CS 507, Topics in Cryptography: Secure Computation
Homework 4

Due: December 10, 2025

Problem 1 (GMW vs GC)**.** Consider the following standard semi-honest 2PC functionality:

- A and B agree on a Boolean circuit C.
- A sends its input $x \in \{0,1\}^n$ to the functionality, and B sends its input $y \in \{0,1\}^n$.
- The functionality computes $C(x,y)$ and sends the output to both A and B.

We have seen two main ways to achieve this functionality: the GMW protocol and a GC-based protocol.

1. Provide short discussion: why might one choose to use GMW over GC, and vice versa?
2. Suppose the parties wish to compute a function that can be cleanly expressed as the *composition* of two circuits. Namely, they wish to compute $C_1(C_0(x,y))$. They plan to use GC to compute C_0 and GMW to compute C_1. Assume you have two realizations of the above functionality (one based on GMW and one based on GC), and design a semi-honest secure 2PC protocol that realizes the following functionality:
   - A and B agree on two Boolean circuits C_0, C_1.
   - A sends its input $x \in \{0,1\}^n$ to the functionality, and B sends its input $y \in \{0,1\}^n$.
   - The functionality computes $C_1(C_0(x,y))$ and sends the output to both A and B.
3. Construct simulators that demonstrate your protocol is secure.
4. Provide short discussion: why might parties wish to use this protocol that mixes GMW and GC in this way?

Answer 1.

Problem 2 (PIR vs ORAM)**.** We have discussed two very different approaches to securely accessing a memory:

- **Oblivious RAM** (ORAM) allows a client to outsource its read-write memory to an untrusted server. One way to formalize ORAM is via the following very simple functionality:
  - The client and server agree on a memory size n.

- The client provides as input to the functionality a RAM program P and its input x. That is, P is a program that reads/writes to a memory of size n.
- The functionality sends $P(x)$ to the client, and it sends m to the server, where m is the number of memory reads/writes performed while running $P(x)$.

- **Private information retrieval** (PIR) allows a client to privately fetch an item from a public database held by one or more servers. Let's assume a single-server variant of PIR. One way to formalize PIR is via the following very simple functionality[1]:
  - The server inputs a database D.
  - The client inputs an index i
  - The functionality sends D_i to the client and $\perp$ to the server.

1. Let us consider a *read-only* RAM program, where the RAM program P has an initial memory state D, but it only reads to memory, and never writes to it. Assuming PIR and a CPA-secure (or CPA$-secure) encryption scheme, construct a semi-honest protocol that achieves the above ORAM functionality for such programs. *Note: Your client should run in low space, offloading storage of D to the server. As a technical point, the low-space client cannot afford to store its own input database D. Therefore, you may assume that the client can "by magic" read each of its database entries D_i exactly once.*
2. Prove your protocol is secure by constructing simulators.
3. Briefly discuss the advantages/disadvantages of your protocol, in terms of efficiency, as compared to ORAMs we saw in class. When might you protocol make sense/not make sense?

Answer 2.

Problem 3. Please complete the following feedback form. Since this form is not anonymous, feel free to mark "I choose not to respond". To emphasize: your grade on this question will not depend on your answers! I encourage you to answer honestly and give genuine criticism![2]

Note: Yes, this counts as a full "problem".

For each of the following statements, please indicate one of the following: **I strongly disagree/I disagree/I am undecided/I agree/I strongly agree/I choose not to respond.**

1. The pace of the course was too fast.
2. The homework increased my understanding of course concepts.
3. The course should have more assignments.
4. The course assignments should be more difficult.
5. The course was interesting.

[1] In fact, this functionality implies is a slightly stronger version of PIR, sometimes called *symmetric* PIR, where the client learns *only* D_i, and nothing more. In other words, this ensures a notion of privacy for the server. In basic PIR, this is not a requirement.

[2] Also, please consider filling out the FLEX form, once it is available: `go.illinois.edu/flex`.

6. Office hours were useful.
7. I understand the definition of semi-honest security.
8. I understand the definition of malicious security.
9. I can at a high level explain the GMW protocol.
10. I can at a high level explain Garbled Circuits.
11. I can at a high level explain the concept of secret sharing/authenticated secret sharing.

Please take the time to write a brief response to the following. Feel free to answer "I choose not to respond".

1. What was been the best part of the course?
2. What was been the least useful part of the course?
3. Is there a topic you wish I covered more/at all?
4. Do you have any other suggestions for the course?

Answer 3.