SE6018 – Blockchains and Cryptocurrency Individual Project

Submission Deadline: 2nd May, 23:59PM

Weight of the course: 50%

Submission Format:

- Submit through NTULearn -> Assignments -> Individual Project -> Upload Files
- Your submission should be a single zip file containing all the programs, text files of the RSA keys, and output.txt file to capture all outputs from your program to demonstrate a trail run of your program.
- Unlimited number of attempts are allowed, only the last attempt is counted

Project Description:

Implement the ring signature with 3 users discussed in the lecture with the following specifications:

1. The width of the operations are 1024-bits

2. The symmetric-key encryption is AES-128 in CBC mode with IV=0: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_block_chaining_(CBC)
   To encrypt 1024 bits, you need to break the plaintext into 8 blocks of 128-bit each.

3. The master key to AES-128 the first 128-bit of SHAKE128 digest with this file as input. Detailed description of SHAKE128 can be found here https://en.wikipedia.org/wiki/SHA-3

4. All additions are bit-wise XOR

5. Generate the (N, p, q, d, e) for all user1, user2 (you), and user3, where p and q should be distinct 512-bit long and N is 1024-bit long, store (N, p, q, e) in a file named user{i}_keys.txt for {i} = 1, 2, and 3. This set of keys including N should be all distinct from each other's among the 3 users.

6. Generate a valid (v, x1, x2, x3) ring signature, where v = your matric card number (all letters are upper cases), x1 = 1, and x3 = 3.

7. Your program should output the ring signature (v, x1, x2, x3), and has a function to verify the correctness of the ring signature and output "verification passed" once successful.

8. Whenever there is a need to extend the length of a string to a fixed length (e.g., 1024 bits), append it with enough “0” bits.

In the program, you are free to use existing libraries to generate or verify prime numbers, however, you are NOT allowed to re-use any RSA library to generate the any of the (N, p, q, d, e), instead you will need to write your own programs to generate these numbers and test if they fulfills the requirements. You should also write your own functions to compute m^d and m^e for the RSA encryptions and decryptions.