Exercise on HMAC Algorithm

March 16, 2018

Name: SONGA MUGABE Fabrice
Click here to Access the Code for HMAC Algorithm

1 Program for HMAC Algorithm

```
import numpy as np

def hmac(key, blocksize, message):
    # If Key is less than the blocksize then zero padding to the left is applied immediately

    I_pad = 54 * blocksize
    O_pad = 92 * blocksize

    # Inner padded key
    I_key_pad = xor_func(key, I_pad)
    print("Key Xor I pad is :", I_key_pad)

    # Outer padded key
    O_key_pad = xor_func(key, O_pad)
    print("Key Xor O pad is :", O_key_pad)

    key_message = []
    for c in message.upper():
        deci = L2I[c]
        print("\n The Decimal is: ", deci)
        print("The corresponding Binary is: ", binary(deci))

        key_mess = binary(deci)
        print("\n The Decimal of the Key and message is : ", int(key_mess, 2))
        print("The corresponding Binary is: ", key_mess)

        key_message.append(key_mess)
    print("Message Before Appending the pad ", np.array(key_message))

    # Appending the ( S1 || M )
    key_message.append(I_key_pad)
    print("After Appending the I pad to the Message ", np.array(key_message))

    # Appending the ( S2 || M )
    key_message.append(O_key_pad)
    print("After Appending the O pad to the Message and I pad ", np.array(key_message))
```

```
def xor_func(key, pad):
    bin_key = binary(key)
    bin_pad = binary(pad)
    print(bin_key)
    print(bin_pad)
    xor_output = binary(int(bin_key, 2) ^ int(bin_pad, 2))
    print("The binary Value is : ", xor_output)
    print("The decimal Value is : ", int(xor_output, 2))

    return xor_output

def binary(num, pre='0b', length=8, spacer=0):
    return '{0}{{:{1}>{2}}}'.format(pre, spacer, length).format(bin(num)[2:])

L2I = dict(zip("ABCDEFGHIJKLMNOPQRSTUVWXYZ", range(0, 26)))
I2L = dict(zip(range(0, 26), "ABCDEFGHIJKLMNOPQRSTUVWXYZ"))
```

2 Main Function Of Our Program Code

```
if __name__ == '__main__':
    try:
        input = raw_input
    except NameError:
        pass
    try:
        chr = unichr
    except NameError:
        pass

    blocksize = int(input("\nPlease Enter Blocksize : "))
    print("The Blocksize value is: ", blocksize)

    key = int(input("\nPlease Enter Key : "))
    print("The Entered Key is :", key)

    PlainText = input("\nPlease Enter PlainText : ")

    print("\nThe Entered PlainText is: \n " + PlainText + "\n" + "\n The PlainText in UPPERCASE is: \n '
          + PlainText.upper() + "\n")


    hmac(key, blocksize, PlainText)
```

2.1 The Output for Our Program

```
/usr/local/Cellar/python3/3.6.4_2/Frameworks/Python.framework/Versions/3.6/bin/python3.6 "/Users/admin/

Please Enter Blocksize : 512
The Blocksize value is:  512
```

```
Please Enter Key : 128
The Entered Key is : 128

Please Enter PlainText : hello

The Entered PlainText is:
 hello

 The PlainText in UPPERCASE is:
 HELLO

0b10000000
0b110110000000000
The binary Value is :  0b110110010000000
The decimal Value is :  27776
Key Xor I pad is : 0b110110010000000
0b10000000
0b1011100000000000
The binary Value is :  0b1011100010000000
The decimal Value is :  47232
Key Xor O pad is : 0b1011100010000000

 The Decimal is:  7
The corresponding Binary is:  0b00000111

 The Decimal of the Key and message is :  7
The corresponding Binary is:  0b00000111

 The Decimal is:  4
The corresponding Binary is:  0b00000100

 The Decimal of the Key and message is :  4
The corresponding Binary is:  0b00000100

 The Decimal is:  11
The corresponding Binary is:  0b00001011

 The Decimal of the Key and message is :  11
The corresponding Binary is:  0b00001011

 The Decimal is:  11
The corresponding Binary is:  0b00001011

 The Decimal of the Key and message is :  11
The corresponding Binary is:  0b00001011

 The Decimal is:  14
The corresponding Binary is:  0b00001110

 The Decimal of the Key and message is :  14
The corresponding Binary is:  0b00001110
Message Before Appending the pad  ['0b00000111' '0b00000100' '0b00001011' '0b00001011' '0b00001110']
After Appending the I pad to the Message  ['0b00000111' '0b00000100' '0b00001011' '0b00001011' '0b000011
 '0b110110010000000']
```

```
After Appending the 0 pad to the Message and I pad  ['0b00000111' '0b00000100' '0b00001011' '0b00001011
 '0b110110010000000' '0b1011100010000000']

Process finished with exit code 0
```