**CS4732/5732 Cryptography Summer 2021**

**Project #2 [75 points]**

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

Due date is Sunday, July 18th

**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 (which will be used in later projects). 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++ (I suggest staying with python or java if possible) 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:**

Start by generating several files of different lengths (one of them very short, shorter than 128 bits), another at least 1Mbyte. These can be generated by programs or manually created. Then, write a program to 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.

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

**Submission:**

For turnin I want a text document or pdf explaining in detail all the steps you completed. So initial environment chosen, what random number generation for the key was used, any source code, results obtained in task 2, etc.