Introduction to Modern Cryptography (Labs)

General Instructions

Requirements

- Programming language: Python 3 (at least Python 3.10; you are encouraged to use the latest version of Python 3)
- Integrated Development Environment: JetBrains PyCharm
    - Hint: You can get a free license through your academic email address.
    - To install JetBrains PyCharm, download JetBrains Toolbox and then select *PyCharm* to be installed. You can also install the official Chinese (Simplified) Language Pack if you are not familiar with English.

Deliverables

- A `main.py` file as the program, as well as any Python files that are also parts of your program.
- If the program needs any third-party libraries that are installed with pip, the `requirements.txt` file must be provided. Refers to Requirements Files for details.
- An `input.txt` file, contains a demo input of the program. Can be omitted if the program does not require inputs.
- An `output.txt` file, contains the corresponding console output of the program.
- A `readme.pdf` file as a documentation, consisting of the following items:
    - Your name, student ID, and the lab name (Lab1, Lab2, etc.).
    - Your operating system version, CPU instruction set (x86-64, arm64, etc.), and Python interpreter version.
    - A screenshot of the console output of the program.
- Pack up all your files above into a `.zip` archive. The filename must be of form `StudentID-StudentName-LabName.zip`. For instance, a student named `李狗蛋` with the student ID `202100010001` should name the zip file as `202100010001-李狗蛋-Lab1.zip` for his lab 1 assignment.
- Submit the zip file as the attachment to the email address `qi@huayi.email` before the deadline.

Grading

Category	Credit Percentage
the demo input	20%
code quality	20%
submit on time	-20%
finish the lab content	60%

The demo input

To get the credits, your program should run successfully under your demo input without runtime errors.

Code quality

To get the credits, write clean and simple code, using comments to explain what is not intuitive. Avoid spaghetti-like code. Name the variables properly. Write functions with no side effects to implement the crypto procedures, such as encryption, decryption, signing, etc.

Submit on time

To avoid lose credits, submit the zip file before the deadline.

Finish the lab content

To get the credits, finish the lab content. Finishing partial contents get the corresponding credits.

Emergency Submit

In the case of an emergency, such as network issues and power failures, you should inform the TA as soon as possible. If possible, you should try to submit a hash digest of your zip file to the TA in any possible way, to prove that you have finished your work before the deadline, and submit the corresponding zip file in a few days later. You will not lose credits if you follow the steps here correctly.