COM S 6830 – Cryptography Oct 29, 2009

Lecture 18: Zero-Knowledge

Instructor: Rafael Pass *Scribe: Matt Weinberg*

1 Recap

Recall what zero-knowledge is:

Definition 1 *The interactive protocol (P, V) is in ZK if:*

$$\forall PPT\ V^*, \exists EPPT\ S, \forall nuPPT\ D, \exists \epsilon, \forall x \in L, w \in R_L(x), \forall z \in \{0,1\}^*$$

ϵ is negligible and D distinguishes:

$$\{View[P(x,w) \leftrightarrow V^*(x,z)]\}$$

from

$$\{S(x,z)\}$$

with probability $\leq \epsilon(|x|)$. In english, for all verifiers, there exists a simulator, such that the view of the simulation is indistinguishable from the view of the actual protocol. So the verifier doesn't learn anything because he could have simulated the interaction himself (even if he were dishonest).

Last class we saw the following two theorems:

Theorem 1 $NP \subseteq ZK$

Theorem 2 $IP \subseteq ZK$

2 NP in fewer rounds

Recall that the proof we saw for $NP \subseteq ZK$ involved linear rounds. This is because for any single round, our soundness was only $1/m$ (because only a single edge would be incorrectly colored). Recall that the protocol for graph isomorphism had soundness $1/2$ (because when the graphs are not isomorphic, the prover will be asked for a non-existent isomorphism with probability $1/2$ and be caught).

2.1 Graph Isomorphism in fewer rounds

The idea to get fewer rounds is to run several independent rounds in parallel. Each round is independent already, and doesn't depend on the outcome of the previous rounds. So why not just run the graph iso protocol n times in parallel to get soundness $1 - 1/2^n$? The problem is that the protocol may no longer be zero-knowledge. Remember that now, instead of sending a single bit, V^* sends k bits, where k is the number of independent rounds we are running in parallel. Remember also that in contstructing the simulator for V^* in the standard graph iso protocol, we had to let the simulator guess V^*s message, and run until it did. However, now that V^*s message is k bits, this takes expected time 2^k. So when $k \in \mathcal{O}(logn)$, we can run k rounds independently and still be in ZK. However, we cannot have n independent rounds run in parallel. This immediately leads us to a protocol for graph iso that runs in $\omega(1)$ rounds, we just run $log(n)$ independent rounds in parallel, and we only need to run $\omega(log(n))$ rounds total to achieve negligible soundness error.

2.2 Hamiltonian Cycle

To make a ZK protocol for NP in fewer rounds, we need to consider a different NP problem. Given a graph, G, on nodes V, a hamiltonian cycle is a cycle of length $|V|$. It is NP-complete to, given a graph G, decide if a hamiltonian cycle exists.

2.3 A new protocol with soundness $1/2$

Now we consider the following protocol for proving that a graph has a hamiltonian cycle. The prover does the following:

1. Generate a permutation, π of V.
2. Commit to π.
3. Generate the adjacency matrix $\pi(G)$.
4. Commit to each $\pi(G)_{ij}$ separately.

The verifier replies by:

1. Randomly sample $b \leftarrow \{0, 1\}$.
2. If $b = 0$, ask for π and $\pi(G)$.
3. If $b = 1$, ask for a hamiltonian path in $\pi(G)$.

The prover replies by complying with the verifiers request. This proof is ZK for the same reason as the graph iso protocol, because the verifier only sends a single bit, we will just run the simulator until it guesses the verifiers message correctly. In addition,

whenever the verifier asks for π and $\pi(G)$, a simulator can generate a random π and compute $\pi(G)$. Whenever the verifier asks for a hamiltonian cycle in $\pi(G)$, a verifier can randomly generate a hamiltonian cycle on n vertices, because all cycles will occur with equal probability over random permutations π. Furthermore, this protocol has completeness 1. Whenever there is a hamiltonian cycle, the prover will generate a valid permutation, correctly apply it to G, and be able to reveal a valid permutation, or a hamiltonian path in $\pi(G)$. Finally, this protocol has soundness 1/2. If there is no hamiltonian path in G, then the prover will do one of two things. either he can choose a valid π, and correctly compute $\pi(G)$. In this case, $\pi(G)$ has no hamiltonian path, so with probability 1/2 the verifier will ask for one and the prover will be caught. Or maybe the prover will cheat and incorrectly compute $\pi(G)$. In this case, with probability 1/2 the verifier will ask for π and $\pi(G)$ and the prover will be caught. So we have a ZK protocol for NP with soundness 1/2 and completeness 1. We can do the same trick for graph iso and run this protocol $log(n)$ times in parallel, and now we only need $\omega(1)$ rounds to achieve negligible soundness error.

2.4 NP in even fewer rounds?

Now that we have a protocol for NP in *basically* constant rounds, is it possible to get a protocol in constant rounds? Under certain reasonable assumptions (the existence of OWFs, existence of OWPs, existence of TDPs), no construction is known. However, there is a construction based on Collision-Resistant Hash functions. In addition, the existence of CRHs is implied by the hardness of discrete log or the hardness of factoring.

Remember in the protocol for Hamiltonian Cycle, that we could only repeat it in parallel $logn$ times without risking the loss of zero-knowledge. What if, to avoid this, we added an extra round at the beginning of the protocol where V commited to his message. This would stop V^* from lying about his message, and now the simulator doesn't have to guess it. The problem is that P is not computationally bounded. So when V commits to a message, P can break the commitment, and cheat in the protocol because he knows Vs message. To avoid this problem, we need a stronger type of commitment. In particular, a commitment that is statistically hiding and computationally binding. This means that P will not be able to break the commitment with any amount of computational power because there are just too many possibilities that Vs message could be that all have the same output. In addition, it is computationally hard for V to find any of these other messages and break the commitment. These types of commitments exist if CRHs exist. Even with this type of commitment, it is hard to show that the protocol works, so the proof is not shown.

3 Lower bounds on number of rounds

Since we think it may be possible to have a ZK protocol for NP that runs in constant number of rounds, how low can that constant possibly be? Below is a theorem that says it is unlikely that a protocl can run in 2 rounds. In particular, no protocol can run in 2 rounds unless $NP \subseteq BPP$.

3.1 A Theorem on 2-round protocols

Theorem 3 *If there exists a 2-round ZK protocol for L, then* $L \in BPP$.

Proof. Becuase $L \in ZK$, there exists a simulator that works for every verifier. In particular, it works for the verifier $V^*(x, z) = z$ (IE: V^* just outputs z). Now we prove two lemmas. In both cases, ϵ just denotes some negligbile function.

Lemma 4 *If* $x \in L$, *then* $Pr[z \leftarrow V(x), S$ *outputs an accepting view*$] \geq 1 - \epsilon$.

Proof. First, observe that the view of V^* and the view of V are exactly the same distribution. Furthermore, we know that with probability 1, the view of V is an accepting view when $x \in L$. Because the output of S is indistinguishable from the view of $V^*(x, z)$, it must be the case that S outputs an accepting view with probability at least $1 - \epsilon$. If not, then we could distinguish S from the view of $V^*(x, z)$ by just checking whether it is accepting or not.

Lemma 5 *If* $x \notin L$, *then* $Pr[z \leftarrow V(x), S$ *outputs an accepting view*$] \leq \epsilon$.

Proof. Assume for contradiction that this were not true. Then we can construct a dishonest prover, P^*. P^* will just run $S(x, V(x))$, and with non-negligible probability have an accepting view. Then P^* will just output his share of that view. So now we have a BPP algorithm that can decide L! Let M run $S(x, V(x))$. If S outputs an accepting view, accept, otherwise reject. If $x \in L$, there is a $1 - \epsilon$ chance that we get an accpeting view by lemma 4. If $x \notin L$, there is a $1 - \epsilon$ chance that we get a non-accepting view by lemma 5. So no matter what, we are wrong with negligible probability, so M is an expected PPT that is wrong in deciding L in all cases with negligible probability, so $L \in BPP$.

4 Witness Indistinguishability

Definition 2 *An interactive protocol* (V, P) *is* ***Witness Indistinguishable*** *if* P *is EFFICIENT, and:*

$$\forall PPT\ V^*, \forall nuPPT\ D, \exists \epsilon, \forall x \in L, w_1, w_2 \in R_L(x), z \in \{0,1\}^*$$

ϵ *is negligble, and* D *distinguishes the following two ensembles with probability* $\leq \epsilon(|x|)$.

$$\{View_{V^*}[P(x, w_1) \leftrightarrow V^*(x, z)]\}\ ,\ \{View_{V^*}[P(x, w_2) \leftrightarrow V^*(x, z)]\}$$

In other words, a protocol is Witness Indistinguishable when no nuPPT can tell whether the prover used w_1 or w_2 in his proof.

Proposition 1 *If (P, V) is a ZK protocol, then (P, V) is witness indistinguishable.*

Proof. By definition of ZK, there exists a simulator S, such that:

$$\{P(x, w_1) \leftrightarrow V^*(x, z)\} \approx \{S(x, z)\} \approx \{P(x, w_2) \leftrightarrow V^*(x, z)\}$$

By the hybrid lemma, $\{P(x, w_1) \leftrightarrow V^*(x, z)\} \approx \{P(x, w_2) \leftrightarrow V^*(x, z)\}$. So (P, V) is witness indistinguishable.
Finally, here is the theorem that says why Witness Indistinguishability is nice to work with.

Theorem 6 *If (P, V) is WI, then (P^n, V^n) is WI. In other words, WI protocols can be repeated polynomially many times in parallel and still be WI.*

Proof. We want to show that $\{P^n(x, w_1) \leftrightarrow V^{*n}(x, z)\} \approx \{P^n(x, w_2) \leftrightarrow V^{*n}(x, z)\}$. To do this, define the following hybrids: Let H_i denote the view where the prover uses w_1 for the first i executions of the protocol, and w_2 for the rest. Then it is clear that our problem is exactly showing that $H_0 \approx H_n$. If they were distinguishable, then by the hybrid lemma, some $H_i \not\approx H_{i+1}$. However, I claim that this is not possible by efficient operations. Because the prover is efficient, we can efficiently simulate the entire protocol where the prover uses w_1 and output the view. We can do this i times. Likewise, we can efficiently simulate the entire protocol where the prover uses w_2 and output the view. We can do this $n - i - 1$ times. So concatenation by views of a WI protocol for a fixed witness is an efficient operation. Now observe that H_i is exactly $\{P(x, w_1) \leftrightarrow V^*(x, z)\}$ with i copies of the view of $\{P(x, w_1) \leftrightarrow V^*(x, z)\}$ pre-concatenated, and $n - i - 1$ copies of the view of $\{P(x, w_2) \leftrightarrow V^*(x, z)\}$ concatenated. Next, observe that H_{i+1} is exactly $\{P(x, w_2) \leftrightarrow V^*(x, z)\}$ with i copies of the view of $\{P(x, w_1) \leftrightarrow V^*(x, z)\}$ pre-concatenated, and $n - i - 1$ copies of the view of $\{P(x, w_2) \leftrightarrow V^*(x, z)\}$ concatenated. So because (P, V) is a WI protocol, $\{P(x, w_1) \leftrightarrow V^*(x, z)\} \approx P(x, w_2) \leftrightarrow V^*(x, z)\}$, and by efficient operations, $H_i \approx H_{i+1}$. So we cannot have any $H_i \not\approx H_{i+1}$, so we have $H_0 \approx H_n$, and (P^n, V^n) is also a WI protocol.