Queen's University
School of Computing

CISC 468/878: Cryptography
Course Project Description
Winter 2025

This handout outlines the separate requirements for the CISC 468 and CISC 878 course projects.

CISC 468: P2P Secure File Sharing Application

CISC 468 projects will design and implement a peer-to-peer secure file sharing application. Each project should be completed in groups of two and should consist of two clients written in two different programming languages, such as Python, Java, Rust, or Go. The clients should be able to communicate with each other despite being implemented with different cryptographic APIs in two different languages. The application must:

1. Support peer discovery on a local network. A simple protocol that can be used for peer discovery on a local network is mDNS. On the Internet, a more sophisticated protocol such as BitTorrent would be required.
2. Support mutual authentication of contacts. In other words, after key verification has been performed, each individual user should be assured of the identity of the user who they are communicating with.
3. Peers should be able to request files from each other, or send a file to another peer; the peer receiving a request or receiving a file should consent before the request is processed.
4. Peers should be able to request a list of files available to be shared by each other (consent is not required).
5. If a peer A is offline, but peer B already had peer A's list of available files, peer B may find another peer C that had previously downloaded the file from peer A, and request the file from them instead. But peer B must be able to verify that the file has not been tampered with (i.e., that it is the same file that peer A was offering).
6. Allow users to migrate to a new key if their old one is compromised. Existing contacts should be notified, in this case, and any necessary steps should be taken to re-establish authenticated and secure communication.
7. Guarantee the confidentiality and integrity of any files that are sent between users.
8. Support perfect forward secrecy. In other words, compromise of a long-term secret should not allow an attacker to decrypt all past communication.
9. Securely store files on the local client device, so that an attacker who steals the device should not be able to read them.
10. Display an appropriate message to the user if any error occurs (e.g., if a file cannot be delivered) or a security check fails (e.g., if a file has been tampered with in transit). **Your repository must contain test cases to check for such relevant scenarios.**

CISC 878: Advanced Cryptographic Techniques

CISC 878 projects will select a zero-knowledge proof or secure multi-party computation scheme. The project should present a problem of the student's choice and present how the chosen scheme can be applied to the problem to achieve some interesting security objectives. The project should include an implementation using a suitable library. Ideally, it should compare two different algorithms; different algorithms can be compared to observe performance differences or other trade-offs.

Here are some links that you might find useful:

- Secure Multiparty Computation (MPC) (introduction by Y. Lindell: must read before diving deeper)
- Private-ID protocol (also see the links under "Additional Resources on Private Computation at Meta" at the end of the repository's `README.md` file)
- Private Join and Compute
- Verifiable Distributed Aggregation Functions (VDAF) IETF draft
- Rust implementation of Prio3 (a VDAF scheme), which was used in a COVID-19 exposure notifications private analytics system
- "Awesome MPC" compilation (see list of frameworks)
- "Awesome Zero Knowledge Proofs (ZKP)" compilation
  - Among other things, this compilation links to some interesting games that use ZK proofs. For example, Zero-Knowledge Wordle uses ZK proofs to prove that a player knows the words without revealing the words to a verifier (see the repository's `README.md` for a detailed explanation).

Deliverables

The project is worth 25% of your final grade, and consists of the following three deliverables.

Proposals

The proposal is worth 5% of the project grade and is due Feb. 25, 2025. CISC 468 proposals should indicate the team members and the two selected programming languages that will be used to write the two clients. CISC 878 proposals should indicate the advanced cryptographic technique that will be studied, the problem that it will be applied to, and the library that is planned to be used for implementation. **Try to be realistic: I am not expecting a massive project.**

Presentations

Presentations are worth (10%) of the project grade. The exact presentation format and dates are TBA but will be in the last 1-2 weeks of class.

Report and Implementation

The final report and implementation, including a link to a Github repository containing all associated code, is worth 85% of the project grade.

The report should be prepared in LaTeX using this ACM template and should be **three pages in length, excluding tables/figures and references**. The template provided is the standard ACM SIG Proceedings double-column template with (i) keywords and CCS concepts removed, and (ii) the copyright box on the first page removed (with the `\documentclass[sigconf,nonacm]{acmart}` command). Please do not adjust margins or font sizes.

Please pay attention to grammar and clarity, academic integrity (do not plagiarize or copy/present others' work or opinions as if they are your own), style (your report layout, references, appearance, etc. should all look professional), and conciseness (say everything that needs to be said, but avoid waffling). Marks may be deducted up to 20% for failure to meet these expectations. You may refer to this document for tips on how to write clearly, concisely, and professionally (it is targeted to graduate theses, but also contains some generally applicable writing advice).

CISC 468 project reports should explain how the application was designed to meet all of the required objectives. It should also discuss all libraries, algorithms, and security parameters (e.g., key lengths) used. For example, explain how you guarantee message integrity. Also explain the protocol and message format you designed for your clients to communicate with each other.

CISC 878 project reports should explain the cryptographic technique chosen and the problem that it was applied to. Sufficient background information should be provided on the problem and on how the chosen technique works, to ensure that a reader without prior background (aside from having taken this course) can understand. The complete methodology and results should also be described.

All project reports should specify (and cite, if applicable) all resources and software used in the project and should discuss limitations of the project.

(Document version: Feb. 11, 2025)