Foundation of Cryptography, Lecture 6
Interactive Proofs and Zero Knowledge

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

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

$\mathcal{NP}$ 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. $\mathsf{V}(x,w) = 1$
- $\mathsf{V}(x,w) = 0$ for every $x \notin \mathcal{L}$ and $w \in \{0,1\}^*$

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

A proof system

$\mathcal{NP}$ 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. $\mathsf{V}(x,w) = 1$
- $\mathsf{V}(x,w) = 0$ 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)

$\mathcal{NP}$ 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. $\mathsf{V}(x,w) = 1$
- $\mathsf{V}(x,w) = 0$ 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.

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

Soundness $\forall x \notin \mathcal{L}$, and any algorithm P^*

$$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle_{\mathsf{V}} = 1] \leq 1/3.$$

IP is the class of languages that have interactive proofs.

[a] $\langle(\mathsf{A}(a), \mathsf{B}(b))(c)\rangle_{\mathsf{B}}$ denote B's view in random execution of $(\mathsf{A}(a), \mathsf{B}(b))(c)$.

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

Soundness $\forall x \notin \mathcal{L}$, and any algorithm P^*

$$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle_{\mathsf{V}} = 1] \leq 1/3.$$

IP is the class of languages that have interactive proofs.

[a] $\langle(\mathsf{A}(a), \mathsf{B}(b))(c)\rangle_{\mathsf{B}}$ denote B's view in random execution of $(\mathsf{A}(a), \mathsf{B}(b))(c)$.

- $\mathsf{IP} = \mathrm{PSPACE}$!

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

Soundness $\forall x \notin \mathcal{L}$, and any algorithm P^*

$$\Pr[\langle(\mathsf{P}^*,\mathsf{V})(x)\rangle_{\mathsf{V}} = 1] \leq 1/3.$$

IP is the class of languages that have interactive proofs.

[a] $\langle(\mathsf{A}(a),\mathsf{B}(b))(c)\rangle_{\mathsf{B}}$ denote B's view in random execution of $(\mathsf{A}(a),\mathsf{B}(b))(c)$.

- $\mathsf{IP} = \mathrm{PSPACE}$!
- We typically consider (and achieve) perfect completeness.

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

Soundness $\forall x \notin \mathcal{L}$, and any algorithm P^*

$$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle_{\mathsf{V}} = 1] \leq 1/3.$$

IP is the class of languages that have interactive proofs.

[a] $\langle(\mathsf{A}(a), \mathsf{B}(b))(c)\rangle_{\mathsf{B}}$ denote B's view in random execution of $(\mathsf{A}(a), \mathsf{B}(b))(c)$.

- $\mathsf{IP} = \mathrm{PSPACE}$!
- We typically consider (and achieve) perfect completeness.
- Negligible “soundness error" achieved via repetition.

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

Soundness $\forall x \notin \mathcal{L}$, and any algorithm P^*

$$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle_{\mathsf{V}} = 1] \leq 1/3.$$

IP is the class of languages that have interactive proofs.

[a] $\langle(\mathsf{A}(a), \mathsf{B}(b))(c)\rangle_{\mathsf{B}}$ denote B's view in random execution of $(\mathsf{A}(a), \mathsf{B}(b))(c)$.

- $\mathsf{IP} = \mathrm{PSPACE}$!
- We typically consider (and achieve) perfect completeness.
- Negligible "soundness error" achieved via repetition.
- Sometime we have efficient provers via "auxiliary input".

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

Soundness $\forall x \notin \mathcal{L}$, and any algorithm P^*

$$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle_{\mathsf{V}} = 1] \leq 1/3.$$

IP is the class of languages that have interactive proofs.

[a] $\langle(\mathsf{A}(a), \mathsf{B}(b))(c)\rangle_{\mathsf{B}}$ denote B's view in random execution of $(\mathsf{A}(a), \mathsf{B}(b))(c)$.

- $\mathsf{IP} = \mathrm{PSPACE}$!
- 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 $(\pi(u), \pi(v)) \in E_1$.

Graph isomorphism

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

Definition 3 (graph isomorphism)

Graphs $\mathsf{G}_0 = ([m], E_0)$ and $\mathsf{G}_1 = ([m], E_1)$ are isomorphic, denoted $\mathsf{G}_0 \equiv \mathsf{G}_1$, if $\exists \pi \in \Pi_m$ such that
$(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.

- $\mathcal{GI} = \{(\mathsf{G}_0, \mathsf{G}_1) \colon \mathsf{G}_0 \equiv \mathsf{G}_1\} \in \mathcal{NP}$

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 $(\pi(u), \pi(v)) \in E_1$.

- $\mathcal{GI} = \{(G_0, G_1) \colon G_0 \equiv G_1\} \in \mathcal{NP}$
- Does $\mathcal{GNI} = \{(G_0, G_1) \colon G_0 \not\equiv G_1\} \in \mathcal{NP}$?

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 $(\pi(u), \pi(v)) \in E_1$.

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

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 $(\pi(u), \pi(v)) \in E_1$.

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

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 $(\pi(u), \pi(v)) \in E_1$.

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

Interactive proof for $\mathcal{GNI}$

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] $\pi(E) = \{(\pi(u), \pi(v) \colon (u, v) \in E\}$.

Interactive proof for $\mathcal{GNI}$

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] $\pi(E) = \{(\pi(u), \pi(v) \colon (u,v) \in E\}$.

Claim 5

The above protocol is IP for $\mathcal{GNI}$, 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)

Proving Claim 5

- Graph isomorphism is an equivalence relation (separates the set of all graph pairs into separate subsets)

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 $[\mathrm{G}_i]$ — the equivalence class of G_i

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

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 $[\mathsf{G}_i]$ — the equivalence class of G_i

Hence,

$\mathsf{G}_0 \equiv \mathsf{G}_1$: $\Pr[b' = b] \leq \frac{1}{2}$.

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 $[\mathsf{G}_i]$ — the equivalence class of G_i

Hence,

$\mathsf{G}_0 \equiv \mathsf{G}_1$: $\Pr[b' = b] \leq \frac{1}{2}$.

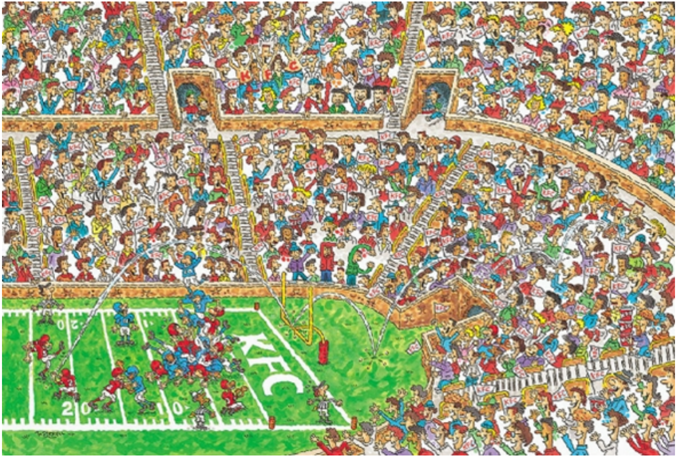
$\mathsf{G}_0 \not\equiv \mathsf{G}_1$: $\Pr[b' = b] = 1$ (i.e., P can, possibly inefficiently, extracted from $\pi(E_i)$)

□

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 addition information.

The concept of zero knowledge

- Proving w/o revealing any addition information.
- What does it mean?

The concept of zero knowledge

- Proving w/o revealing any addition information.
- What does it mean?

  Simulation paradigm.

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.
2. $\mathcal{ZK}$ only required to hold wrt. true statements.

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.
2. $\mathcal{ZK}$ only required to hold wrt. true statements.
3. Trivial to achieve for $\mathcal{L} \in \mathcal{BPP}$.

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.
2. $\mathcal{ZK}$ 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.

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.
2. $\mathcal{ZK}$ 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 $\mathcal{NP}$.

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.
2. $\mathcal{ZK}$ 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

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.
2. $\mathcal{ZK}$ 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

Zero-knowledge proofs

Definition 7 (zero-knowledge proofs)

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

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

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

Perfect $\mathcal{ZK}$ ($\mathcal{PZK}$)/statistical $\mathcal{ZK}$ ($\mathcal{SZK}$) — the above distributions are identicallly/statistically close.

1. $\mathcal{ZK}$ is a property of the prover.
2. $\mathcal{ZK}$ 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 $V^* = V$.

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.

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

Zero-knowledge proof for $\mathcal{GI}$

Idea: route finding

Protocol 8 ((P, V))

Common input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

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 $\pi''(E_b) = E$.

Zero-knowledge proof for $\mathcal{GI}$

Idea: route finding

Protocol 8 ((P, V))

Common input: $x = (\mathrm{G}_0 = ([m], E_0), \mathrm{G}_1 = ([m], E_1))$

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

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 $\frac{1}{2}$.

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).

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

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

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.

The simulator

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

Algorithm 10 (S)

Input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

Do $|x|$ times:

1. Choose $b' \leftarrow \{0, 1\}$ and $\pi \leftarrow \Pi_m$, and “send" $\pi(E_{b'})$ to $\mathsf{V}^*(x)$.
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.

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

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

The simulator

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

Algorithm 10 (S)

Input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

Do $|x|$ times:

1. Choose $b' \leftarrow \{0,1\}$ and $\pi \leftarrow \Pi_m$, and "send" $\pi(E_{b'})$ to $\mathsf{V}^*(x)$.
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

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

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.

Proving Claim 11

Consider the following inefficient simulator:

Algorithm 12 (S′)

Input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$.

Do $|x|$ times:

1. Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
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

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

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: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

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

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. $\langle (P, V^*(x)) \rangle_{V^*} \equiv S''(x)$.

Proving Claim 11 cont.

Consider a second inefficient simulator:

Algorithm 14 (S'')

Input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
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. $\langle(\mathsf{P}, \mathsf{V}^*(x))\rangle_{\mathsf{V}^*} \equiv \mathsf{S}''(x)$.
2. $\mathsf{SD}(\mathsf{S}''(x), \mathsf{S}'(x)) \leq 2^{-|x|}$.

Proving Claim 11 cont.

Consider a second inefficient simulator:

Algorithm 14 (S'')

Input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
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. $\langle(\mathsf{P}, \mathsf{V}^*(x))\rangle_{\mathsf{V}^*} \equiv \mathsf{S}''(x)$.
2. $\mathsf{SD}(\mathsf{S}''(x), \mathsf{S}'(x)) \leq 2^{-|x|}$.

Proving Claim 11 cont.

Consider a second inefficient simulator:

Algorithm 14 (S'')

Input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
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. $\langle(\mathsf{P}, \mathsf{V}^*(x))\rangle_{\mathsf{V}^*} \equiv \mathsf{S}''(x)$.
2. $\mathsf{SD}(\mathsf{S}''(x), \mathsf{S}'(x)) \leq 2^{-|x|}$.

Proof: ?

Proving Claim 11 cont.

Consider a second inefficient simulator:

Algorithm 14 (S$''$)

Input: $x = (\mathsf{G}_0 = ([m], E_0), \mathsf{G}_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and send $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
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. $\langle(\mathsf{P}, \mathsf{V}^*(x))\rangle_{\mathsf{V}^*} \equiv \mathsf{S}''(x)$.
2. $\mathsf{SD}(\mathsf{S}''(x), \mathsf{S}'(x)) \leq 2^{-|x|}$.

Proof: ? (1) is clear.

Proving Claim 15(2)

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

Proving Claim 15(2)

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

$$\Pr_{\mathsf{S}'(x)}[t] = \alpha \cdot \sum_{i=1}^{|x|} (1 - \frac{1}{2})^{i-1} \cdot \frac{1}{2}$$
$$= (1 - 2^{-|x|}) \cdot \alpha$$

Proving Claim 15(2)

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

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

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

Remarks

1. Perfect $\mathcal{ZK}$ for “expected polynomial-time" simulators.

Remarks

1. Perfect $\mathcal{ZK}$ for “expected polynomial-time" simulators.
2. Aborting verifiers.

Remarks

1. Perfect $\mathcal{ZK}$ for “expected polynomial-time" simulators.
2. Aborting verifiers.
3. Randomized verifiers.

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.

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.

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?

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

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!

“Transcript simulation" might not suffice!

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

"Transcript simulation" might not suffice!

Let $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ 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 $(d, e) \leftarrow \mathsf{G}(1^{|x|})$ and sends e to P
2. P sends $c = \mathsf{E}_e(w)$ to V
3. V accepts iff $\mathsf{D}_d(c) \in R_{\mathcal{L}}(x)$

"Transcript simulation" might not suffice!

Let $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ 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 $(d, e) \leftarrow \mathsf{G}(1^{|x|})$ and sends e to P
2. P sends $c = \mathsf{E}_e(w)$ to V
3. V accepts iff $\mathsf{D}_d(c) \in R_{\mathcal{L}}(x)$

- The above protocol has perfect completeness and soundness.

"Transcript simulation" might not suffice!

Let $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ 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 $(d, e) \leftarrow \mathsf{G}(1^{|x|})$ and sends e to P
2. P sends $c = \mathsf{E}_e(w)$ to V
3. V accepts iff $\mathsf{D}_d(c) \in R_{\mathcal{L}}(x)$

- The above protocol has perfect completeness and soundness.
- Is it zero-knowledge?

“Transcript simulation" might not suffice!

Let $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ 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 $(d, e) \leftarrow \mathsf{G}(1^{|x|})$ and sends e to P
2. P sends $c = \mathsf{E}_e(w)$ to V
3. V accepts iff $\mathsf{D}_d(c) \in R_{\mathcal{L}}(x)$

- The above protocol has perfect completeness and soundness.
- Is it zero-knowledge?
- It has “transcript simulator" (at least for honest verifiers): exits PPT S such that $\{\langle(\mathsf{P}(w \in R_{\mathcal{L}}(x)), \mathsf{V})(x)\rangle_{\text{trans}}\}_{x\in\mathcal{L}} \approx_c \{\mathsf{S}(x)\}_{x\in\mathcal{L}}$,

  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?

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.

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

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

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

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.

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

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

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

- Strengthening of the standard definition.

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.

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

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

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 $\mathcal{SZK}$

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.

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

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

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?

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.

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

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

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.

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.

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

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

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.

Is zero-knowledge maintained under composition?, cont.

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

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:

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

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.

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

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

Prefect and statistical variants are defined analogously.

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.

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

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

Prefect and statistical variants are defined analogously.

1. “Most simulators" are black box

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.

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

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

Prefect and statistical variants are defined analogously.

1. “Most simulators" are black box

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.

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

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

Prefect and statistical variants are defined analogously.

1. “Most simulators" are black box
2. Strictly weaker then 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 .

$\mathcal{CZK}$ 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 $\in \mathcal{NPC}$).

$\mathcal{CZK}$ 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 $\in \mathcal{NPC}$).

$\mathcal{CZK}$ 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 $\exists\, \phi \colon M \mapsto [3]$ s.t. $\phi(u) \neq \phi(v)$ for every $(u, v) \in E$.

$\mathcal{CZK}$ 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 $\exists\, \phi \colon M \mapsto [3]$ 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 $[3]$.

The protocol

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

The protocol

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

Protocol 20 ((P, V))

Common input: Graph $\mathsf{G} = (M, E)$ with $n = |\mathsf{G}|$

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 $1/|E|$.

Claim 21

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

- Completeness: Clear

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^*.

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^*.

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 $\phi \colon M \mapsto [3]$ 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 $\phi(v) = 1$).

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 $\phi \colon M \mapsto [3]$ 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 $\phi(v) = 1$).

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 $\phi\colon M \mapsto [3]$ 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 $G \notin$ 3COL, then $\exists (u,v) \in E$ s.t. $\psi(u) = \psi(v)$.

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 $\phi \colon M \mapsto [3]$ 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 $G \notin$ 3COL, then $\exists (u, v) \in E \text{ s.t. } \psi(u) = \psi(v)$.

  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.

Proving $\mathcal{ZK}$

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

Algorithm 22 (S)

Input: A graph $\mathsf{G} = (M, E)$ with $n = |\mathsf{G}|$

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 $\psi(w) = 1$ 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 ($\widetilde{\mathsf{S}}$)

Input: $\mathsf{G} = (V, E)$ with $n = |\mathsf{G}|$, 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.

Proving $\mathcal{ZK}$ cont.

Algorithm 23 ($\widetilde{\mathsf{S}}$)

Input: $\mathsf{G} = (V, E)$ with $n = |\mathsf{G}|$, 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

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

Proving $\mathcal{ZK}$ cont.

Algorithm 23 ($\widetilde{\mathsf{S}}$)

Input: $\mathsf{G} = (V, E)$ with $n = |\mathsf{G}|$, 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

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

Proof: ?

Proving $\mathcal{ZK}$ cont..

Claim 25

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

Proving $\mathcal{ZK}$ cont..

Claim 25

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

Proof:

Proving $\mathcal{ZK}$ cont..

Claim 25

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

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

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

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

Proving $\mathcal{ZK}$ cont..

Claim 25

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

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

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

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

Hence, $\exists$ PPT R^* and $b \in [3] \setminus \{1\}$ such that

$$\Pr\left[\left\langle (\mathsf{Snd}(1), \mathsf{R}^*(x, w(x)))\,(1^{|x|})\right\rangle_{\mathsf{R}^*} = 1\right] - \Pr\left[\left\langle (\mathsf{Snd}(b), \mathsf{R}^*(x, w(x)))\,(1^{|x|})\right\rangle_{\mathsf{R}^*} = 1\right]$$
$$\geq \frac{1}{|x|^2 \cdot p(|x|)}$$

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

Proving $\mathcal{ZK}$ cont..

Claim 25

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

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

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

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

Hence, $\exists$ PPT R^* and $b \in [3] \setminus \{1\}$ such that

$$\Pr\left[\left\langle(\mathsf{Snd}(1), \mathsf{R}^*(x, w(x)))\,(1^{|x|})\right\rangle_{\mathsf{R}^*} = 1\right] - \Pr\left[\left\langle(\mathsf{Snd}(b), \mathsf{R}^*(x, w(x)))\,(1^{|x|})\right\rangle_{\mathsf{R}^*} = 1\right] \geq \frac{1}{|x|^2 \cdot p(|x|)}$$

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

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

Remarks

- Aborting verifiers

Remarks

- Aborting verifiers
- Auxiliary inputs

Remarks

- Aborting verifiers
- Auxiliary inputs
- Soundness amplification

Extending to all $\mathcal{NP}$

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.

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

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.

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

We assume for simplicity that Map_X is injective.

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.

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

We assume for simplicity that Map_X is injective.

Let (P, V) be a $\mathcal{CZK}$ for $\mathsf{3COL}$.

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.

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

We assume for simplicity that Map_X is injective.

Let (P, V) be a $\mathcal{CZK}$ for $\mathsf{3COL}$.

Protocol 26 ($(\mathsf{P}_{\mathcal{L}}, \mathsf{V}_{\mathcal{L}})$)

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

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

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

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.
- Zero knowledge: Let S (an efficient) $\mathcal{ZK}$ simulator for (P, V) (for 3COL).

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.
- Zero knowledge: Let S (an efficient) $\mathcal{ZK}$ simulator for (P, V) (for 3COL).

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.
- Zero knowledge: Let S (an efficient) $\mathcal{ZK}$ simulator for (P, V) (for 3COL).

  On input (x, z_x) and verifier V^*, let $\mathsf{S}_\mathcal{L}$ output $\mathsf{S}^{\mathsf{V}^*(x,z_x)}(\mathsf{Map}_X(x))$.

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.
- Zero knowledge: Let S (an efficient) $\mathcal{ZK}$ simulator for (P, V) (for 3COL).

  On input (x, z_x) and verifier V^*, let $\mathsf{S}_{\mathcal{L}}$ output $\mathsf{S}^{\mathsf{V}^*(x,z_x)}(\mathsf{Map}_X(x))$.

Claim 28

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

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.
- Zero knowledge: Let S (an efficient) $\mathcal{ZK}$ simulator for (P, V) (for 3COL).

  On input (x, z_x) and verifier V^*, let $\mathsf{S}_{\mathcal{L}}$ output $\mathsf{S}^{\mathsf{V}^*(x,z_x)}(\mathsf{Map}_X(x))$.

Claim 28

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

Proof:

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.
- Zero knowledge: Let S (an efficient) $\mathcal{ZK}$ simulator for (P, V) (for 3COL).

  On input (x, z_x) and verifier V^*, let $\mathsf{S}_{\mathcal{L}}$ output $\mathsf{S}^{\mathsf{V}^*(x,z_x)}(\mathsf{Map}_X(x))$.

Claim 28

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

Proof: Assume $\{\langle(\mathsf{P}_{\mathcal{L}}(w(x)), \mathsf{V}^*_{\mathcal{L}}(z(x))(x)\rangle_{\mathsf{V}^*_{\mathcal{L}}}\}_{x\in\mathcal{L}} \not\approx_c \{\mathsf{S}_{\mathcal{L}}^{\mathsf{V}^*_{\mathcal{L}}(x,z(x))}(x)\}_{x\in\mathcal{L}}$.

Extending to all $\mathcal{L} \in \mathcal{NP}$ cont.

Claim 27

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

- Completeness and soundness: Clear.
- Zero knowledge: Let S (an efficient) $\mathcal{ZK}$ simulator for (P, V) (for 3COL).

  On input (x, z_x) and verifier V^*, let $\mathsf{S}_{\mathcal{L}}$ output $\mathsf{S}^{\mathsf{V}^*(x,z_x)}(\mathsf{Map}_X(x))$.

Claim 28

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

Proof: Assume $\{\langle(\mathsf{P}_{\mathcal{L}}(w(x)), \mathsf{V}^*_{\mathcal{L}}(z(x))(x)\rangle_{\mathsf{V}^*_{\mathcal{L}}}\}_{x\in\mathcal{L}} \not\approx_c \{\mathsf{S}_{\mathcal{L}}^{\mathsf{V}^*_{\mathcal{L}}(x,z(x))}(x)\}_{x\in\mathcal{L}}$.

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

where $\mathsf{V}^*(x, z'_x = (z_x, x^{-1}))$ acts like $\mathsf{V}^*_{\mathcal{L}}(x^{-1}, z_x)$, and $z'(x) = (z(x^{-1}), x^{-1})$ for $x^{-1} = \mathsf{Map}_X^{-1}(x)$.