

ASSIGNMENT 4

One Pad Cipher

Sukh Atwal
10-21-2020

Contents

Introduction	2
How to Use.....	3
How XOR Works.....	4
Testing.....	5
Test 1.....	6
Test 2.....	7
Test 3.....	8
Test 4.....	9

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.

```
Please input 'file' if it's a file or 'input' if it's an input? input
Please enter the text: test
Key: UsY'
Ciphertext: !^*S
-----
Do you want to decrypt using the above ciphertext (Y/N)?
```

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’.

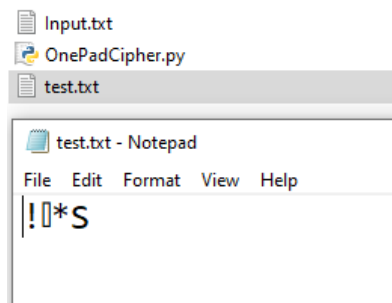
```
Do you want to decrypt using the above ciphertext (Y/N)? Y
Please enter the ciphertext: !^V*S
Please enter the key: UsY'
This is the decryption of the ciphertext and key entered: test
-----
```

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.

```
-----
Do you want to decrypt using the above ciphertext (Y/N)? Y
Please enter the ciphertext: !^V*S
Please enter the key: UsY'
This is the decryption of the ciphertext and key entered: test
-----
Do you want to save the ciphertext (Y/N)? Y
Please enter the name of the file: test.txt
```

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.

1	1	1	0	0	0	1	1	PLAINTEXT
0	0	1	1	0	1	0	1	SECRET KEY
1	1	0	1	0	1	1	0	CIPHERTEXT

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.

1	1	0	1	0	1	1	0	CIPHERTEXT
0	0	1	1	0	1	0	1	SECRET KEY
1	1	1	0	0	0	1	1	PLAINTEXT

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.

```
Please input 'file' if it's a file or 'input' if it's an input? input
Please enter the text: Internet of Threats
Key: J."{$bOL'#M/_$SiOwn
Ciphertext: @@VV^*8LL+LL!^..
-----
Do you want to decrypt using the above ciphertext (Y/N)? |
```

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.

```
Do you want to decrypt using the above ciphertext (Y/N)? Y
Please enter the ciphertext: @@VV^*8LL+LL!^..-----
Please enter the key: J."{$bOL'#M/_$SiOwn
This is the decryption of the ciphertext and key entered: Internet of Threats
-----
```

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.

```
Please input 'file' if it's a file or 'input' if it's an input? file
Please enter the file name: Input.txt
Key: uPvxKVx&!X]aUvx?$![~
Ciphertext: 8[X, #[TE+}]=[XXQ@)
-----
Do you want to decrypt using the above ciphertext (Y/N)? |
```

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.

```
Do you want to decrypt using the above ciphertext (Y/N)? Y
Please enter the ciphertext: 8[X, #[TE+}]=[XXQ@)
Please enter the key: uPvxKVx&!X]aUvx?$![~
This is the decryption of the ciphertext and key entered:  who guards the guard
-----
```




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.

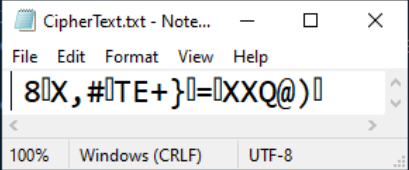
```
Please input 'file' if it's a file or 'input' if it's an input? file
Please enter the file name: Input.txt
Key: uPvxKVx&!X]aUvx?$![~
Ciphertext: 8[X, #TE+}=XXQ@)
-----
Do you want to decrypt using the above ciphertext (Y/N)? Y
Please enter the ciphertext: 8[X, #TE+}=XXQ@)
Please enter the key: uPvxKVx&!X]aUvx?$![~
This is the decryption of the ciphertext and key entered:  who guards the guard
-----
Do you want to save the ciphertext (Y/N)? Y
Please enter the name of the file: CipherText.txt
>>>
```

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.

	CipherText.txt	10/21/2020 10:08 PM	Text Document	1 KB
	Input.txt	10/21/2020 2:56 PM	Text Document	1 KB
	OnePadCipher.py	10/21/2020 3:33 PM	Python File	3 KB

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

```
Please input 'file' if it's a file or 'input' if it's an input? file
Please enter the file name: Input.txt
Key: uPvxKVx&!X]aUvx?$![~
Ciphertext: 8[X, #TE+}=XXQ@)
-----
Do you want to decrypt using the above ciphertext (Y/N)? Y
Please enter the ciphertext: 8[X, #TE+}=XXQ@)
Please enter the key: uPvxKV
This is the decryption of the ciphertext and key entered:  who guards the guard
-----
Do you want to save the ciphertext (Y/N)? Y
Please enter the name of the file: CipherText.txt
>>>
```



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.

```
Please input 'file' if it's a file or 'input' if it's an input? test
Your response was invalid, please enter 'file' or 'input'
Please input 'file' if it's a file or 'input' if it's an input? does this not work
Your response was invalid, please enter 'file' or 'input'
Please input 'file' if it's a file or 'input' if it's an input? i guess it doesnt
Your response was invalid, please enter 'file' or 'input'
Please input 'file' if it's a file or 'input' if it's an input? why not
Your response was invalid, please enter 'file' or 'input'
Please input 'file' if it's a file or 'input' if it's an input? cause conditionals
Your response was invalid, please enter 'file' or 'input'
Please input 'file' if it's a file or 'input' if it's an input? input
Please enter the text: having an input does work
Key: S$i09l{eK(p0gZF:"6%D"w37%
Ciphertext: ;EYW[[]%[]!/2[]FY@7 \EN
-----
Do you want to decrypt using the above ciphertext (Y/N)? Y
Please enter the ciphertext: ;EYW[[]%[]!/2[]FY@7 \EN
Please enter the key: S$i09l{eK(p0gZF:"6%D"w37%
This is the decryption of the ciphertext and key entered:  having an input does work
-----
Do you want to save the ciphertext (Y/N)? N
Shutting Down.
>>> |
```

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