

INTERNET SECURITY (M1522.002300 001)

# **PROGRAMMING ASSIGNMENT #1**

## **RSA ENCRYPTION / DECRYPTION**

**Deadline: 2022. 10. 16 (Sun) 23:59**

**TA: Hyunsoo Kim (hskim@mmlab.snu.ac.kr)**

# INTRODUCTION

- Write a program to implement the RSA public-key cryptosystem
  1. Choose two large prime numbers  $p$  and  $q$ .
  2. Compute  $n = pq$ .
  3. Compute  $\phi(n) = (p - 1)(q - 1)$
  4. Choose an integer  $e$  such that  $1 < e < \phi(n)$  and  $\gcd(e, \phi(n)) = 1$
  5. Determine  $d$  as  $d \equiv e^{-1} \pmod{\phi(n)}$ ; that is,  $d$  is the modular multiplicative inverse of  $e \pmod{\phi(n)}$
- Public Key:  $(e, n)$       Private Key:  $d$       key: 1024 bit
- Encryption       $c \equiv m^e \pmod{n}$ .
- Decryption       $c^d \equiv (m^e)^d \equiv m \pmod{n}$ .
- Since you will have to compute large exponents as part of the RSA, the built-in C type integers (int - 16 or 32 bits, long int - 32 bits and even long long int - 64 bits) will not suffice

# ENVIRONMENT

## ■ Write in C, C++

- Compiler

- ✓ Debian Bullseye: gcc/g++ 10.2.1
- ✓ Ubuntu 20.04 LTS: gcc/g++ 9.3.0
- ✓ Apple clang 14.0.0

- Libraries: **crypto, rsa libraries are NOT allowed**

big intger 사용할 수 있는 library

- ✓ stdio, stdlib, math, GMP (GNU Multiple Precision Arithmetic Library), ...
- ✓ iostream, cstdlib, math, GMP, NTL (Library for doing Number Theory), ...

## ■ Execute from command line

- ./rsa -keygen {prime\_p prime\_q, primes.txt, ...}
- ./rsa -{encrypt/decrypt} {public.key, private.key} {plaintext.txt, ciphertext.txt}

- Choose from menu style

```
Enter your choice:    1.Encrypt    2.Decrypt    3.Exit
```

# WHAT YOU SHOULD DO

## ■ keygen

- Validates the given two prime numbers
- Generates a (public, private) key pair (e.g., integer, hexadecimal, base64)
  - ✓ Public:  $e, n$     두개의 소수로부터 key 값이 나와야 한다 : 나오는 형태는 `readme` 에 작성
  - ✓ Private:  $d$

## ■ Encrypt

- Given a plaintext (integer) and the public key
- Encrypts the plaintext and returns a ciphertext

## ■ Decrypt

- Given the ciphertext and the private key
- Decrypts the ciphertext and returns the plaintext

# SCORING CRITERIA (FULL SCORE: 5PTS)

## ■ Implement either

- Supports 1024-bit primes

- ✓ Key generator +2 pts

- ✓ Encryptor/decryptor +2 pts

math 로 int 범위 내에서만 할 경우 +1점씩

- Supports only built-in C type integers

- ✓ Key generator +1 pts

- ✓ Encryptor/decryptor +1 pts

## ■ Exception handling +1 pts

- p or q are not prime
- Invalid argument, type, format, length, etc.

# SUBMISSION GUIDELINES

- Upload your compressed archive file (e.g., .zip, tar, gz) to myETL
  - 파일 이름 양식: 이름\_학번\_HW01.zip
- Include the following items in your submission
  - README
    - ✓ 학번, 이름, 이메일, 전화번호 기입
    - ✓ 개발 환경, 사용한 라이브러리, 컴파일 명령어 등 실행에 필요하다고 생각되는 내용들은 상세히 작성
    - ✓ 본인이 테스트할 때 사용한 소수, plaintext
  - rsa.c or rsa.cpp source code
    - ✓ 작성한 함수들의 기능 및 입출력 정의 작성. 주석 혹은 README
  - Compiled executable
  - (opt.) Plaintext file, Prime numbers file

감사합니다  
Thank you~!