ICT2213—Applied Cryptography

AY2024-2025 Trimester 2

Team Project (Weightage: 25%)

1 Important dates

- **Tuesday, Mar 11, 11:59pm:** Submit your report+code via LMS Dropbox. Only one member of the team is required to submit for the whole team. Please write the names of all the team members clearly on the front page of your report. Marks will be deducted for submissions without names.
- **Thursday, Mar 13 (during lab):** Demo.

2 Scenario

Your team has been hired to implement a novel social networking application, where friends can check whether they are physically close to each other without disclosing their exact locations. For simplicity, assume that users may only roam within a two-dimensional space with (integer) coordinates ranging from 0 to 99999 in both the x and y coordinates. You should implement the following two applications:

1. **Server**: Allows users to login/register and maintains information about user friendships. Besides that, it merely acts as a proxy when users communicate with each other (forwards messages between users).
2. **Client**: Clients will register with the server and establish friend relationships with other users. It should also implement the following two functions: (i) location update (i.e., latest x and y coordinates of the user's location); and (ii) proximity request, i.e., determine whether a specific friend is nearby based on some proximity metric. Note that the latter will require the two users to exchange multiple messages, using the server as a proxy.

The *minimum* requirement is for your team to implement the following proximity metric. Divide the space into a 100×100 grid (i.e., each cell will have a size of 1000×1000 distance units) and declare two users as close if they are located within the same cell.

Besides the above metric (which is not very accurate), you are encouraged to explore other ones, such as Euclidean distance. For example, two users are close if their distance is within d units.

To satisfy the *privacy* requirements of the application, your protocols should **not** disclose the cell where a user is located or the exact distance between two users (for example, the exact value of the Euclidean distance).

In addition to the proximity protocol, your application should address other important issues as well, such as user authentication, key management and distribution, message integrity, etc. Your team should only focus on developing an application that is secure. There will not be any bonus points for GUI, so keep everything simple under the command line interface.

3 Learning outcomes

Upon completion of this project, you should be able to:

- Investigate novel ways in which cryptography can be used in the real world.
- Correctly employ basic cryptographic primitives to solve a particular real world problem.
- Implement complex cryptographic protocols in software.

4 Submission and Grading

A report covering:

- A theoretical explanation of how cryptography is used to achieve your project's goals, with detailed elaboration on your chosen cryptographic algorithms, cryptographic protocols, key management and distribution.
- Relevant screen shots of the protocol's output, including Wireshark packets.
- Timing information regarding the CPU overhead of the proximity protocol.
- Instructions on how to compile and run your code.

A demo covering:

- A demonstration of your application.

A text file containing:

- The source code of your application.

Grading scheme:

- Correctness of the proposed protocols.
- Quality of the report (presentation, clarity, language, etc.).
- Theoretical understanding of the underlying cryptographic concepts.
- Effectiveness of the demo.
- Design logic, programming style, and readability of your code.