Homework 1

Advanced Tools From Modern Cryptography
CS 758 : Autumn 2017

Released: August 19 Saturday
Due: September 6 Wednesday

Secret-Sharing and MPC [Total 100 pts]

1. **Secure Switching of Linear Secret-Sharing.** [25 pts]

   Suppose Σ_1 and Σ_2 are two n-party linear secret-sharing schemes for messages in a set $\mathcal{M}$, with access structures $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively.

   Consider the functionality $\mathcal{F}_{\Sigma_1 \to \Sigma_2}$ which interacts with parties $P_1, \ldots, P_n$ as follows: for each i, it accepts w_i from party P_i, where w_i is in the share-space of Σ_1. Then it computes $m := \Sigma_1.\mathsf{recon}(w_1, \ldots, w_n)$, and using fresh randomness, computes $(z_1, \ldots, z_n) \leftarrow \Sigma_2.\mathsf{share}(m)$. Finally, for each $i \in [n]$, it sends z_i to P_i.

   (Here recon denotes the deterministic reconstruction algorithm and share denotes the randomized sharing algorithm, for a secret-sharing scheme.)

   Recall the protocol from the lectures for share-switching: Each party P_i sets $(\sigma_{i,1}, \ldots, \sigma_{i,n}) \leftarrow \Sigma_2.\mathsf{share}(w_i)$, and sends $\sigma_{i,j}$ to P_j. Then, each party P_i computes and outputs $z_i = \Sigma_1.\mathsf{recon}(\sigma_{1,i}, \ldots, \sigma_{n,i})$.

   (a) In order to show that the above is a passive-secure protocol for $\mathcal{F}_{\Sigma_1 \to \Sigma_2}$ against adversaries who corrupt only sets not in $\mathcal{A}_2$, describe a simulator. (You need not prove that the simulation is good.)

   (b) Now consider a set $S \in \mathcal{A}_2$. In the ideal world the adversary can learn m. Does that make the above protocol secure against passive corruption of parties in S? Justify your answer by either describing a simulator, or by arguing that there is no good simulator.

   Note. The Passive-BGW protocol from class can be formulated modularly as carrying out all communication after the initial input sharing phase, up till the final output phase, only through the share-switching functionality.

2. **Power of 2-party SFE with only one output.** [25 pts]

   In this problem we shall see how deterministic secure function evaluation (SFE) functionalities in which only one party receives the outcome can be easily used to realize more general functionalities securely, against passive (honest-but-curious) adversaries.

   (a) Suppose $\mathcal{R}$ is an arbitrary *randomized* 2-party functionality which takes x and y from Alice and Bob respectively, and samples a uniform random string r (of a fixed length) and gives $R_A(x, y, r)$ and $R_B(x, y, r)$ respectively to Alice and Bob (where R_A, R_B are two determinitic functions). Describe a *deterministic* 2-party SFE functionality $\mathcal{F}$ (which takes x^* and y^* from Alice and Bob respectively, and gives $f_A(x^*, y^*)$ and $f_B(x^*, y^*)$ to them respectively; you can specify what the functions f_A, f_B are), and a protocol $\pi^{\mathcal{F}}$ (i.e., a protocol in which Alice and Bob can access a trusted party implementing $\mathcal{F}$), such that $\pi^{\mathcal{F}}$ securely realizes $\mathcal{R}$. Security needs to hold only against passive adversaries.

   In your protocol $\pi^{\mathcal{F}}$, Alice and Bob should access $\mathcal{F}$ exactly once.

(b) Suppose $\mathcal{F}$ is an arbitrary 2-party SFE functionality which takes x and y from Alice and Bob respectively, and gives $f_A(x, y)$ and $f_B(x, y)$ to them respectively. Describe another 2-party SFE functionality $\mathcal{G}$ which provides output only to Bob (i.e., Alice gets a dummy output $\perp$), and a protocol $\rho^{\mathcal{G}}$ (i.e., a protocol in which Alice and Bob can access a trusted party implementing $\mathcal{G}$), such that $\rho^{\mathcal{G}}$ securely realizes $\mathcal{F}$. In your protocol $\rho^{\mathcal{G}}$, Alice and Bob should access $\mathcal{G}$ exactly once. Security needs to hold only against passive adversaries.

3. **OT, OLE and Correlated Random Variables.** [25 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 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.

Give a protocol τ^{RandOLE} that securely realizes OLE, by accessing RandOLE exactly once at the beginning of the protocol. [Extra Credit]

4. **1-out-of-n OT from 1-out-of-2 OT.** [25 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?]