CS276: Cryptography October 29, 2015

Zero Knowledge Proofs

Instructor: Alessandro Chiesa *Scribe: Chenyang Yuan*

1 Interactive Proofs

An interactive proof system consists of two parties: a prover and an verifier. Given a statement, the verifier makes adaptive queries to the prover, and after a certain number of exchanges decides if the statement is true. Intuitively, such proof systems must satisfy two properties:

1. Completeness: All true statements must have proofs. That is, after the interaction between the prover and the verifier, the verifier must be able to conclude that the statement is true.
2. Soundness: All false statements do not have proofs. No malicious prover can convince the verifier that a false statement is true.

Now we can formally define an interactive proof:

Definition 1 *An interactive proof for a language L is a pair of algorithms (P, V) where P is unbounded and V is ppt. satisfying two properties:*

1. $\forall x \in L, \Pr[\langle P, V\rangle(x) = 1] = 1$ *(Completeness)*
2. $\forall x \notin L, \forall P^* \in \mathcal{P}, \Pr[\langle P^*, V\rangle(x) = 1] \leq \frac{1}{2}$ *(Soundness)*

Where $\langle P, V\rangle(x)$ is the verifier's output after a series of interactions with the prover on input x, and $\mathcal{P}$ is the class of all provers.

There are two key ingredients necessary in interactive proofs: interaction and randomness. If we remove randomness, we get the complexity class NP. If we remove interaction and keep randomness, we get the complexity class MA.

2 Honest Verifier Zero-Knowledge Proofs

A zero-knowledge proof is an interactive proof where the verifier cannot learn anything from the interaction except the fact that the theorem to be proved is true. That is if we can construct a simulator that gives the same probability distribution of the view of the interaction without invoking the prover. First we give the definition when we assume that the verifier is honest, therefore not considering the possibility that a malicious verifier can extract extra information from the prover.

Definition 2 *An interactive proof (P, V) is an honest verifier zero-knowledge proof if there exists an expected ppt simulator S such that:*

$$\forall x \in L, S(x) = View_V(\langle P, V\rangle(x))$$

Where $S(x)$ and $View_V(\langle P, V\rangle(x))$ are distributions, and $View_V(\langle P, V\rangle(x))$ contains the randomness used by V and all entries in the transcript of the interaction.

3 Zero-Knowledge Proofs

We now give a definition for zero-knowledge proofs for all (potentially malicious) verifiers, by giving oracle access of the verifier to the simulator.

Definition 3 *An interactive proof (P, V) is a zero-knowledge proof if there exists an expected ppt simulator S such that:*

$$\forall x \in L, \forall V^* \in \mathcal{V}, S^{V^*}(x) = View_{V^*}(\langle P, V^* \rangle(x))$$

Where $S^{V^*}(x)$ and $View_{V^*}(\langle P, V^* \rangle(x))$ are distributions, $View_{V^*}(\langle P, V^* \rangle(x))$ contains the randomness used by V^* and all entries in the transcript of the interaction, and $\mathcal{V}$ is the class of all verifiers.

Remark 4 *The definition of a zero-knowledge proof given is for perfect zero-knowledge proofs (PZK), if the equality between distributions is replaced by statistical or computational indistinguishability, we get the definitions for statistical (SZK) or computational (CZK) zero-knowledge proofs.*

Theorem 5 *The language* Graph-Isomorphism*, which is the set $\{(G_0, G_1) \mid \exists \phi \text{ s.t. } G_0 = \phi(G_1)\}$ where ϕ is a permutation of vertices, has a zero-knowledge proof.*

Proof: We first construct a prover and verifier, show that it is an interactive proof then prove that it is a zero-knowledge proof. Suppose both the prover and verifier know G_0 and G_1.

$P: G_0, G_1$		$V: G_0, G_1$
Compute ϕ s.t. $G_0 = \phi(G_1)$		
$\psi \leftarrow$ Random permutation		
$G \leftarrow \psi(G_0)$	$\xrightarrow{G}$	
	$\xleftarrow{b}$	$b \leftarrow \{0,1\}$
$\alpha \leftarrow \psi \circ \phi^b$	$\xrightarrow{\alpha}$	Output $\alpha(G_b) \stackrel{?}{=} G$

We prove that (P, V) is an interactive proof.

1. Completeness: Note that $\phi^b(G_b) = G_0$. Then $\alpha(G_b) = \psi(\phi^b(G_b)) = \psi(G_0) = G$.
2. Soundness: Suppose G_0 is not isomorphic to G_1. Then the received H is either in the isomorphism class of G_1, G_0 or neither. Since b is chosen at random, the probability that the verifier accepts the proof is at most $1/2$.

First we prove that (P, V) is honest verifier zero-knowledge. We construct a ppt S that reproduces the distribution of the view of the interaction:
$S(G_1, G_2) :=$

1. Sample random permutation α.
2. Sample $b \leftarrow \{0, 1\}$
3. Compute $G \leftarrow \alpha(G_b)$
4. Output (G, b, α)

Notice that in this interaction, if we are given G, computing b and α is believed to be inefficient. However, if we are given b and α, computing G is efficient. Informally, this notion of directionality contributes to the soundness of the proof, but it does not matter when we want to sample the view of the interaction ourselves.

Now we prove that (P, V) is zero-knowledge. The difference here is that a verifier V^* does not need to output b^* following a uniform distribution. Now we face a problem if we attempt a similar construction as before (sampling b first uniformly), since b^* can have any arbitrary distribution. Thus we want to run V^* first to get b^*, however to do that we need to first compute G. To solve this chicken-and-egg problem, we allow S^{V^*} to fail, but it runs in expected polynomial time. First we define the view of the interaction in this case:

$$\text{View}_{V^*}(\langle P, V^* \rangle (G_0, G_1)) := (r, G, b^*, \alpha)$$

1. $G = \phi(G_0)$ for a random permutation ϕ
2. r is the private randomness of V^*
3. $b^* = V^*(r, G_0, G_1, G)$
4. α such that $G = \alpha(G_{b^*})$

$S^{V^*}(G_0, G_1) :=$

1. Sample b, α, r at random
2. $G := \alpha(G_b)$
3. $b^* \leftarrow V^*(r, G_0, G_1, G)$
4. If $b^* = b$, return (r, G, b^*, α). Otherwise fail.

This scheme works because conditioning a random variable (b^*) with another independent random variable (b) does not change its distribution. Therefore if S^{V^*} succeeds, then (r, G, b^*, α) will have the same distribution as $\text{View}_{V^*}(\langle P, V^* \rangle (G_0, G_1))$. The probability of failure is $1/2$, thus S^{V^*} runs in expected polynomial time. □

4 Zero-Knowledge Proof Systems for NP

Unfortunately, the classes of problems that have perfect zero-knowledge proofs is small. To consider problems in NP, we relax our definition to computational zero-knowledge proofs instead. If we can construct a zero-knowledge proof for a NP-complete problem, say 3-Coloring, then we can do it for all of NP.

Theorem 6 *Let the language* 3-Coloring *be the set* $\{G \mid \exists$ *3-coloring* $\alpha : [n] \to [3]$ *for* $G\}$*. If there exists a computationally hiding and statistically binding commitment scheme* (S, R)*, then there exists a computational zero-knowledge proof for* 3-Coloring*.*

Informally, the main idea is that the prover sends a scrambled version of the proof, and the verifier picks a randomly chosen part of the proof for the prover to unscramble and then checks that part. This scheme is zero-knowledge if the part revealed doesn't give the verifier any information about the proof. The commitment scheme serves two purposes here. First, it is used hide the proof from the verifier. Second, the prover should not be able modify the proof later on to fool the verifier.

Suppose G is the graph, E are its edges, and all sampling are done uniformly randomly. Now we provide the construction of the prover and verifier: