Lecture A.5

Zero-Knowledge IPs

Summer Graduate School on
Foundations and Frontiers of Probabilistic Proofs
2021.07.30

Zero Knowledge

Benefits of interaction and randomness so far:

- capture many languages beyond NP (coNP, $P^{\#P}$, PSPACE)
- delegate certain classes of computations (bounded depth circuits)

Today we learn about an additional benefit: ZERO KNOWLEDGE.

Informally, this means that we want to protect the privacy of the prover, by designing interactive proofs that do not "give away" <u>why</u> a statement is true.

We will illustrate this notion by focusing on the language of graph isomorphism. Recall that $GI = \{(G_0, G_1) \mid G_0 \equiv G_1\}$ is in NP (the NP proof is the isomorphism) and GI is not known to be in BPP.

The "trivial" interactive proof is for the prover to send an isomorphism between G_0 and G_1.

New challenge: what if we consider the isomorphism a private input of the prover?

Namely, how do we design an alternative IP for GI that is still complete and sound and yet reveals no information about any isomorphism between G_0 and G_1?

An Alternative IP for Graph Isomorphism

$\sigma : [n] \to [n]$
s.t. $G_0 = \sigma(G_1)$

$P((G_0, G_1))$		$V((G_0, G_1))$
sample random permutation $\phi : [n] \to [n]$ and set $H := \phi(G_0)$	$\xrightarrow{H}$	
	$\xleftarrow{b}$	$b \leftarrow \{0,1\}$
$\Psi := \phi \circ \sigma^b$	$\xrightarrow{\Psi}$	$H \overset{?}{=} \Psi(G_b)$

We first argue that this is an interactive proof for GI:

- <u>completeness</u>: Suppose $(G_0, G_1) \in GI$, and that $G_0 = \sigma(G_1)$.
  Then, $\forall b \in \{0,1\}$, $H \overset{?}{=} \Psi(G_b) \iff H \overset{?}{=} (\phi \circ \sigma^b)(G_b) \iff H \overset{?}{=} \phi(G_0)$.
- <u>soundness</u>: Suppose $(G_0, G_1) \notin GI$.
  Then H can be isomorphic to at most one of G_0 or G_1.
  So any malicious prover gets caught w.p. $\geq 1/2$.

Zero Knowledge against Honest Verifiers

An interactive proof (P,V) for L is honest-verifier zero knowledge if

$\exists$ probabilistic polynomial-time simulator S such that

$$\forall x \in L \quad S(x) \equiv \mathrm{View}_V(\langle P,V\rangle(x))$$

Here $\mathrm{View}_V(\langle P,V\rangle(x)) := (r, x, a_1, \ldots, a_k)$ is all information seen by V: its randomness r, its input x, and the prover's messages $a_1, \ldots, a_k$.

Interpretation:

The honest verifier could have simulated the whole interaction by himself without talking to the honest prover.
The simulator formalizes this by sampling in polynomial time the view of the honest verifier.

Note: HVZK is a joint property of honest prover P & honest verifier V

Honest-Verifier ZK for Graph Isomorphism

<u>claim:</u> (P,V) is honest-verifier ZK

$P((G_0,G_1),\sigma)$ $\qquad\qquad$ $V((G_0,G_1))$

sample random permutation $\phi:[n]\to[n]$ and set $H:=\phi(G_0)$

$\xrightarrow{\quad H \quad}$

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

$\xleftarrow{\quad b \quad}$

$\psi:=\phi\circ\sigma^{b}$ $\xrightarrow{\quad \psi \quad}$ $H\stackrel{?}{=}\psi(G_b)$

<u>proof:</u> Fix $(G_0,G_1)\in GI$.

The honest verifier's view consists of

$((G_0,G_1),H,b,\psi)$ where $H=\phi(G_0)$ for random $\phi:[n]\to[n]$, $b\in\{0,1\}$ is random, and ψ is random such that $H=\psi(G_b)$.

Consider the following probabilistic polynomial-time algorithm:

$S((G_0,G_1)):=$
1. sample $b\leftarrow\{0,1\}$
2. sample random $\psi:[n]\to[n]$
3. compute $H:=\psi(G_b)$
4. output $((G_0,G_1),H,b,\psi)$.

Since $G_0\equiv G_1$ the two tuples are equidistributed. ∎

Zero Knowledge against Malicious Verifiers

An interactive proof (P,V) for L is (malicious-verifier) zero knowledge if

$\exists$ probabilistic polynomial time (in expectation) simulator S such that

$$\forall x \in L \ \forall \text{ ppt } \tilde{V} \quad S(\tilde{V}, x) \equiv \mathrm{View}_{\tilde{V}}(\langle P, \tilde{V} \rangle(x))$$

Interpretation: even verifiers that deviate from the prescribed protocol cannot learn any information besides the bit "$x \in L$".

Note: (malicious-verifier) ZK is a property of the honest prover alone.

Compare with: completeness (P&V), soundness (V alone), HVZK (P&V).

Note: consider running time of simulator in expectation because it is a useful (and still meaningful) relaxation

Note: other flavors of definition are also used, e.g., one can modify the requirement on the simulator to "$\forall$ ppt $\tilde{V}$ $\exists$ eppt S $\forall x \in L$ $S(x) \equiv \mathrm{View}_{\tilde{V}}(\langle P, \tilde{V} \rangle(x))$".

Malicious-Verifier ZK for Graph Isomorphism

$P((G_0,G_1),\sigma)$ | $V((G_0,G_1))$

sample random permutation $\phi:[n]\to[n]$ and set $H:=\phi(G_0)$

$\xrightarrow{H}$

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

$\xleftarrow{b}$

$\psi:=\phi\circ\sigma^{-b}$ $\xrightarrow{\psi}$ $H\overset{?}{=}\psi(G_b)$

claim: (P,V) is malicious-verifier ZK

proof: Fix $(G_0,G_1)\in GI$ and ppt $\tilde{V}$.

The malicious verifier's view consists of

$((G_0,G_1),H,\tilde{b},\psi)$ where $H=\phi(G_0)$ for random $\phi:[n]\to[n]$, $\tilde{b}$ is whatever $\tilde{V}$ outputs given H, and ψ is random such that $H=\psi(G_{\tilde{b}})$.

Consider the following probabilistic algorithm:

$S(\tilde{V},(G_0,G_1)):=$
1. sample random $b\in\{0,1\}$
2. sample random $\psi:[n]\to[n]$
3. compute $H:=\psi(G_b)$
4. give H to $\tilde{V}$ and get $\tilde{b}$
5. if $\tilde{b}\neq b$ go to $\boxed{1}$
6. output $((G_0,G_1),H,\tilde{b},\psi)$

Since $G_0\equiv G_1$, H is independent of b, and so is $\tilde{b}$.
Hence $\Pr[\tilde{b}=b]=1/2$, and $\mathbb{E}[\#\text{rewinds}]=2$, so S runs in expected polynomial time.
Also, S works since it is doing rejection sampling:

$$\Pr[\tilde{b}=0\mid\tilde{b}=b]=\frac{\Pr[\tilde{b}=0\wedge\tilde{b}=b]}{\Pr[\tilde{b}=b]}=\frac{\Pr[\tilde{b}=0]\cdot 1/2}{1/2}=\Pr[\tilde{b}=0].$$

Note: S only makes black-box use of malicious verifier

Limitations of Zero Knowledge

What happens more generally?

def: • HVZK-IP to be all languages having IPs with honest-verifier ZK
• (MV)ZK-IP to be all languages having IPs with malicious-verifier ZK

It is straightforward to see that $(MV)ZK\text{-}IP \subseteq HVZK\text{-}IP \subseteq IP$.
And also that $BPP \subseteq (MV)ZK\text{-}IP$, as the simulator has nothing to do.
We have already established that $GI \in (MV)ZK\text{-}IP$.

What other languages have zero knowledge interactive proofs?

theorem: $HVZK\text{-}IP \subseteq AM \cap coAM$

In particular, we do not expect e.g. NP to have (even HV) zero knowledge IPs.

The limitation holds even if we relax the requirement on the (even HV) simulator, to require that $S(x)$ and $View_V(\langle P, V\rangle(x))$ are statistically close rather than equal.

Overcoming Limitations of Zero Knowledge

We are still interested in zero knowledge proofs for NP (and more!).
What do we do?

<u>Option #1:</u> relax requirement on the simulator further to

$S(\tilde{V}, x)$ and $\mathrm{View}_{\tilde{V}}(\langle P, \tilde{V}\rangle(x))$ are computationally close

This leads to corresponding complexity classes, HV CZK-IP & (MV) CZK-IP.

<u>theorem:</u> if OWFs exist then (MV) CZK-IP = IP ← very strong result!

<u>Option #2:</u> consider a different model of probabilistic proof

We will see various other models of probabilistic proof in this course, and each of them behaves differently with respect to zero knowledge.

For now we informally mention one result about multi-prover interactive proofs:

<u>theorem:</u> PZK-MIP = MIP

intuitively, cryptography is replaced by a "physical assumption" (two provers can't communicate)

Some Intuition on the Limits of HVZK-IP

Lemma: if $L \in \text{HVZK-IP}[k]$ then $\bar{L} \in IP[O(k)]$

Suppose that (P,V) is HVZK-IP for L and let S be the simulator.
We know that $\forall x \in L \quad S(x) \equiv View_V(\langle P,V \rangle(x))$.

What does $S(x)$ do if $x \notin L$?

① $S(x)$ outputs a view that is rejecting (e.g. garbage or fails some check of V)

② $S(x)$ outputs a view that is accepting

In ① we can efficiently tell that $x \notin L$ so we don't expect this if $L \notin BPP$.
So for languages not in BPP we will be (almost always) in ②.

Observation: in ② it MUST be that $S(x)$ is not close (in statistical distance) to $View_V(\langle P,V \rangle(x))$ because this latter is whp a rejecting transcript.

This can be used by $\bar{V}$ for $\bar{L}$ by sampling a view from $S(x)$ and asking $\bar{P}$ to prove that this sample is not from the distribution when $x \in L$.

Example: from HVZK-IP for GI to IP for GNI

Consider the zero knowledge IP for GI and its honest-verifier simulator:

$P_{GI}((G_0, G_1), \sigma)$ | $V_{GI}((G_0, G_1))$

sample random permutation $\phi : [n] \to [n]$ and set $H := \phi(G_0)$

$\xrightarrow{\quad H \quad}$

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

$\xleftarrow{\quad b \quad}$

$\Psi := \phi \circ \sigma^b$

$\xrightarrow{\quad \Psi \quad}$

$H \stackrel{?}{=} \Psi(G_b)$

$S((G_0, G_1)) :=$
1. sample $b \leftarrow \{0,1\}$
2. sample random $\Psi : [n] \to [n]$
3. compute $H := \Psi(G_b)$
4. output $((G_0, G_1), H, b, \Psi)$.

We have already analyzed that if $G_0 \equiv G_1$ then $S((G_0, G_1)) \equiv \mathrm{View}_V(\langle P, V \rangle(x))$.
If $G_0 \not\equiv G_1$ then $S((G_0, G_1))$ still outputs accepting views but with a different distribution.

We can use this to recover the protocol for GNI (the complement of GI)!

$P_{GNI}((G_0, G_1))$ | $V_{GNI}((G_0, G_1))$

$b \leftarrow \{0,1\}$
$\Psi \leftarrow$ (permutations on V)
$H := \Psi(G_b)$

$\xleftarrow{\quad H \quad}$

choose $\tilde{b}$ s.t. H is in equiv class of $G_{\tilde{b}}$

$\xrightarrow{\quad \tilde{b} \quad}$

$\tilde{b} \stackrel{?}{=} b$

the verifier for GNI is running the simulator for GI and challenging the prover to show that the distribution is different from the case when $G_0 \equiv G_1$ (thus proving that $G_0 \not\equiv G_1$) by asking for the bit b.