

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.


```
(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ python3 otp_cipher.py
Please enter the file you would like to encrypt: pp.txt
Thank you for uploading your file.

Please enter where to save the cipher [example: foo.txt]: encrypted_file.txt
Saving data...
Done

Your cipher text is
3b154e0d1c4d13411a161a191a411b0a061b17131d0503010b410f0704031d1602010b0a170542441b0513154e054f1e1b0f09080a4d1f00
00440603521101171c080112070b014d1d074e054f0a1d0e0a44090200151b0a0a41520c1b171b4d10044e0d014d050000104f0214410f44
1804140440

This cipher has been saved to: encrypted_file.txt

Now decrypting cipher ...
Original text:
It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a
wife.

(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$
```

The new, encrypted file is shown here:

```
3b154e0d1c4d13411a161a191a411b0a061b17131d0503010b410f0704031d1602010b0a170542441b0513154e054f1e1b0f09080a4d1f00
00440603521101171c080112070b014d1d074e054f0a1d0e0a44090200151b0a0a41520c1b171b4d10044e0d014d050000104f0214410f44
1804140440
~
~
~
~
"encrypted_file.txt" 1L, 235B written                                1,234      All
```

References:

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

https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_one_time_pad_cipher.htm#