**CS4732/5732 Cryptography Summer 2019**

**Project #2 [75 points]**

---------------------------------------------------------------------------------------------------------

Due date is Thursday, July 25th

**Overview:**

In this project you will be working with a cryptographic library, doing various tasks with the goal of becoming more comfortable with using a cryptographic library.

**Task 1:**

Your first task is to select an environment and a cryptographic library. This library needs support for at least AES, RSA and SHA-256. Once you have selected a library, make a test to make sure you can compile some code and do some rudimentary task with it. Keep in mind that depending on the environment, you might not have one installed natively in your environment. In that case you will need to install that library.

If you are using Java, one library that is easy to use and has decent documentation is at:

<http://tutorials.jenkov.com/java-cryptography/index.html>

Python also has an easy to use library simply called *cryptography*.

If you prefer C++ you will usually need to install your own library. As all students have access to the delmar server, I will suggest if you do not have a better idea to use the OpenSSL library.

https://www.openssl.org/

You will then need to install it from source on delmar,

https://wiki.openssl.org/index.php/Compilation\_and\_Installation

You will then want to test the environment by writing a program that does some basic operation. I have example code to do this on delmar, located at

/accounts/facstaff/hauschildm/Crypto/proj2test

which has a Makefile and some source files for a program that takes in a string from the user and hashes it in C.

Note that depending on how you install it you might need to export the library location.

**Task 2:**

Your second task is to test some PRNGs. Given two randomly chosen integers, x and y, the probability that gcd(x,y) = 1 is 6/𝜋2. I want you to use this idea to estimate 𝜋. The trick is that if you use a “bad” random number generator, this result will not be actually 3.14159265…. For this task you will need a function to calculate the GCD of two integers and a function to calculate square roots (depending on the environment, this might already be available for you). I want you to do this first using a LCG, which should result in a number that is not actually correct. Then use a cryptographic random number generator from your library. Document the results. If you cannot find a bad random number generator, create your own “bad” prng using a LCG. Note that it is possible that the default LCG might pass this test.

**Task 3:**

Start by generating several files of different lengths (one of them very short, shorter than 128 bits), another at least 1Mbyte. Encrypt them using AES, with 128 and 256 bit keys. This key should be generated randomly, using a cryptographically secure random number generator. Determine how many times you can encrypt and decrypt a particular file in a second. Make a calculation to determine what size of file you could decrypt in one second with various settings. Then do the above operation using RSA, generating a public/private key pair. How large a file could you decrypt in a second?

Suppose it were necessary to brute-force the key to break one of your files. How long would it take?

**Submission:**

For turnin I want a text document or pdf explaining in detail all the steps you completed. So initial environment chosen, any source code, results obtained in task 2 and 3, etc.