CS783: Theoretical Foundations of Cryptography Fall 2024

Assignment 4

October 9, 2024

Instructor: Chethan Kamath

Exercise 1 (Robustness of definition of interactive proof (IP))**.** *Recall Definition 1 (IP) from Lecture 14. The correctness error ϵ_c and soundness error ϵ_s in that definition are both set to* constant $1/3$*. Show that if a language $\mathcal{L}$ has an IP Π according to Definition 1, then it has an IP Π' according to a definition where ϵ_c and ϵ_s are set to*

1. *negligible: $\epsilon_c = \epsilon_s \leq 1/2^n$*
2. *noticeable: $\epsilon_c = \epsilon_s \leq 1/2 - 1/n$,*

where n denotes the size of the instance. (Hint: just like in Exercise 3 of Assignment 3, Π' reduces the error by repeating Π and then taking a majority vote. The analysis is then using Chernoff bound.)

Exercise 2 (Randomness is useful for verification)**.**

1. *Show that an IP with deterministic verifier can only exist for $\mathcal{L} \in \mathbf{NP}$. (Hint: Observe that prover's messages are fixed if the verifier is deterministic.)*
2. *A language $\mathcal{L}$ is in the class* **BPP** *(bounded-error probabilistic polynomial-time) if there exists a probabilistic polynomial-time decider D such that:*

   - $\forall x \in \mathcal{L}$: $\Pr[\mathsf{D}(x) = 1] \geq 2/3$
   - $\forall x \notin \mathcal{L}$: $\Pr[\mathsf{D}(x) = 1] \leq 1/3$,

   where the probabilities are over random coins of D. Note that **BPP** *has a trivial ZK protocol: the prover sends nothing and the verifier simply decides the membership of an instance x in $\mathcal{L}$* on her own*. Show that a zero knowledge proof (ZKP) with deterministic verifier can only exist for $\mathcal{L} \in$* **BPP***.*

Exercise 3 (A sanity check)**.** *Recall from Lecture 14 that the* **NP** *proof for graph isomorphism of (G_0, G_1) involves the prover sending the witness, i.e., the isomorphism π between G_0 and G_1. As noted in the lecture this protocol is a perfect IP for graph isomorphism problem (where the IP verifier simply runs the* **NP** *verifier). However, this protocol* should not *satisfy Definition 2 (ZKP) in Lecture 14. Point out where exactly it fails to satisfy Definition 2. (Hint: Write down the contrapositive of Definition 2.)*

Exercise 4 (More trivial ZK)**.** *In this exercise, we will see more cases where ZK can only exist for trivial languages, i.e, those in* **BPP***.*

1. *Recall that when defining the simulator for honest-verifier (HV) computational ZK in Lecture 14 (Definition 2), the real view and simulated view are only required to be indistinguishable for $x \in \mathcal{L}$. Show that if the distribution of simulator's output*

on $x \in \mathcal{L}$ and $x \notin \mathcal{L}$ are computationally distinguishable*, then $\mathcal{L} \in \mathbf{BPP}$. (Hint: use simulator and distinguisher for the simulator's output to construct a decider for $\mathcal{L}$.)*

2. *Show that non-interactive (NI) perfect ZK proof can only exist for $\mathcal{L} \in \mathbf{BPP}$. (Hint: keeping in mind Exercise 4.1, analyse what happens when your run the verifier on simulated transcript)*

3. *Extend Exercise 4.2 to NI* computational *ZK proof.*

Exercise 5 (Non-interactive bit commitments (NIBC))**.** *Recall that in Definition 1 (NIBC) from Lecture 16, our requirements were computational hiding and perfect binding. In the following exercises, we will try to better understand NIBC.*

1. *Given an NIBC $\Sigma = (\mathsf{S}, \mathsf{R})$, construct a OWF* f*. Explain why* f *is one-way.*

2. *Consider the dual of Definition 1, where we require computational binding (i.e., a* PPT adversary *should not be able to find commitment c and decommitments r_0 and r_1 such that $\mathsf{R}(c, r_0, 0) = 1$ and $\mathsf{R}(c, r_1, 1) = 1$ both hold) and perfect hiding (i.e., commitments to 0 and 1 are identically distributed). Given an NIBC Σ that satisfies the dual definition, construct a OWF* f*. Explain why* f *is one-way.*

3. *We have now seen two definitions of NIBC. Now consider a strengthening of the two where we require both binding* and *hiding to be perfect. Show that perfectly binding and perfectly hiding NIBCs* cannot *exist.*

4. *Construct a NIBC in the random-oracle model (ROM). Recall that in ROM, all parties have oracle access to a random function, say, $\mathrm{H}_n : \{0,1\}^n \to \{0,1\}^n$.*

Exercise 6. *In Lecture 16, we showed that Schnorr's protocol is HVZK. What happens to ZK when you consider Schnorr's protocol with malicious verifiers?*