CS783: Theoretical Foundations of Cryptography Fall 2024

Assignment 5

October 12, 2024

Instructor: Chethan Kamath

Exercise 1 (Functionalities with trivial MPC protocols)**.** *We saw in Lecture 17 that* $\oplus$ *has a trivial 2PC protocol:*

- P_1 *sends his input bit* x_1 *to* P_2
- P_2*, with input bit* x_2*, outputs* $x_1 \oplus x_2$*.*

Consider any deterministic functionality $\mathrm{f} = (\mathrm{f}_1, \mathrm{f}_2)$ *for which output depends only on one party's input, say* P_1*'s:* $(x_1, x_2) \mapsto \mathrm{f}_1(x_1), \mathrm{f}_2(x_1)$*, Show that* f *also has a trivial MPC protocol.*

Exercise 2 (Two-party computation (2PC) and zero-knowledge proof (ZKP))**.** *Recall from Lecture 17 that for an* **NP** *relation* $\mathcal{R}$ *corresponding to an* **NP** *language* $\mathcal{L}$*,* $(x, w) \in \mathcal{R}$ *if and only if* $x \in \mathcal{L}$*. You are given a 2PC protocol* $\Pi = (\mathsf{P}, \mathsf{V})$ *that privately computes* $\mathcal{R}$ *in the following sense:*

- *The common input to* P *and* V *is the instance* x*.*
- P*'s private input is the witness* w*;* V *has no private input.*
- V*'s output in the protocol is the predicate corresponding to* $\mathcal{R}(x, w)$*, i.e., the output is* 1 *if and only if* $(x, w) \in \mathcal{R}$*.*

Show formally that Π *is a ZKP for* $\mathcal{L}$*. That is, show that* Π *is correct, sound and zero-knowledge based on its its 2PC properties (correctness and privacy).*

Exercise 3 (Linear functions)**.** *Recall from Lecture 17 that a function* $\mathrm{f} : \mathbb{F}_p^n \to \mathbb{F}_p$*, where* $\mathbb{F}_p = (\mathbb{Z}_p, +, \cdot)$ *is a finite field of order* p*, is linear if for all* $\bar{a}, \bar{b} \in \mathbb{F}_p^n$*,* $\mathrm{f}(\bar{a} + \bar{b}) = \mathrm{f}(\bar{a}) + \mathrm{f}(\bar{b})$*. Show that any linear function* f *can be represented an algebraic circuit with only addition and multiply-by-constant gates. (Hint: show that* f *can be represented in the canonical basis* $(\bar{e}_1, \ldots, \bar{e}_n)$*, where* $\bar{e}_i$ *is the vector with* $1 \in \mathbb{F}_p$ *in position* i *and* $0 \in \mathbb{F}_p$ *everywhere else.)*

Exercise 4 (Return of the combiner)**.** *Recall from Assignment 2 and Quiz 1, that a combiner for shared-key encryption (SKE) schemes* $\Sigma_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ *and* $\Sigma_2 = (\mathsf{Gen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$ *is another SKE scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *constructed using* Σ_1 *and* Σ_2 *that is secret against eavesdroppers as long as* at least one out of Σ_1 *or* Σ_2 *is secret against eavesdroppers. To be precise,* Σ *is said to be a* $(1, 2)$*-combiner. Using threshold secret-sharing scheme, your goal is to now construct a* (t, n)*-combiner: given SKEs* $(\Sigma_1, \ldots, \Sigma_n)$*, construct SKE* Σ *that is secret against eavesdroppers as long as* at least t out of n *given SKEs are secret against eavesdroppers. (Hint: you need to extend the solution in the solution set for Quiz 1.)*

Exercise 5 (Oblivious transfer (OT) and public-key encryption (PKE))**.** *In this exercise, we will try to better understand the relationship between OT and PKE. In Part 1, we will show that OT implies PKE. In Part 2, we will show that PKE with some additional properties implies OT. (Additional properties are necessary as OT* cannot *be constructed from plain PKE!)*

1. *Given any one-round* 1*-out-of-*2 *OT protocol* $\Pi = (\mathsf{S}, \mathsf{R})$ *(where the sender speaks first and then the receiver replies), construct a PKE scheme* $\Sigma := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*.*

2. *In Lecture 18, we saw that the construction of* 1*-out-of-*2 *OT from trapdoor permutation (TDP) looked very similar to the construction of PKE from TDP. In this exercise, we will make this connection a bit more concrete. A PKE scheme* $\Sigma := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to have* oblivious key-sampling *if there exists additional algorithms* SCoins *(sample random coins) and* SKey *(sample key) with the following syntax:*

   - $r \leftarrow \mathsf{SCoins}(pk)$
   - $pk \leftarrow \mathsf{SKey}(1^n; r)$

   and such that the following distributions are computationally indistinguishable:

   - (r, pk)*, where* $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$ *and* $r \leftarrow \mathsf{SCoins}(pk)$
   - (r, pk)*, where* $r \leftarrow \{0,1\}^n$ *and* $pk := \mathsf{SKey}(1^n; r)$*.*

   Construct 1*-out-of-*2 *OT from key-oblivious PKE that is secure against eavesdroppers. Prove that is secure. (Hint: Think in terms of Protocol 1 from Lecture 18.)*

Exercise 6 (1-out-of-2 OT $\rightarrow$)**.**

1. *Given* 1*-out-of-*2 *OTs, construct a* 1*-out-of-*4 *OT. First think of a construction that uses four* 1*-out-of-*2 *OTs. Then come up with a construction using only three* 1*-out-of-*2 *OTs (Hint: think serially). Do you think this is possible using just two* 1*-out-of-*2 *OTs?*

2. *In Lecture 18, we saw how to privately compute* $\wedge$ *using one* 1*-out-of-*2 *OT. Show how to privately compute* n*-bit comparison using (as many as you like)* 1*-out-of-*2 *OTs. In more details, the parties* P_1 *and* P_2 *have inputs* $x_1, x_2 \in [0, 2^n - 1]$*, respectively. At the end of the protocol they locally output whether or not* $x_1 \leq x_2$*, but should not have learnt anything else about the other party's input. (You will have solved Yao's Millionaires' problem using OT.)*