Assignment 1:

In this Assignment, you will be applying symmetric encryption, decryption, and hashing of data files using the Python programming language.

Please note that we have changed what needs to be uploaded. This is not any major structural difference to the assignment though. It does help to clarify an issue that some students have identified with task 2.

Learning Outcomes linked to this assessment

- ULO1: Explain the concepts and principles on which modern cryptography relies upon.
- ULO2: Employ adapted cryptographic tools and techniques to encrypt, decrypt and sign messages.
- ULO3: Decipher simple encrypted messages using a range of cryptanalysis methods.

Key Assignment Details

Due date: Sunday 24 September 11:55pm Sydney time (note, this is extended - end of the mid-session teaching break).

Submission method: in the iLearn A1 submission box, there are ~~two~~ ***five*** files to submit... 1 txt file (assignment_answers_YOURSTUDENTID.txt) containing your answers to the tasks below and 1 python file containing your updated code from the crypto_a1_activity template. DO NOT zip or compress the files. submit the two files as is. No folders, no zip, no rar, and no compressed formats. ***You also need to upload task2.txt, task4.txt and task5.txt as per the instructions in each task.***

Number of submissions: You can submit / update as many times as you like between now and the deadline. You are encouraged to submit early drafts in case of any last-minute technical issues (even if only partially completed).

Late submissions, illness / special consideration cases: (from the unit guide) Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark) will be applied each day a written assessment is not submitted, up until the 7th day (including weekends). After the 7th day, a grade of '0' will be awarded even if the assessment is submitted. Submission time for all written assessments is set at 11:55 pm. A 1- hour grace period is provided to students who experience a technical concern.

The Assignment

Your mission, should you choose to accept it, is to take a given data file and a partially complete python file and finish the encryption / decryption / hashing / and collision checking activities outlined below.

While this is not a programming unit, you may, in the future need to explore how to use a new library to achieve a task. This assignment will get you to look at the documentation for a small number of Python cryptographic libraries and apply those libraries to your problem. the main cryptography python library - https://cryptography.io/en/latest/ can be used to provide some functionality to encrypt, decrypt, and hash files with symmetric cryptography and hash functions.

What we've provided for you:

1. **birthday_attack_statistics.py -** A file that runs a simulated 'birthday attack' scenario on a random sample of dates in a given date range and reports on the average number of attempts to get a collision.
2. **crypto_a1_activity.py** - A skeleton python file is provided in iLearn just under this specification . It contains functions to-be completed for this assignment.
3. **assignment_answers.txt** - A template text file for you to replace selected lines with your answers / outputs from programs.
4. two files, but you only need to use one of them depending on your OS.
    1. **windows_data_encrypted.txt**
    2. **mac_linux_data_encrypted.txt**

Instructions

Setup: Put all of the files (birthday_attack_statistics.py, crypto_a1_activity.py, assignment_answers.txt, windows_data_encrypted.txt, mac_linux_data_encrypted.txt) into a folder and open that folder up in your chosen python IDE (e.g. visual studio code).

Task 0: [2 marks]

a) run birthday_attack_statistics.py and copy the last line of the console output into the assignment_answers.txt file for subtask a. The console line should start with
"average number of attempts ...."

b) now, modify birthday_attack_statistics.py to start a search from your birthday date up until the assignment due date (24 September 2023). Use the lecture slides on the birthday paradox to calculate what your expected size should be for a 50% chance of a collision to have occured with this new date range. Run your updated birthday_attack_statistics.py and copy the new updated console output (just the last line) for subtask b. In the line underneath, indicate if the average was close (within +-2 of what you calculated) or not close (outside the +-2 of what you calcualted).

example for b below:

average number of attempts to find a collision over the 1000 runs is 5.532
not close

Task 1: [4 marks]

You have been be given an encrypted file that a student has encrypted, but they forgot which student ID they typed. You suspect they might have used the student ID 4000000 instead of their student number.

- **modify the task_1(...) function in crypto_a1_activity.py**
- **modify the decrypt_file(...) function in crypto_a1_activity.py**

to call the functions you need to try and decrypt a file. We've provided some empty functionsin the file for you to complete with some guidance. You will also need to implmenet the decrypt_file(...) function for your task 1 to be completed.

Once you've completed the code, run your python code to decrypt the file and see what the output was: (how to run it...)

Windows users in vscode terminal:

```
python crypto_a1_activity.py 40000000 task1 windows_data_encrypted.txt
windows_data_decrypted.txt
```

Mac / Linux users in vscode:

```
python crypto_a1_activity.py 40000000 task1 mac_linux_data_encrypted.txt
mac_linux_data_decrypted.txt
```

If you've succesfully decrypted the file, then copy line 5 of the text file and paste it as your answer. Hopefully the text should be readable in English...

Task 2: [3 marks]

Now that you've sucessfully decrypted the file, what we would like to you to do is re-encrypt the file. You will need to add code into the task_2 function and also implement the encrypt_file function

- **modify the task_2(...) function in crypto_a1_activity.py**
- **modify the encrypt_file(...) function in crypto_a1_activity.py**

1) If you've successfully implemented the function, then you should be able to run the python file with student number 40000000, take in your decrypted file, and then output to a newly encrypted txt. ~~file and compare the newly encrypted file with the original file downloaded from iLearn. If they are the same, then you are ready to run task 2.~~ ***[assignment 2 - task 2 modification]. You will notice that the file is not the same as the original encrypted file that we provided you. That is because the fernet encrypt function actually does some cheeky things under the hood with the choice of cypher block mode (see week 5 lectures)! Use the fernet library link here [https://cryptography.io/en/latest/fernet/] to find out what block mode fernet uses for encryption and tell us what block mode it uses in your answer for task 2.***

```
python crypto_a1_activity.py 40000000 task2 windows_data_decrypted.txt
windows_data_decrypted_encrypted.txt
```
or
```
python crypto_a1_activity.py 40000000 task2 mac_linux_data_decrypted.txt
mac_linux_data_decrypted_encrypted.txt
```

2) Take the same decrypted file and encrypt it using your own student number. save the output file as task2.txt
```
python crypto_a1_activity.py YOUR SID HERE... task2 windows_data_decrypted.txt task2.txt

python crypto_a1_activity.py YOUR SID HERE... task2 mac_linux_data_decrypted.txt task2.txt
```

3) ~~copy the first 20 characters of the task2.txt file and paste it in the answers txt file for task 2.~~ We will check the start of your has with our version run with your student number. ***[assignment 2 - task 2 modification]: upload the task2.txt as one of the files you submit. We will then check that it is able to be decrypted using our decryption algorithm.[note: If you've been able to decrypt the file from task 1, and are able to decrypt your task2 file, then you should be ok]***

Task 3: [2 mark]

use your student number to encrypt and decrypt the decrypted file. Write "Success:" or "not success" in your answer file for task 3. provide the first 20 characters of the encrypted file (which should be the same as task 2). ***[assignment 2 - task 3 clarification]:* we will check your task2 file and run some checks on it, we will aslo you YOUR decryption function to decrypt your task 2 file**.

- **modify the task_3(...) function in crypto_a1_activity.py**

Task 4: [2 marks]

hash the unencrypted data file using SHA256 and copy the hash into task 4
```
python crypto_a1_activity.py YOUR SID HERE... task4 windows_data_decrypted.txt task4.txt
```
or
```
python crypto_a1_activity.py YOUR SID HERE... task4 mac_linux_data_decrypted.txt task4.txt
```

~~write the hash into the assignment_answers.txt file.~~ ***[assignment 2 - task 2 modification]: upload task4.txt with your assignment submission***

- **modify the task_4(...) function in crypto_a1_activity.py**
- **modify the generate_hash(...) function in crypto_a1_activity.py**

Task 5: [2 marks]

hash the file that was encrypted with your student number (task2.txt) **and save the resultsin task5.txt**
copy the hash stored in the output file into the assignment_answers.txt file in task 5.

[assignment 2 - task 2 modification]: upload task5.txt with your assignment submission

- **In theory, task 5 can call task 4...**

What to Submit

1. Completed python file named as 'crypto_a1_activity_[ID].py' where [ID] needs to be replaced with your student ID number.
2. Text file containing the answers requested for task 0, 1, 2, 3, 4, 5. The filename should be of the format 'assignment_answers_[ID].txt' where [ID] needs to be replaced with your student ID number.

3. ***task2.txt - the file that gets generated from using the encryption using your student number***
4. ***task4.txt - the file that gets generated from hashing the file discussed in task 4***
5. ***task5.txt - the file that gets generated from hashing the file discussed in task 5***

Background

The Cryptography Library in Python

The cryptography Python package/library provides cryptographic recipes and primitives to Python developers. It is a commonly used and a standard cryptographic library and it supports Python 3.6+.

It includes both high level recipes and low level interfaces to common cryptographic algorithms such as symmetric ciphers, asymmetric algorithms, message digests, and key derivation functions.

You will learn the basics of cryptography library and the Fernet encryption and authentication algorithm in Week 3 lectures.

References and links:
https://pypi.org/project/cryptography/

https://cryptography.io/en/latest/#

https://cryptography.io/en/latest/hazmat/primitives/

https://docs.python-guide.org/scenarios/crypto/

Marking Rubric

The assignment is marked out of a total of 15.

The marking rubric for this assignment is as below:

Task 0: (2 marks)

task 0 has been answered in the submitted .txt file and shows the variance between the initial run and the second run.

Task 1: (4 marks)

3 marks for the completed, readable and working code in the python file and 1 marks for the correct execution/output

Task 2: (3 marks)
2 marks for the correct and readable code ，1 mark for the copy of the first 20 characters of the task2.txt matching the marker's run of their code using your student number ***[modification] and the answer for what block mode is being used in fernet's encryption and decryption algorithms***.

Task 3: (2 marks)
2 marks for the completed and readable code in the task_3 function.

Task 4: (2 marks)
1 mark for the completed and readable code 1 mark for the correct execution and output

Task 5: (2 marks)
1 mark for the completed and readable code 1 mark for the correct execution and output

Changelog:
1 Sep 2023: updated reference to assignment1_answers.txt in "task 0 a" to reflect the filename assignment_answers.txt

6 Sep 2021: updated files to be submitted - it doesn't change the code that needs to be written, just that there are more of the output files needed (updates are in blue bold italics).

Last modified: Thursday, 7 September 2023, 6:45 PM

◄ Module Exam 1

Jump to...

birthday attack statistics ►

Contact site support

You are logged in as Rafeul Islam (Log out)