Infinite Possibilities

MSc. Applied Cyber Security **Secure Communications & Cryptography**

Analysis of Next Generation Cryptography

Requirements:

You are requested to investigate, build a prototype, and evaluate one of the following:

a) **Zero-knowledge proof (ZKP) methods.** This involves using cryptography to allow users to prove shared knowledge without having to reveal the original information. This might involve passing knowledge of a password, without revealing the password.

b) **Homomorphic encryption.** This involves processing encrypted data and can be partial homomorphic encryption (where a few mathematical methods can be implemented) or full homomorphic encryption.

c) **Light-weight cryptography.** This involves cryptography methods that supports encryption which are suitable for IoT devices. This is normally optimised in terms of reducing processing requirements, memory utilization and energy drain.

d) **Quantum-robust cryptography.** Quantum computers are likely to crack existing public key methods, and thus we need to move toward public key methods which are quantum robust.

e) **Key Distribution Centres**. The complexity of handling keys is an increasing challenge for many companies. Within a KDC, we can implement methods which will generate encryption keys for users. This topic will involve investigating possible methods that could be used.

f) **Blockchain integration.** Many applications are moving towards storing data on a blockchain and in using smart contracts. This area will investigate new areas of improving the security and/or performance of blockchain related methods.

There are many different methods involved in each of these areas, so while your literature review might have a relatively wide scope, you might want to focus in on one or more methods for your implementation and evaluation. You may also pick another related cryptographic area, but this would have to be approved, so check with me first.

Tasks:

You should pick one of the areas defined, and then:

- **Perform a literature review** around the required application area. This will involve using sources such as Google Scholar and outlining some of the key research in the area.

- **Produce a basic prototype** of your application area. This might involve gathering code from GitHub sources, and/or creating your own code. Make sure you keep the references to any code you have used.

- **Evaluate the performance/security** of the system. This will involve both the validation of your implemented code – such as using test vectors – and also the evaluation of the method(s) using one or more experiments. If you are building you own code, this will be more of a validation test.

A possible structure for your report:

The coursework should have up to 15 pages, and will include the sections of:

- **Introduction.** This will involve an outline of the area you are addressing and in defining the key aim of the coursework.
- **Literature Review.** This will involve an outline literature review of your selected area, with integrated references.
- **Implementation.** This will implement one or more of the methods you have outlined in the literature review, and which demonstrate the operation of the area you have selected. The report will perhaps include a few snippets of key elements of code, with an outline of their operation.
- **Evaluation.** This will outline the validation and evaluation methods you have implemented with some results of the key aspects of the methods (or of your application).
- **Conclusions.** This will include your key findings, and pointers to future work.

The marking schedule is:

- Quality of the research undertaken. 20%.
- Quality and clarity of the implementation. 25%.
- Level of innovation applied. 20%.
- Quality of the evaluation or application. 20%.
- Overall presentation and use of references. 15%.

Submission of coursework

Your coursework document should be created either as a PDF document or as a Word document. The preferred method of submitting the code you have used, and associated documentation, is to use a private GitHub. This should then be shared with mark.a.cummins@tudublin.ie. Do not make your repository public until after the submission of your coursework.