Lab 4: Programming Symmetric & Asymmetric Crypto

Objectives:

- To program symmetric & asymmetric cryptography and study their operations.

Submission:

- A lab report, a source file and other associated files as required.

Instruction:

In your previous labs you have carried out symmetric and asymmetric encryption and decryption using state-of-the-art tool *openssl*. Even though it is quite useful, it, however, does not reveal the exact mechanisms by which different crypto operations are carried out. In this lab, you will be required to code a single program for different crypto operations in any of your favourite programming languages, such as Python, Java, C++ or so on. In addition, you will measure and study the execution time of these crypto operations.

Please follow the instructions, complete the tasks and prepare a report as per instructed.

Tasks:

Write a single program with the following functionalities:

- AES encryption/decryption with two key lengths, 128 and 256 bits, and two modes ECB and CFB (**5 marks**).
- RSA encryption and decryption (**4 marks**).
- RSA Signature (**4 marks**).
- SHA-256 hashing (**3 marks**).

All these functionalities need to be carried out via command line much like openssl. When your program starts, it will give options for the functionalities to the user and the user can choose one after another functionality.

Regarding the keys, the keys must be generated at the first instance and stored in files. Your program then needs to read the keys from the files and use it in the program.

Regarding the encryption/decryption process:

- The program encrypts and stores the encrypted result in a file
- Use the encrypted file to decrypt its contents and display it on the console

Similarly, for RSA signature:

- The signature is generated for a file. The generated signature is then stored in another file.
- During the verification process, the content file and the signature file are passed for verification.

Finally, the program generates a SHA-256 hash of a file which is displayed in the console.

Your program also needs to have another functionality: to calculate the execution time of each operation. Start a timer in the program before any of these cryptographic functionalities (encryption/decryption of AES or RSA) and end the timer once the functionality ends. Determine the elapsed time for the performed operation as a function of the number of bits

N for key. Try at least **five** different N values starting from 16 bits for AES and RSA and plot graphs. Add it in the report with your observations and explanations. This functionality carries **4 Marks**.

To write this program, you can utilise either C/C++ or Java. There are several online resources with comprehensive tutorials for crypto in different programming languages. For example:

- For Java: http://tutorials.jenkov.com/java-cryptography/index.html

You can use any online resource to write your program, but it must be properly credited in your report.

Submission

- Your program.
- Any other file that might be required to execute the program.
- A lab report detailing how to execute your program and the options it has.
- Your report must contain the link(s) of the website(s) from where you have gathered the code snippet(s) utilised in your program. If you fail to do so, you will be **heavily penalised**.