

Foundation of Cryptography, Lecture 6

Interactive Proofs and Zero Knowledge

Handout Mode

Benny Applebaum & Iftach Haitner, Tel Aviv University

Tel Aviv University.

December 22, 2016

Part I

Interactive Proofs

\mathcal{NP} as a Non-interactive Proofs

Definition 1 (\mathcal{NP})

$\mathcal{L} \in \mathcal{NP}$ iff \exists and poly-time algorithm V such that:

- $\forall x \in \mathcal{L}$ there exists $w \in \{0, 1\}^*$ s.t. $V(x, w) = 1$
- $V(x, w) = 0$ for every $x \notin \mathcal{L}$ and $w \in \{0, 1\}^*$

Only $|x|$ counts for the running time of V .

A proof system

- Efficient verifier, efficient prover (given the witness)
- Soundness holds **unconditionally**

Interactive proofs

Protocols between **efficient** verifier and **unbounded** provers.

Definition 2 (Interactive proof)

A protocol (P, V) is an **interactive proof** for \mathcal{L} , if V is PPT and:

Completeness $\forall x \in \mathcal{L}, \Pr[\langle (P, V)(x) \rangle_V = 1] \geq 2/3$.^a

Soundness $\forall x \notin \mathcal{L}$, and **any** algorithm P^*
 $\Pr[\langle (P^*, V)(x) \rangle_V = 1] \leq 1/3$.

IP is the class of languages that have interactive proofs.

^a $\langle (A(a), B(b))(c) \rangle_B$ denote B 's view in random execution of $(A(a), B(b))(c)$.

- **IP = PSPACE!**
- We typically consider (and achieve) perfect completeness.
- Negligible "soundness error" achieved via repetition.
- Sometime we have efficient provers via "auxiliary input".
- Relaxation: *Computationally sound proofs* [also known as, *interactive arguments*]: soundness only guaranteed against **efficient** (PPT) provers.

Section 1

Interactive Proof for Graph Non-Isomorphism

Graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are **isomorphic**, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.

- $\mathcal{GI} = \{(G_0, G_1) : G_0 \equiv G_1\} \in \mathcal{NP}$
- Does $\mathcal{GNI} = \{(G_0, G_1) : G_0 \not\equiv G_1\} \in \mathcal{NP}$?
- We will show a simple interactive proof for \mathcal{GNI}
Idea: Beer tasting...

Interactive proof for $\mathcal{GN}\mathcal{I}$

Protocol 4 ((P, V))

Common input: $G_0 = ([m], E_0), G_1 = ([m], E_1)$.

- 1 V chooses $b \leftarrow \{0, 1\}$ and $\pi \leftarrow \Pi_m$, and sends $\pi(E_b)$ to P.^a
- 2 P send b' to V (tries to set $b' = b$).
- 3 V accepts iff $b' = b$.

$$^a \pi(E) = \{(\pi(u), \pi(v)) : (u, v) \in E\}.$$

Claim 5

The above protocol is **IP** for $\mathcal{GN}\mathcal{I}$, with perfect completeness and soundness error $\frac{1}{2}$.

Proving Claim 5

- Graph isomorphism is an equivalence relation (separates the set of all graph pairs into separate subsets)
- $([m], \pi(E_i))$ is a random element in $[G_i]$ — the equivalence class of G_i

Hence,

$$G_0 \equiv G_1: \Pr[b' = b] \leq \frac{1}{2}.$$

$$G_0 \not\equiv G_1: \Pr[b' = b] = 1 \text{ (i.e., } P \text{ can, possibly inefficiently, extracted from } \pi(E_i))$$

□

Part II

Zero knowledge Proofs

Where is Waldo?



Question 6

Can you prove you know where Waldo is **without** revealing his location?

The concept of zero knowledge

- Proving w/o revealing any additional information.
- What does it mean?
Simulation paradigm.

Distribution ensembles, revisited

We will consider distribution ensembles indexed by arbitrary sets.

Let $\mathcal{L} \subseteq \{0, 1\}^*$, and let $P = \{P_x\}_{x \in \mathcal{L}}$ and $Q = \{Q_x\}_{x \in \mathcal{L}}$ be two distribution ensembles.

P is computationally indistinguishable from Q , denoted $P \approx_c Q$, means that

$$\left| \Pr_{y \leftarrow P_x} [D(1^{|x|}, y) = 1] - \Pr_{y \leftarrow Q_x} [D(1^{|x|}, y) = 1] \right| \leq \text{neg}(|x|)$$

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

An interactive proof (P, V) is **computational zero-knowledge proof (CZK)** for $\mathcal{L} \in \mathcal{NP}$, if \forall PPT V^* , \exists PPT S (i.e., simulator) such that

$$\{ \langle (P(w(x)), V^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{ S(x) \}_{x \in \mathcal{L}}. \quad (1)$$

for any poly-bounded function w with $w(x) \in R_{\mathcal{L}}(x)$.

Perfect ZK (PZK)/statistical ZK (SZK) — the above distributions are identically/statistically close.

- 1 ZK is a property of the **prover**.
- 2 ZK only required to hold wrt. **true** statements.
- 3 Trivial to achieve for $\mathcal{L} \in \mathcal{BPP}$.
- 4 The \mathcal{NP} proof system is typically **not** zero knowledge.
- 5 Meaningful also for languages outside \mathcal{NP} .
- 6 Auxiliary input (will give formal def later)

Zero-knowledge proofs, cont.

- 1 ZK for **honest verifiers**: (1) only holds for $V^* = V$.
- 2 We sometimes assume for notational convenient, and wlg, that a cheating V^* **outputs** its view.
- 3 **Statistical** ZK proofs are believed to to exists only for a restricted subclass of \mathcal{NP} , so to go beyond that we settle for **computational** ZK (as in this course). or for arguments.

Section 2

Zero-Knowledge Proof for Graph Isomorphism

Zero-knowledge proof for \mathcal{GI}

Idea: route finding

Protocol 8 ((P, V))

Common input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

P's input: a permutation π over $[m]$ such that $\pi(E_1) = E_0$.

- 1 P chooses $\pi' \leftarrow \Pi_m$ and sends $E = \pi'(E_0)$ to V.
- 2 V sends $b \leftarrow \{0, 1\}$ to P.
- 3 If $b = 0$, P sets $\pi'' = \pi'$, otherwise, it sends $\pi'' = \pi' \circ \pi$ to V.
- 4 V accepts iff $\pi''(E_b) = E$.

Claim 9

Protocol 8 is a \mathcal{SZK} for \mathcal{GI} , with perfect completeness and soundness $\frac{1}{2}$.

Proving Claim 9

- Completeness: Clear
- Soundness: If exist $j \in \{0, 1\}$ for which $\nexists \pi' \in \Pi_m$ with $\pi'(E_j) = E$, then V rejects w.p. at least $\frac{1}{2}$.

Assuming V rejects w.p. less than $\frac{1}{2}$ and let π_0 and π_1 be the values guaranteed by the above observation (i.e., mapping E_0 and E_1 to E respectively).

Then $\pi_0^{-1}(\pi_1(E_1)) = \pi_0 \implies (G_0, G_1) \in \mathcal{GI}$.

- \mathcal{ZK} : Idea – for $(G_0, G_1) \in \mathcal{GI}$, it is easy to generate a random transcript for Steps 1–2, and to be able to open it with prob $\frac{1}{2}$.

The simulator

For a start, consider a deterministic cheating verifier V^* that never aborts.

Algorithm 10 (S)

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

Do $|x|$ times:

- 1 Choose $b' \leftarrow \{0, 1\}$ and $\pi \leftarrow \Pi_m$, and “send” $\pi(E_{b'})$ to $V^*(x)$.
- 2 Let b be V^* 's answer. If $b = b'$, send π to V^* , output V^* 's view and halt. Otherwise, **rewind** V^* to its initial step, and go to step 1.

Abort.

Claim 11

$$\{ \langle (P, V^*)(x) \rangle_{V^*} \}_{x \in GI} \approx \{ S(x) \}_{x \in GI}$$

Claim 11 implies that **Protocol 8** is zero knowledge.

Proving Claim 11

Consider the following inefficient simulator:

Algorithm 12 (S')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$.

Do $|x|$ times:

- 1 Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $V^*(x)$.
- 2 Let b be V^* 's answer.

W.p. $\frac{1}{2}$,

- 1 Find π' such that $E = \pi'(E_b)$, and send it to V^* .
- 2 Output V^* 's view and halt.

Otherwise, rewind V^* to its initial step, and go to step 1.

Abort.

Claim 13

$S(x) \equiv S'(x)$ for any $x \in \mathcal{GI}$.

Proof: ?

Proving Claim 11 cont.

Consider a second inefficient simulator:

Algorithm 14 (S'')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

- 1 Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $V^*(x)$.
- 2 Find π' such that $E = \pi'(E_b)$ and send it to V^*
- 3 Output V^* 's view and halt.

Claim 15

$\forall x \in \mathcal{GI}$ it holds that

- 1 $\langle (P, V^*(x)) \rangle_{V^*} \equiv S''(x)$.
- 2 $SD(S''(x), S'(x)) \leq 2^{-|x|}$.

Proof: ? (1) is clear.

Proving Claim 15(2)

Fix $t \in \{0, 1\}^*$ and let $\alpha = \Pr_{S''(x)}[t]$.

It holds that

$$\begin{aligned}\Pr_{S'(x)}[t] &= \alpha \cdot \sum_{i=1}^{|x|} \left(1 - \frac{1}{2}\right)^{i-1} \cdot \frac{1}{2} \\ &= (1 - 2^{-|x|}) \cdot \alpha\end{aligned}$$

Hence, $SD(S''(x), S'(x)) \leq 2^{-|x|} \square$

Remarks

- 1 Perfect \mathcal{ZK} for “expected polynomial-time” simulators.
- 2 Aborting verifiers.
- 3 Randomized verifiers.
 - 1 The simulator first **fixes** the coins of V^* at random.
 - 2 Same proof goes through.
- 4 Negligible soundness error?
 - 1 Amplify by repetition
 - 2 But what about the ZK?

“Transcript simulation” might not suffice!

Let (G, E, D) be a public-key encryption scheme and let $\mathcal{L} \in \mathcal{NP}$.

Protocol 16 $((P, V))$

Common input: $x \in \{0, 1\}^*$

P’s input: $w \in R_{\mathcal{L}}(x)$

- 1 V chooses $(d, e) \leftarrow G(1^{|x|})$ and sends e to P
- 2 P sends $c = E_e(w)$ to V
- 3 V accepts iff $D_d(c) \in R_{\mathcal{L}}(x)$

- The above protocol has perfect completeness and soundness.
- Is it zero-knowledge?
- It has “transcript simulator” (at least for honest verifiers): exists PPT S such that $\{ \langle (P(w \in R_{\mathcal{L}}(x)), V)(x) \rangle_{\text{trans}} \}_{x \in \mathcal{L}} \approx_c \{ S(x) \}_{x \in \mathcal{L}}$,
where **trans** stands for the transcript of the protocol (i.e., the **messages** exchange through the execution).

Section 3

Composition of Zero-Knowledge Proofs

Is zero-knowledge maintained under composition?

- Sequential repetition?
- Parallel repetition?

Zero-knowledge proof, auxiliary input variant

Definition 17 (zero-knowledge proofs, auxiliary input)

An interactive proof (P, V) is **auxiliary-input** computational zero-knowledge proof (\mathcal{CZK}) for $\mathcal{L} \in \mathcal{NP}$, if \forall **deterministic** poly-time V^* , \exists PPT S s.t.

$$\{ \langle (P(w(x)), V^*(z(x)))(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{ S(x, z(x)) \}_{x \in \mathcal{L}}.$$

for any poly-bounded functions w with $w(x) \in R_{\mathcal{L}}(x)$ and $z: \mathcal{L} \mapsto \{0, 1\}^*$.

Perfect \mathcal{ZK} (\mathcal{PZK})/statistical auxiliary-input \mathcal{ZK} (\mathcal{SZK}) — the above distributions are identically/statistically close.

- Strengthening of the standard definition.
- The protocol for \mathcal{GI} we just saw, is also auxiliary-input \mathcal{SZK}
- What about randomized verifiers?
- Necessary for proving that zero-knowledge proof compose sequentially.
- To keep things simple, we will typically prove the non-auxiliary zero-knowledge, but all proofs we present can easily modified to achieve the stronger auxiliary input variant.

Is zero-knowledge maintained under composition?, cont.

- Auxiliary-input zero-knowledge is maintained under **sequential** repetition.
- Zero-knowledge might not maintained under **parallel** repetition.

Examples:

- ▶ Chess game
- ▶ Signature game

Section 4

Black-box Zero Knowledge

Black-box simulators

Definition 18 (Black-box simulator)

(P, V) is \mathcal{CZK} with **black-box simulation** for $\mathcal{L} \in \mathcal{NP}$, if \exists **oracle-aided** PPT S s.t.

$$\{ \langle (P(w(x)), V^*(z(x)))(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{ S^{V^*}(x, z(x))(x) \}_{x \in \mathcal{L}}$$

for any deterministic polynomial-time V^* , any w with $w(x) \in R_{\mathcal{L}}(x)$ and any $z: \mathcal{L} \mapsto \{0, 1\}^*$.

Perfect and statistical variants are defined analogously.

- 1 “Most simulators” are black box
- 2 Strictly **weaker** than general simulation!

Section 5

Zero-knowledge proofs for all NP

\mathcal{CZK} for 3COL

- Assuming OWFs exists, we give a (black-box) \mathcal{CZK} for 3COL .
- We show how to transform it for any $\mathcal{L} \in \mathcal{NP}$ (using that 3COL $\in \mathcal{NPC}$).

Definition 19 (3COL)

$G = (M, E) \in 3COL$, if $\exists \phi: M \mapsto [3]$ s.t. $\phi(u) \neq \phi(v)$ for every $(u, v) \in E$.

We use [commitment schemes](#).

The protocol

Let π_3 be the set of all permutations over $[3]$. We use perfectly binding commitment $\text{Com} = (\text{Snd}, \text{Rcv})$.

Protocol 20 ((P, V))

Common input: Graph $G = (M, E)$ with $n = |G|$

P's input: a (valid) coloring ϕ of G

- 1 P chooses $\pi \leftarrow \Pi_3$ and sets $\psi = \pi \circ \phi$
- 2 $\forall v \in M$: P commits to $\psi(v)$ using Com (with security parameter 1^n).
Let c_v and d_v be the resulting commitment and decommitment.
- 3 V sends $e = (u, v) \leftarrow E$ to P
- 4 P sends $(d_u, \psi(u)), (d_v, \psi(v))$ to V
- 5 V verifies that
 - 1 Both decommitments are valid,
 - 2 $\psi(u), \psi(v) \in [3]$, and
 - 3 $\psi(u) \neq \psi(v)$.

Claim 21

The above protocol is a \mathcal{CZK} for 3COL , with perfect completeness and soundness $1/|E|$.

- Completeness: Clear
- Soundness: Let $\{c_v\}_{v \in M}$ be the commitments resulting from an interaction of V with an arbitrary P^* .

Define $\phi: M \mapsto [3]$ as follows:

$\forall v \in M$: let $\phi(v)$ be the (single) value that it is possible to decommit c_v into (if not in $[3]$, set $\phi(v) = 1$).

If $G \notin 3\text{COL}$, then $\exists (u, v) \in E$ s.t. $\psi(u) \neq \psi(v)$.

Hence, V rejects such x w.p. at least $1/|E|$.

Proving \mathcal{ZK}

Fix a deterministic, non-aborting V^* that gets no auxiliary input.

Algorithm 22 (S)

Input: A graph $G = (M, E)$ with $n = |G|$

Do $n \cdot |E|$ times:

- 1 Choose $e' = (u, v) \leftarrow E$.
 - 1 Set $\psi(u) \leftarrow [3]$,
 - 2 Set $\psi(v) \leftarrow [3] \setminus \{\psi(u)\}$, and
 - 3 Set $\psi(w) = 4$ for $w \in M \setminus \{u, v\}$.
- 2 $\forall v \in M$: commit to $\psi(v)$ to V^* (resulting in c_v and d_v)
- 3 Let e be the edge sent by V^* .
If $e = e'$, send $(d_u, \psi(u)), (d_v, \psi(v))$ to V^* , output V^* 's view and halt.
Otherwise, **rewind** V^* to its initial step, and go to step 1.

Abort.

Proving \mathcal{ZK} cont.

Algorithm 23 (\tilde{S})

Input: $G = (V, E)$ with $n = |G|$, and a (valid) coloring ϕ of G .

Do for $n \cdot |E|$ times:

- 1 Choose $e' \leftarrow E$.
- 2 Act like the honest prover does given private input ϕ .
- 3 Let e be the edge sent by V^* . If $e = e'$
 - 1 Send $(\psi(u), d_u), (\psi(v), d_v)$ to V^* ,
 - 2 Output V^* 's view and halt.

Otherwise, **rewind** V^* to its initial step, and go to step 1.

Abort.

Claim 24

$\{ \langle (P(w(x)), V^*)(x) \rangle_{V^*} \}_{x \in 3\text{COL}} \approx \{ \tilde{S}^{V^*(x)}(x, w(x)) \}_{x \in 3\text{COL}}$,
for any w with $w(x) \in R_{\mathcal{L}}(x)$.

Proof: ?

Proving \mathcal{ZK} cont..

Claim 25

$\{S^{V^*(x)}(x)\}_{x \in 3\text{COL}} \approx_c \{\tilde{S}^{V^*(x)}(x, w(x))\}_{x \in 3\text{COL}}$, for any w with $w(x) \in R_{\mathcal{L}}(x)$.

Proof: Assume \exists PPT D , $p \in \text{poly}$, $w(x) \in R_{\mathcal{L}}(x)$ and an infinite set $\mathcal{I} \subseteq 3\text{COL}$ s.t.

$$\Pr \left[D(S^{V^*(x)}(x)) = 1 \right] - \Pr \left[D(\tilde{S}^{V^*(x)}(x, w(x))) = 1 \right] \geq \frac{1}{p(|x|)}$$

for all $x \in \mathcal{I}$.

Hence, \exists PPT R^* and $b \in [3]$ such that

$$\begin{aligned} & \Pr \left[\left\langle (Snd(4), R^*(x, w(x))) (1^{|x|}) \right\rangle_{R^*} = 1 \right] - \Pr \left[\left\langle (Snd(b), R^*(x, w(x))) (1^{|x|}) \right\rangle_{R^*} = 1 \right] \\ & \geq \frac{1}{|x| \cdot p(|x|)} \end{aligned}$$

for all $x \in \mathcal{I}$.

In contradiction to the (non-uniform) security of Com .

Remarks

- Aborting verifiers
- Auxiliary inputs
- Soundness amplification

Extending to all \mathcal{NP}

For $\mathcal{L} \in \mathcal{NP}$, let Map_X and Map_W be two poly-time computable functions s.t.

- $x \in \mathcal{L} \iff \text{Map}_X(x) \in 3\text{COL}$
- $w \in R_{\mathcal{L}}(x) \iff \text{Map}_W(w) \in R_{3\text{COL}}(\text{Map}_X(x))$.

Let (P, V) be a \mathcal{CZK} for 3COL with black-box simulation.

Protocol 26 $((P_{\mathcal{L}}, V_{\mathcal{L}}))$

Common input: $x \in \{0, 1\}^*$.

$P_{\mathcal{L}}$'s input: $w \in R_{\mathcal{L}}(x)$.

- 1 The two parties interact in $(P(\text{Map}_W(w)), V)(\text{Map}_X(x))$, where $P_{\mathcal{L}}$ and $V_{\mathcal{L}}$ taking the role of P and V respectively.
- 2 $V_{\mathcal{L}}$ accepts iff V accepts in the above execution.

Claim 27

$(P_{\mathcal{L}}, V_{\mathcal{L}})$ is a \mathcal{CZK} for \mathcal{L} with the same completeness and soundness as (P, V) as for 3COL .

Completeness and soundness are clear (?)

Proving zero knowledge of $(P_{\mathcal{L}}, V_{\mathcal{L}})$

- For simplicity, Map_X is injective and Map_X^{-1} is poly-time computable.
- Assume a view of a (deterministic) verifier is of the form: input, transcript.
- Given a view $v = (x, t)$, let $v_{\text{trans}} = t$
- Let S be a black-box simulator of (P, V) .
- The oracle-aided $S_{\mathcal{L}}$ is defined by $S_{\mathcal{L}}^O(x) = x, S^O(\text{Map}_X(x))_{\text{trans}}$.

Claim 28

$$\{\langle (P_{\mathcal{L}}(w(x)), V_{\mathcal{L}}^*(x)) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{S_{\mathcal{L}}^{V_{\mathcal{L}}^*(x)}(x)\}_{x \in \mathcal{L}} \quad \forall \text{ PPT } V_{\mathcal{L}}^* \text{ and } w.$$

Proof: Otherwise, $\{x, \langle (P(w'), V_{\mathcal{L}}^*(x')) \rangle_{\text{trans}}\}_{x \in \mathcal{L}} \not\approx_c \{x, S_{\mathcal{L}}^{V_{\mathcal{L}}^*(x)}(x')_{\text{trans}}\}_{x \in \mathcal{L}}$, for $x' = \text{Map}_X(x)$ and $w' = \text{Map}_W(w(x))$.

$$\Rightarrow \{x, x', \langle (P(w'), V_{\mathcal{L}}^*(x')) \rangle_{\text{trans}}\}_{x \in \mathcal{L}} \not\approx_c \{x, S_{\mathcal{L}}^{V_{\mathcal{L}}^*(x)}(x')\}_{x \in \mathcal{L}}$$

$$\Rightarrow (?) \{x', \langle (P(w'), V_{\mathcal{L}}^*(x')) \rangle_{\text{trans}}\}_{x \in \mathcal{L}} \not\approx_c \{S_{\mathcal{L}}^{V_{\mathcal{L}}^*(x)}(x')\}_{x \in \mathcal{L}}$$

$$\Rightarrow \{\langle (P(w'), V^*(x')) \rangle_{V^*}\}_{x' \in 3\text{COL}} \not\approx_c \{S^{V^*(x')}(x')\}_{x' \in 3\text{COL}}$$

for $x = \text{Map}_X^{-1}(x')$, $w' = \text{Map}_W(w(x))$ and $V^*(x')$ that acts like $V_{\mathcal{L}}^*(x)$.

Is non-auxiliary-input black-box ZK is auxiliary input ZK?

- Let $\mathcal{L} = \{1^n : n \in \mathbb{N}\}$ and consider the \mathcal{NP} -relation for \mathcal{L} defined as $R = \{(1^n, 0), (1^n, 1) : n \in \mathbb{N}\}$.
- Assume exists commitment scheme Com , that is computationally hiding against PPT *uniform* receivers, but **not** hiding against *non-uniform* PPT receivers.
- The following protocol is non-auxiliary input \mathcal{CZK} for \mathcal{L} with black-box simulator, but **not** auxiliary-input \mathcal{CZK} (it is not even witness hiding).

Protocol 29 ((P, V))

Common input: $x \in \{0, 1\}^*$

P's input: $w \in R_{\mathcal{L}}(x)$

- 1 P commits to w using Com
- 2 V accepts if $x \in \mathcal{L}$.