Seminar 2 Preparation (and e-portfolio entry)

Read the Cryptography with Python blog at tutorialspoint.com (link is in the reading list). Select one of the methods described/ examples given and create a python program that can take a short piece of text and encrypt it.

Create a python program in Codio (you can use the Jupyter Notebooks space provided in the Codio resources section) that can take a text file and output an encrypted version as a file in your folder on the Codio system. Demonstrate your program operation in this week's seminar session.

Answer the following questions in your e-portfolio:

- Why did you select the algorithm you chose?
- Would it meet the GDPR regulations? Justify your answer.

We will review your work from Unit 4 in this week's seminar, as well as this cryptography activity. There will also be an opportunity to review your team's assignment progress during the seminar.

Remember to record your results, ideas and team discussions in your e-portfolio. You also need to ensure your initial design has been reviewed and approved by your tutor **BEFORE** you start work on the coding exercise for the module assessment due in Unit 6.

Learning Outcomes

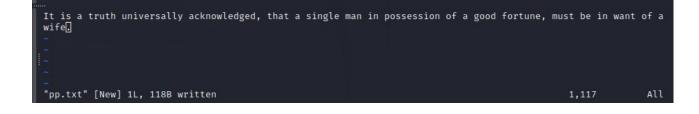
- Identify and manage security risks as part of a software development project.
- Critically analyse development problems and determine appropriate methodologies, tools and techniques (including program design and development) to solve them.
- Design and develop/adapt computer programs and to produce a solution that meets the design brief and critically evaluate solutions that are produced.
- Why did you select the algorithm you chose?
 - The program's encryption is not deprecated, unlike code for Caesar's Cipher or other such programs.
 - It looked complicated enough to be a challenge to compose without being so complicated as to prove impossible for a beginner like me.
- Would it meet the GDPR regulations? Justify your answer.
 - Yes, for the following reasons (Tutorialspoint. 2023):
 - It is an unbreakable cipher
 - The key is as long as the message encrypted
 - The key is made up of random symbols
 - The key is used once and never again.

L. M. Saxton

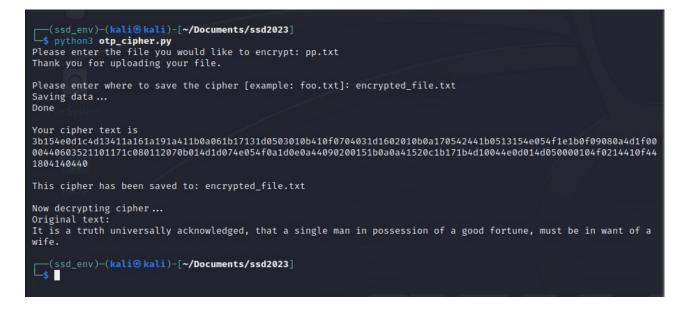
The python program for One Time Pad encryption:

```
import onetimepad
input_file = input("Please enter the file you would like to encrypt: ")
# Open file with reading privileges
with open(input_file, 'r') as data_file:
     data = [list(line.rstrip('\n')) for line in data_file]
flat_data_list = []
for element in data:
     if type(element) is list:
          for item in element:
              flat_data_list.append(item)
result = flat_data_list
print("Thank you for uploading your file.\n")
cipher = onetimepad.encrypt(result, "random")
output_file = input("Please enter where to save the cipher [example: foo.txt]: ")
print("Saving data...")
with open(output_file, 'w') as encrypted_file:
     encrypted_file.writelines(cipher)
print("Done\n")
msq = onetimepad.decrypt(cipher, "random")
print(cipher)
print(f"\nThis cipher has been saved to: {output_file}\n")
print("Now decrypting cipher ... ")
print(f"Original text: \n{msq}")
```

using the following text to encrypt:



results in the following output:



The new, encrypted file is shown here:



References:

TutorialPoint (2020) Cryptography with Python Tutorial. Available at:

https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_one_tim e_pad_cipher.htm#