Homework 2

Advanced Tools From Modern Cryptography
CS 758 : Spring 2018

Released: September 27 Wednesday
Due: October 15 Sunday

Secure Multiparty Computation [Total 100 pts]

1. **Semi-Honest to Semi-Malicious Security.** [20 pts]

   In the *semi-malicious* corruption model, the corrupt parties follow the protocol honestly, *except in the choice of randomness*, which may be arbitrary. Note that it has a stronger adversary than the semi-honest (or passive) corruption model.

   (a) Argue that in general a semi-honest secure protocol need not be semi-malicious. Specifically, assume that you are given a semi-honest secure protocol for (say) OT. Convert it into a protocol that remains semi-honest secure, but is *not semi-malicious secure*.

   (b) Given a 2-party semi-honest secure protocol Π, show how it can be transformed into a semi-malicious secure protocol Π^* for the same functionality, using an extra round of interaction. You may assume that the parties are given access to an ideal commitment functionality.

   Can you prove the semi-malicious security of Π^* by showing how to transform a simulator for Π (against semi-honest adversaries) into a simulator for Π^*? [Extra Credit]

2. **OT, OLE and Correlated Random Variables.** [40 pts]

   Define Oblivious Transfer (OT) functionality over a field $\mathbb{F}$ (or, over a ring) as an SFE in which Alice inputs $(x_0, x_1) \in \mathbb{F}^2$ and Bob inputs $b \in \{0, 1\}$; then Alice gets $\perp$ as output, but Bob gets x_b.

   (a) Consider an inputless, randomized functionality RandOT, which outputs a random pair $(z_0, z_1) \in \mathbb{F}^2$ to Alice and (c, z_c) to Bob, where $c \in \{0, 1\}$ is a random bit. Give a protocol π^{RandOT} that UC-securely realizes OT, by accessing RandOT exactly once at the beginning of the protocol.

   (b) Oblivious Linear-function Evaluation (OLE) functionality over a field $\mathbb{F}$ (or, over a ring) is a generalization of OT. It accepts $(a, b) \in \mathbb{F}^2$ from Alice and $x \in \mathbb{F}$ from Bob and sends $y = ax - b$ as output to Bob (and $\perp$ to Alice). Give a protocol ρ^{OLE} that passive-securely realizes OT (over the same field) by accessing OLE.

   (c) Define an inputless, randomized version of OLE, called RandOLE, which outputs $(s_A, p_A) \in \mathbb{F}^2$ to Alice and $(s_B, p_B) \in \mathbb{F}^2$ to Bob, where (s_A, s_B, p_A, p_B) are uniformly random conditioned on the relation $s_A + s_B = p_A p_B$. (This distribution corresponds to picking p_A, p_B uniformly from the field, and setting s_A, s_B to be an additive sharing of p_A, p_B.)

   For the case when $\mathbb{F} = GF(2)$ (the field of the two elements $\{0, 1\}$), give a *deterministic, non-interactive* protocol σ^{RandOLE} that UC-securely realizes RandOT, by accessing RandOLE exactly once.

   (d) Generalize Part (a) to OLE (for any field): i.e., give a protocol τ^{RandOLE} that UC-securely realizes OLE, by accessing RandOLE exactly once at the beginning of the protocol.

3. **1-out-of-n OT from 1-out-of-2 OT.** [20 pts]

   In this problem you shall construct protocols for 1-out-of-n OT (which takes n bits $(x_1, \ldots, x_n)$ from Alice, an index $i \in \{1, \ldots, n\}$ from Bob and gives x_i to Bob), by accessing 1-out-of-2 OT.

   (a) Give a *simple, deterministic* protocol for 1-out-of-n OT, when security is required only against passive (honest-but-curious) corruption. In your protocol, Alice and Bob can access the 1-out-of-2 functionality n times.

   (b) Give a protocol that is secure against active corruption as well.

   [Hint: Consider $n = 3$. Suppose Alice and Bob carry out two 1-out-of-2 OTs: the first with Alice's inputs being (x_1, r) and the second with (y_2, y_3), where r is a random bit and $y_i = x_i \oplus r$. What should Bob's inputs in the two OTs be?]

4. **OT from Smooth Projective Hash** [20 pts]

   Describe a UC-secure $(n-1)$-out-of-n OT protocol (in the common reference string model) from Smooth Projective Hash (SPH). You should describe the protocol (including the setup) in detail, using the syntax for SPH from class. Also, briefly sketch a proof of security.