CSA2203/CSA5221 Assignment

This assignment carries 20% of the final CSA2203/CSA5221 grade.

Report Deliverable Instructions:

The first page in your report must be the title sheet of your assignment clearly showing your name, surname and study unit code. The second page must the table of contents.

Software Artefact Deliverables Instructions

All software deliverables must be delivered in a CD attached to the report, in a properly compiled form and an associated read me file containing set-up instructions. Only binaries executable within either windows or linux/x86 environments are accepted.

Artifacts that are unnecessarily difficult to set up will be penalized. In particular these include un-compiled programs, missing external libraries or the requirement for an IDE or some testing environment to execute the program.

You may utilize any programming language.

Title: Programming Cryptographic Applications.

Context:

You are required to familiarize yourself with a cryptographic library/framework classes, available for your preferred programming language, and then use the offered cryptographic primitives to develop a cryptographic application.

Assignment tasks:

1) **Symmetric encryption.** Write a small program that encrypts and decrypts a test string using AES. Give a step-by-step explanation of the source code.

(10 marks)

2) **Asymmetric encryption.** Write a small program that encrypts and decrypts a test string using RSA. Give a step-by-step explanation of the source code.

(10 marks)

3) **Cryptographic hash functions.** Write a small program that produces a SHA-2 digest. Give a step-by-step explanation of the source code.

(10 marks)

4) **Application development.** Develop a 'secure file storage application' that offers file/encryption services both for securely storing personal files, as well as for encrypting files intended for sharing. Securing personal files should be password-driven, whilst securing shared files should be public/private key pair driven. Document the main source code fragments.

(50 marks)

5) **Application testing.** Test the application, demonstrating the main application's functionality with well-explained screen-shots included as part of the report deliverable.

(20 marks)

Notes:

- You may use any AES key size. Use the default encryption mode and padding scheme.
- You may any SHA-2 output size.
- Public/private keys are not required to be stored in certificates.
- Framework run-times are not required to be supplied as part of the deliverable. A 'readme' file clearly indicating the runtime required for executing the application suffices.