CS 507, Topics in Cryptography: Secure Computation
Homework 2

Due: October 22, 2025

Problem 1. It is often useful to build a secure computation from another already-designed protocol. In class, we have discussed 1-out-of-2 OT, where the receiver selects one of two secrets; it is natural to consider generalizations of OT, e.g. 1-out-of-N OT, where the receiver selects one of N secrets. Many variants of OT are closely related. In the following, you will construct protocols in a *hybrid world* where parties are allowed to interact with the ideal functionality for some problem.

1. **Warm-up:** Suppose you have a semi-honest secure 1-out-of-4 OT functionality. Construct a semi-honest 1-out-of-2 OT protocol.

2. Suppose you have a semi-honest secure 1-out-of-4 OT functionality. Construct a semi-honest protocol that simultaneously executes two 1-out-of-2 OTs. Your protocol *may not* make more than one call to the 1-out-of-4 functionality.

3. Suppose you have a semi-honest secure 1-out-of-2 OT functionality. Construct a semi-honest 1-out-of-4 OT protocol. Your protocol *may* make more than one call to the 1-out-of-2 functionality.

4. Suppose you have access to a semi-honest secure 1-out-of-2 *random* OT functionality. Namely, the functionality delivers to the sender to uniformly random messages m_0, m_1, and it delivers to the receiver a uniform bit b and m_b. Construct a semi-honest 1-out-of-2 *chosen input* OT, where both parties select their inputs.

Prove your protocols are secure in the **semi-honest model** by constructing simulators and arguing indistinguishability.

Answer 1.

Problem 2. Consider an arbitrary two-party protocol Π, and suppose that Π is secure in the malicious model. One might think that Π is *also* secure in the *semi-honest* model. Perhaps surprisingly, this is not necessarily the case.

Consider the following one-sided AND functionality:

PARAMETERS:

1. Let P_0, P_1 be two parties.
2. Each party P_i has input $x_i \in \{0, 1\}$.

FUNCTIONALITY:

1. P_0 outputs $\perp$.
2. P_1 outputs $x_0 \wedge x_1$.

I.e., the parties compute AND, but only P_1 receives output. Consider the following protocol Π_{AND} for the above functionality:

PROTOCOL:

1. P_0 sends x_0 to P_1 and outputs $\perp$.
2. P_1 outputs $x_0 \wedge x_1$.

Π_{AND} is not secure in the semi-honest model, but it *is secure* in the malicious model.

1. Give a brief and informal argument that explains why this protocol is secure in the presence of a malicious adversary, but not a semi-honest adversary.
2. Prove that Π_{AND} is not secure in the semi-honest model.
3. Prove that Π_{AND} is secure in the malicious model by constructing simulators for P_0 and P_1.
4. **Bonus.** Suppose we are willing to adjust our definition of semi-honest security to ensure that malicious security *does* imply semi-honest security. How would you adjust the definition? Informally argue (1) that your change still captures the notion of an adversary that is honest but curious and (2) that malicious security implies security under your adjusted definition.

Answer 2.

Problem 3. In this problem, suppose we have access to a maliciously secure protocol Π for 1-out-of-2 OT.

1. Suppose we would like to use Π to design a maliciously secure protocol for computing arbitrary functions. To do so, we take the semi-honest GMW protocol as specified in class, and we substitute semi-honest OT by malicious OT. Is this modified GMW protocol maliciously secure? If so, argue why. If not, briefly describe an attack by a malicious adversary.
2. Recall the coin flip functionality, where the ideal functionality flips a coin x, delivers x to the adversary, and then delivers x to the honest party, iff the adversary does not abort. Recall that in class we showed a maliciously-secure protocol for this functionality, based on a commitment scheme. Construct a maliciously secure protocol for the coin flip functionality that *does not* use commitments, but you may invoke malicious OT once. Argue informally that your protocol is secure. There is no need to construct formal simulators. *Hint: a main challenge here is in delivering output to both parties. Remember that in the malicious model, It is okay if the malicious adversary can launch an "attack" that succeeds only with negligible probability.*