CS 600.442 – Modern Cryptography October 12, 2016

Lecture 10: Zero Knowledge Proofs (II)

Instructor: Abhishek Jain *Scribe: Arka Rai Choudhuri*

In this class we will first establish the zero-knowledge property of the interactive proof for graph isomorphism discussed in the previous class. Later, we will prove that every language in NP has a zero-knowledge proof[1]. Along the way we will also define and construct commitment schemes.

1 Zero-knowledge Proof for Graph Isomorphism

We recall the definition of zero-knowledge from the last class.

Definition 1 (Zero-knowledge) *An interactive proof* (P, V) *for a language* L *with witness relation* R *is said to be zero-knowledge if for every non-uniform PPT adversary* V^**, there exists a PPT simulator* $\mathcal{S}$ *such that for every non-uniform PPT distinguisher* D*, there exists a negligible function* $\nu(\cdot)$ *such that for every* $x \in L, w \in R(x), z \in \{0,1\}^*$*,* D *distinguishes between the following distributions with probability at most* $\nu(|x|)$*:*

$$\Big\{\mathsf{View}_{V^*}[\mathsf{P}(x,w) \leftrightarrow \mathsf{V}^*(x,z)]\Big\} \textit{ and } \Big\{\mathcal{S}(1^n, x, z)\Big\}.$$

This definition is based on the computational indistinguishability of the two distributions, and is thus also referred to as *computational zero-knowledge.* In similar vein, we have two other variants of zero-knowledge:

- If the distributions are statistically close, then we call it *statistical zero-knowledge.*
- If the distributions are identical, then we call it *perfect zero-knowledge.*

In the last class we described the protocol for an interactive proof for graph isomorphism. This protocol is reproduced below:

Interactive proof for Graph Isomorphism

Repeat the following procedure n times using *fresh randomness*

$\underline{\mathsf{P}((G_0, G_1), \pi)}$ $\qquad\qquad\qquad\qquad$ $\underline{\mathsf{V}((G_0, G_1), z)}$

$\sigma \xleftarrow{\$} \Pi_n$

$H = \sigma(G_0)$

$$\xrightarrow{\quad H \quad}$$

$b \xleftarrow{\$} \{0, 1\}$

$$\xleftarrow{\quad b \quad}$$

If $b = 0,\ \varphi = \sigma$

Otherwise, $\varphi = \sigma \cdot \pi^{-1}$

$$\xrightarrow{\quad \varphi \quad}$$

Output 1 iff $H = \varphi(G_b)$

We already discussed that the protocol is an interactive proof by demonstrating its *completeness* and *soundness* properties. We now turn our attention to showing that it also satisfies the zero-knowledge property.

Note that the protocol is iterated n times for the *soundness* property. We will prove that a single iteration of the proof is *perfect zero-knowledge*. This extends to the full protocol from the following result:

Theorem 1 *Sequential repetition of any zero-knowledge protocol is also zero-knowledge.*

A proof sketch, which is skipped here, can be seen in section 7.2.1 of [3].

To prove a single iteration of the interactive proof is perfect zero-knowledge, we need to perform the following steps:

- Construct a simulator $\mathcal{S}$ for every PPT V^*.
- Prove that the expected run time of $\mathcal{S}$ is polynomial.
- Prove that the output distribution of $\mathcal{S}$ is indistinguishable from the real execution.

The simulator is defined below,

```
$\mathcal{S}(x, z)$
$b' \xleftarrow{\$} \{0,1\}, \sigma \xleftarrow{\$} \Pi_n$
$H = \sigma(G_{b'})$
Emulate execution of $V^*(x, z)$ by feeding it $H$. Let $b$ be its response.
If $b = b'$
  Feed $\sigma$ to $V^*$ and output its view.
Else
  Restart above procedure.
```

We briefly discuss the need to restart the procedure in the simulator. In the case that $b \neq b'$, the correct response would require the knowledge of π, which the simulator doesn't know (if it did, the adversary has all the 'knowledge' it needs). So the procedure is restarted with the hope that we will have $b = b'$ eventually.

The following lemma will aid us in performing the remaining two steps.

Lemma 2 *In the execution of $\mathcal{S}(x, z)$,*

- *H is identically distributed to $\sigma(G_0)$, and*
- $\Pr[b = b'] = \frac{1}{2}$

Proof. Since G_0 is isomorphic to G_1, for a random $\sigma \xleftarrow{\$} \Pi_n$, the distributions $\sigma(G_0)$ and $\sigma(G_1)$ are identically distributed. Thus, H has a distribution that is independent of b'. Therefore, H has the same distribution as $\sigma(G_0)$.

The simulator chooses b' independently from x and z. When we emulate the execution of $V^*(x, z)$ on feeding H, x, z and H (as argued earlier) are independent of b'. Thus the output of $V*$ will be independent of b'. Since b' is chosen at random, $\Pr[b = b'] = \frac{1}{2}$. ■

Run time: From the above lemma, we see that a single iteration of $\mathcal{S}$ has a success probability of $\frac{1}{2}$. Thus the expected number of iterations before $\mathcal{S}$ succeeds is 2. Since each iteration emulates a PPT adversary V^* in addition to some other polynomial time operations, it takes polynomial time. This in turn implies that the expected running time of $\mathcal{S}$ is polynomial.

Indistinguishability of Simulated View: The above lemma also shows that H has the same distribution as $\sigma(G_0)$. If we could always output σ, then the output distribution of $\mathcal{S}$ would match the distribution in the real execution. This is taken care of when we check if $b = b'$, and outputs H and σ only if it is true. But since H is independent of b', this does not change the output distribution. ■

2 Reflections on Zero Knowledge

The proof of zero-knowledge property using a simulator may seem a little paradoxical for the following reasons:

- Protocol execution convinces V of the validity of x.
- But V could have generate the protocol transcript on his own.

To understand why there is no paradox, consider the following story:

> *Alice and Bob run the above protocol on input (G_0, G_1) where Alice acts as* P *and Bob as* V*. Now, Bob goes to Eve and tells her that G_0 and G_1 are isomorphic. Eve is skeptical about what Bob knows and asks how he knew this to be true. Bob then shows her the accepting transcript. But Eve knows all about simulators and doesn't believe Bob. She tells him that anyone could have come up with the transcript without actually knowing the isomorphism. Bob is now annoyed, and persists by telling her that he computed the transcript talking to Alice, who answered his every query correctly. But Eve remains unmoved.*

The two most important points of the above story are:

- Bob participated in a "live" conversation with Alice, and was convinced *how* the transcript was generated.
- Eve on the other hand did not see the live conversation, and has no way to tell if the transcript is from a real execution or produced by a simulator.

Thus, zero-knowledge is about transcripts while soundness is about "live" executions because of the random challenges.

3 Zero-knowledge Proofs for NP

We now prove a powerful theorem assuming the existence of one-way permutations. The theorem essentially states that anything that can be proved (and verified efficiently), can also be proved in zero-knowledge. The formal statement is,

Theorem 3 *If one-way permutations exist, then every language in* NP *has a zero-knowledge interactive proof.*

Remark 1 *The assumption of one-way permutations in the above theorem can be relaxed to only one-way functions.*

Now, let us consider how we could prove the above theorem. Could we achieve this by constructing a zero-knowledge proof for each language in NP? That would be ridiculously inefficient. Instead we focus on NP-complete languages, and rely on their 'completeness' property.

Proof. The proof proceeds in two steps:

Step 1: Construct a zero-knowledge proof for an NP-complete language. We will consider *Graph 3-Coloring*, which is the language of all graphs whose vertices can be colored using only three colors such that no two connected vertices have the same color.

Step 2: To construct zero-knowledge proof for any NPlanguage L, do the following

- Given instance x and witness w, P and V reduce x into an instance x' of *Graph 3-Coloring* using Cook's deterministic reduction. The determinism ensures that both P and V end up with the same value x'.

- P also applies the reduction to the witness w to obtain witness w' for x'.
- Now, P and V can run the zero-knowledge proof from Step 1 on the common input x'.

We shall first show a "physical" proof. Here the colors are represented by numbers $\{1,2,3\}$. Let $\Pi_{\{1,2,3\}}$ denote the set of all permutations over $\{1,2,3\}$ and color_i refers to the 'color' of vertex $v_i \in V$ where $|V| = n$. We also define $\boxed{\mathsf{color}_i}$ to be a locked box containing color_i. As with any locked box, it has a key key_i which locks and unlocks it. Obviously, it should be hard, if not impossible, to open or view the contents of this locked box without the key (we assume the box is opaque). The physical interactive proof follows below,

Interactive proof for Graph Isomorphism

Repeat the following procedure $n|E|$ times using *fresh randomness*

$\underline{\mathsf{P}(G, (\mathsf{color}_1, \cdots, \mathsf{color}_n))}$ $\qquad\qquad$ $\underline{\mathsf{V}(G, z)}$

$\pi \overset{\$}{\leftarrow} \Pi_{\{1,2,3\}}$

$\forall i \in [n], \widetilde{\mathsf{color}_i} = \pi(\mathsf{color}_i)$

$\forall i \in [n],\ \widetilde{\mathsf{color}_i} \xrightarrow[\mathsf{key}_i]{\mathsf{lock}} \boxed{\widetilde{\mathsf{color}_i}}$

$$\xrightarrow{\left(\boxed{\widetilde{\mathsf{color}_i}}, \cdots, \boxed{\widetilde{\mathsf{color}_i}}\right)}$$

$(u, v) \overset{\$}{\leftarrow} E$

$$\xleftarrow{(u,v)}$$

$$\xrightarrow{\mathsf{key}_u, \mathsf{key}_v}$$

$\boxed{\widetilde{\mathsf{color}_u}} \xrightarrow[\mathsf{key}_u]{\mathsf{unlock}} \widetilde{\mathsf{color}_u}$

$\boxed{\widetilde{\mathsf{color}_v}} \xrightarrow[\mathsf{key}_v]{\mathsf{unlock}} \widetilde{\mathsf{color}_v}$

If $\widetilde{\mathsf{color}_u} \neq \widetilde{\mathsf{color}_v}$ Accept; else Reject

The completeness is trivial, and the intuitions for soundness follows from the fact that in each iteration, a cheating prover is caught with probability $\frac{1}{|E|}$ (we shall explain this later). For zero-knowledge, in each iteration, V only sees something it knew before - two random but different colors.

To "digitize" the above proof, we need some way to implement these locked boxes. Specifically, we need the two following properties about locked boxes:

- **Hiding:** V should not be able to see the contents inside a locked box.

- **Binding:** P should not be able to modify the content inside a box once it is locked.

Why do we even care about the second property? It's something that's so obvious about "physical" locked boxes that we often forget it exists. But this is what stops P from cheating when it doesn't know a correct coloring. If this property wasn't present, P on receiving (u, v) could modify the contents in the locked boxes containing the colorings of u and v to always unlock to different values. This would let P always convince V even without knowing the solution (coloring), thus violating the soundness requirement.

4 Commitment Schemes

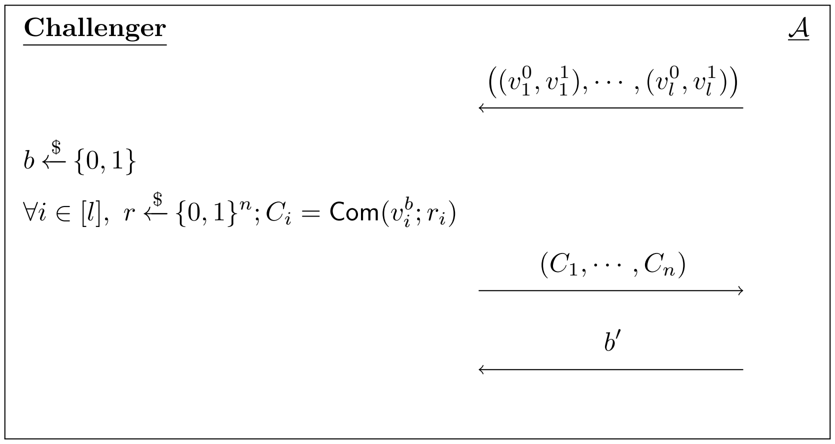
The digital analogue of 'locked' boxes contains of two phases: A **Commit phase:** where the sender locks a value v inside a box. And a **reveal phase:** where the sender unlocks the box and reveals v. This can be implemented using interactive protocols, but we will consider the non-interactive case where both commit and reveal phases will consist of a single message. We call the digital analogue "commitment schemes" and formally define them,

Definition 2 (Commitment) *A randomized polynomial-time algorithm* Com *is called a* **commitment scheme** *for n-bit strings if it satisfies the following properties:*

- **Binding:** *For all $v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^n$ it holds that*
$$\mathsf{Com}(v_0; r_0) \neq \mathsf{Com}(v_1; r_1).$$
- **Hiding:** *For every non-uniform PPT distinguisher D, there exists a negligible function $\nu(\cdot)$ such that for every $v_0, v_1 \in \{0,1\}^n$, D distinguishes between the following distribution with probability at most $\nu(n)$*
$$\left\{r \xleftarrow{\$} \{0,1\}^n : \mathsf{Com}(v_0; r)\right\} \text{ and } \left\{r \xleftarrow{\$} \{0,1\}^n : \mathsf{Com}(v_1; r)\right\}.$$

The above definition talks about *perfect binding* and *computational hiding.* Why don't we have *perfect binding* and *perfect hiding*? This is unfortunately impossible. It's left as an exercise to the reader to see why this is the case.

This definition only guarantees hiding for a single commitment. What about *multi-value hiding*? We sketch its definition here, similar to *multi-message secure* encryption schemes. Specifically, the adversary $\mathcal{A}$ needs to guess the bit b chosen by the challenger below.


Challenger
$\mathcal{A}$
$((v_1^0, v_1^1), \cdots, (v_l^0, v_l^1))$
$b \xleftarrow{\$} \{0,1\}$
$\forall i \in [l],\ r \xleftarrow{\$} \{0,1\}^n; C_i = \mathsf{Com}(v_i^b; r_i)$
$(C_1, \cdots, C_n)$
b'

For security we require that the adversary $\mathcal{A}$ has only a negligible advantage in guessing b. We now claim that any commitment scheme satisfies *multi-value hiding.* Like public-key encryption, commitment schemes do not have any 'key', and we follow the same technique. The formal proof is left as an exercise. The corollary to this claim is that one-bit commitment implies string commitment.

Construction. The following theorem shows us how to construct a bit commitment scheme based on one-way permutations. Other such constructions based on pseudorandom generators, among others, exist too[2].

Theorem 4 *If one-way permutations exist, then commitment schemes exist.*

Proof. Let f be a one-way permutation and h be the hard core predicate for f. We shall use these primitives to construct a bit-commitment scheme.

Commit phase: For this phase the sender computes $C = \mathsf{Com}(b; r) = (f(r), h(r) \oplus b)$. The bit b is being masked by $h(r)$.

Open phase: Sender reveals (b, r). Receiver accepts if $C = (f(r), h(r) \oplus b)$, and reject otherwise.

Binding follows from the fact that f is a permutation. Hence $f(r_1) = f(r_2) \Longleftrightarrow r_1 = r_2$. The hard core bit is deterministic once r is fixed, thus ensuring binding.

For *hiding* we follow the proof for proving a secure single bit encryption scheme based on trapdoor permutation. For us, the trapdoor makes no difference to the binding or hiding properties and thus follows immediately.

Now that we have our digital "locked boxes", we can proceed to our zero-knowledge proof for *Graph 3-coloring.*

5 Zero-knowledge Proof for Graph 3-coloring

The protocol for the interactive proof is presented below, Interactive proof for Graph Isomorphism

Repeat the following procedure $n|E|$ times using *fresh randomness*

$\underline{\mathsf{P}(G, (\mathsf{color}_1, \cdots, \mathsf{color}_n))}$ $\qquad\qquad$ $\underline{\mathsf{V}(G, z)}$

$\pi \xleftarrow{\$} \Pi_{\{1,2,3\}}$

$\forall i \in [n], \widetilde{\mathsf{color}}_i = \pi(\mathsf{color}_i)$

$\forall i \in [n],\ C_i = \mathsf{Com}(\widetilde{\mathsf{color}}_i;\ r_i)$

$$\xrightarrow{(C_1, \cdots, C_n)}$$

$(u, v) \xleftarrow{\$} E$

$$\xleftarrow{(u, v)}$$

$$\xrightarrow{(\widetilde{\mathsf{color}}_u;\ r_u), (\widetilde{\mathsf{color}}_v;\ r_v)}$$

If C_u, C_v are valid
and $\widetilde{\mathsf{color}}_u \neq \widetilde{\mathsf{color}}_v$ Accept
else Reject

The completeness trivially follows from the fact that knowledge of the right coloring ensures that the prover never fails. We need to prove the soundness and zero-knowledge for for the interactive proof.

Proof of Soundness:

Let G be the graph that is not 3-colorable. Then any coloring $\mathsf{color}_1, \cdots, \mathsf{color}_n$ will have at least one edge which have the same colors on both endpoints. Let one such edge be (i^*, j^*). From the binding property of Com, we know that $C_1, \cdots, C_n$ have unique openings $\widetilde{\mathsf{color}}_1 \cdots, \widetilde{\mathsf{color}}_n$. In each iteration P chooses $(u, v) = (i^*, j^*)$ with probability $\frac{1}{|E|}$. There might be other such edges too, but we take the pessimistic approach of assuming only one such edge. The cheating P only succeeds if it is able to cheat in every iteration. Therefore, the probability P successfully cheats in every one of the $n|E|$ iterations is at most:

$$\left(1 - \frac{1}{|E|}\right)^{n|E|} \approx e^{-n}.$$

As required, this satisfies the soundness requirement.

We are left with the proof for zero-knowledge, which we shall complete in the next class.

References

[1] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[2] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[3] Rafael Pass and Abhi Shelat. A Course In Cryptography, 2010. `http://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf`.