Encryption Key: 0111010101100101

Decrypted Message:

Always go to other people's funerals, otherwise they won't go to yours.

- Yogi Berra

Code:

'''

Homework Number: 1

Name: Alex Goebel

ECN Login: goebel2

Due Date: 1/28/2021

'''

#!/usr/bin/env python3

import sys #Do I need this or was this only for the example? Do the arguments need to be checked?

from BitVector import \*

def cryptBreak(ciphertextFile, key\_bv):

PassPhrase = "Hopes and dreams of a million years"

BLOCKSIZE = 16

numbytes = BLOCKSIZE // 8

bv\_iv = BitVector(bitlist = [0]\*BLOCKSIZE)

for i in range(0, len(PassPhrase) // numbytes):

textstr = PassPhrase[i\*numbytes:(i+1)\*numbytes]

bv\_iv ^= BitVector(textstring = textstr)

inFile = open(ciphertextFile)

encrypted\_bv = BitVector(hexstring = inFile.read())

decryptedMessage\_bv = BitVector(size = 0)

previousDecryptedBlock = bv\_iv

for i in range(0, len(encrypted\_bv) // BLOCKSIZE):

bv = encrypted\_bv[i\*BLOCKSIZE:(i+1)\*BLOCKSIZE]

temp = bv.deep\_copy()

bv ^= previousDecryptedBlock

previousDecryptedBlock = temp

bv ^= key\_bv

decryptedMessage\_bv += bv

outputText = decryptedMessage\_bv.get\_text\_from\_bitvector()

inFile.close()

return outputText

if \_\_name\_\_ == '\_\_main\_\_':

ciphertextFile = 'encrypted.txt'

keyRangeMax = 2 \*\* 16

for i in range(28000, keyRangeMax):

key\_bv = BitVector(intVal = i, size=16)

decryptedMessage = cryptBreak(ciphertextFile, key\_bv)

if 'Yogi Berra' not in decryptedMessage:

print('Not decrypted yet')

print(key\_bv)

else:

print('Encryption broken!')

print(key\_bv)

print(decryptedMessage)

break

Code Explanation:

In my main function I have the brute force analysis where I go through a for loop that creates a BitVector object for each key option and sends that object, along with the input file, to cryptBreak. The cryptBreak function is mostly taken from the code given in DecryptForFun.py. It first gets the passphrase into a bit array. Then, it uses the ciphertext file to create a BitVector. After, it creates a place to store the decrypted message. Finally, it XORs the bit blocks with the decryption and outputs the decrypted message. My main file then checks to see if ‘Yogi Berra’ is part of that decrypted message and if it isn’t, it tries the next key. If it is there, the loop ends and the key, message, and alert is printed to let the user know the encryption has been broken and what it says.