Final Project

Theme Song: Redo

In the final project, you'll use what you learned throughout the semester to implement a final project of your choice! We've provided some ideas below and left the specifications open for you to decide and discover.

Contents

1 Project Specification

In this project, you will get to implement one of two project types: an extension or a new project. An extension project takes a prior project and implements one or more extensions or optimizations on top of it. A new project implements a protocol that we discussed in class but did not get to implement in our prior projects. Both types will culminate in a codebase, report, and presentation.

You may partner up with **one** other student for this project!

1.1 Grading

- Project proposal: 20%
- Final report: 20%
- Recorded presentation: 20%
- Codebase (a working demo): 40%

1.2 Timeline

Project proposals are due on **Wednesday, April 22nd.** We will review them promptly and assign each project a mentor TA. Final reports and presentations are then due on **Friday, May 8th.**

Late submissions will NOT be accepted.

1.3 Github, Gradescope, and Working in Pairs

The final project is intended to be done in pairs. GitHub has been set up so that you can make your own repository (see 5), and Gradescope has been set up so that you can add up to one other classmate to a submission. Please use both of these features! The Gradescope for final submissions will become available after project proposals have been graded. The Gradescope for project proposals is live.

If you prefer to work by yourself, you may do so as well.

Important: all repositories should be private. Github will auto-configure this for you, but it is important you do not change settings to make it public. If you have questions or issues with access, please reach out to the TA staff for assistance.

1.4 Project Proposal

To match you with an appropriate mentor TA and to ensure that the scope of your final project is suitable, please write a **one-page** project proposal by **Wednesday, April 22nd. Late submissions will NOT be accepted.** (If you submit late, you will lose the points for the proposal, be assigned a TA at random and without proposal feedback.)

Your proposal should include an outline of the project you're interested in implementing, stretch goals that your group might deem difficult but worthwhile, a list of any libraries that you'd like to use, and any open questions about your project that you'd like a TA to help you answer.

If you're not sure about all of the details of your project, that's fine! Let us know what you'd like to learn about so we can pair you with a suitable TA. Your TA will be a resource for you to learn more about your project domain and get coding help when appropriate.

1.5 Final Report

In addition to your project's codebase, please write a final report of no more than **5 pages** long (including the bibliography), detailing the findings of your final project. **Late submissions will NOT be accepted.**

Your final report should include any prerequisite knowledge necessary to understand the project, an overview of the project itself, difficulties your group faced in implementation and design, and any experiments or benchmarks you may have conducted. Note that this report does not have to be incredibly formal. However, it should explain the project in detail.

1.6 Recorded Presentation

Finally, in addition to a final report, please prepare a recorded presentation of at most **5 minutes** long. Your presentation should include an overview of your project and a

working demo. You may also choose to walk through your codebase and any notable design decisions. You may have some overlap between your presentation and report; this is perfectly fine and expected.

1.7 PDF Format

The project proposal and final report must be PDF files in double-column ACM format: see ACM Proceedings Template, using the **sigconf** style.

2 Extension Project Ideas

2.1 Post-Quantum Secure Communication

Extend the Signal project to use post-quantum cryptography using the SEAL library (or other equivalent).

2.2 Double Ratchet

Extend the Signal project to support double ratchet by adding the symmetric-key ratchet. Use the symmetric-key ratchet to handle lost or out-of-order messages in the communication.

2.3 Group Chat

Implement a messaging protocol that supports group chats, ideally hiding the group structure from the server.

2.4 Voting for Multiple Candidates

Extend the Vote project to support voting for multiple candidates. You may choose to enforce each voter to vote for exactly k candidates or at most k candidates, or both.

2.5 Yao's Optimizations

Implement point-and-permute, free XOR, and row reduction (GRR3) in the Yao's project. Read more about all of these optimizations here.

2.6 PIR Optimizations

Implement the ciphertext compression in the PIR project. Read more about this optimization here.

3 New Project Ideas

3.1 Secure Shell

Implement a secure shell (SSH) protocol between a client and a server. You may find the RFC useful.

3.2 Zero-Knowledge Proofs for Graph 3-Coloring

Implement a zero-knowledge proof (ZKP) for graph 3-coloring, which implies ZKPs for all NP languages.

3.3 Succinct Commitment via Merkle Tree

Implement a succinct commitment scheme via Merkle Tree, leveraging a (hash-based or Pedersen) commitment scheme and a hash function.

3.4 Multi-Party Computation

Implement the GMW protocol for semi-honest MPC among any number of parties to compute any function.

3.5 Private Set Intersection

Implement the DDH-based private set intersection (PSI) and PSI-CA.

3.6 PIR from Additive Homomorphism

Learn about and implement an efficient PIR protocol from additively homomorphic encryption (e.g., Regev encryption). Read more here.

4 Proposing your own project

We strongly encourage you to look outside of what we have brainstormed and seek out a project idea of your own! We are currently living in a cryptographic renaissance, and there are countless ideas that would be interesting to explore. Check out some project ideas from a similar course here or here (note that not all of these project ideas are suitable).

If you choose to go down this route, be sure to have scoped out the project so that you have a good idea of the amount of work that this will be. Too much or too little and it may not be a suitable project.

5 Getting Started

To get started, begin from this template and clone it into the `devenv/home` folder. You can either create a new team when joining the assignment, or join an existing one that your project partner has created. From here you can access the code from both your computer and from the Docker container. The initial repository is empty to allow you to build what you'd like - good luck!