CS 598DH Secure Computation – Homework 3

Professor David Heath

Due: April 25

Problem 1 (GMW vs GC)**.** In class, we considered two basic approaches to secure computation, the secret-sharing-based GMW protocol, and the encryption-based Garbled Circuit technique. These techniques are incomparable in terms of their properties, trading off in the number of supported parties, the amount of bandwidth consumed, the amount of latency consumed, and the natural security they provide.

1. Explain the advantages of GMW as compared to GC.
2. Explain the advantages of GC as compared to GMW.

Answer 1.

Problem 2 (Multiparty AND)**.** In class, we in detail explored the GMW protocol for two parties, and we also briefly discussed how GMW naturally generalizes to more than two parties. Let's understand this generalization in more detail. We will not consider the full GMW protocol, but rather a special case that involves multiplying bits. Consider the following functionality:

PARAMETERS:

1. Let $P_0, ..., P_{n-1}$ be n parties.

FUNCTIONALITY:

1. Each party P_i has input $x_i \in \{0, 1\}$.
2. Each P_i outputs the AND of all bits: $\bigwedge_{i \in [n]} x_i$

1. Construct a protocol Π that implements the above functionality in the following setting:
   - You have black-box access to the 1-out-of-2 OT functionality (i.e., construct Π in the OT hybrid model).
   - Π should be secure against a semi-honest adversary.
   - Π should tolerate up to $n - 1$ semi-honest corruptions.

   Clarification. Please do not write "use GMW". I would like you to explain how the parties sample randomness/call OT to achieve the above functionality.

   Hint. Recall how we multiplied pairs of bits in the GMW protocol. Remember that it is *not* secure for the adversary to learn the value of any intermediate wire.

2. **Bonus.** If you implement Π naïvely, how many calls to the OT functionality are required? Achieve a protocol Π that requires only $O(n^2)$ calls to the OT functionality.

   Hint. If it is helpful, you may assume that n is a power of two (i.e., $n = 2^k$ for some k).

3. Prove security by writing out simulators for all possible subsets of corrupted parties.

   Hint. Because we parameterize over n, it is not possible to write out individual simulators, since you cannot even enumerate all possible corruptions. Instead, your simulator(s) will need to be general, explaining how to simulate some parameterized number of corruptions.

Answer 2.

Problem 3 (Power and Limitations of Circuits)**.** In this class, we focused our discussion on Boolean circuits, and we constructed several powerful techniques for securely implementing Boolean circuits. Circuits form a complete model of computation, and some computations with circuits are efficient, but some are not. In this question, you will construct Boolean circuits that solve relatively interesting problems.

Clarification. I am not looking for a gate-by-gate circuit descriptions, but rather high level descriptions of the circuits.

1. Consider the following simple functionality that accesses an element from a list:

   PARAMETERS:

   (a) Let P_0, P_1 be two parties.

   (b) Let n denote a list size, and let $n = 2^k$ for some k. I.e., $k = \log_2 n$.

   FUNCTIONALITY:

   (a) P_0 inputs a list of n bits $X \in \{0,1\}^n$.

   (b) P_1 inputs an index $i \in \{0,1\}^k$.

   (c) Party P_1 outputs $X[i]$, where i is interpreted as an integer index.

   (d) Party P_0 outputs $\perp$.

   - **Base protocol.** First, suppose that we do not care about security (i.e., suppose each party is honest). Describe a simple protocol that implements the above functionality in $O(k)$ time.
   - Now, suppose we care about protecting (1) P_0's unaccessed bits, and (2) P_1's index. I.e., we wish to implement the functionality with a semi-honest protocol. We decide to implement the functionality via GMW or GC, and so we must prepare a Boolean circuit that implements the functionality. Design a Boolean circuit with $O(n)$ gates that implements array lookup; i.e., on input (X, i) the circuit outputs $X[i]$.
   - Is it possible to build a circuit that (1) correctly solves the same problem and (2) has size (i.e., number of gates) that scales only sublinearly with n (i.e., has $o(n)$ gates)? Why/why not?

2. Consider the following set equality functionality:

> PARAMETERS:
>
> (a) Let P_0, P_1 be two parties.
>
> (b) Let w denote the bit-length (width) of set elements.
>
> (c) Let n denote a set size.
>
> FUNCTIONALITY:
>
> (a) Each party P_i holds a size-n set X_i of elements:
>
> $$X_i \subseteq \{0, ..., 2^w - 1\} \qquad \text{such that } |X_i| = n$$
>
> (b) Each P_i outputs 1 if the sets are equal and 0 otherwise: They output $X_0 \stackrel{?}{=} X_1$

- **Base protocol.** First, suppose we do not care about security (i.e., suppose each party is honest). Describe a simple protocol that implements the above functionality. Asymptotically, how long does your protocol take to run?
- Now, we wish to implement the above functionality using GMW/GC, and so we must prepare a Boolean circuit that implements our functionality. Design a Boolean circuit that implements set equality; i.e., on input two sorted lists X_0, X_1, the the circuit outputs a single bit indicating if the sets are equal. Ensure that your circuit has size $O(w \cdot n)$.
  Hint. There exists a circuit of size $O(w)$ that checks if two length-w bitstrings are equal.

3. Consider the functionality which computes the *size* of the intersection of two sets:

> PARAMETERS:
>
> (a) Let P_0, P_1 be two parties.
>
> (b) Let w denote the bit-length (width) of set elements.
>
> (c) Let n denote a set size.
>
> FUNCTIONALITY:
>
> (a) Each party P_i holds a size-n set X_i of elements:
>
> $$X_i \subseteq \{0, ..., 2^w - 1\} \qquad \text{such that } |X_i| = n$$
>
> (b) Each P_i outputs the size of the intersection of the two sets: $|X_0 \cap X_1|$

- **Base protocol.** First, suppose we do not care about security (i.e., suppose each party is honest). Describe a simple protocol that implements the above functionality. Asymptotically, how long does your protocol take to run?

- Describe a circuit that takes as input the two sets (represented as sorted lists) and computes the size of the intersection. Your circuit should have size at most $O(w \cdot n \cdot \log^2 n)$.

  Hint 1. There exists a circuit of size $O(w)$ for comparing two w-bit elements.

  Hint 2. There exists a circuit of size $O(\log n)$ for adding two $(\log_2 n)$-bit numbers.

  Hint 3. We looked at a particularly helpful primitive for solving this problem when we first discussed Oblivious RAM.

Answer 3.

Problem 4. Please complete the following feedback form. Since this form is not anonymous, feel free to mark "I choose not to respond". I cannot emphasize strongly enough: your grade on this question will not depend on your answers! I encourage you to answer honestly and give genuine criticism![1]

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.
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.

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?

[1] Also, please consider filling out the ICES form, once it is available: `go.illinois.edu/ices-online`.