600.641 Special Topics in Theoretical Cryptography 01/29/07

Lecture 3: Zero-Knowledge Proofs

Instructor: Susan Hohenberger *Scribe: Kevin Snow*

1 Review

Last week we described interactive proofs, or proofs that use a wizard (prover) to convince a verifier of a truth. Two properties define interactive proofs:

1. Completeness - Any true statement can be proven.
2. Soundness - Any false statement cannot be proven.

Formally, these properties are defined as:

1. Completeness - $\forall x \in L,\ Pr[(P,V)[x] = accept] \geq 1 - negl(k)$
2. Soundness - $\forall P^* \forall x \notin L,\ Pr[(P^*,V)[x] = accept] \leq 1 - negl(k)$

2 Zero-Knowledgeness

There is an additional property that is necessary to make an interactive proof zero-knowledge. This property is called *zero-knowledgeness*.

2.1 Introduction

Consider the following example:

Prove a tube is in a class of hollow tubes.

1. Prover turns his back and places a coin in one side of the tube.
2. Verifier chooses one side of the tube.
3. Prover tilts the tube to the side chosen, the coin falls out.

This proof meets the desired properties of an interactive proof:

1. Completeness - suppose the tube is hollow.
   (a) Verifier will eventually be convinced after k trials.
2. Soundness - suppose the tube is not hollow.
   (a) Prover would have to guess which side the verifier will choose to tilt with probability $\frac{1}{2}$.
   (b) When repeated, the probability is negligable that the prover will guess correctly every time.

Can this proof be used to convince a third party? For example, the process could be video taped and watched by someone else – would they be convinced? No. The verifier and prover could have collaborated to create a fake video or the the video could have been tampered with. The proof is only meant to convince the verifier that the tube is hollow, not a third party.

Is this a zero-knowledge proof? Does it meet the zero-knowledgeness property? We start by giving an informal definition of zero-knowledgeness:

Zero-Knowledgeness - any verifier does not learn anything except that a statement is true.

The task is to formalize this notion of zero knowledge.

2.2 Formalizing the Definition

What does "not learn anything" mean? What does it really mean to "know" something? These questions must be answered to formally define the zero-knowledgeness property. As an example, consider that the verifier knows that $n = pq$, while given the values for n and p. Does the verfier "know" the value of q? Intuitively we would say yes, the verifier does know the value of q because it could be easily computed given the other known values, $q = \frac{n}{p}$. The intuition of knowledge needs to be captured in the definition; we use the notion of a $VIEW$:

- The $VIEW$ is a probability distribution over all $P, V^*(x)$; that is, over all possible conversations that P and V^* might have about x.
- $VIEW_{P,V^*(x)} =< messages\ from\ P,\ randomness\ of\ V^* >$

Given a probabalistic verifier, this definition encapsulates its whole view because the entire set of messages with the prover can be reconstructed. To achieve zero-knowledgeness we want the $VIEW$ to remain the same before and after communication with the prover. In other words, the view will only consist of messages the verifier could have simulated themselves. We want $VIEW_{P,V^*(x)} = S(x)$ where $S(x)$ is the view of a *simulator* that is capable of producing conversations selected from the same distribution as those of as $P, V^*(x)$ and, as with the verifier, is bound on probabilistic polynomial time. This gives our first formal notion of zero-knowledgeness:

$$\forall V^*_{ppt}\ \exists S_{ppt}\ \forall x \in L\ s.t.\ VIEW_{P,V^*}(x) = S(x)$$

This is known as the *Full/Code Access* definition for zero-knowledgeness because the simulator's actions can depend on the code of the verifier.

2.3 ZK Example

Consider the isomorphic graph example from last week:

Prove $(G, H) \in ISO$

Prover	**Message**	**Verifier**
knows $\phi(G) = H$		
pick random permutation π.	$C = \pi(G) \rightarrow$	
		pick G or H at rand., say H.
	$\leftarrow$ "H"	Show H is iso to C
if "G" is received, set $\alpha = \pi$		
if "H" is received, set $\alpha = (\pi \cdot \phi^{-1})$	$\alpha \rightarrow$	Check that $\alpha(H) = C$.

Recall that this proof meets the requirements for an interactive proof:

1. Completeness - the verifier will be convinced that the graphs are isomorphic.
2. Soundness - suppose G and H are not isomorphic.
   (a) Prover not be able to generate a C isomorphic to both G and H.
   (b) Prover has $\frac{1}{2}$ probability to choose same graph as the verifier.
   (c) Probability of successful cheating is $\frac{1}{2^k}$ after k attempts.

Additionally, we can show that this proof meets our zero-knowdgeness property by showing that $VIEW_{P,V^*(x)} = S(x)$:

1. We start by constructing the simulator:
   (a) Pick either graph G or H at random. (Suppose G is chosen.)
   (b) Choose a random permutation α.
   (c) Compute $C = \alpha(G)$.
   (d) Now, choose a random tape r for V^* and run it on input C.
   (e) If V^*, responds with "G", return α and record (C, α) as the messages from the "prover" and the random tape r as "V^*"'s randomness.
   (f) If V^* responds with "H", abort, don't record anything and start over at the beginning.

   Remarks:
   (a) Note that the simulator does not need to actually prove anything, it must just generate conversations that look like those between the prover and verifier.
   (b) Simulators are granted the extra power of generating messages in a different order than that of the actual messages.
   (c) Out of order messages allow the simulator to construct $C = \alpha(G)$.
2. Next, the simulator is analyzed:

(a) Simulator runs in *expected* polynomial time – each step of a single run is polynomial. Takes an expected two runs to get one good conversation.

(b) How do k trials of the actual runs compare with k trials of the simulated runs?

i. Simulator guesses G or H as V^* will with $\frac{1}{2}$ probability.

ii. No problem, throw out incorrect runs (and start over just repeating correct runs up to the run before the last abort). The claim is that it will now look like the long continuous view from before.

iii. Expected [trials for k conversations] $= 2k$ (since simulator is correct w/ $\frac{1}{2}$ probability).

2.4 A Stronger Definition of Zero-Knowledgeness

Let's consider our previous definition of zero-knowledgeness. Can we prove anything with it? It states that for every V^* there is a simulator, but there are so many V^*'s! So, now let's consider another definition – one that works with all V^*.

Alternative, stronger definition of zero-knowledgeness:

$$\exists S_{ppt}\, \forall V^*_{ppt}\, \forall x \in L \ s.t.\ VIEW_{P,V^*}(x) = S^{V^*}(x)$$

This is known as the *Black Box Access* definition of zero-knowledgeness. The earlier example meets this definition and most zero-knowledge proofs actually prove this stronger property, because it is easier to work with. Additionally, if this strong definition is satisfied, so is the weaker definition.

3 Tommorrow

- How happy are we with the definitions?
  - Does it capture everthing?
  - Is it too restrictive?
- A proof protocol is often run sequentially many times to decrease the soundness error, but how does that effect its' zero-knowledgeness?
- ZK $\subseteq$ IP, but how big is ZK?

References

[1] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design. In FOCS '86, pages 174-187, 1986.

[2] O. Goldreich, S. Micali, and A. Wigderson. How to prove all NP statements in zeroknowledge and a methodology of cryptographic protocol design. In A. M. Odlyzko, editor, Advances in Cryptology — CRYPTO '86, volume 263 of Lecture Notes in Computer Science, pages 171–185. Springer-Verlag, 1987

[3] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In STOC '85, pages 291-304, 1985.