, it is now mainly used for educational purposes and not for securing sensitive data.

2. **How can the Vigenère cipher be broken or attacked? Describe one method.**
   The Vigenère cipher can be broken using Kasiski Examination. Here's how it works:
   - Find Repeated Sequences: Look for repeated patterns in the ciphertext caused by repeating key segments.
   - Measure Gaps: Note the distances between these repeats. These are likely multiples of the key length.
   - Guess Key Length: Use the GCD of the distances to estimate the key length.
   - Frequency Analysis: Split the ciphertext based on key length and apply frequency analysis to each part to find the Caesar shift and recover the key.

   This method works because repeating keys create patterns that can be statistically analyzed.

3. **What is the significance of the key length in the security of the Vigenère cipher?**
   The key length in the Vigenère cipher plays a crucial role in its security. A longer key increases the complexity of the cipher and makes it harder to break using frequency analysis. When the key is short or repeated frequently, patterns begin to emerge in the ciphertext, which can be exploited by cryptanalysts. On the other hand, if the key is as long as the plaintext and completely random (like in a one-time pad), the cipher becomes theoretically unbreakable. Therefore, the longer and more random the key, the more secure the Vigenère cipher becomes.

4. **Compare and contrast the Vigenère cipher with other polyalphabetic ciphers.**

Polyalphabetic ciphers enhance security over monoalphabetic ciphers by using multiple substitution alphabets. The Vigenère cipher is a classic example, but others like the Autokey, Beaufort, and Running Key ciphers offer distinct approaches.

Feature / Cipher	**Vigenère**	**Beaufort**	**Autokey**
Key Type & Structure	Repeating alphabetic key	Repeating alphabetic key	Short key + plaintext as key
Key Length	Usually short & repeats	Usually short & repeats	Potentially very long
Main Difference	Standard tabula recta with cyclic key	Reversed alphabet in tableau	Key stream changes with plaintext
Strengths	Simple & easy to implement	Adds extra confusion	Reduces key repetition, more secure
Weaknesses	Broken by Kasiski/Friedman if short key	Same weaknesses as Vigenère	Vulnerable if part of key is known

5. **Using the Vigenere cipher , encrypt the word “explanation” using the key leg.**

To encrypt the word "explanation" using the Vigenère cipher with the key "leg", follow these steps:
Repeat the key to match the plaintext length:
Plaintext: explanation (11 letters)
Key: leg → Repeated key: l e g l e g l e g l e
Convert letters to numerical values (A=0, B=1, ..., Z=25):
Plaintext:
e → 4, x → 23, p → 15, l → 11, a → 0, n → 13, a → 0, t → 19, i → 8, o → 14, n → 13
Key:
l → 11, e → 4, g → 6
Encrypt each letter using the formula:

$$C_i = (P_i + K_i) \bmod 26$$

Where P_i is the plaintext number and K_i is the key number.

- e (4) + l (11) = 15 → P
- x (23) + e (4) = 27 → 27 mod 26 = 1 → B
- p (15) + g (6) = 21 → V
- l (11) + l (11) = 22 → W
- a (0) + e (4) = 4 → E
- n (13) + g (6) = 19 → T
- a (0) + l (11) = 11 → L
- t (19) + e (4) = 23 → X
- i (8) + g (6) = 14 → O
- o (14) + l (11) = 25 → Z

- n (13) + e (4) = 17 → R

Combine the ciphertext letters:
Result: P B V W E T L X O Z R

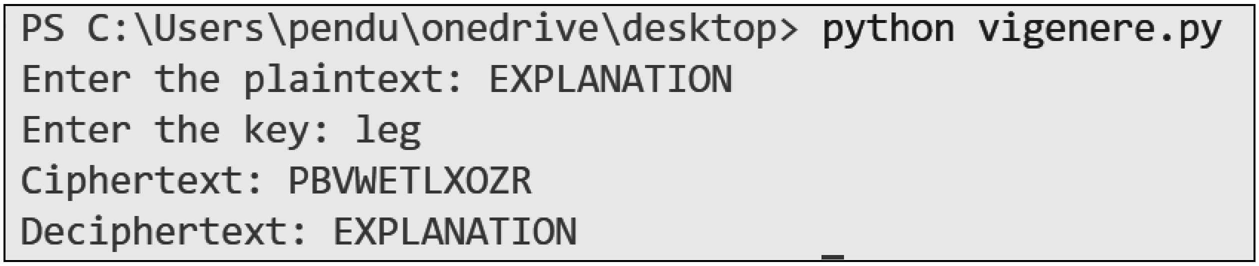
Final Ciphertext:
The encrypted word is PBVWETLXOZR.

Code:

```
def format_key(plaintext, key):
    key = key.upper()
    formatted_key = ''
    key_index = 0
    for char in plaintext:
        if char.isalpha():
            formatted_key += key[key_index % len(key)]
            key_index += 1
        else:
            formatted_key += char
    return formatted_key

def vigenere_encrypt(plaintext, key):
    plaintext = plaintext.upper()
    formatted_key = format_key(plaintext, key)
    ciphertext = ''
    for p, k in zip(plaintext, formatted_key):
        if p.isalpha():
            c = chr(((ord(p) - ord('A') + ord(k) - ord('A')) % 26) + ord('A'))
            ciphertext += c
        else:
            ciphertext += p
    return ciphertext

def vigenere_decrypt(ciphertext, key):
    ciphertext = ciphertext.upper()
    formatted_key = format_key(ciphertext, key)
    deciphertext = ''
    for c, k in zip(ciphertext, formatted_key):
        if c.isalpha():
            p = chr(((ord(c) - ord(k) + 26) % 26) + ord('A'))
            deciphertext += p
        else:
            deciphertext += c
    return deciphertext

plaintext = input("Enter the plaintext: ")
```

```
key = input("Enter the key: ")

ciphertext = vigenere_encrypt(plaintext, key)
print("Ciphertext:", ciphertext)

deciphertext = vigenere_decrypt(ciphertext, key)
print("Deciphertext:", deciphertext)
```

Output:



```
PS C:\Users\pendu\onedrive\desktop> python vigenere.py
Enter the plaintext: EXPLANATION
Enter the key: leg
Ciphertext: PBVWETLXOZR
Deciphertext: EXPLANATION
```