Cryptography with Python - Multiplicative Cipher

1. Algorithm of Multiplicative

1.1 Encryption Phase

For each letters P in the plaintext, compute a corresponding letter C of the ciphertext using the equation below:

$$C = E(\mathrm{P} * \mathrm{K}) \bmod 26$$

Where C: Cipher Text

P: Plain Text

E: Encryption algorithm

K: Key

➢ **To implement encryption algorithm, follow these steps:**

- ✓ **Input :** Plain text, Key
- ✓ **Output:** Cipher text
- ✓ Encryption Algorithm = (P * K) mod 26

Step 1) Taking input in Python: using **input ( )** function to read the **Plaintext** and **Key** from the keyboard.

```
# Python program showing
# a use of input()

PlainText = input("Enter The PlainText: ")
print(PlainText)
print(type(PlainText))

Key = int(input("Enter Key value: "))
print(Key)
print(type(Key))
```

Step 2) Encryption code:

```
PlainText = input("Enter The PlainText: ")
Key = int(input("Enter Key Value: "))
print("Plain Text : " + PlainText)

def Encrypt(PlainText, Key):
    CipherText = ""
# transverse the plain text
    for char in PlainText:
        # Encrypt uppercase characters in plain text
        if (char.isupper()):
            CipherText += chr((ord(char) * Key - 65) % 26 + 65)

        # Encrypt lowercase characters in plain text
        else:
            CipherText += chr((ord(char) * Key - 97) % 26 + 97)
    return CipherText

# call the above function
CipherText = Encrypt(PlainText, Key)

print("Cipher: " + CipherText)
```

Result:-

```
Enter The PlainText: HelloGroupB
Enter Key Value: 7
Plain Text : HelloGroupB
Cipher: XmjjeQzeulH
```

1.2 Decryption Phase

For each letters C in the ciphertext, compute a corresponding letter P of the plaintext using the equation below:

$$P = D(C * \mathrm{K}^{-1}) \bmod 26$$

Where P: Plain Text

C: Cipher Text

D: Decryption algorithm

K^{-1}: Key Inverse

➢ **To implement decryption algorithm, follow these steps:**

- ✓ **Input :** Cipher text, Key
- ✓ **Output:** Plain text
- ✓ Decryption Algorithm = (C * K^{-1}) mod 26

Step 1) Calculate the inverse of the key to be used in the decryption algorithm:

```
# multiplicative inverse of 'k'
# under modulo 'm'
def modInverse(k, m):
    k = k % m;
    for x in range(1, m):
        if((k * x)% m == 1):
            return x
    return 1

InKey = modInverse(Key,26)
```

Step 2) Decryption code:

```
def Decrypt(CipherText, InKey):
    PlainText = ""
# transverse the cipher text
    for char in CipherText:
        # Decrypt uppercase characters in Cipher text
        if (char.isupper()):
            PlainText += chr((ord(char) * InKey - 65) % 26 + 65)

        # Decrypt lowercase characters in cipher text
        else:
            PlainText += chr((ord(char) * InKey - 97) % 26 + 97)
    return PlainText

PlainText = Decrypt(CipherText, InKey)

print("Plain Text : " + PlainText)
```

Result:-

```
Enter The PlainText: HelloGroupB
Enter Key Value: 7
Plain Text : HelloGroupB
Cipher: XmjjeQzeulH
########################################################################
Key Inverse:  15
Plain Text : HelloGroupB
```

❖ Assignment No. 1:

Implement **Affine Algorithm** for both encryption and decryption using Python language, based on ASCII code.

- Please send the assignment to my Email **(suadadsafaa@itnet.uobabylon.edu.iq)**.