Foundation of Cryptography, Lecture 7
Non-Interactive ZK and Proof of Knowledge

Benny Applebaum & Iftach Haitner, Tel Aviv University

Tel Aviv University.

December 29, 2016

Part I

Non-Interactive Zero Knowledge

Interaction is crucial for $\mathcal{ZK}$

Claim 1

Assume that $\mathcal{L} \subseteq \{0,1\}^*$ has a one-message $\mathcal{ZK}$ proof (even computational), with standard completeness and soundness,[a] then $\mathcal{L} \in \mathcal{BPP}$.

[a] That is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Interaction is crucial for $\mathcal{ZK}$

Claim 1

Assume that $\mathcal{L} \subseteq \{0,1\}^*$ has a one-message $\mathcal{ZK}$ proof (even computational), with standard completeness and soundness,[a] then $\mathcal{L} \in \mathcal{BPP}$.

[a] That is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

Interaction is crucial for $\mathcal{ZK}$

Claim 1

Assume that $\mathcal{L} \subseteq \{0,1\}^*$ has a one-message $\mathcal{ZK}$ proof (even computational), with standard completeness and soundness,[a] then $\mathcal{L} \in \mathcal{BPP}$.

[a] That is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

1. To reduce interaction, we relax the zero-knowledge requirement

Interaction is crucial for $\mathcal{ZK}$

Claim 1

Assume that $\mathcal{L} \subseteq \{0,1\}^*$ has a one-message $\mathcal{ZK}$ proof (even computational), with standard completeness and soundness,[a] then $\mathcal{L} \in \mathcal{BPP}$.

[a] That is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

1. To reduce interaction, we relax the zero-knowledge requirement
   1. Witness Indistinguishability
      $\{\langle (\mathsf{P}(w_x^1), \mathsf{V}^*)(x)\rangle_{\mathsf{V}^*}\}_{x\in\mathcal{L}} \approx_c \{\langle (\mathsf{P}(w_x^2), \mathsf{V}^*)(x)\rangle_{\mathsf{V}^*}\}_{x\in\mathcal{L}}$,
      for any $\{w_x^1 \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$ and $\{w_x^2 \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$

Interaction is crucial for $\mathcal{ZK}$

Claim 1

Assume that $\mathcal{L} \subseteq \{0,1\}^*$ has a one-message $\mathcal{ZK}$ proof (even computational), with standard completeness and soundness,[a] then $\mathcal{L} \in \mathcal{BPP}$.

[a] That is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

1. To reduce interaction, we relax the zero-knowledge requirement
   1. Witness Indistinguishability
      $\{\langle(\mathsf{P}(w_x^1),\mathsf{V}^*)(x)\rangle_{\mathsf{V}^*}\}_{x\in\mathcal{L}} \approx_c \{\langle(\mathsf{P}(w_x^2),\mathsf{V}^*)(x)\rangle_{\mathsf{V}^*}\}_{x\in\mathcal{L}}$,
      for any $\{w_x^1 \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$ and $\{w_x^2 \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$
   2. Witness hiding

Interaction is crucial for $\mathcal{ZK}$

Claim 1

Assume that $\mathcal{L} \subseteq \{0,1\}^*$ has a one-message $\mathcal{ZK}$ proof (even computational), with standard completeness and soundness,[a] then $\mathcal{L} \in \mathcal{BPP}$.

[a] That is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

1. To reduce interaction, we relax the zero-knowledge requirement
    1. Witness Indistinguishability
       $\{\langle (\mathsf{P}(w_x^1), \mathsf{V}^*)(x) \rangle_{\mathsf{V}^*}\}_{x \in \mathcal{L}} \approx_c \{\langle (\mathsf{P}(w_x^2), \mathsf{V}^*)(x) \rangle_{\mathsf{V}^*}\}_{x \in \mathcal{L}}$,
       for any $\{w_x^1 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$ and $\{w_x^2 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$
    2. Witness hiding
    3. Non-interactive "zero knowledge"

Non-Interactive Zero Knowledge ($\mathcal{NIZK}$)

Definition 2 ($\mathcal{NIZK}$)

A pair of non interactive PPTM's (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L} \in \mathcal{NP}$, if $\exists \ell \in \text{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, for any $x \in \mathcal{L}$ and $w(x) \in R_{\mathcal{L}}(x)$.
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$.
- **Zero knowledge:** $\exists$ PPTM S s.t.

  $$\{(x, c, \mathsf{P}(x, w(x), c))_{c \leftarrow \{0,1\}^{\ell(|x|)}}\}_{x \in \mathcal{L}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$$

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

Non-Interactive Zero Knowledge ($\mathcal{NIZK}$)

Definition 2 ($\mathcal{NIZK}$)

A pair of non interactive PPTM's (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L} \in \mathcal{NP}$, if $\exists \ell \in \mathsf{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, for any $x \in \mathcal{L}$ and $w(x) \in R_{\mathcal{L}}(x)$.
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$.
- **Zero knowledge:** $\exists$ PPTM S s.t.

  $\{(x, c, \mathsf{P}(x, w(x), c))_{c \leftarrow \{0,1\}^{\ell(|x|)}}\}_{x \in \mathcal{L}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

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

- c – common (random) reference string (CRS)

Non-Interactive Zero Knowledge ($\mathcal{NIZK}$)

Definition 2 ($\mathcal{NIZK}$)

A pair of non interactive PPTM's (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L} \in \mathcal{NP}$, if $\exists \ell \in \text{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, for any $x \in \mathcal{L}$ and $w(x) \in R_{\mathcal{L}}(x)$.
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$.
- **Zero knowledge:** $\exists$ PPTM S s.t.

  $\{(x, c, \mathsf{P}(x, w(x), c))_{c \leftarrow \{0,1\}^{\ell(|x|)}}\}_{x \in \mathcal{L}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

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

- c – common (random) reference string (CRS)
- In the ZK part, CRS is chosen by the simulator.

Non-Interactive Zero Knowledge ($\mathcal{NIZK}$)

Definition 2 ($\mathcal{NIZK}$)

A pair of non interactive PPTM's (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L} \in \mathcal{NP}$, if $\exists \ell \in \mathsf{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, for any $x \in \mathcal{L}$ and $w(x) \in R_{\mathcal{L}}(x)$.
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$.
- **Zero knowledge:** $\exists$ PPTM S s.t.

  $$\{(x, c, \mathsf{P}(x, w(x), c))_{c \leftarrow \{0,1\}^{\ell(|x|)}}\}_{x \in \mathcal{L}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$$

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

- c – common (random) reference string (CRS)
- In the ZK part, CRS is chosen by the simulator.
- What does this definition (intuitively) mean?

Non-Interactive Zero Knowledge ($\mathcal{NIZK}$)

Definition 2 ($\mathcal{NIZK}$)

A pair of non interactive PPTM's (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L} \in \mathcal{NP}$, if $\exists \ell \in \mathrm{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, for any $x \in \mathcal{L}$ and $w(x) \in R_{\mathcal{L}}(x)$.
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$.
- **Zero knowledge:** $\exists$ PPTM S s.t.

  $\{(x, c, \mathsf{P}(x, w(x), c))_{c \leftarrow \{0,1\}^{\ell(|x|)}}\}_{x \in \mathcal{L}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

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

- c – common (random) reference string (CRS)
- In the ZK part, CRS is chosen by the simulator.
- What does this definition (intuitively) mean?
- Auxiliary information.

Non-Interactive Zero Knowledge ($\mathcal{NIZK}$)

Definition 2 ($\mathcal{NIZK}$)

A pair of non interactive PPTM's (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L} \in \mathcal{NP}$, if $\exists \ell \in \mathrm{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, for any $x \in \mathcal{L}$ and $w(x) \in R_{\mathcal{L}}(x)$.
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$.
- **Zero knowledge:** $\exists$ PPTM S s.t.

  $$\{(x, c, \mathsf{P}(x, w(x), c))_{c \leftarrow \{0,1\}^{\ell(|x|)}}\}_{x \in \mathcal{L}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$$

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

- c – common (random) reference string (CRS)
- In the ZK part, CRS is chosen by the simulator.
- What does this definition (intuitively) mean?
- Auxiliary information.
- Amplification?

Non-Interactive Zero Knowledge ($\mathcal{NIZK}$)

Definition 2 ($\mathcal{NIZK}$)

A pair of non interactive PPTM's (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L} \in \mathcal{NP}$, if $\exists \ell \in \text{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, for any $x \in \mathcal{L}$ and $w(x) \in R_{\mathcal{L}}(x)$.
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$.
- **Zero knowledge:** $\exists$ PPTM S s.t.

  $\{(x, c, \mathsf{P}(x, w(x), c))_{c \leftarrow \{0,1\}^{\ell(|x|)}}\}_{x \in \mathcal{L}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

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

- c – common (random) reference string (CRS)
- In the ZK part, CRS is chosen by the simulator.
- What does this definition (intuitively) mean?
- Auxiliary information.
- Amplification?
- What happens when applying S on $x \notin \mathcal{L}$?

Non-Interactive Zero Knowledge, cont.

- Statistical/Perfect zero knowledge?

Non-Interactive Zero Knowledge, cont.

- Statistical/Perfect zero knowledge?
- Non-interactive Witness Hiding (WI)

Section 1

NIZK in HBM

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the “hidden" CRS:

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the "hidden" CRS:

1. Prover sees c^H, and outputs a proof π and a set of indices $\mathcal{I}$.

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the “hidden" CRS:

1. Prover sees c^H, and outputs a proof π and a set of indices $\mathcal{I}$.
2. Verifier only sees π and the bits in c^H indexed by $\mathcal{I}$.

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the “hidden" CRS:

1. Prover sees c^H, and outputs a proof π and a set of indices $\mathcal{I}$.
2. Verifier only sees π and the bits in c^H indexed by $\mathcal{I}$.
3. Simulator outputs a proof π, a set of indices $\mathcal{I}$ and a partially hidden CRS c^H.

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the “hidden" CRS:

1. Prover sees c^H, and outputs a proof π and a set of indices $\mathcal{I}$.
2. Verifier only sees π and the bits in c^H indexed by $\mathcal{I}$.
3. Simulator outputs a proof π, a set of indices $\mathcal{I}$ and a partially hidden CRS c^H.

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the “hidden" CRS:

1. Prover sees c^H, and outputs a proof π and a set of indices $\mathcal{I}$.
2. Verifier only sees π and the bits in c^H indexed by $\mathcal{I}$.
3. Simulator outputs a proof π, a set of indices $\mathcal{I}$ and a partially hidden CRS c^H.

Soundness, completeness and ZK are naturally defined.

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the “hidden" CRS:

1. Prover sees c^H, and outputs a proof π and a set of indices $\mathcal{I}$.
2. Verifier only sees π and the bits in c^H indexed by $\mathcal{I}$.
3. Simulator outputs a proof π, a set of indices $\mathcal{I}$ and a partially hidden CRS c^H.

Soundness, completeness and ZK are naturally defined.

- We give a $\mathcal{NIZK}$ for $\mathcal{HC}$, Directed Graph Hamiltonicity, in the HBM, and then transfer it into a $\mathcal{NIZK}$ for $\mathcal{HC}$ in the standard model.

Hidden Bits Model (HBM)

A CRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof.

Let c^H be the “hidden" CRS:

1. Prover sees c^H, and outputs a proof π and a set of indices $\mathcal{I}$.
2. Verifier only sees π and the bits in c^H indexed by $\mathcal{I}$.
3. Simulator outputs a proof π, a set of indices $\mathcal{I}$ and a partially hidden CRS c^H.

Soundness, completeness and ZK are naturally defined.

- We give a $\mathcal{NIZK}$ for $\mathcal{HC}$, Directed Graph Hamiltonicity, in the HBM, and then transfer it into a $\mathcal{NIZK}$ for $\mathcal{HC}$ in the standard model.
- The latter implies a $\mathcal{NIZK}$ for all $\mathcal{NP}$.

Useful matrix

Useful matrix

- Permutation matrix: an $n \times n$ Boolean matrix, each row/column contains a single 1.

Useful matrix

- Permutation matrix: an $n \times n$ Boolean matrix, each row/column contains a single 1.
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that is an Hamiltonian cycle of all nodes (note that Hamiltonian matrix is also a permutation matrix).

Useful matrix

- Permutation matrix: an $n \times n$ Boolean matrix, each row/column contains a single 1.
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that is an Hamiltonian cycle of all nodes (note that Hamiltonian matrix is also a permutation matrix).
- Useful matrix: an $n^3 \times n^3$ Boolean matrix that contains an Hamiltonian generalized $n \times n$ sub-matrix, and all other entries are zeros.

Useful matrix

- Permutation matrix: an $n \times n$ Boolean matrix, each row/column contains a single 1.
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that is an Hamiltonian cycle of all nodes (note that Hamiltonian matrix is also a permutation matrix).
- Useful matrix: an $n^3 \times n^3$ Boolean matrix that contains an Hamiltonian generalized $n \times n$ sub-matrix, and all other entries are zeros.

Useful matrix

- Permutation matrix: an $n \times n$ Boolean matrix, each row/column contains a single 1.
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that is an Hamiltonian cycle of all nodes (note that Hamiltonian matrix is also a permutation matrix).
- Useful matrix: an $n^3 \times n^3$ Boolean matrix that contains an Hamiltonian generalized $n \times n$ sub-matrix, and all other entries are zeros.

Claim 3

Let T be a random $n^3 \times n^3$ Boolean matrix s.t. each entry is 1 w.p n^{-5}. Then, $\Pr[T \text{ is useful}] \in \Omega(n^{-3/2})$.

Proving $\Pr[T \text{ is useful}] \in \Omega(n^{-3/2})$

Proving $\Pr[T \textbf{ is useful}] \in \Omega(n^{-3/2})$

- The expected # of ones (entries) in T is $n^6 \cdot n^{-5} = n$.

Proving $\Pr[T \textbf{ is useful}] \in \Omega(n^{-3/2})$

- The expected $\#$ of ones (entries) in T is $n^6 \cdot n^{-5} = n$.
- By (extended) Chernoff bound, T contains exactly n ones w.p. $\theta(1/\sqrt{n})$.

Proving $\Pr[T \text{ is useful}] \in \Omega(n^{-3/2})$

- The expected # of ones (entries) in T is $n^6 \cdot n^{-5} = n$.
- By (extended) Chernoff bound, T contains exactly n ones w.p. $\theta(1/\sqrt{n})$.
- Each row/colomn of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.

Proving $\Pr[T \textbf{ is useful}] \in \Omega(n^{-3/2})$

- The expected # of ones (entries) in T is $n^6 \cdot n^{-5} = n$.
- By (extended) Chernoff bound, T contains exactly n ones w.p. $\theta(1/\sqrt{n})$.
- Each row/colomn of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.

  Hence, wp at least $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$, no raw or column of T contains more than a single one entry.

Proving $\Pr[T \textbf{ is useful}] \in \Omega(n^{-3/2})$

- The expected $\#$ of ones (entries) in T is $n^6 \cdot n^{-5} = n$.
- By (extended) Chernoff bound, T contains exactly n ones w.p. $\theta(1/\sqrt{n})$.
- Each row/colomn of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.

  Hence, wp at least $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$, no raw or column of T contains more than a single one entry.
- Hence, wp $\theta(1/\sqrt{n})$ the matrix T contains a permutation matrix and all its other entries are zero.

Proving $\Pr[T \textbf{ is useful}] \in \Omega(n^{-3/2})$

- The expected # of ones (entries) in T is $n^6 \cdot n^{-5} = n$.
- By (extended) Chernoff bound, T contains exactly n ones w.p. $\theta(1/\sqrt{n})$.
- Each row/colomn of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.

  Hence, wp at least $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$, no raw or column of T contains more than a single one entry.
- Hence, wp $\theta(1/\sqrt{n})$ the matrix T contains a permutation matrix and all its other entries are zero.
- A random permutation matrix forms a cycle wp $1/n$ (there are $n!$ permutation matrices and $(n-1)!$ of them form a cycle)

$\mathcal{NIZK}$ for Hamiltonicity in HBM

$\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$

$\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph $\mathsf{G} = ([n], E)$
- We assume wlg. that n is a power of 2 (?)

$\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$
- We assume wlg. that n is a power of 2 (?)
- Common reference string T viewed as a $n^3 \times n^3$ Boolean matrix, where each entry is 1 w.p n^{-5} (?)

$\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$
- We assume wlg. that n is a power of 2 (?)
- Common reference string T viewed as a $n^3 \times n^3$ Boolean matrix, where each entry is 1 w.p n^{-5} (?)

$\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$
- We assume wlg. that n is a power of 2 (?)
- Common reference string T viewed as a $n^3 \times n^3$ Boolean matrix, where each entry is 1 w.p n^{-5} (?)

Algorithm 4 (P)

Input: n-node graph $G = ([n], E)$ and a cycle C in G.

CRS: $T \in \{0, 1\}_{n^3 \times n^3}$.

1. If T not useful, set $\mathcal{I} = n^3 \times n^3$ (i.e., reveal all T) and $\pi = \perp$.
2. Otherwise, let H be the (generalized) $n \times n$ sub-matrix containing the hamiltonian cycle in T.
    1. Set $\mathcal{I} = T \setminus H$ (i.e., reveal the bits of T outside of H).
    2. Choose $\phi \leftarrow \Pi_n$ s.t. C is mapped to the cycle in H.
    3. Add the entries in H corresponding to non edges in G (wrt. ϕ) to $\mathcal{I}$.
3. Output $\pi = \phi$ and $\mathcal{I}$.

$\mathcal{NIZK}$ for Hamiltonicity in HBM cont.

Algorithm 5 (V)

Input: n-node graph $G = ([n], E)$, mapping ϕ, index set $\mathcal{I} \subseteq [n^3] \times [n^3]$ and an ordered set $\{T_i\}_{i \in \mathcal{I}}$.

`Accept` if $\phi = \perp$, all the bits of T are revealed and T is not useful.

Otherwise,

1. Verify that $\phi \in \Pi_n$.
2. Verify that exists a single $n \times n$ generalized submatrix $H \subseteq T$ s.t. all entries in $T \setminus H$ are zeros.
3. Verify that all entries of H not corresponding to edges of G according to ϕ, are zeros: $\forall (u, v) \notin E$, the entry $(\phi(u), \phi(v))$ in H is opened to 0.

$\mathcal{NIZK}$ for Hamiltonicity in HBM cont.

Algorithm 5 (V)

Input: n-node graph $\mathsf{G} = ([n], E)$, mapping ϕ, index set $\mathcal{I} \subseteq [n^3] \times [n^3]$ and an ordered set $\{T_i\}_{i \in \mathcal{I}}$.

`Accept` if $\phi = \perp$, all the bits of T are revealed and T is not useful.

Otherwise,

1. Verify that $\phi \in \Pi_n$.
2. Verify that exists a single $n \times n$ generalized submatrix $H \subseteq T$ s.t. all entries in $T \setminus H$ are zeros.
3. Verify that all entries of H not corresponding to edges of G according to ϕ, are zeros: $\forall (u, v) \notin E$, the entry $(\phi(u), \phi(v))$ in H is opened to 0.

Claim 6

The above protocol is a perfect $\mathcal{NIZK}$ for $\mathcal{HC}$ in the HBM, with perfect completeness and soundness error $1 - \Omega(n^{-3/2})$.

Proving Claim 6

- Completeness: Clear.

Proving Claim 6

- Completeness: Clear.
- Soundness: Assume T is useful and V accepts. Then ϕ^{-1} maps the unrevealed "edges" of H to the edges of G.

Proving Claim 6

- Completeness: Clear.
- Soundness: Assume T is useful and V accepts. Then ϕ^{-1} maps the unrevealed "edges" of H to the edges of G.

  Hence, ϕ^{-1} maps the cycle in H to an Hamiltonian cycle in G.

Proving Claim 6

- Completeness: Clear.
- Soundness: Assume T is useful and V accepts. Then ϕ^{-1} maps the unrevealed "edges" of H to the edges of G.

  Hence, ϕ^{-1} maps the cycle in H to an Hamiltonian cycle in G.
- Zero knowledge?

Algorithm 7 (S)

Input: G

1. Choose T at random (i.e., each entry is one wp n^{-5}).
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$.
3. Otherwise,
   1. Set $\mathcal{I} = T \setminus H$ (where H is the hamiltonian sub-matrix in T).
   2. Let $\phi \leftarrow \Pi_n$. Replace all entries of H with zeros.
   3. Add the entries in H corresponding to non edges in G to $\mathcal{I}$.
4. Output $\pi = (T, \mathcal{I}, \phi)$.

Algorithm 7 (S)

Input: G

1. Choose T at random (i.e., each entry is one wp n^{-5}).
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$.
3. Otherwise,
   1. Set $\mathcal{I} = T \setminus H$ (where H is the hamiltonian sub-matrix in T).
   2. Let $\phi \leftarrow \Pi_n$. Replace all entries of H with zeros.
   3. Add the entries in H corresponding to non edges in G to $\mathcal{I}$.
4. Output $\pi = (T, \mathcal{I}, \phi)$.

- Perfect simulation for non-useful T's.

Algorithm 7 (S)

Input: G

1. Choose T at random (i.e., each entry is one wp n^{-5}).
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$.
3. Otherwise,
   1. Set $\mathcal{I} = T \setminus H$ (where H is the hamiltonian sub-matrix in T).
   2. Let $\phi \leftarrow \Pi_n$. Replace all entries of H with zeros.
   3. Add the entries in H corresponding to non edges in G to $\mathcal{I}$.
4. Output $\pi = (T, \mathcal{I}, \phi)$.

- Perfect simulation for non-useful T's.
- For useful T, the location of H is uniform in the real and simulated case.

Algorithm 7 (S)

Input: G

1. Choose T at random (i.e., each entry is one wp n^{-5}).
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$.
3. Otherwise,
   1. Set $\mathcal{I} = T \setminus H$ (where H is the hamiltonian sub-matrix in T).
   2. Let $\phi \leftarrow \Pi_n$. Replace all entries of H with zeros.
   3. Add the entries in H corresponding to non edges in G to $\mathcal{I}$.
4. Output $\pi = (T, \mathcal{I}, \phi)$.

- Perfect simulation for non-useful T's.
- For useful T, the location of H is uniform in the real and simulated case.
- ϕ is a random element in Π_n in both (real and simulated) cases (?)

Algorithm 7 (S)

Input: G

1. Choose T at random (i.e., each entry is one wp n^{-5}).
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$.
3. Otherwise,
    1. Set $\mathcal{I} = T \setminus H$ (where H is the hamiltonian sub-matrix in T).
    2. Let $\phi \leftarrow \Pi_n$. Replace all entries of H with zeros.
    3. Add the entries in H corresponding to non edges in G to $\mathcal{I}$.
4. Output $\pi = (T, \mathcal{I}, \phi)$.

- Perfect simulation for non-useful T's.
- For useful T, the location of H is uniform in the real and simulated case.
- ϕ is a random element in Π_n in both (real and simulated) cases (?)
- Hence, the simulation is perfect!

Section 2

From HBM to Standard NIZK

Subsection 1

TDP

Trapdoor permutations

Definition 8 (trapdoor permutations)

A triplet $(\mathsf{G}, f, \mathsf{Inv})$, where G is a PPTM, and f and Inv are poly-time computable, is a family of trapdoor permutation (TDP), if:

1. On input 1^n, $\mathsf{G}(1^n)$ outputs a pair (sk, pk).
2. $f_{pk} = f(pk, \cdot)$ is a permutation over $\{0,1\}^n$, for every $n \in \mathbb{N}$ and $pk \in \mathsf{Supp}(\mathsf{G}(1^n)_2)$.
3. $\mathsf{Inv}_{sk} = \mathsf{Inv}(sk, \cdot) \equiv f_{pk}^{-1}$ for every $(sk, pk) \in \mathsf{Supp}(\mathsf{G}(1^n))$
4. For any PPTM A,
   $\Pr_{x \leftarrow \{0,1\}^n, pk \leftarrow \mathsf{G}(1^n)_2}\left[\mathsf{A}(pk, x) = f_{pk}^{-1}(x)\right] = \mathsf{neg}(n)$

Hardcore Predicates for Trapdoor Permutations

Definition 9 (hardcore predicates for TDP)

A polynomial-time computable $b \colon \{0,1\}^n \mapsto \{0,1\}$ is a hardcore predicate of a TDP $(\mathsf{G}, f, \mathsf{Inv})$, if

$$\Pr_{pk \leftarrow \mathsf{G}(1^n)_2, x \leftarrow \{0,1\}^n}[\mathsf{P}(pk, f_{pk}(x)) = b(x)] \leq \frac{1}{2} + \mathsf{neg}(n),$$

for any PPTM P.

Hardcore Predicates for Trapdoor Permutations

Definition 9 (hardcore predicates for TDP)

A polynomial-time computable $b\colon \{0,1\}^n \mapsto \{0,1\}$ is a hardcore predicate of a TDP $(\mathsf{G}, f, \mathsf{Inv})$, if

$$\Pr_{pk \leftarrow \mathsf{G}(1^n)_2, x \leftarrow \{0,1\}^n}[\mathsf{P}(pk, f_{pk}(x)) = b(x)] \leq \frac{1}{2} + \mathrm{neg}(n),$$

for any PPTM P.

Goldreich-Levin: any TDP has an hardcore predicate (ignoring padding issues)

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

- $\mathbb{Z}_N = [N]$ and $\mathbb{Z}_N^* = \{x \in [N] \colon \operatorname{gcd}(x, N) = 1\}$

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

- $\mathbb{Z}_N = [N]$ and $\mathbb{Z}_N^* = \{x \in [N] \colon \gcd(x, N) = 1\}$
- $\phi(N) = |\mathbb{Z}_N^*|$ (equals $(P-1)(Q-1)$ for $N = PQ$ with $P, Q \in \mathcal{P}$)

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

- $\mathbb{Z}_N = [N]$ and $\mathbb{Z}_N^* = \{x \in [N] \colon \gcd(x, N) = 1\}$
- $\phi(N) = |\mathbb{Z}_N^*|$ (equals $(P-1)(Q-1)$ for $N = PQ$ with $P, Q \in \mathcal{P}$)
- For every $e \in \mathbb{Z}_{\phi(N)}^*$, the function $f(x) \equiv x^e \bmod N$ is a permutation over $\mathbb{Z}_N^*$.
  In particular, $(x^e)^d \equiv x \bmod N$, for every $x \in \mathbb{Z}_N^*$, where $d \equiv e^{-1} \bmod \phi(N)$

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

- $\mathbb{Z}_N = [N]$ and $\mathbb{Z}_N^* = \{x \in [N] \colon \gcd(x, N) = 1\}$
- $\phi(N) = |\mathbb{Z}_N^*|$ (equals $(P-1)(Q-1)$ for $N = PQ$ with $P, Q \in \mathcal{P}$)
- For every $e \in \mathbb{Z}_{\phi(N)}^*$, the function $f(x) \equiv x^e \bmod N$ is a permutation over $\mathbb{Z}_N^*$.
  In particular, $(x^e)^d \equiv x \bmod N$, for every $x \in \mathbb{Z}_N^*$, where $d \equiv e^{-1} \bmod \phi(N)$

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

- $\mathbb{Z}_N = [N]$ and $\mathbb{Z}_N^* = \{x \in [N] \colon \gcd(x, N) = 1\}$
- $\phi(N) = |\mathbb{Z}_N^*|$ (equals $(P-1)(Q-1)$ for $N = PQ$ with $P, Q \in \mathcal{P}$)
- For every $e \in \mathbb{Z}_{\phi(N)}^*$, the function $f(x) \equiv x^e \bmod N$ is a permutation over $\mathbb{Z}_N^*$.
  In particular, $(x^e)^d \equiv x \bmod N$, for every $x \in \mathbb{Z}_N^*$, where $d \equiv e^{-1} \bmod \phi(N)$

Definition 10 (RSA)

- $\mathsf{G}(P, Q)$ sets $pk = (N = PQ, e)$ for some $e \in \mathbb{Z}_{\phi(N)}^*$, and $sk = (N, d \equiv e^{-1} \bmod \phi(N))$
- $f(pk, x) = x^e \bmod N$
- $\mathsf{Inv}(sk, x) = x^d \bmod N$

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

- $\mathbb{Z}_N = [N]$ and $\mathbb{Z}_N^* = \{x \in [N] \colon \ \gcd(x, N) = 1\}$
- $\phi(N) = |\mathbb{Z}_N^*|$ (equals $(P-1)(Q-1)$ for $N = PQ$ with $P, Q \in \mathcal{P}$)
- For every $e \in \mathbb{Z}_{\phi(N)}^*$, the function $f(x) \equiv x^e \bmod N$ is a permutation over $\mathbb{Z}_N^*$.
  In particular, $(x^e)^d \equiv x \bmod N$, for every $x \in \mathbb{Z}_N^*$, where $d \equiv e^{-1} \bmod \phi(N)$

Definition 10 (RSA)

- $\mathrm{G}(P, Q)$ sets $pk = (N = PQ, e)$ for some $e \in \mathbb{Z}_{\phi(N)}^*$, and $sk = (N, d \equiv e^{-1} \bmod \phi(N))$
- $f(pk, x) = x^e \bmod N$
- $\mathrm{Inv}(sk, x) = x^d \bmod N$

Factoring is easy $\implies$ RSA is easy.

Example, RSA

In the following $N \in \mathbb{N}$ is a large number (n-bit long) and all operations are modulo N.

- $\mathbb{Z}_N = [N]$ and $\mathbb{Z}_N^* = \{x \in [N] \colon \gcd(x, N) = 1\}$
- $\phi(N) = |\mathbb{Z}_N^*|$ (equals $(P-1)(Q-1)$ for $N = PQ$ with $P, Q \in \mathcal{P}$)
- For every $e \in \mathbb{Z}_{\phi(N)}^*$, the function $f(x) \equiv x^e \bmod N$ is a permutation over $\mathbb{Z}_N^*$.
  In particular, $(x^e)^d \equiv x \bmod N$, for every $x \in \mathbb{Z}_N^*$, where $d \equiv e^{-1} \bmod \phi(N)$

Definition 10 (RSA)

- $\mathrm{G}(P, Q)$ sets $pk = (N = PQ, e)$ for some $e \in \mathbb{Z}_{\phi(N)}^*$, and $sk = (N, d \equiv e^{-1} \bmod \phi(N))$
- $f(pk, x) = x^e \bmod N$
- $\mathrm{Inv}(sk, x) = x^d \bmod N$

Factoring is easy $\implies$ RSA is easy. The other direction?

Subsection 2

The Transformation

The transformation

- Let $(\mathsf{P}_H, \mathsf{V}_H)$ be a HBM $\mathcal{NIZK}$ for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.

The transformation

- Let $(\mathsf{P}_H, \mathsf{V}_H)$ be a HBM $\mathcal{NIZK}$ for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let $(\mathsf{G}, f, \mathsf{Inv})$ be a TDP and let b be an hardcore bit for it.

  For simplicity, assume that $\mathsf{G}(1^n)$ chooses (sk, pk) as follows:

The transformation

- Let $(\mathsf{P}_H, \mathsf{V}_H)$ be a HBM $\mathcal{NIZK}$ for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let $(\mathsf{G}, f, \mathsf{Inv})$ be a TDP and let b be an hardcore bit for it.

  For simplicity, assume that $\mathsf{G}(1^n)$ chooses (sk, pk) as follows:

  1. $sk \leftarrow \{0,1\}^n$

The transformation

- Let $(\mathsf{P}_H, \mathsf{V}_H)$ be a HBM $\mathcal{NIZK}$ for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let $(\mathsf{G}, f, \mathsf{Inv})$ be a TDP and let b be an hardcore bit for it.

  For simplicity, assume that $\mathsf{G}(1^n)$ chooses (sk, pk) as follows:

  1. $sk \leftarrow \{0,1\}^n$
  2. $pk = PK(sk)$

The transformation

- Let $(\mathsf{P}_H, \mathsf{V}_H)$ be a HBM $\mathcal{NIZK}$ for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let $(\mathsf{G}, f, \mathsf{Inv})$ be a TDP and let b be an hardcore bit for it.

  For simplicity, assume that $\mathsf{G}(1^n)$ chooses (sk, pk) as follows:

  1. $sk \leftarrow \{0,1\}^n$
  2. $pk = PK(sk)$

The transformation

- Let $(\mathsf{P}_H, \mathsf{V}_H)$ be a HBM $\mathcal{NIZK}$ for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let $(\mathsf{G}, f, \mathsf{Inv})$ be a TDP and let b be an hardcore bit for it.

  For simplicity, assume that $\mathsf{G}(1^n)$ chooses (sk, pk) as follows:

  1. $sk \leftarrow \{0,1\}^n$
  2. $pk = PK(sk)$

  where $PK \colon \{0,1\}^n \mapsto \{0,1\}^n$ is a polynomial-time computable function.

The transformation

- Let $(\mathsf{P}_H, \mathsf{V}_H)$ be a HBM $\mathcal{NIZK}$ for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let $(\mathsf{G}, f, \mathsf{Inv})$ be a TDP and let b be an hardcore bit for it.

  For simplicity, assume that $\mathsf{G}(1^n)$ chooses (sk, pk) as follows:

  1. $sk \leftarrow \{0,1\}^n$
  2. $pk = PK(sk)$

  where $PK \colon \{0,1\}^n \mapsto \{0,1\}^n$ is a polynomial-time computable function.

We construct a $\mathcal{NIZK}$ (P, V) for $\mathcal{L}$, with the same completeness and “not too large" soundness error.

The protocol

Algorithm 11 (P)

Input: $x \in \mathcal{L}$, $w \in R_{\mathcal{L}}(x)$ and CRS $c = (c_1, \ldots, c_\ell) \in \{0,1\}^{n\ell}$, where $n = |x|$ and $\ell = \ell(n)$.

1. Choose $(sk, pk) \leftarrow \mathrm{G}(sk)$ and compute $c^H = (b(z_1 = f_{pk}^{-1}(c_1)), \ldots, b(z_{\ell(n)} = f_{pk}^{-1}(c_\ell)))$
2. Let $(\pi_H, \mathcal{I}) \leftarrow \mathrm{P}_H(x, w, c^H)$ and output $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$

The protocol

Algorithm 11 (P)

Input: $x \in \mathcal{L}$, $w \in R_{\mathcal{L}}(x)$ and CRS $c = (c_1, \ldots, c_\ell) \in \{0,1\}^{n\ell}$, where $n = |x|$ and $\ell = \ell(n)$.

1. Choose $(sk, pk) \leftarrow \mathsf{G}(sk)$ and compute $c^H = (b(z_1 = f_{pk}^{-1}(c_1)), \ldots, b(z_{\ell(n)} = f_{pk}^{-1}(c_\ell)))$
2. Let $(\pi_H, \mathcal{I}) \leftarrow \mathsf{P}_H(x, w, c^H)$ and output $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$

Algorithm 12 (V)

Input: $x \in \mathcal{L}$, CRS $c = (c_1, \ldots, c_\ell) \in \{0,1\}^{np}$, and $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$, where $n = |x|$ and $\ell = \ell(n)$.

1. Verify that $pk \in \{0,1\}^n$ and that $f_{pk}(z_i) = c_i$ for every $i \in \mathcal{I}$
2. Return $\mathsf{V}_H(x, \pi_H, \mathcal{I}, c^H)$, where $c_i^H = b(z_i)$ for every $i \in \mathcal{I}$.

Claim 13

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a $\mathcal{NIZK}$ for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L}$ with the same completeness, and soundness error α.

Claim 13

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a $\mathcal{NIZK}$ for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L}$ with the same completeness, and soundness error α.

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).

Claim 13

Assuming that (P_H, V_H) is a $\mathcal{NIZK}$ for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L}$ with the same completeness, and soundness error α.

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).
For every $pk \in \{0,1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \ldots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0,1\}^{np}}$ is uniformly distributed in $\{0,1\}^\ell$.

Claim 13

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a $\mathcal{NIZK}$ for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L}$ with the same completeness, and soundness error α.

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).
For every $pk \in \{0,1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \ldots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0,1\}^{np}}$ is uniformly distributed in $\{0,1\}^\ell$.

- Completeness: clear

Claim 13

Assuming that (P_H, V_H) is a $\mathcal{NIZK}$ for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L}$ with the same completeness, and soundness error α.

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).
For every $pk \in \{0,1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \ldots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0,1\}^{np}}$ is uniformly distributed in $\{0,1\}^\ell$.

- Completeness: clear
- Soundness: follows by a union bound over all possible choice of $pk \in \{0,1\}^n$.

Claim 13

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a $\mathcal{NIZK}$ for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a $\mathcal{NIZK}$ for $\mathcal{L}$ with the same completeness, and soundness error α.

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).
For every $pk \in \{0,1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \ldots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0,1\}^{np}}$ is uniformly distributed in $\{0,1\}^\ell$.

- Completeness: clear
- Soundness: follows by a union bound over all possible choice of $pk \in \{0,1\}^n$.
- Zero knowledge:?

Proving zero knowledge

Algorithm 14 (S)

Input: $x \in \{0,1\}^n$ of length n.

- Let $(\pi_H, \mathcal{I}, c^H) = \mathsf{S}_H(x)$, where S_H is the simulator of $(\mathsf{P}_H, \mathsf{V}_H)$
- Output $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
  - $pk \leftarrow \mathsf{G}(U_n)$
  - Each z_i is chosen at random in $\{0,1\}^n$ such that $b(z_i) = c_i^H$
  - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0,1\}^n$ otherwise.

Proving zero knowledge

Algorithm 14 (S)

Input: $x \in \{0,1\}^n$ of length n.

- Let $(\pi_H, \mathcal{I}, c^H) = \mathsf{S}_H(x)$, where S_H is the simulator of $(\mathsf{P}_H, \mathsf{V}_H)$
- Output $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
  - $pk \leftarrow \mathsf{G}(U_n)$
  - Each z_i is chosen at random in $\{0,1\}^n$ such that $b(z_i) = c_i^H$
  - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0,1\}^n$ otherwise.

- The above implicitly describes an efficient M s.t. $M(\mathsf{S}_H(x)) \equiv \mathsf{S}(x)$ and $M(\mathsf{P}_H(x, w(x))) \approx_c \mathsf{P}(x, w(x))$

Proving zero knowledge

Algorithm 14 (S)

Input: $x \in \{0,1\}^n$ of length n.

- Let $(\pi_H, \mathcal{I}, c^H) = \mathsf{S}_H(x)$, where S_H is the simulator of $(\mathsf{P}_H, \mathsf{V}_H)$
- Output $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
  - $pk \leftarrow \mathsf{G}(U_n)$
  - Each z_i is chosen at random in $\{0,1\}^n$ such that $b(z_i) = c_i^H$
  - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0,1\}^n$ otherwise.

- The above implicitly describes an efficient M s.t. $M(\mathsf{S}_H(x)) \equiv \mathsf{S}(x)$ and $M(\mathsf{P}_H(x, w(x))) \approx_c \mathsf{P}(x, w(x))$
- Hence, distinguishing $\mathsf{P}(x, w(x))$ from $\mathsf{S}(x)$ is hard

Proving zero knowledge

Algorithm 14 (S)

Input: $x \in \{0,1\}^n$ of length n.

- Let $(\pi_H, \mathcal{I}, c^H) = \mathsf{S}_H(x)$, where S_H is the simulator of $(\mathsf{P}_H, \mathsf{V}_H)$
- Output $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
  - $pk \leftarrow \mathsf{G}(U_n)$
  - Each z_i is chosen at random in $\{0,1\}^n$ such that $b(z_i) = c_i^H$
  - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0,1\}^n$ otherwise.

- The above implicitly describes an efficient M s.t. $M(\mathsf{S}_H(x)) \equiv \mathsf{S}(x)$ and $M(\mathsf{P}_H(x, w(x))) \approx_c \mathsf{P}(x, w(x))$
- Hence, distinguishing $\mathsf{P}(x, w(x))$ from $\mathsf{S}(x)$ is hard
- Direct solution for our $\mathcal{NIZK}$

Proving zero knowledge

Algorithm 14 (S)

Input: $x \in \{0,1\}^n$ of length n.

- Let $(\pi_H, \mathcal{I}, c^H) = \mathsf{S}_H(x)$, where S_H is the simulator of $(\mathsf{P}_H, \mathsf{V}_H)$
- Output $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
  - $pk \leftarrow \mathsf{G}(U_n)$
  - Each z_i is chosen at random in $\{0,1\}^n$ such that $b(z_i) = c_i^H$
  - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0,1\}^n$ otherwise.

- The above implicitly describes an efficient M s.t. $M(\mathsf{S}_H(x)) \equiv \mathsf{S}(x)$ and $M(\mathsf{P}_H(x, w(x))) \approx_c \mathsf{P}(x, w(x))$
- Hence, distinguishing $\mathsf{P}(x, w(x))$ from $\mathsf{S}(x)$ is hard

- Direct solution for our $\mathcal{NIZK}$
- An “adaptive" $\mathcal{NIZK}$

Section 3

Adaptive NIZK

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x = f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f \colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x = f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$
- **Soundness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^n$ and P^*
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}^*(c)) = 1 \wedge x \notin \mathcal{L}] \leq 1/3$

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$
- **Soundness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^n$ and P^*
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}^*(c)) = 1 \wedge x \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$**:** $\exists$ pair of PPTM's $(\mathsf{S}_1, \mathsf{S}_2)$ s.t. $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$

  $$\{(c \leftarrow \{0,1\}^{\ell(n)}, x = f(c), \mathsf{P}(x, w(x)))\}_{n\in\mathbb{N}} \approx_c \{\mathsf{S}^f(n)\}_{n\in\mathbb{N}}.$$

  where $\mathsf{S}^f(n)$ is the output of the following process

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$
- **Soundness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^n$ and P^*
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}^*(c)) = 1 \wedge x \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$**:** $\exists$ pair of PPTM's $(\mathsf{S}_1, \mathsf{S}_2)$ s.t. $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$

$$\{(c \leftarrow \{0,1\}^{\ell(n)}, x = f(c), \mathsf{P}(x, w(x)))\}_{n \in \mathbb{N}} \approx_c \{\mathsf{S}^f(n)\}_{n \in \mathbb{N}}.$$

  where $\mathsf{S}^f(n)$ is the output of the following process

  1. $(c, s) \leftarrow \mathsf{S}_1(1^n)$

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$
- **Soundness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^n$ and P^*
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}^*(c)) = 1 \wedge x \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$**:** $\exists$ pair of PPTM's $(\mathsf{S}_1, \mathsf{S}_2)$ s.t. $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$

  $$\{(c \leftarrow \{0,1\}^{\ell(n)}, x = f(c), \mathsf{P}(x, w(x)))\}_{n\in\mathbb{N}} \approx_c \{\mathsf{S}^f(n)\}_{n\in\mathbb{N}}.$$

  where $\mathsf{S}^f(n)$ is the output of the following process

  1. $(c, s) \leftarrow \mathsf{S}_1(1^n)$
  2. $x = f(c)$

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$
- **Soundness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^n$ and P^*
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}^*(c)) = 1 \land x \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$**:** $\exists$ pair of PPTM's $(\mathsf{S}_1, \mathsf{S}_2)$ s.t. $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$

  $$\{(c \leftarrow \{0,1\}^{\ell(n)}, x = f(c), \mathsf{P}(x, w(x)))\}_{n \in \mathbb{N}} \approx_c \{\mathsf{S}^f(n)\}_{n \in \mathbb{N}}.$$

  where $\mathsf{S}^f(n)$ is the output of the following process

  1. $(c, s) \leftarrow \mathsf{S}_1(1^n)$
  2. $x = f(c)$
  3. Output $(c, x, \mathsf{S}_2(x, c, s))$

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
$\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x = f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$
- **Soundness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^n$ and P^*
$\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x = f(c)}[\mathsf{V}(x, c, \mathsf{P}^*(c)) = 1 \wedge x \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$**:** $\exists$ pair of PPTM's $(\mathsf{S}_1, \mathsf{S}_2)$ s.t. $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$

$$\{(c \leftarrow \{0,1\}^{\ell(n)}, x = f(c), \mathsf{P}(x, w(x)))\}_{n \in \mathbb{N}} \approx_c \{\mathsf{S}^f(n)\}_{n \in \mathbb{N}}.$$

where $\mathsf{S}^f(n)$ is the output of the following process

1. $(c, s) \leftarrow \mathsf{S}_1(1^n)$
2. $x = f(c)$
3. Output $(c, x, \mathsf{S}_2(x, c, s))$

Adaptive $\mathcal{NIZK}$

x is chosen after the CRS.

- **Completeness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$ and $w(x) \in R_{\mathcal{L}}(x)$:
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$
- **Soundness:** $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^n$ and P^*
  $\Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)}[\mathsf{V}(x, c, \mathsf{P}^*(c)) = 1 \land x \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$**:** $\exists$ pair of PPTM's $(\mathsf{S}_1, \mathsf{S}_2)$ s.t. $\forall f\colon \{0,1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0,1\}^n$

  $$\{(c \leftarrow \{0,1\}^{\ell(n)}, x = f(c), \mathsf{P}(x, w(x)))\}_{n\in\mathbb{N}} \approx_c \{\mathsf{S}^f(n)\}_{n\in\mathbb{N}}.$$

  where $\mathsf{S}^f(n)$ is the output of the following process

  1. $(c, s) \leftarrow \mathsf{S}_1(1^n)$
  2. $x = f(c)$
  3. Output $(c, x, \mathsf{S}_2(x, c, s))$

Why do we need s?

Adaptive $\mathcal{NIZK}$, cont.

- Adaptive completeness and soundness are easy to achieve from any non-adaptive $\mathcal{NIZK}$.(?)

Adaptive $\mathcal{NIZK}$, cont.

- Adaptive completeness and soundness are easy to achieve from any non-adaptive $\mathcal{NIZK}$.(?)
- Not every $\mathcal{NIZK}$ is adaptive $\mathcal{ZK}$.

Adaptive $\mathcal{NIZK}$, cont.

- Adaptive completeness and soundness are easy to achieve from any non-adaptive $\mathcal{NIZK}$.(?)
- Not every $\mathcal{NIZK}$ is adaptive $\mathcal{ZK}$.

Adaptive $\mathcal{NIZK}$, cont.

- Adaptive completeness and soundness are easy to achieve from any non-adaptive $\mathcal{NIZK}$.(?)
- Not every $\mathcal{NIZK}$ is adaptive $\mathcal{ZK}$.

Theorem 15

Assume TDP exist, then every $\mathcal{NP}$ language has an adaptive $\mathcal{NIZK}$ with perfect completeness and negligible soundness error.

Adaptive $\mathcal{NIZK}$, cont.

- Adaptive completeness and soundness are easy to achieve from any non-adaptive $\mathcal{NIZK}$.(?)
- Not every $\mathcal{NIZK}$ is adaptive $\mathcal{ZK}$.

Theorem 15

Assume TDP exist, then every $\mathcal{NP}$ language has an adaptive $\mathcal{NIZK}$ with perfect completeness and negligible soundness error.

In the following, when saying adaptive $\mathcal{NIZK}$, we mean negligible completeness and soundness error.

Section 4

Simulation-Sound NIZK

Simulation soundness

A $\mathcal{NIZK}$ system (P, V) for $\mathcal{L}$ has (one-time) simulation soundness, if $\exists$ a pair of PPTM's $\mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2)$ that satisfies the $\mathcal{ZK}$ property of P with respect to $\mathcal{L}$, and in addition

Simulation soundness

A $\mathcal{NIZK}$ system (P, V) for $\mathcal{L}$ has (one-time) simulation soundness, if $\exists$ a pair of PPTM's $\mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2)$ that satisfies the $\mathcal{ZK}$ property of P with respect to $\mathcal{L}$, and in addition

$$\Pr_{(c,x,\pi,x',\pi') \leftarrow \mathrm{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}} [x' \notin \mathcal{L} \wedge \mathsf{V}(x', \pi', c) = 1 \wedge (x', \pi') \neq (x, \pi)] = \mathrm{neg}(n)$$

for any pair of PPTM's $\mathsf{P}^* = (\mathsf{P}^*_1, \mathsf{P}^*_2)$.

Experiment 16 ($\mathrm{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$)

1. $(c, s) \leftarrow \mathsf{S}_1(1^n)$
2. $(x, p) \leftarrow \mathsf{P}^*_1(1^n, c)$
3. $\pi \leftarrow \mathsf{S}_2(x, c, s)$
4. $(x', \pi') \leftarrow \mathsf{P}^*_2(p, \pi)$
5. Output (c, x, π, x', π')

Simulation soundness

A $\mathcal{NIZK}$ system (P, V) for $\mathcal{L}$ has (one-time) simulation soundness, if $\exists$ a pair of PPTM's $\mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2)$ that satisfies the $\mathcal{ZK}$ property of P with respect to $\mathcal{L}$, and in addition

$$\Pr_{(c,x,\pi,x',\pi') \leftarrow \mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}} [x' \notin \mathcal{L} \wedge \mathsf{V}(x', \pi', c) = 1 \wedge (x', \pi') \neq (x, \pi)] = \mathsf{neg}(n)$$

for any pair of PPTM's $\mathsf{P}^* = (\mathsf{P}^*_1, \mathsf{P}^*_2)$.

Experiment 16 ($\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$)

1. $(c, s) \leftarrow \mathsf{S}_1(1^n)$
2. $(x, p) \leftarrow \mathsf{P}^*_1(1^n, c)$
3. $\pi \leftarrow \mathsf{S}_2(x, c, s)$
4. $(x', \pi') \leftarrow \mathsf{P}^*_2(p, \pi)$
5. Output (c, x, π, x', π')

Simulation soundness, cont.

- After seeing a simulated (possibly false) proof, hard to generate an additional false proof

Simulation soundness, cont.

- After seeing a simulated (possibly false) proof, hard to generate an additional false proof
- Definition only considers efficient provers

Simulation soundness, cont.

- After seeing a simulated (possibly false) proof, hard to generate an additional false proof
- Definition only considers efficient provers
- (P,V) might be adaptive or non-adaptive

Simulation soundness, cont.

- After seeing a simulated (possibly false) proof, hard to generate an additional false proof
- Definition only considers efficient provers
- (P, V) might be adaptive or non-adaptive
- Standard $\mathcal{NIZK}$ guarantees weak type of simulation soundness (hard to fake proofs for simulated CRS and predefined x') (?)

Simulation soundness, cont.

- After seeing a simulated (possibly false) proof, hard to generate an additional false proof
- Definition only considers efficient provers
- (P, V) might be adaptive or non-adaptive
- Standard $\mathcal{NIZK}$ guarantees weak type of simulation soundness (hard to fake proofs for simulated CRS and predefined x') (?)
- Does the adaptive $\mathcal{NIZK}$ we seen have simulation soundness?

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ (one-time scheme suffices)

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ (one-time scheme suffices)
2. Non-interactive, perfectly-binding commitment Com.

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme (Gen, Sign, Vrfy) (one-time scheme suffices)
2. Non-interactive, perfectly-binding commitment Com.
   - Pseudorandom range: for some $\ell \in \text{poly}$
     $\{\text{Com}(w, r \leftarrow \{0,1\}^{\ell(|w|)})\}_{w \in \{0,1\}^*} \approx_c \{u \leftarrow \{0,1\}^{\ell(|w|)}\}_{w \in \{0,1\}^*}$

Construction

We present a simulation sound $\mathcal{NIZK}$ (P,V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme $(\mathsf{Gen},\mathsf{Sign},\mathsf{Vrfy})$ (one-time scheme suffices)
2. Non-interactive, perfectly-binding commitment Com.
   - Pseudorandom range: for some $\ell \in \mathsf{poly}$ $\{\mathsf{Com}(w, r \leftarrow \{0,1\}^{\ell(|w|)})\}_{w\in\{0,1\}^*} \approx_c \{u \leftarrow \{0,1\}^{\ell(|w|)}\}_{w\in\{0,1\}^*}$
   - * achieved by the standard OWP (or TDP) based perfectly-binding commitment.

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ (one-time scheme suffices)
2. Non-interactive, perfectly-binding commitment Com.
   - Pseudorandom range: for some $\ell \in \mathrm{poly}$
     $\{\mathsf{Com}(w, r \leftarrow \{0,1\}^{\ell(|w|)})\}_{w \in \{0,1\}^*} \approx_c \{u \leftarrow \{0,1\}^{\ell(|w|)}\}_{w \in \{0,1\}^*}$
     * achieved by the standard OWP (or TDP) based perfectly-binding commitment.
   - Negligible support: a random string is a valid commitment only with negligible probability.

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ (one-time scheme suffices)
2. Non-interactive, perfectly-binding commitment Com.

   - Pseudorandom range: for some $\ell \in \mathrm{poly}$
     $\{\mathsf{Com}(w, r \leftarrow \{0,1\}^{\ell(|w|)})\}_{w \in \{0,1\}^*} \approx_c \{u \leftarrow \{0,1\}^{\ell(|w|)}\}_{w \in \{0,1\}^*}$
   * achieved by the standard OWP (or TDP) based perfectly-binding commitment.
   - Negligible support: a random string is a valid commitment only with negligible probability.
   * achieved by using the standard OWP (or TDP) based perfectly-binding commitment, and committing to the same value many times.

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme (Gen, Sign, Vrfy) (one-time scheme suffices)
2. Non-interactive, perfectly-binding commitment Com.
   - Pseudorandom range: for some $\ell \in$ poly $\{\mathsf{Com}(w, r \leftarrow \{0,1\}^{\ell(|w|)})\}_{w \in \{0,1\}^*} \approx_c \{u \leftarrow \{0,1\}^{\ell(|w|)}\}_{w \in \{0,1\}^*}$
   * achieved by the standard OWP (or TDP) based perfectly-binding commitment.
   - Negligible support: a random string is a valid commitment only with negligible probability.
   * achieved by using the standard OWP (or TDP) based perfectly-binding commitment, and committing to the same value many times.
3. Adaptive $\mathcal{NIZK}$ $(\mathsf{P}_A, \mathsf{V}_A)$ for $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\} \in \mathcal{NP}$

Construction

We present a simulation sound $\mathcal{NIZK}$ (P, V) for $\mathcal{L} \in \mathcal{NP}$

Ingredients:

1. Strong signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ (one-time scheme suffices)
2. Non-interactive, perfectly-binding commitment Com.
   - Pseudorandom range: for some $\ell \in \mathsf{poly}$ $\{\mathsf{Com}(w, r \leftarrow \{0,1\}^{\ell(|w|)})\}_{w \in \{0,1\}^*} \approx_c \{u \leftarrow \{0,1\}^{\ell(|w|)}\}_{w \in \{0,1\}^*}$

   * achieved by the standard OWP (or TDP) based perfectly-binding commitment.
   - Negligible support: a random string is a valid commitment only with negligible probability.

   * achieved by using the standard OWP (or TDP) based perfectly-binding commitment, and committing to the same value many times.
3. Adaptive $\mathcal{NIZK}$ $(\mathsf{P}_A, \mathsf{V}_A)$ for $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\} \in \mathcal{NP}$

* adaptive WI suffices

Construction, cont.

Recall $\mathcal{L}_A := \{(x, \mathrm{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathrm{com} = \mathrm{Com}(w, r)\}$.

Construction, cont.

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \lor \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

Algorithm 17 (P)

Input: $x \in \mathcal{L}$ and $w \in R_{\mathcal{L}}(x)$, and CRS $c = (c_1, c_2)$

1. $(sk, vk) \leftarrow \text{Gen}(1^{|x|})$
2. $\pi_A \leftarrow \text{P}_A((x, c_1, vk), w, c_2)$
3. $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
4. Output $\pi = (vk, \pi_A, \sigma)$

Construction, cont.

Recall $\mathcal{L}_A := \{(x, \text{com}, w)\colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^*\colon \text{com} = \text{Com}(w, r)\}$.

Algorithm 17 (P)

Input: $x \in \mathcal{L}$ and $w \in R_{\mathcal{L}}(x)$, and CRS $c = (c_1, c_2)$

1. $(sk, vk) \leftarrow \text{Gen}(1^{|x|})$
2. $\pi_A \leftarrow \text{P}_A((x, c_1, vk), w, c_2)$
3. $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
4. Output $\pi = (vk, \pi_A, \sigma)$

Algorithm 18 (V)

Input: $x \in \{0,1\}^*$, $\pi = (vk, \pi_A, \sigma)$ and a CRS $c = (c_1, c_2)$
Verify that $\text{Vrfy}_{vk}((x, \pi_A), \sigma) = 1$ and $\text{V}_A((x, c_1, vk), c_2, \pi_A) = 1$

Construction, cont.

Recall $\mathcal{L}_A := \{(x, \mathrm{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathrm{com} = \mathrm{Com}(w, r)\}$.

Algorithm 17 (P)

Input: $x \in \mathcal{L}$ and $w \in R_{\mathcal{L}}(x)$, and CRS $c = (c_1, c_2)$

1. $(sk, vk) \leftarrow \mathrm{Gen}(1^{|x|})$
2. $\pi_A \leftarrow \mathrm{P}_A((x, c_1, vk), w, c_2)$
3. $\sigma \leftarrow \mathrm{Sign}_{sk}(x, \pi_A)$
4. Output $\pi = (vk, \pi_A, \sigma)$

Algorithm 18 (V)

Input: $x \in \{0,1\}^*$, $\pi = (vk, \pi_A, \sigma)$ and a CRS $c = (c_1, c_2)$
Verify that $\mathrm{Vrfy}_{vk}((x, \pi_A), \sigma) = 1$ and $\mathrm{V}_A((x, c_1, vk), c_2, \pi_A) = 1$

Claim 19

The proof system (P, V) is an adaptive $\mathcal{NIZK}$ for $\mathcal{L}$, with one-time simulation soundness.

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{ com} = \text{Com}(w, r)\}$.

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \mathrm{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathrm{com} = \mathrm{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathrm{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \mathrm{Com}(vk, z)$.

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \mathrm{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathrm{com} = \mathrm{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathrm{P}_A, \mathrm{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathrm{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathrm{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \mathrm{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \text{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \text{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.
  - $\mathsf{S}_2(x, c = (c_1, c_2), s = (z, sk, vk))$:

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \text{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \text{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.
  - $\mathsf{S}_2(x, c = (c_1, c_2), s = (z, sk, vk))$:
    1. Let $\pi_A \leftarrow \mathsf{P}_A((x, c_1, vk), z, c_2)$

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathsf{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \mathsf{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.
  - $\mathsf{S}_2(x, c = (c_1, c_2), s = (z, sk, vk))$:
    1. Let $\pi_A \leftarrow \mathsf{P}_A((x, c_1, vk), z, c_2)$
    2. $\sigma \leftarrow \mathsf{Sign}_{sk}(x, \pi_A)$

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathsf{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \mathsf{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.
  - $\mathsf{S}_2(x, c = (c_1, c_2), s = (z, sk, vk))$:
    1. Let $\pi_A \leftarrow \mathsf{P}_A((x, c_1, vk), z, c_2)$
    2. $\sigma \leftarrow \mathsf{Sign}_{sk}(x, \pi_A)$
    3. Output $\pi = (vk, \pi_A, \sigma)$

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of (P_A, V_A).
- **Adaptive $\mathcal{ZK}$:**
  - $\text{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \text{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \text{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.
  - $\text{S}_2(x, c = (c_1, c_2), s = (z, sk, vk))$:
    1. Let $\pi_A \leftarrow \text{P}_A((x, c_1, vk), z, c_2)$
    2. $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
    3. Output $\pi = (vk, \pi_A, \sigma)$

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \text{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \text{com} = \text{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \text{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \text{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.
  - $\mathsf{S}_2(x, c = (c_1, c_2), s = (z, sk, vk))$:
    1. Let $\pi_A \leftarrow \mathsf{P}_A((x, c_1, vk), z, c_2)$
    2. $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
    3. Output $\pi = (vk, \pi_A, \sigma)$

    Proof follows by the adaptive WI of $(\mathsf{P}_A, \mathsf{V}_A)$ and the pseudorandomness of Com

Proving Claim 19

Recall $\mathcal{L}_A := \{(x, \mathrm{com}, w)\colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^*\colon \mathrm{com} = \mathrm{Com}(w, r)\}$.

- **Adaptive completeness:** Follows by the adaptive completeness of $(\mathsf{P}_A, \mathsf{V}_A)$.
- **Adaptive $\mathcal{ZK}$:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathrm{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $c_1 = \mathrm{Com}(vk, z)$.
    2. Output $(c = (c_1, c_2), s = (z, sk, vk))$, where c_2 is chosen uniformly at random.
  - $\mathsf{S}_2(x, c = (c_1, c_2), s = (z, sk, vk))$:
    1. Let $\pi_A \leftarrow \mathsf{P}_A((x, c_1, vk), z, c_2)$
    2. $\sigma \leftarrow \mathrm{Sign}_{sk}(x, \pi_A)$
    3. Output $\pi = (vk, \pi_A, \sigma)$

  Proof follows by the adaptive WI of $(\mathsf{P}_A, \mathsf{V}_A)$ and the pseudorandomness of Com
- **Adaptive soundness:** Implicit in the proof of simulation soundness, given next slide.

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \text{com}, w)\colon\ x \in \mathcal{L} \vee \exists r \in \{0,1\}^*\colon\ \text{com} = \text{Com}(w, r)\}$.

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathrm{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathrm{com} = \mathrm{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathrm{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathrm{com} = \mathrm{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assume $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$.

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assume $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$.

Then with all but negligible probability:

- vk' is not the verification key appeared in π ($(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is a strong signature)

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assume $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$.

Then with all but negligible probability:

- vk' is not the verification key appeared in π ($(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is a strong signature)

  $\implies \nexists r \in \{0,1\}^* \text{ s.t. } c_1 = \mathsf{Com}(vk', r)$ (Com is perfectly binding)

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assume $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$.

Then with all but negligible probability:

- vk' is not the verification key appeared in π ($(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is a strong signature)

  $\Longrightarrow \nexists r \in \{0,1\}^* \text{ s.t. } c_1 = \mathsf{Com}(vk', r)$ (Com is perfectly binding)

  $\Longrightarrow x'_A = (x', c_1, vk') \notin \mathcal{L}_A$ (above and $x' \notin \mathcal{L}$)

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assume $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$.

Then with all but negligible probability:

- vk' is not the verification key appeared in π ($(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is a strong signature)

  $\implies \nexists r \in \{0,1\}^* \text{ s.t. } c_1 = \mathsf{Com}(vk', r)$ (Com is perfectly binding)

  $\implies x'_A = (x', c_1, vk') \notin \mathcal{L}_A$ (above and $x' \notin \mathcal{L}$)

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assume $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$.

Then with all but negligible probability:

- vk' is not the verification key appeared in π ($(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is a strong signature)

  $\implies \nexists r \in \{0,1\}^* \text{ s.t. } c_1 = \mathsf{Com}(vk', r)$ (Com is perfectly binding)

  $\implies x'_A = (x', c_1, vk') \notin \mathcal{L}_A$ (above and $x' \notin \mathcal{L}$)

Since c_2 was chosen at random by S_1, the adaptive soundness of $(\mathsf{P}_A, \mathsf{V}_A)$ yields that $\Pr[\mathsf{V}_A(x'_A, c_2, \pi'_A) = 1] = \mathsf{neg}(n)$.

Proving simulation soundness

Recall $\mathcal{L}_A := \{(x, \mathsf{com}, w) \colon x \in \mathcal{L} \vee \exists r \in \{0,1\}^* \colon \mathsf{com} = \mathsf{Com}(w, r)\}$.

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPTM's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $c = (c_1, c_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assume $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$.

Then with all but negligible probability:

- vk' is not the verification key appeared in π ($(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is a strong signature)

  $\implies \nexists r \in \{0,1\}^* \text{ s.t. } c_1 = \mathsf{Com}(vk', r)$ (Com is perfectly binding)

  $\implies x'_A = (x', c_1, vk') \notin \mathcal{L}_A$ (above and $x' \notin \mathcal{L}$)

Since c_2 was chosen at random by S_1, the adaptive soundness of $(\mathsf{P}_A, \mathsf{V}_A)$ yields that $\Pr[\mathsf{V}_A(x'_A, c_2, \pi'_A) = 1] = \mathsf{neg}(n)$.

Adaptive soundness?

Part II

Proof of Knowledge

Proof of Knowledge

The protocol (P,V) is a proof of knowledge for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x, then P^* "knows" $w \in R_{\mathcal{L}}(x)$.

Proof of Knowledge

The protocol (P, V) is a proof of knowledge for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x, then P^* "knows" $w \in R_{\mathcal{L}}(x)$.

Definition 20 (knowledge extractor)

Let (P, V) be an interactive proof for $\mathcal{L} \in \mathcal{NP}$. A probabilistic algorithm E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^*, $\mathsf{E}^{\mathsf{P}^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x)-\eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(\mathsf{P}^*, \mathsf{V})(x) = 1]$.

(P, V) is a proof of knowledge for $\mathcal{L}$ with error η,

Proof of Knowledge

The protocol (P,V) is a proof of knowledge for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x, then P^* “knows" $w \in R_{\mathcal{L}}(x)$.

Definition 20 (knowledge extractor)

Let (P,V) be an interactive proof for $\mathcal{L} \in \mathcal{NP}$. A probabilistic algorithm E is a knowledge extractor for (P,V) and $R_{\mathcal{L}}$ with error $\eta\colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \mathrm{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^*, $\mathsf{E}^{\mathsf{P}^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x)-\eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(\mathsf{P}^*,\mathsf{V})(x) = 1]$.

(P,V) is a proof of knowledge for $\mathcal{L}$ with error η,

- A property of V

Proof of Knowledge

The protocol (P,V) is a proof of knowledge for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x, then P^* “knows" $w \in R_{\mathcal{L}}(x)$.

Definition 20 (knowledge extractor)

Let (P,V) be an interactive proof for $\mathcal{L} \in \mathcal{NP}$. A probabilistic algorithm E is a knowledge extractor for (P,V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \mathrm{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^*, $\mathsf{E}^{\mathsf{P}^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x)-\eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(\mathsf{P}^*,\mathsf{V})(x) = 1]$.

(P,V) is a proof of knowledge for $\mathcal{L}$ with error η,

- A property of V
- Why do we need it?

Proof of Knowledge

The protocol (P,V) is a proof of knowledge for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x, then P^* “knows" $w \in R_{\mathcal{L}}(x)$.

Definition 20 (knowledge extractor)

Let (P,V) be an interactive proof for $\mathcal{L} \in \mathcal{NP}$. A probabilistic algorithm E is a knowledge extractor for (P,V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^*, $\mathsf{E}^{\mathsf{P}^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x)-\eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(\mathsf{P}^*,\mathsf{V})(x) = 1]$.

(P,V) is a proof of knowledge for $\mathcal{L}$ with error η,

- A property of V
- Why do we need it?

Proof of Knowledge

The protocol (P, V) is a proof of knowledge for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x, then P^* "knows" $w \in R_{\mathcal{L}}(x)$.

Definition 20 (knowledge extractor)

Let (P, V) be an interactive proof for $\mathcal{L} \in \mathcal{NP}$. A probabilistic algorithm E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^*, $\mathsf{E}^{\mathsf{P}^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(\mathsf{P}^*, \mathsf{V})(x) = 1]$.

(P, V) is a proof of knowledge for $\mathcal{L}$ with error η,

- A property of V
- Why do we need it? Authentication schmes

Proof of Knowledge

The protocol (P, V) is a proof of knowledge for $\mathcal{L} \in \mathcal{NP}$, if a P^* convinces V to accept x, then P^* “knows" $w \in R_{\mathcal{L}}(x)$.

Definition 20 (knowledge extractor)

Let (P, V) be an interactive proof for $\mathcal{L} \in \mathcal{NP}$. A probabilistic algorithm E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^*, $\mathsf{E}^{\mathsf{P}^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x)-\eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(\mathsf{P}^*, \mathsf{V})(x) = 1]$.

(P, V) is a proof of knowledge for $\mathcal{L}$ with error η,

- A property of V
- Why do we need it? Authentication schmes
- Why only deterministic P^*?

Examples

Claim 21

The $\mathcal{ZK}$ proof we've seen in class for $\mathcal{GI}$, has a knowledge extractor with error $\frac{1}{2}$.

Examples

Claim 21

The $\mathcal{ZK}$ proof we've seen in class for $\mathcal{GI}$, has a knowledge extractor with error $\frac{1}{2}$.

Proof: ?

Examples

Claim 21

The $\mathcal{ZK}$ proof we've seen in class for $\mathcal{GI}$, has a knowledge extractor with error $\frac{1}{2}$.

Proof: ?

Claim 22

The $\mathcal{ZK}$ proof we've seen in class for 3COL, has a knowledge extractor with error $\frac{1}{|E|}$.

Examples

Claim 21

The $\mathcal{ZK}$ proof we've seen in class for $\mathcal{GI}$, has a knowledge extractor with error $\frac{1}{2}$.

Proof: ?

Claim 22

The $\mathcal{ZK}$ proof we've seen in class for 3COL, has a knowledge extractor with error $\frac{1}{|E|}$.

Proof: ?