# CS 507, Topics in Cryptography: Secure Computation
# Homework 1

Due: September 24, 2025

**Problem 1.** Suppose Alice has an input $x \in \{0, 2, 4, ..., 8\}$ and Bob has an input $y \in \{1, 3, 5, ..., 9\}$. Here is a protocol that computes the function $\max(x, y)$:

- If Bob has input $y = 9$, he announces "yes" and both parties output 9 and halt. Otherwise he announces "no" and the protocol continues.
- If Alice has input $x = 8$, she announces "yes" and both parties output 8 and halt. Otherwise, she announces "no" and the protocol continues.
- If Bob has input $y = 7$, he announces "yes" and both parties output 7 and halt. Otherwise he announces "no" and the protocol continues.
- If Alice has input $x = 6$, she announces "yes" and both parties output 6 and halt. Otherwise, she announces "no" and the protocol continues.
- ...
- The protocol continues until some party says "yes", at which point the output is determined and the protocol is finished.

Construct simulators that demonstrate this protocol is secure in the presence of a semi-honest adversary. Prove that each simulation perfectly captures the view of the adversary.

**Answer 1.**

**Problem 2.** Consider Problem 1, modified as follows: Alice has an input $x \in \{0, 1, ..., 9\}$ and Bob has an input $y \in \{0, 1, ..., 9\}$. The protocol is amended as follows:

- If Bob has input $y = 9$, he announces "yes" and both parties output 9 and halt. Otherwise he announces "no" and the protocol continues.
- If Alice has input $x = 9$, she announces "yes" and both parties output 9 and halt. Otherwise, she announces "no" and the protocol continues.
- If Bob has input $y = 8$, he announces "yes" and both parties output 8 and halt. Otherwise he announces "no" and the protocol continues.
- ...

- The protocol continues until some party says “yes”, at which point the output is determined and the protocol is finished.

Is the modified protocol still secure in the semi-honest model? If so, prove security; if not, explain where simulation fails.

**Answer 2.**

**Problem 3.** We often assume that in 2PC, each party outputs the same value. Let’s instead consider the case where the ideal functionality delivers separate outputs to each party.

Suppose there are two functions $f_A$ and $f_B$ and that in the ideal world, the functionality receives $x$ from Alice and $y$ from Bob, then delivers only $f_A(x, y)$ to Alice and only $f_B(x, y)$ to Bob.

1. Give an example $f_A$ and $f_B$ where it is demonstrably insecure (i.e., less secure than the ideal world described above) if in the real world *both* parties learn $f_A(x, y)$ and $f_B(x, y)$. *Hint: there are extremely simple choices of $f_A$ and $f_B$ that meet the criteria.*

2. Suppose we have access to a semi-honest secure protocol that computes any function $f(x, y)$ and delivers this output to both Alice and Bob.

   (a) Formalize a new protocol that uses the above protocol as a black-box. This new protocol should deliver $f_A(x, y)$ to Alice and $f_B(x, y)$ to Bob.

   (b) Prove your new protocol secure in the semi-honest model by constructing simulators.

   You may assume that $x$, $y$, $f_A(x, y)$, and $f_B(x, y)$ are each $n$-bit strings.

**Answer 3.**

**Problem 4.** The strongest definition of semi-honest security requires us to simulate *each possible subset* of corrupted parties.

Consider the 3-party setting. One might naively assume that it suffices to simulate each *size-two* subset of corrupted parties, without needing to simulate subsets of size one. *This is wrong.*

1. Construct a three-party functionality and a corresponding protocol such that the protocol (1) is secure against an adversary who corrupts exactly two parties, but (2) is insecure against an adversary who corrupts exactly one party.

   *Hint: There exist very simple protocols that meet these requirements.*

2. Construct simulators that prove that your protocol securely achieves your functionality when a semi-honest adversary corrupts an arbitrary size-two subset of the parties.

3. Prove that your protocol is insecure under the full definition of semi-honest security (by demonstrating that some single party cannot be simulated).

**Answer 4.**