COMSATS UNIVERSITY
ISLAMABAD

LAB ASSIGNMENT: 01

Submitted By: Manahil Kamran

Registration no.: FA24-BSE-025

Submitted To: Ma’am Ambreen Gul

Date: 28TH February, 2026

Course: Information Security

Table Of Contents

Task 1: Implement the Caesar Cipher

Write a Python program that encrypts and decrypt a message using the Caesar Cipher.

1. Introduction

The Caesar cipher is one of the simplest and oldest encryption techniques. It works by shifting each letter of the plaintext by a fixed number of positions in the alphabet. This Python implementation demonstrates both encryption and decryption using a shift of 3

2. Line-by-Line Explanation

Key and Plaintext

```
key = 'abcdefghijklmnopqrstuvwxyz'   # alphabetic key
plaintext = "My name is Manahil Kamran!" # original text
```

key: Defines the alphabet used for shifting.

plaintext: The message we want to encrypt.

Encryption Function

```
def enc_caesar(plaintext, shift):  # Function for encryption
 result = ‘’   # empty string to store encrypted text
```

- Defines enc_caesar function.
- Initializes result to store the encrypted output.

```
for l in plaintext: # go through each character
if l.isalpha(): # check if character is a letter
```

- Loops through each character.

- Checks if it’s alphabetic (ignores spaces/punctuation).

if l.islower(): # ***if lowercase letter***

i = (key.index(l) + shift) % 26 # ***find position and shift***

result += key[i] # ***add shifted letter***

- Handles lowercase letters.
- Finds index in key, shifts by shift, wraps around using % 26.

else: # ***if uppercase letter***

i = (key.index(l.lower()) + shift) % 26 # ***shift using lowercase index***

result += key[i].upper() # ***convert back to uppercase***

- Handles uppercase letters by converting to lowercase for indexing.
- Converts back to uppercase after shifting.

else: result += l # ***keep spaces/punctuation unchanged***

return result # ***return encrypted text***

- Non-alphabetic characters remain unchanged.
- Returns the encrypted string.

Encrypting the Plaintext

ciphertext = enc_caesar(plaintext, 3) # ***Encrypt the plaintext with a shift of 3***

Calls the encryption function with a shift of 3.

Decryption Function

def dec_caesar(ciphertext, shift): # ***Function for decryption***

result = '' # ***empty string to store decrypted text***

- Defines dec_caesar function.
- Initializes result.

```
for l in ciphertext: # go through each character
 if l.isalpha(): # check if character is a letter
```

- Loops through ciphertext.
- Checks if character is alphabetic.

```
if l.islower(): # if lowercase letter
 i = (key.index(l) - shift) % 26 # shift backwards
result += key[i] # add original letter
```

- Handles lowercase letters.
- Shifts backwards by subtracting shift.

```
else: # if uppercase letter
 i = (key.index(l.lower()) - shift) % 26 # shift backwards using lowercase index
result += key[i].upper() # convert back to uppercase
```

- Handles uppercase letters.
- Converts back to uppercase after shifting.

```
else: result += l  # keep spaces/punctuation unchanged
return result # return decrypted text
```

- Non-alphabetic characters remain unchanged.
- Returns decrypted string.

Decrypting the Ciphertext

```
decrypted = dec_caesar(ciphertext, 3) # Decrypt the ciphertext with the shift of 3
```

Calls decryption with the same shift.

Output

```
print("Plaintext:", plaintext)
print("Ciphertext:", ciphertext)
print("Decrypted:", decrypted)
```

Displays original, encrypted, and decrypted text.

3. Security Analysis

Strengths

- Simple and easy to implement.
- Good for learning basic cryptography concepts.

Weaknesses

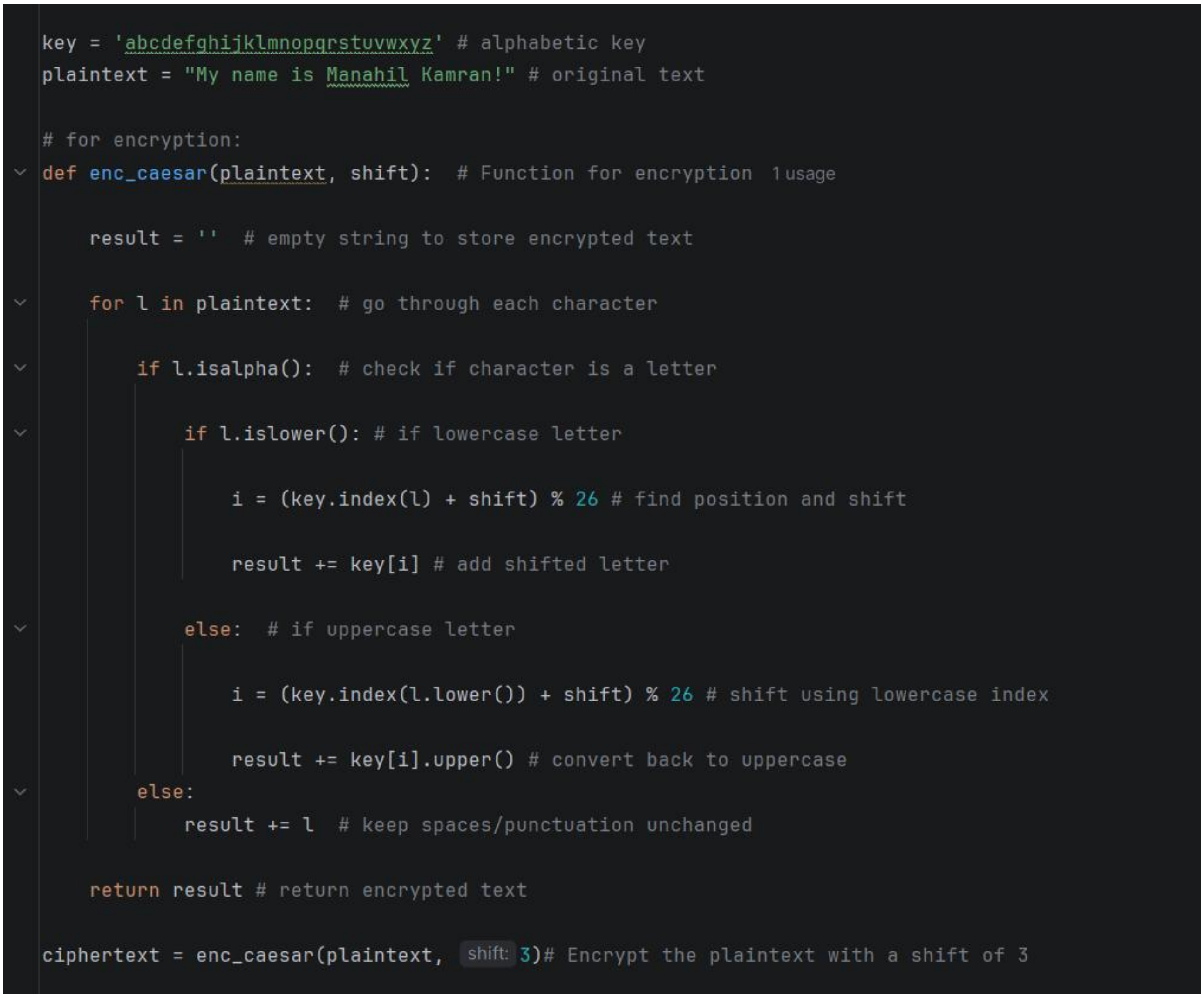
- Extremely insecure: only 25 possible shifts.
- Vulnerable to brute force attacks.
- Easily broken using frequency analysis.

Modern Alternatives

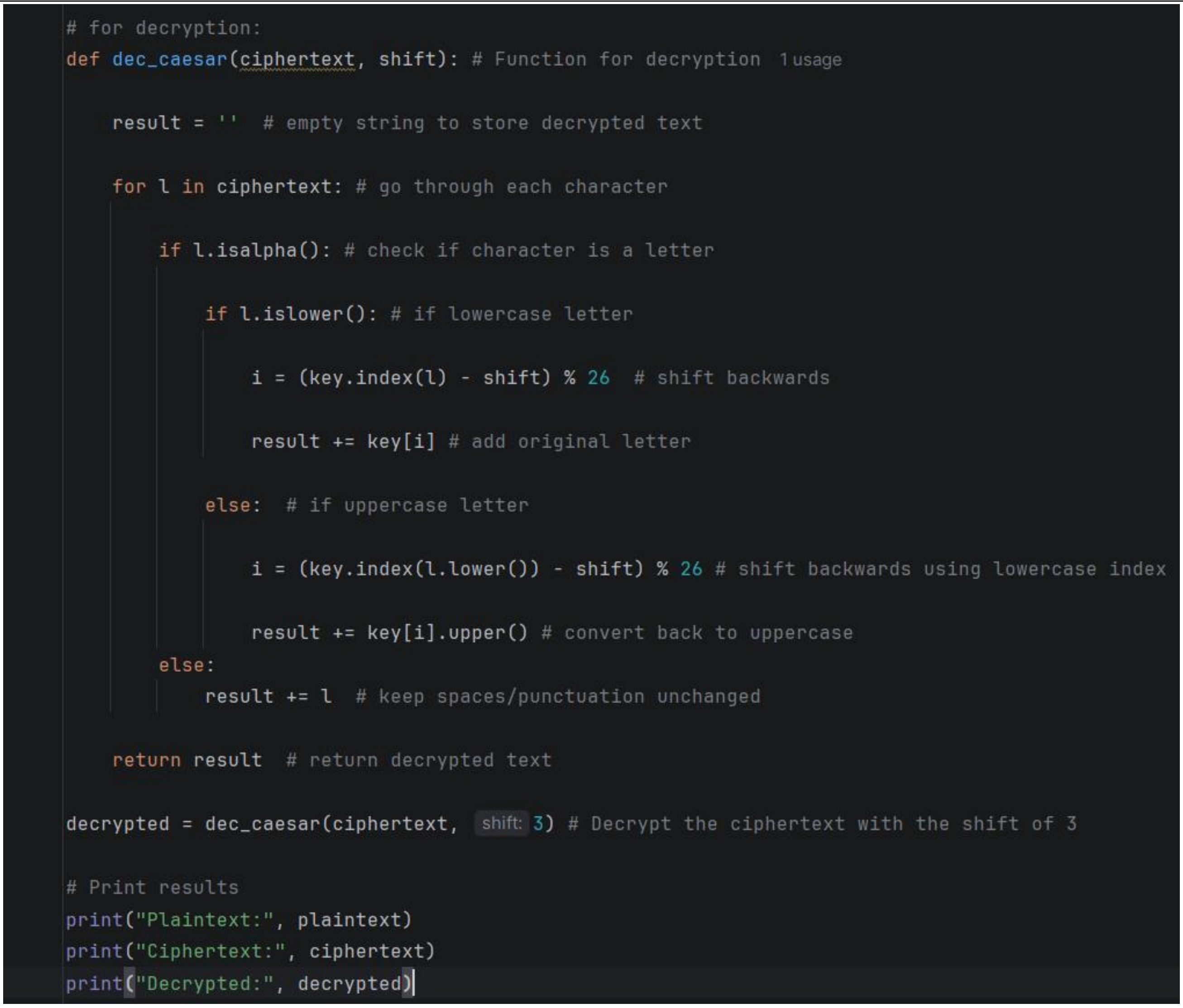
- AES (Advanced Encryption Standard).
- RSA (public-key cryptography).
- Python’s cryptography library for secure implementations.

Code Screenshots:

CODE:

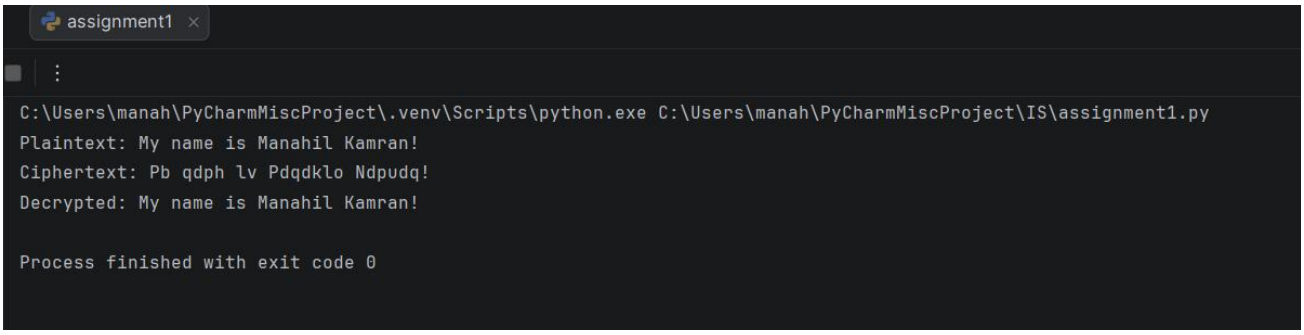


```
key = 'abcdefghijklmnopqrstuvwxyz' # alphabetic key
plaintext = "My name is Manahil Kamran!" # original text

# for encryption:
def enc_caesar(plaintext, shift):  # Function for encryption  1 usage

    result = ''  # empty string to store encrypted text

    for l in plaintext:  # go through each character

        if l.isalpha():  # check if character is a letter

            if l.islower(): # if lowercase letter

                i = (key.index(l) + shift) % 26 # find position and shift

                result += key[i] # add shifted letter

            else:  # if uppercase letter

                i = (key.index(l.lower()) + shift) % 26 # shift using lowercase index

                result += key[i].upper() # convert back to uppercase
        else:
            result += l  # keep spaces/punctuation unchanged

    return result # return encrypted text

ciphertext = enc_caesar(plaintext, shift: 3)# Encrypt the plaintext with a shift of 3
```



```python
# for decryption:
def dec_caesar(ciphertext, shift): # Function for decryption  1 usage

    result = ''  # empty string to store decrypted text

    for l in ciphertext: # go through each character

        if l.isalpha(): # check if character is a letter

            if l.islower(): # if lowercase letter

                i = (key.index(l) - shift) % 26  # shift backwards

                result += key[i] # add original letter

            else:  # if uppercase letter

                i = (key.index(l.lower()) - shift) % 26 # shift backwards using lowercase index

                result += key[i].upper() # convert back to uppercase
        else:
            result += l  # keep spaces/punctuation unchanged

    return result  # return decrypted text

decrypted = dec_caesar(ciphertext, shift: 3) # Decrypt the ciphertext with the shift of 3

# Print results
print("Plaintext:", plaintext)
print("Ciphertext:", ciphertext)
print("Decrypted:", decrypted)
```

OUTPUT:



```
C:\Users\manah\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\manah\PyCharmMiscProject\IS\assignment1.py
Plaintext: My name is Manahil Kamran!
Ciphertext: Pb qdph lv Pdqdklo Ndpudq!
Decrypted: My name is Manahil Kamran!

Process finished with exit code 0
```