ISEC2000 Fundamental Concepts of Cryptography & ISEC5002 Introduction to Cryptography
Assignment 2, 2022
@ Computing, Curtin University

Weighting:
This assignment contains 3 questions, for a total of 100 points, which weights for 25% of the final mark.

Submission:
You should submit **a single ZIP** file to Blackboard. Name the file as `<studentID>_<name>_assignment02.zip`. It should contain the code, report, and text files. Use the `Declaration_of_originality.pdf` as the cover page of your report. The due date is **22 May 2022 11:59 PM**.

Academic Integrity:
This is an **individual** assignment so that any form of collaboration is not permitted. This is an **open-book** assignment so that you are allowed to use external materials, but make sure you properly **cite the references**. It is your responsibility to understand Curtin's Academic Misconduct Rules, for example, post assessment questions online and ask for answers is considered as contract cheating and not permitted.

Question answering

1. (10 points) The Euclidean algorithm is based on the following assertion. Given two integers a, b, $(a > b)$,

$$\gcd(a, b) = \gcd(b, a \bmod b). \tag{1}$$

   Prove the assertion (1) **mathematically**. (Note that proof by example is NOT appropriate here)

2. (20 points) Assuming that Alice signed a document m using the RSA signature scheme. (You should describe the RSA signature structure first with a diagram and explain the authentication principle). The signature is sent to Bob. Accidentally Bob found one message m' $(m \neq m')$ such that $H(m) = H(m')$, where $H()$ is the hash function used in the signature scheme. Describe clearly how Bob can forge a signature of Alice with such m'.

Programming

3. (50 points) Implement the RSA algorithm (C/C++, Java, Python). The requirements are as follows:

   - Implement each component as a separate function, such as key schedule, prime test, the extended Euclidean algorithm, binary modular exponentiation, and so on.
   - Implement both encryption and decryption of RSA. Encryption takes a txt file as input and output another txt file containing ciphertext (use hexadecimal for easy readability). Decryption should recover the plaintext.
   - Your code should encrypt and decrypt standard keyboard characters, including letters, numbers, and symbols.
   - The prime numbers p and q should be larger than 2^{64}. (you are allowed to use libraries to handle large numbers, such as BigInteger in Java)

- The strategy of source coding (converting characters to integers in RSA) is up to you. You can encrypt one or more characters at a time, but make sure the constraint $m < n$ is satisfied.
- Use the provided file RSA-test.txt to test your code.

After implementing your code, please **answer the following questions** in your report:

(a) (10 points) What are the lessons you learned, and difficulties you met, in the process of implementing RSA?

(b) (10 points) Describe what you have done for source coding and decoding.

END OF ASSIGNMENT