CSC 2426: Fundamentals of Cryptography **Fall 2023**

Lecture 8: Zero-Knowledge Proofs

Instructor: Akshayaram Srinivasan *Scribe: Yiqing Xia*

Date: November 13, 2023

8.1 Recap

In the last lecture, we talked about Collision-Resistant Hash Function and the construction of a q-Time signature.

In this lecture, we will look at zero-knowledge proofs.

8.2 Interactive Proofs

Definition. A pair of interactive machines (Prover, Verifier), or (P,V), is an interactive proof system for a language $\mathcal{L}$ if V is a p.p.t. machine and the following properties hold:

- Completeness: $\exists$ Prover P s.t. $\forall x \in \mathcal{L}$, the verifier V accepts this proof at the end of interaction;
- Soundness: $\forall x \notin \mathcal{L}$, $\forall$ Prover P^*, the probability that the verifier V accepts is $\leq negl(n)$.

Example. Consider $\mathcal{L} \in$NP, i.e. $\mathcal{L} = \{x \in \mathcal{L} | \exists w \in \{0,1\}^{poly(|x|)}$ s.t. $\mathcal{R}(x,w) = 1\}$. Here, w is the witness string defined by the definition of NP, and $\mathcal{R}$ is a public p.p.t machine which can be used by the Verifier. Then, a valid (Prover, Verifier) could have the Prover sending the witness w to the Verifier and the Verifier outputs $\mathcal{R}(x,w)$.

Given the definition of Interactive Proofs, now we want to construct a type of proof where the prover could convince the verifier about the truth of some statement without revealing any other information. For example, for $\mathcal{L} \in \mathsf{NP}$, we don't want the verifier to learn any information about the witness but at the same time, get convinced that x belongs to NP. Such interactive proofs are called zero-knowledge proofs. How do we formally define that the verifier learns no information about the witness?

To build-up to the actual definition, consider the following attempts:

Attempt 1: "Witness hiding" This requires that the verifier cannot output the witness at the end of the interaction. However, this does not prevent the verifier from learning partial information about the witness and hence, this is unsuitable.

Attempt 2: "Witness indistinguishability" This requires that for any given two witnesses w_0, w_1, the view of the verifier when interacting with a prover that uses either w_0 or w_1 should be indistinguishable.

However, for languages with unique witness, sending the witness in the clear satisfies witness indistinguishability but does not satisfy zero-knowledge as the verifier learns the witness completely.

Actual Definition: To define zero-knowledge, consider an ideal world where the verifier interacts with another PPT machine called the simulator. The simulator takes only the statement as input and consider the verifier running the simulator in its head. By definition, as the simulator has no information about the witness, the verifier learns no information from its interaction with the simulator. Now, if we somehow prove that whatever the verifier "sees" in the real world is indistinguishable to whatever verifier "sees" in the ideal world, then we can infer that the verifier gets no information about the witness in the real world. This is the intuition behind the zero-knowledge definition.

Definition 8.1 *An Interactive Proof (P, V) for language $\mathcal{L} \in NP$ with witness w is called a Zero-Knowledge Proof if $\forall$ p.p.t.$V^*, \exists$ p.p.t. Simulator Sim s.t. $\forall x \in \mathcal{L}$, there is $View_{V^*}(P(w), V^*) \approx_c Sim^{V^*}(x)$. Here, $View$ comprises of the verifier's input, its random coin tosses, and the messages sent by the prover during its interaction.*

8.3 Commitment Schemes

To construct a zero-knowledge proof, we need a tool called commitments.

$Com(1^n, m) \to (c, r)$, where r is the "opening" of black box; and when it was returned a (m, r),
$Verify(c, (m, r)) \to acc/rej$.

Definition. A polynomial-time machine Com is called a Commitment Scheme i the following two properties hold:

- Hiding: $\forall m_0, m_1, \{(c, r) \leftarrow Com(1^n, m_0) : c\} \approx_c \{(c, r) \leftarrow Com(1^n, m_1) : c\}$;
- Binding: $\nexists\, c, (m_0, r_0), (m_1, r_1)$ s.t. $m_0 \neq m_1$ but $Verify(c, (m_0, r_0)) = acc \wedge Verify(c, (m_1, r_1)) = acc$.

Theorem. One-way Permutation $\mapsto$ Commitments
Let $\{f_n : \{0,1\}^n \to \{0,1\}^n\}$ and $\{h_n : \{0,1\}^n \to \{0,1\}^n\}$ be the OWP. A single-bit Commitment Scheme $(Com, Verify)$ works like:

$Com(1^n, b \in \{0,1\}) = c = \{f_n(r), h_n(r) \oplus b\}, r$; and
$Verify(c, (b', r'))$ outputs 1 iff $f_n(r') = f_n(r)$ and $h_n(r') \oplus h_n(r) \oplus b = b'$.

Proof. The Binding property is straightforward given how the Commitment Scheme is constructed. We will prove the Hiding property using Hybrid Arguments.

$H_0 : f_n(r), h_n(r) \oplus b_0$
$H_1 : u_1, u_2 \oplus b_0$. $H_1 \approx_c H_0$ since $f_n(r), h_n(r)$ is a PRG.
$H_2 : u_1, u_2 \oplus b_1$. $H_2 \approx_c H_1$ given that u_1, u_2 are random.
$H_3 : f_n(r), h_n(r) \oplus b_1$. $H_3 \approx_c H_2$ given the definition of OWP.
Therefore the Hiding property holds.

8.4 Zero-Knowledge Proof for NP

We will construct a zero-knowledge proof for NP based on a commitment scheme.

Consider the 3-Coloring problem: Given graph $G = (V, E)$, is there a $C : V \to \{R, G, B\}$ s.t. $\forall e \in E = (i, j)$, there is $C(i) \neq C(j)$?
We will use a commitment to do the ZKP for this specific problem.

Common input	$G = (V, E)$ where $\|V\| = n, \|E\| = m$.
Prover input	Witness $w = C : V \to \{R, G, B\}$
$P \to V$	Let π be a random permutation of $\{R, G, B\}$.
	Let $C' = \pi \circ C$. $\forall v \in V, (\alpha_v, r_v) \leftarrow Com(C'(v))$.
	Sends $C'(v) \forall v$.
$V \to P$	Sends a randomly selected edge $e = (i, j) \in E$.
$P \to V$	Sends $(C'(i), r_i), (C'(j), r_j)$.
V	Accepts iff $C'(i) \neq C'(j)$.
P, V	Repeats the procedure for mn times.

Proposition. The proof above is a valid ZKP.

Proof. Firstly, prove its soundness. If the Prover is cheating, it means that exists at least one edge (i, j) s.t. $C(i) = C(j)$. Given the binding property of Commitment Scheme, in each iteration, there is at least $1/m$ probability where the Prover is caught. Let's see how to bring down the probability that the prover could cheat to negligible.

Next, prove its Zero-Knowledge. We start with the description of the simulator. The simulator chooses a random edge (i, j) and chooses two random colors to color the vertices i and j. For every other vertex, it colors using an arbitrary color. Note that this coloring may not be a valid 3-coloring. It commits to the coloring and then waits until the verifier sends its edge. If this edge matches with the previously chosen edge, then the simulator proceeds with completing the protocol and outputting the view of the verifier. Else, it starts once again by choosing a random edge and repeats the above process. If the simulator fails in each one of the $n \cdot m$ repetitions, it outputs a special symbol fail. We now show that the view generated by the simulator is indistinguishable to the real world by a hybrid argument.

- Hyb_0 : This corresponds to the view of the verifier in the real world interaction with the prover.
- Hyb_1 : In this hybrid, we modify the original prover to initially guess an edge (i, j) before the start of the interaction. It commits to the colors as in the previous hybrid, but if the verifier sends a different edge than the one it guessed initially, it starts the interaction once again. It outputs a special symbol fail, if it fails to correctly guess the verifier's edge in each one of $n \cdot m$ iterations. Note that the probability that the prover fails to correctly guess the edge in a single iteration is $1 - \frac{1}{m}$. Since each repetition is independent, the probability that the prover fails to guess the verifier's edge in each one of the $n \cdot m$ iterations is $(1 - \frac{1}{m})^{nm} \leq e^{-n}$ which is negligible. Note that the only difference between Hyb_1 and Hyb_0 is that Hyb_1 outputs fail if the prover fails in each of the repetitions. However, the probability that the prover fails is at most e^{-n}. Hence, Hyb_1 is computationally indistinguishable to Hyb_0.
- Hyb_2 : In this hybrid, we change the prover from Hyb_1 to behave exactly like the simulator. Note that the only difference between Hyb_1 and Hyb_2 is that in Hyb_2, the coloring of other vertices are generated arbitrarily whereas in Hyb_1, we committed to the correct coloring. Furthermore, in the rest of the protocol interaction, we did not use the openings to any of these commitments. We only needed the openings to the vertices (i, j) but since we generated these to be commitments to random colors,

we know the openings. Thus, the indistinguishability of this hybrid from the previous hybrid follows directly from the hiding property of the commitment scheme.

Remark. To reduce the soundness error in the above protocol, we need to repeat the protocol independently. If we repeat nm times, the soundness error becomes $(1-\frac{1}{m})^{mn} \leq e^{-n}$. We can repeat this either sequentially or parallely. But it is known that only sequential repetition preserves zero-knowledge property.