

Code a Python Playfair Cipher

Task

1h5m - 1h30m

✓ Status

Incomplete

Introduction

In this activity, you will analyse a code that uses a cryptographic algorithm. In fact, the algorithm shared in this activity is a practical example of using a cryptographic algorithm.

Instruction

Given below is a code that creates a 5x5 matrix using a secret key. Unfortunately, this code has bugs and cannot generate the matrix successfully. Therefore, you will try to analyse every step of this simple cryptographic code and debug, wherever required.



Follow the instructions given below to debug the code.

- Copy and paste the code given below into VSCode.
- You may also copy and paste the code into any text editor of your choice and save the file.
- Run the code using Python.
- Debug the code line by line until the 5X5 matrix is successfully generated.



After you are done with this exercise, reflect on the debugging process you followed. You may record any learnings in your PKM.

The Python Code



The following is a Python code that creates a 5x5 matrix using a secret key.

```

def create_matrix(key):
    key = key.upper()
    matrix = [[0 for i in range(5)] for j in range(5)]
    letters_added = []
    row = 0
    col = 0

    # add the key to the matrix
    for letter in key:
        if letter not in letters_added:
            matrix[row][col] = letter
            letters_added.append(letter)
        else:
            continue
        if (col==4):
            col = 0
            row += 1
        else:
            col += 1

    #Add the rest of the alphabet to the matrix
    # A=65 ... Z=90
    for letter in range(65,91):
        if letter==74: # I/J are in the same position
            continue
        if chr(letter) not in letters_added: # Do not add repeated letters
            letters_added.append(chr(letter))

    index = 0
    for i in range(5):
        for j in range(5):
            matrix[i][j] = letters_added[index]
            index+=1

    return matrix

#Add fillers if the same letter as a pair
def separate_same_letters(message):
    index = 0
    while (index<len(message)):
        l1 = message[index]
        if index == len(message)-1:
            message = message + 'X'
            index += 2
            continue
        l2 = message[index+1]
        if l1==l2:
            message = message[:index+1] + "X" + message[index+1:]
        index +=2
    return message

#Return the index of a letter in the matrix
#This will be used to know what rule (1-4) to apply
def indexOf(letter,matrix):
    for i in range(5):
        try:
            index = matrix[i].index(letter)
            return (i,index)
        except:
            continue

# Function to Implementation of the playfair cipher
# If encrypt=True the method will encrypt the message
# otherwise the method will decrypt

def playfair(key, message, encrypt=True):
    inc = 1
    if encrypt==False:

```

```
inc = -1

matrix = create_matrix(key)
message = message.upper()
message = message.replace(' ', '')
message = separate_same_letters(message)
cipher_text= ''

for (l1, l2) in zip(message[0::2], message[1::2]):
    row1,col1 = indexOf(l1,matrix)
    row2,col2 = indexOf(l2,matrix)
    if row1==row2: #Rule 2, the letters are in the same row
        cipher_text += matrix[row1][(col1+inc)%5] + matrix[row2][(col2+inc)%5]
    elif col1==col2:# Rule 3, the letters are in the same column
        cipher_text += matrix[(row1+inc)%5][col1] + matrix[(row2+inc)%5][col2]
    else: #Rule 4, the letters are in a different row and column
        cipher_text += matrix[row1][col2] + matrix[row2][col1]


return cipher_text

## Calling the main application
if __name__=='__main__':
    # a sample of encryption and decryption
    print ('Type the Key')
    secret = input ()
    print ('Type the Message')
    message = input ()
    encrypted_message = playfair(secret, message)
    print ('Decrypting the message ->', encrypted_message)
    print ('Result', playfair(secret, encrypted_message, False))
```

Reference

[Just Cryptography - How to implement the Playfair cipher in python?](#)

✓ Mark Completed




Previous

Code a Python Playfair Cipher

Next

Cryptographic Encryption



How well did this activity help you to understand the content?
Let us know how we're doing



W06D5 📅
Fri Aug 2

- > Outline & Notes (1)
- > Lectures (1)

v
 Work (10)

7 hrs


[Cryptanalysis](#)


[The Use of the Historical vs. Modern Encryption](#)


[Cryptanalysis Quiz](#)


[Cryptography Features and Objectives](#)


[Typical Levels of Cryptography.](#)


[The Use of Cryptography.](#)


[Code a Python Playfair Cipher](#)


[Code a Python Playfair Cipher](#)


[Cryptographic Encryption](#)


[Cryptographic Methods with GPG](#)

[W06D5 Schedule »](#)