OTP Encipher-decipher Example

Updated 10-23-2023.

This example assumes that you have read the following documentation and have edited the settings files to reflect the name and location of your Encipher Decipher directory.

0-Shannon-nformation-theory.pdf

1-Python One Time Pad Software.pdf

2-Installing Python Encryption files.pdf

3-Cipher Machines and Cryptology-OTP.pdf

1. Start **Python IDLE** and Open OTP-Encipher-decipher.py
2. Use **Run** or F5 on the top menu bar and launch the application
3. Use **operation 10** to load example.txt. If you have not edited this settings file use operation 9 to save it. Exit Python and use a text editor such as **NotePad** to open **example.txt.** At the end of the file, you will find the path where you have installed encipher-decipher. Use search and replace to update the directory paths. Save the file. Run OTP-Encipher-decipher.py Use **operation 10** to load example.txt.
4. The Command Menu will reflect the files and their correct locations.
5. Let's verify one of the settings. **Choose Command 1**
    a. It will show the current working directory where the file **Engineering Ethics.txt** is located
    b. Hit ***enter*** to accept this directory or enter 'c' to change it. Enter 'c'. You will see a list of directories. Enter the number corresponding to **Plain Text**.
    c. You should see files in this directory including **Engineering Ethics.txt**.
    d. Enter the number next to it to select it.
    e. You will see a confirmation
    f. Hit enter to accept it. (It is good practice to check the rest of the files as well) this will take you back to the Command Menu
6. Use function 6 to enter a pass phrase if needed. In this example we use 'The Quick Brown Fox.' This sets the cipher key to 478510. The cipher key can be set to any value between 0 and 1999999 for this OTP file.
7. When we choose **Command 7** the plain text file **Engineering Ethics.txt** will be XORed byte by byte with the OTP file (OTP-2MB.bin) starting at position 478510. The resulting enciphered file will be **engineering ethics.bin** placed in the **Enciphered** directory.
8. When we use **Command 7** you will get a verification list showing what is about to happen. If everything is correct hit **Enter**. You will see the encryption proceed and get a final result. We are now back where we started. The file **engineering ethics.bin is the enciphered file** and can be sent to recipients who have the **same OTP and know the cipher key or pass phrase.**
9. Now let's decipher **engineering ethics.bin** Choose **Command 3** which lets you choose the file to be deciphered. **Command 4** lets you choose the deciphered file destination file. Note that we

are placing the deciphered file in the **Deciphered** directory so there is no chance of destroying our original plain text file.

10. Choose **Command 8.** Again, we get a verification list and a chance to exit. If everything looks *OK* hit ***Enter*.** We see the decryption progress and get a result report.
11. Once you have made changes to a group of settings **use Command 9 to save the** settings to a settings file. The default is settings.txt. Choose new names for settings so you can recall important settings for regular tasks quickly.
12. Go to the Deciphered folder and verify that **engineering ethics.txt** is identical to the original file in the Plain Text directory.
13. Use Command 10 to load a previously saved setting.
14. Use Command 11 to reset all values to 'none'.
15. Use Command 12 to exit the program.

Additional Comments

Once you become accustomed to using this tool you can create an encipher-decipher communication network that even government agencies such as the CIA or NSA cannot decipher. This is completely **Open Software. There can never be any hidden backdoors.** You can use a huge number of ordinary files as One Time Pads. Use the **OTP-Compute-Entropy.py** program to measure their randomness. You will be surprised at how random **Photo JPG** files are. You can share photos with recipients for use as OTP files. Out of a gallery of 1000 photos only you and the others in your network know which ones are being used as OTP's. In addition, the cipher key or pass phrase is required to decipher an enciphered file.

As an experiment take one of your photo jpg files and encipher it using another photo jpg file as the OTP. Then reverse the process and decipher the enciphered file to the Deciphered directory. The resulting deciphered jpg file will be identical to the original file.

A number of JPG photo files and a settings file named **photo.txt** are provoded if you want to experiment.

What if you need to encipher a large number of files and directories of files?

If you need to encipher a large number of files and directories, put them in a **ZIP** file and then encipher the ZIP file. It is that easy. There is almost no limit to what you can do with this tool to protect your privacy and security. Python is well worth every engineer's time to learn. It is a powerful programming environment capable of solving very complex problems. You can learn a lot about Python by studying this program.

The Bad Actors do not know what they are up against!

Have fun

Bill Marcy