Code:

```
#####
#Homework Number: 1
#Name: Prekshaa Veeraragavan
#ECN login: pveerar
#Due Date: January 28, 2021
#####

import sys
from BitVector import *

def cryptBreak(ciphertextFile, key_bv):

    PassPhrase = "Hopes and dreams of a million years"

    BLOCKSIZE = 16
    numbytes = BLOCKSIZE // 8

    # Reduce the passphrase to a bit array of size BLOCKSIZE:
    bv_iv = BitVector(bitlist=[0] * BLOCKSIZE)  # (F)
    for i in range(0, len(PassPhrase) // numbytes):  # (G)
        textstr = PassPhrase[i * numbytes:(i + 1) * numbytes]  # (H)
        bv_iv ^= BitVector(textstring=textstr)  # (I)

    # Create a bitvector from the ciphertext hex string:
    FILEIN = open(ciphertextFile)  # (J)
    encrypted_bv = BitVector(hexstring=FILEIN.read())

    # Create a bitvector for storing the decrypted plaintext bit array:
    msg_decrypted_bv = BitVector(size=0)  # (T)

    # Carry out differential XORing of bit blocks and decryption:
    previous_decrypted_block = bv_iv  # (U)
    for i in range(0, len(encrypted_bv) // BLOCKSIZE):  # (V)
        bv = encrypted_bv[i * BLOCKSIZE:(i + 1) * BLOCKSIZE]  # (W)
        temp = bv.deep_copy()  # (X)
        bv ^= previous_decrypted_block  # (Y)
        previous_decrypted_block = temp  # (Z)
        bv ^= key_bv  # (a)
        msg_decrypted_bv += bv  # (b)

    # Extract plaintext from the decrypted bitvector:
    outputtext = msg_decrypted_bv.get_text_from_bitvector()  # (c)

    # return output text
    return outputtext

if __name__ == '__main__':
    for i in range(0,65536):
            trykey = chr(i)
            key_bv = BitVector(intVal=i, size=16)
            decryptedMessage = cryptBreak('encrypted.txt', key_bv)
            if ('Yogi Berra' in decryptedMessage):
                print('Encryption Broken!')
                print(i)
                print(decryptedMessage)
                break
            else:
                print('Not decrypted yet')
```

Recovered Plaintext Quote:

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

- Yogi Berra

Encryption Key: 30053

Method Used: Brute force

Since the encryption key is an integer between 0 and 2^{16}, the brute force method goes through all values in that range (0-65536) and implements them till the encryption is broken and the plaintext is recovered.