# INFA723 Cryptography and Network Security

# Lab5 Use OpenSSL to Generate Random Number and Test Primality

# 1 Introduction

OpenSSL toolkit provides a command line tool and a crypto library used for a wide range of crypto operations. In this exercise, we are going to learn how to use the command line tool to generate random numbers and test if a number is a prime number. We will also learn how to write Linux shell scripts to use OpenSSL embedded functions. Note that for all the functions implemented in the shell scripts, you can also find a way to implement them using the OpenSSL crypto library.

A lab package has been created for the lab. Go to the class website and download the lab package. Assume your Cygwin installation folder is c:\cygwin. Unzip the package to your Cygwin home folder, e.g., C:\cygwin\home\user name.

# 2 Objectives

* Learn how to use OpenSSL command line tool to generate random numbers
* Learn how to use OpenSSL command line tool to test if a number is a prime number
* Learn and understand how to use OpenSSL functions in Linux shell scripts

If you want to change a shell script or create a new one, be sure that the script is saved as **unix format (LF Only)**. If the file is saved as a dos format (CR and LF), you may have some troubles when you run the script in the Cygwin window.

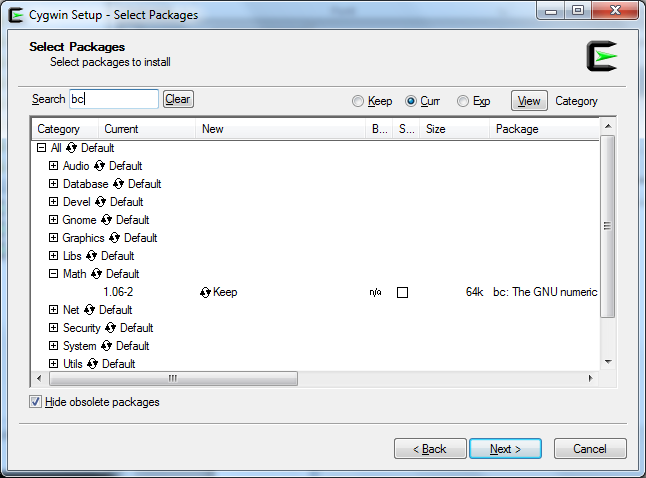
# 3 Setup

Before starting the lab, you need to install an extra package. Follow the steps below to install “bc: The GNU numeric processing language and reverse polish calculator” package.

1. Click Cygwin setup.exe
2. Click “Next” button until you are in the “Select Package” window;

# 

1. Enter “bc” in the Search window;



1. Click Math and enable “bc: The GNU numeric processing language and reverse polish calculator” package
2. Click “Next” button;
3. Proceed to install this package.

# 4 Use OpenSSL to Generate Random Numbers and Test Primality

1. Use Openssl to generate a random number

**$ openssl rand -hex 3**

**a5b8ea**

3 is the number of bytes of the pseudo-random you want to generate. You can replace 3 using another value.

1. Use Openssl to test if a number is prime number

**$ openssl prime 77**

**4D is not prime**

**$ openssl prime 79**

**4F is prime**

The input number is decimal but the output is in hex. In the examples below, we will show how to convert a number from hex to decimal using the bc package we just installed.

1. print prime numbers between 1 and 100

Go to the lab5 folder and run the script prime100 in a Cygwin window. The script will print all the prime numbers between 1 and 100.

**$ ./prime100**

Enter the command “cat prime100” and check how the script is structured.

**$ cat prime100**

1. print prime numbers between two numbers in decimal

Go to the lab5 folder and run the script printPrimeDec in a Cygwin window. The script will print all the prime numbers between two assigned numbers in decimal. The two numbers are defined by two variables, *start* and *end*, in the script.

**$ ./printPrimeDec**

The output number is in decimal.

Enter the command “cat printPrimeDec” and check how the script is structured.

**$ cat printPrimeDec**

1. print prime numbers between two numbers in hex

Go to the lab5 folder and run the script printPrimeHex in a Cygwin window. The script will print all the prime numbers between two assigned numbers in hex. The two numbers are defined by two variables, *start* and *end*, in the script.

**$ ./printPrimeHex**

Enter the command “cat printPrimeHex” and check how the script is structured.

**$ cat printPrimeHex**

1. use OpenSSL to generate a prime numer

Go to th lab5 folder and run the script primeTest. The script will generate random numbers and test if the number is a prime number. The script will stop until a prime number is found

**$ ./primeTest**

The random number generated in the script is set to 3 bytes in default. You can change the number to another value.

Enter the command “cat primeTest” and check how the script is structured.

**$ cat primeTest**

# 4 Question

1. (10 points) Based on the script primeTest, write another shell script using OpenSSL command line. The script is called isprime. It requires an integer as an input parameter. If the input integer is a prime, the script will show “\*\*\* is a prime”. If not, the script will show “\*\*\* is not a prime”. For example,

**$ isprime 8**

**8 is not a prime number**

**$ isprime 17**

**17 is a prime number**

A body of isprime is included in the lab directory. You can start with the sample script.

1. (5 points) Use the program to decide if the following numbers are prime numbers:

a. 2685457421

b. 4294967295

1. (5 points ) If an integer passes the Miller-Rabin primality test, does that guarantee the number is a prime number? Why?