Lab 4

Substitution Ciphers

In the field of cryptography, the substitution cipher simply substitutes one letter in the alphabet for another based upon a crypto variable. The substitution involves shifting positions in the alphabet. This includes the Caesar Cipher and ROT-13. Examine the following example:

- *Plaintext*: WE HOLD THESE TRUTHS TO BE SELF-EVIDENT, THAT ALL MEN ARE CREATED EQUAL.
- *Ciphertext*: ZH KROG WKHVH WUXWKV WR EH VHOI-HYLGHQW, WKDW DOO PHQ DUH FUHDWHG HTXDO.

In lab3, you implemented a custom algorithm for Caesar Cipher.

Some other types of substitution ciphers include the the Vigenere Cipher, and the Playfair Cipher.

The Hill 2x2 cipher is a polygraphic substitution cipher based on linear algebra. The inventor of this cipher, Lester S. Hill, created the cipher in 1929. The cipher uses matrices and matrix multiplication to mix the plaintext to produce the ciphertext. This cipher is based on a branch of mathematics known as number theory. The key used here is HILL, which corresponds to the numbers 7, 8, 11, and 11.

In this lab, you will develop two Python programs.

1. A function to perform encryption of a plaintext input to cipher text using the Hill 2x2 technique.
2. A function to perform decryption of a plaintext input to cipher text using the Hill 2x2 technique.

A skeleton code for the encryption/decryption algorithm is shown below:
(The below code is just for your reference, you can write your own). Do not hard code.

```
import sys
import numpy as np

def cipher_encryption(plain, key):

    # Handle condition if the message length is odd.
    #<Enter code here>

    # Convert msg to matrices
    #Enter code here>

    # Convert key to 2x2
    #Enter code here>

    print (key2d)
```

```
    # checking validity of the key
    # finding determinant
    #Enter code here>

    # finding multiplicative inverse and implementing steps to encrypt text
    #<Enter code here>

    print("Encrypted text: {}".format(encryp_text))
    return encryp_text


def cipher_decryption(cipher, key):

    # Handle condition if the message length is odd.
    #<Enter code here>

    # Convert msg to matrices
    #<Enter code here>

    # Convert key to 2x2
    #<Enter code here>

    # finding determinant
    #<Enter code here>

    # finding multiplicative inverse
    #<Enter code here>

    # find transpose
    # find minor
    #<Enter code here>

    #changing signs
    #<Enter code here>

    # multiplying multiplicative inverse with adjugate matrix
    #<Enter code here>

    # Calculate modulo
    #<Enter code here>

    # Convert cipher to plaintext
    #<Enter code here>

    print("Decrypted text: {}".format(decryp_text))


Example:
plaintext = "Secret Message"
plaintext = plaintext.upper().replace(" ","")
key = "hill"
key = key.upper().replace(" ","")
ciphertext = cipher_encryption(plaintext, key)
cipher_decryption(ciphertext, key)
```

What to Submit

Upload these deliverables to Canvas.

Deliverables:

1. Jupyter Notebook file <Lab4-netid.ipynb> with two functions `cipher_Encryption` and `cipher_Decryption` and example shown above.
2. Screenshot of the terminal where the code was written.

Rubric

Task	**Points**
Encryption	40
Decryption	40
Example	20