Name: Tejas Redkar

Batch: C2; Roll no: 44

PRN: 1032210937

**ICS Lab Assignment 1**

**Lab A1:** Implement any classical cryptographic technique using java or python or C++

**Objective of Lab**

1. To study understand and implement at least two classical cryptographic algorithms

**Theory :**

**Explain classical algorithm: Caesar cipher and Mono alphabetic cipher compulsory with example**

Classical Cryptographic Algorithms: Caesar Cipher and Monoalphabetic Cipher

Introduction to Classical Cryptography:

Classical cryptography refers to the early methods of encrypting and decrypting messages to ensure their confidentiality. In this discussion, we will explore two classical cryptographic algorithms: the Caesar Cipher and the Monoalphabetic Cipher. These techniques have played a significant role in the history of cryptography and provide valuable insights into the fundamentals of encryption.

**Caesar Cipher:**

The Caesar Cipher is one of the simplest and most well-known substitution ciphers. It is named after Julius Caesar, who is believed to have used this method to protect sensitive military messages. The Caesar Cipher is a type of symmetric key encryption, meaning the same key is used for both encryption and decryption.

How the Caesar Cipher Works:

In the Caesar Cipher, each letter in the plaintext is shifted a fixed number of positions down or up the alphabet. This shift is defined by the "key," which is an integer between 1 and 25. For example, with a key of 3:

- A becomes D

- B becomes E

- C becomes F

- ...

- X becomes A

- Y becomes B

- Z becomes C

This simple algorithm provides a basic level of security, but it is vulnerable to brute-force attacks since there are only 25 possible keys. Let's illustrate the Caesar Cipher with an example:

Example:

Suppose we want to encrypt the message "HELLO" using a Caesar Cipher with a key of 3.

1. Start with the plaintext message: HELLO

2. Apply the Caesar Cipher with a key of 3:

- H -> K

- E -> H

- L -> O

- L -> O

- O -> L

3. The encrypted message is "KHOOR."

To decrypt the message, you would simply reverse the process by shifting the letters back by 3 positions.

**Monoalphabetic Cipher:**

The Monoalphabetic Cipher is another type of substitution cipher, but it offers more complexity than the Caesar Cipher. In the Monoalphabetic Cipher, each letter in the plaintext is replaced by another letter from the alphabet. However, unlike the Caesar Cipher, there is no fixed shift value. Instead, each letter is substituted with a corresponding letter from a randomly generated or predefined substitution key.

How the Monoalphabetic Cipher Works:

A monoalphabetic substitution key is a one-to-one mapping of letters from the plaintext alphabet to letters in the ciphertext alphabet. Here's a simplified example of a monoalphabetic substitution key:

Plaintext alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ciphertext alphabet: XYZABCDEFGHIJKLMNOPQRSTUVW

Using this key, the letter 'A' in the plaintext would be replaced by 'X' in the ciphertext, 'B' would become 'Y,' and so on.

Example:

Let's encrypt the message "HELLO" using a monoalphabetic substitution key:

Plaintext alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ciphertext alphabet: XYZABCDEFGHIJKLMNOPQRSTUVW

1. Start with the plaintext message: HELLO

2. Apply the Monoalphabetic Cipher using the given substitution key:

- H -> X

- E -> A

- L -> O

- L -> O

- O -> L

3. The encrypted message is "XAOLLO."

To decrypt the message, you would need to use the same substitution key in reverse.

Strengths and Weaknesses:

- Caesar Cipher Strengths:

- Simplicity: Easy to understand and implement.

- Speed: Fast encryption and decryption.

- Caesar Cipher Weaknesses:

- Low security: Vulnerable to brute-force attacks.

- Limited key space: Only 25 possible keys.

- Monoalphabetic Cipher Strengths:

- Greater complexity: More secure than Caesar Cipher.

- Key variability: Different substitution keys provide increased security.

- Monoalphabetic Cipher Weaknesses:

- Frequency analysis vulnerability: Vulnerable to frequency analysis attacks, as the frequency of letters in the plaintext may be reflected in the ciphertext.

- Limited security: Modern cryptanalysis techniques can still break it.

**Conclusion:**

The Caesar Cipher and Monoalphabetic Cipher are classical cryptographic algorithms that demonstrate the fundamental principles of encryption. While they are no longer considered secure for protecting sensitive information, they serve as essential building blocks for understanding more advanced encryption techniques. Modern cryptographic systems rely on mathematical algorithms and large key spaces to ensure the security of data, but learning about these classical ciphers provides valuable insights into the history and evolution of cryptography.

**Code (caesar cipher)**

choice = int(input("Enter choice \n1. Encrypt \n2. Decrypt \n"))

key = 5

def encrypt():

    plain\_text = input("Enter text to encrypt: ")

    cipher = ""

    for c in plain\_text.upper():

        if (not c.isalpha()):

            cipher += c

            continue

        cypher\_of\_c = chr((ord(c) - 65 + key) % 26 + 65)

        cipher += cypher\_of\_c

    print(f"Cipher text for {plain\_text} is {cipher}")

def decrpyt():

    cipher = input("Enter text to decrypt: ")

    plain\_text = ""

    for c in cipher.upper():

        if (not c.isalpha()):

            plain\_text += " "

            continue

        cypher\_of\_c = chr((ord(c) - 65 - key) % 26 + 65)

        plain\_text += cypher\_of\_c

    print(f"Plain text for {cipher} is {plain\_text}")

if choice == 1:

    encrypt()

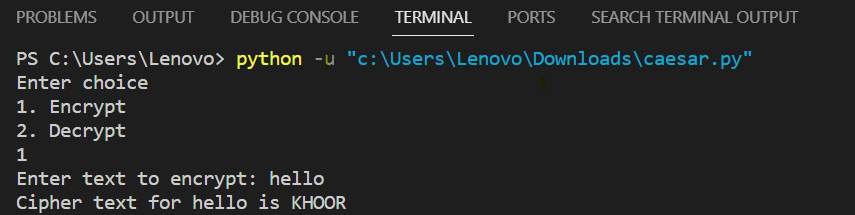
elif choice == 2:

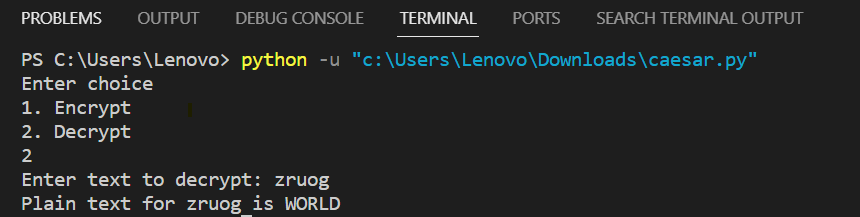
    decrpyt()

else:

    exit()

**Output Screen shots (Caesar Cipher)**





**Code (Mono alphabetic cipher)**

choice = int(input("Enter choice \n1. Encrypt \n2. Decrypt \n"))

key = "LKJHGFDSAQWERTYUIOPMNBVCXZ"

def encrypt():

    plain\_text = input("Enter text to encrypt: ")

    cipher = ""

    for c in plain\_text.upper():

        if (not c.isalpha()):

            cipher += c

            continue

        cipher\_of\_c = key[ord(c) - 65]

        cipher += cipher\_of\_c

    print(f"Cipher text for {plain\_text} is {cipher}")

def decrypt():

    cipher = input("Enter text to decrypt: ")

    plain\_text = ""

    for c in cipher.upper():

        if (not c.isalpha()):

            cipher += c

            continue

        plain\_of\_c = chr(key.index(c) + 65)

        plain\_text += plain\_of\_c

    print(f"Plain text for {cipher} is {plain\_text}")

if choice == 1:

    encrypt()

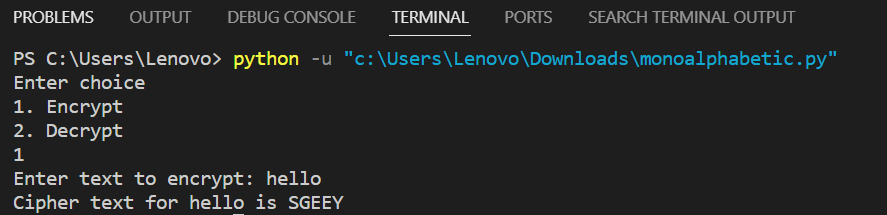
elif choice == 2:

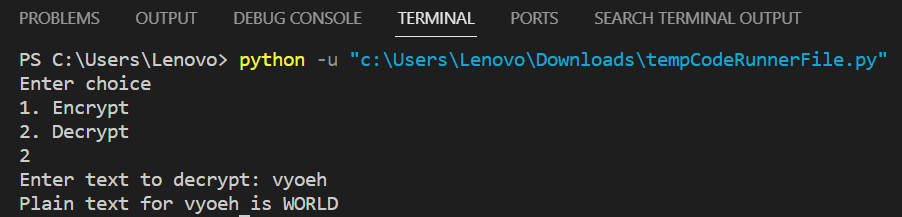
    decrypt()

else:

    exit()

**Output Screen shots (Monoalphabetic cipher)**





**Conclusion**:

We successfully implemented two classical cryptographic algorithms, the Caesar Cipher and the Monoalphabetic Cipher, in Python. These implementations provide a practical demonstration of how these algorithms work for encrypting messages.

# FAQs:

* 1. What are various classical ciphers?
  2. Compare steganography and Cryptography.
  3. State the reasons why classical ciphers are obsolete.
  4. How to carry our cryptanalysis of classical cryptography?
  5. Write how different disciplines of art, science and engineering have contributed for information security.