|  |
| --- |
|  |
| ASSIGNMENT 4 |
| One Pad Cipher |

|  |
| --- |
| Sukh Atwal  10-21-2020 |

Contents

[Introduction 2](#_Toc54215439)

[How to Use 3](#_Toc54215440)

[How XOR Works 4](#_Toc54215441)

[Testing 5](#_Toc54215442)

[Test 1 6](#_Toc54215443)

[Test 2 7](#_Toc54215444)

[Test 3 8](#_Toc54215445)

[Test 4 9](#_Toc54215446)

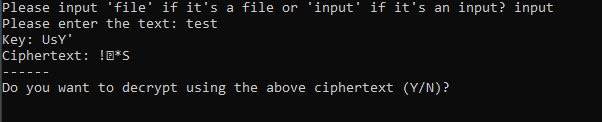
# Introduction

The objective of this assignment is to design and implement a one-time pad cipher using bit manipulation. The bit manipulator that will be used is XOR (Exclusive Or) which is a logical operation that outputs true only when the inputs are different. The script will also include input by either file or user with the option to decrypt as well as save the ciphertext.

The language I’ll be using is Python.

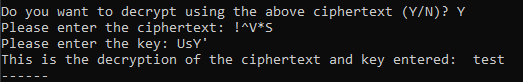
# How to Use

The script can be located in the “Source” folder. You can run it through a terminal, where it will prompt you for user inputs.



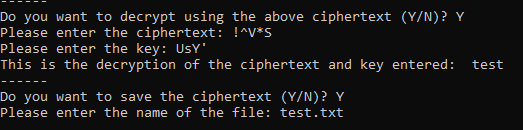
It should first ask you to choose either file or input, in this case I chose input. It then tells you to enter the text (Or file if you chose file), displaying the Key and Ciphertext afterwards.

For the purposes of this assignment, I also added if you want to decrypt the ciphertext. If you do, enter ‘Y’.

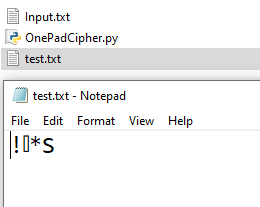


You would enter the relevant ciphertext and key and it will decrypt the message.

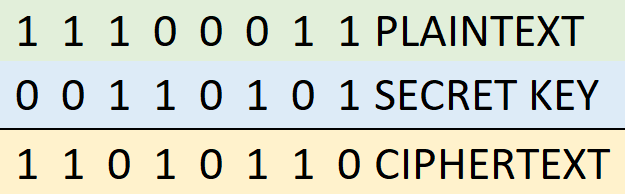
Finally, it’ll ask if you want to save the ciphertext.



Here, I chose yes or ‘Y’ again which prompted me to enter what I would want to name the file. I chose test.txt. It will save the file in the same directory as the script and end.

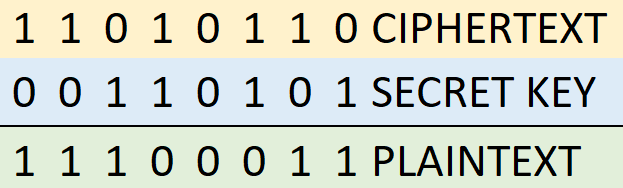


# How XOR Works

If we look back, we can see that XOR is a logical operation that outputs true only when the inputs differ. It would output ‘1’ if true and ‘0’ as false. Let’s look at how we encrypt below. 

If we look at the first line, we can see the plaintext is 1 and the secret key is 0. Since they are different, the output is ‘true.’ XOR outputs **TRUE** only when the **INPUT** **DIFFERS**. Similar to the second line but if we look at the third line we can see both plaintext and secret key are the same, ‘1’. Therefore, the ciphertext is 0 as the inputs are the same. This is how XOR encryption works.

Now to decrypt, we simply switch it up.



Using the same numbers as above, we can figure out the plaintext if we know the ciphertext and key. Similarly, we can look at each line and see that if the ciphertext and plaintext are different, the plaintext would be 1 and if they’re the same, it would be 0. This is how XOR decryption works.

# Testing

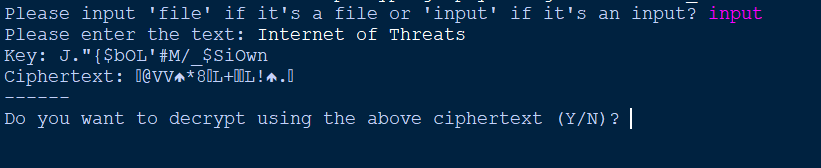
My testing setup is a Windows computer running Windows 10 using IDLE as my Python development platform of choice. You can also view all the tests in the ‘Videos’ folder.

For the tests I’ll be using the text “Internet of Threats” as my input and in a file called Input.txt I’ll have the text “who guards the guard”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test #** | **Tools Used** | **Expectation** | **Actuality** | **Pass or Fail** |
| 1 | IDLE | The ‘input’ option should allow me to input text and should output unintelligible encrypted text back. I should also be able to decrypt it using the key and ciphertext. | I was able to input any text and it output back unintelligible ciphertext. Decrypting using the ciphertext and key, I was able to go back to the original plaintext. | Pass |
| 2 | IDLE | The ‘file’ option should allow me to input text and should output unintelligible encrypted text back. I should also be able to decrypt it using the key and ciphertext. | I was able to input a file and it output back unintelligible ciphertext. Decrypting using the ciphertext and key, I was able to go back to the original plaintext. | Pass |
| 3 | IDLE | You should be able to save the same ciphertext that was output in the terminal. | The script asked whether I wanted to save the file and it did. | Pass |
| 4 | IDLE | The script should not work if the user does not enter ‘file’ or ‘input’ and output a message saying to enter the correct inputs | The script asked for the correct inputs when the user input ones that weren’t listed | Pass |

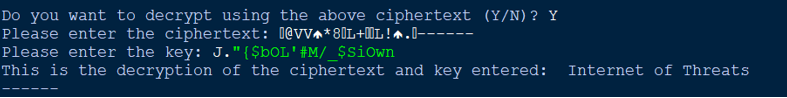
## Test 1

I’ll be testing the input function of the script.



Using the input ‘Internet of Threats’ it outputted the key as well as ciphertext.

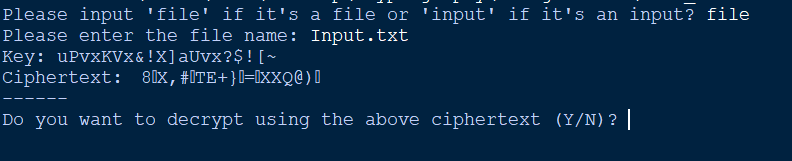
Using the key and ciphertext, I’ll test if we can decrypt it to the same plaintext as we originally had.



As we can see, it successfully decrypted the original plaintext using the ciphertext and key.

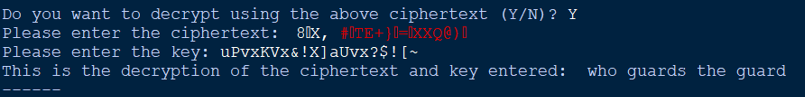
## Test 2

Similar to Test 1 but I’ll be testing the file functionality of the script instead.



Here we enter the file Input.txt which has the text “who guards the guard”. It successfully outputted the key and ciphertext.

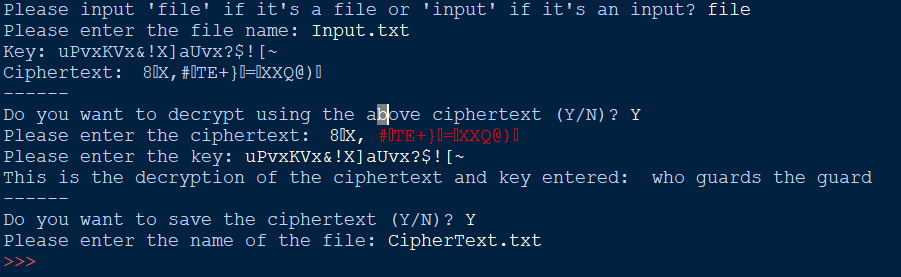
Using the above key and ciphertext, we get the following.



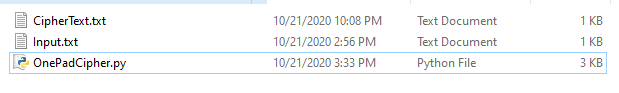
As we can see, the plaintext is “who guards the guards” which is what we had in the file originally.

## Test 3

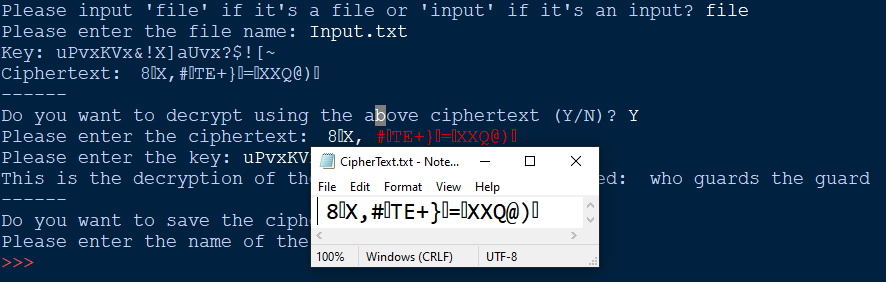
Here I’ll be testing the saving functionality of the script.



Here I save the ciphertext to a file called ‘Ciphertext.txt’ which should be located in the same directory as the script which we can see here.



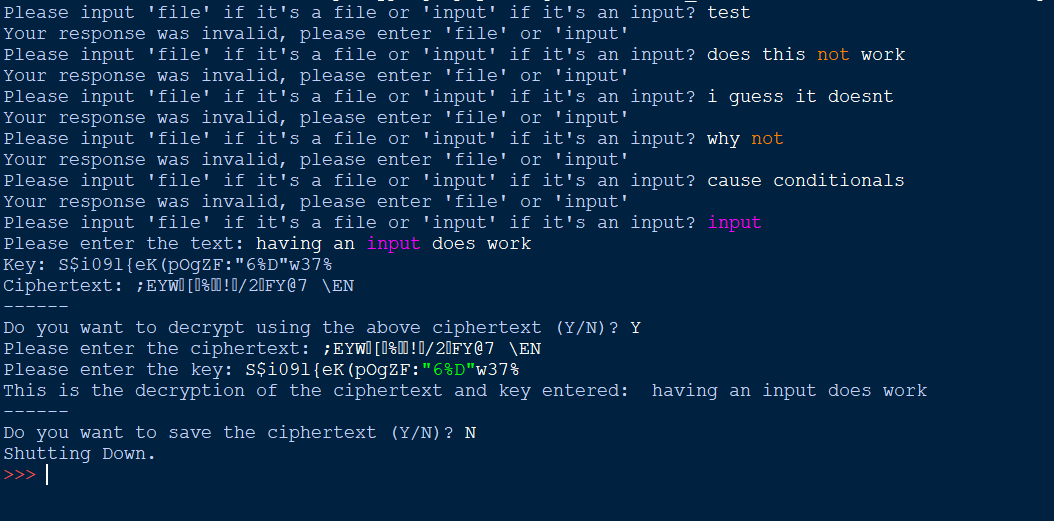
And if we open the CipherText.txt file, it should output the same ciphertext.



Which it does. We can see that the text file saved the ciphertext correctly.

## Test 4

I’ll be testing whether the script will still run if what I enter at the start isn’t what’s specified.



We can see that it does not work and keeps repeating to the original line asking to enter ‘file’ or ‘input’.