Foundation of Cryptography (0368-4162-01), Lecture 6

More on Zero Knowledge

Iftach Haitner, Tel Aviv University

December 20, 2011

Part I

Non-Interactive Zero Knowledge

Interaction is crucial for ZK

Claim 1

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

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

Interaction is crucial for ZK

Claim 1

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

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

Proof: HW

Interaction is crucial for ZK

Claim 1

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

[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
       $\{\langle(\mathrm{P}(w_x^1),\mathrm{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{\langle(\mathrm{P}(w_x^2),\mathrm{V}^*)(x)\rangle\}_{x\in\mathcal{L}}$,
       for any $\{w_x^1 \colon (x,w_x) \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$ and $\{w_x^2 \colon (x,w_x) \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$

Interaction is crucial for ZK

Claim 1

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

[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
      $\{\langle(\mathsf{P}(w_x^1), \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{\langle(\mathsf{P}(w_x^2), \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}}$,
      for any $\{w_x^1 \colon (x, w_x) \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$ and
      $\{w_x^2 \colon (x, w_x) \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$
   2. Witness Hiding

Interaction is crucial for ZK

Claim 1

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

[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 $\{\langle(\mathsf{P}(w_x^1),\mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{\langle(\mathsf{P}(w_x^2),\mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}}$, for any $\{w_x^1 : (x,w_x) \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$ and $\{w_x^2 : (x,w_x) \in R_{\mathcal{L}}(x)\}_{x\in\mathcal{L}}$
   2. Witness Hiding
   3. Non-interactive “zero knowledge"

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

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

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, where $w(x) \in R_{\mathcal{L}}(x)$ for any $x \in \mathcal{L}$ (w is an arbitrary function)
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$
- **ZK:** $\exists$ PPT S s.t. $\{(x, c, \mathsf{P}(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

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

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, where $w(x) \in R_{\mathcal{L}}(x)$ for any $x \in \mathcal{L}$ (w is an arbitrary function)
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$
- **ZK:** $\exists$ PPT S s.t. $\{(x, c, \mathsf{P}(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

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

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

The *non interactive* PPT's (P, V) is a NIZK for $\mathcal{L} \in$ NP, if $\exists \ell \in$ poly s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, where $w(x) \in R_{\mathcal{L}}(x)$ for any $x \in \mathcal{L}$ (w is an arbitrary function)
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$
- **ZK:** $\exists$ PPT S s.t. $\{(x, c, \mathsf{P}(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRS)
- CRS is chosen by the simulator

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

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

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}(x, w(x), c)) = 1] \geq 2/3$, where $w(x) \in R_{\mathcal{L}}(x)$ for any $x \in \mathcal{L}$ (w is an arbitrary function)
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathsf{V}(x, c, \mathsf{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$
- ZK: $\exists$ PPT S s.t. $\{(x, c, \mathsf{P}(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, \mathsf{S}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRS)
- CRS is chosen by the simulator
- What does the definition stand for?

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

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

- **Completeness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathrm{V}(x, c, \mathrm{P}(x, w(x), c)) = 1] \geq 2/3$, where $w(x) \in R_{\mathcal{L}}(x)$ for any $x \in \mathcal{L}$ (w is an arbitrary function)
- **Soundness:** $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}}[\mathrm{V}(x, c, \mathrm{P}^*(x, c)) = 1] \leq 1/3$, for any P^* and $x \notin \mathcal{L}$
- **ZK:** $\exists$ PPT S s.t. $\{(x, c, \mathrm{P}(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, \mathrm{S}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRS)
- CRS is chosen by the simulator
- What does the definition stand for?

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:

- Prover sees c^H, and outputs a proof π and a set on indices $\mathcal{I}$
- Verifier only sees the bits in c^H that are indexed by $\mathcal{I}$
- 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:

- Prover sees c^H, and outputs a proof π and a set on indices $\mathcal{I}$
- Verifier only sees the bits in c^H that are indexed by $\mathcal{I}$
- 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:

- Prover sees c^H, and outputs a proof π and a set on indices $\mathcal{I}$
- Verifier only sees the bits in c^H that are indexed by $\mathcal{I}$
- 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 NIZK for HC - Directed Graph Hamiltonicity, in the HBM, and then transfer it into a NIZK in the standard model.

Implies a (standard model) NIZK for all NP

Useful Matrix

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

Useful Matrix

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

Useful Matrix

- Permutation matrix: an $n \times n$ Boolean matrix, where each row/column contains a single 1
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)
- An $n^3 \times n^3$ Boolean matrix is called *useful*: if it contains a generalized $n \times n$ Hamiltonian sub matrix, and all the other entries are zeros

Useful Matrix

- Permutation matrix: an $n \times n$ Boolean matrix, where each row/column contains a single 1
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)
- An $n^3 \times n^3$ Boolean matrix is called *useful*: if it contains a generalized $n \times n$ Hamiltonian sub matrix, and all the other entries are zeros

Claim 3

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

Proving Claim 3

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

Proving Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.
- 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 Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.
- 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 Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.
- 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 Claim 3

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

NIZK for Hamiltonicity in HBM

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

NIZK for Hamiltonicity in HBM

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

NIZK for Hamiltonicity in HBM

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

NIZK for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$
- 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: G and a cycle C in G. A 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 $\phi = \perp$ Otherwise, let H be the (generalized) $n \times n$ sub matrix containing the hamiltonian cycle in T.
2. Set $\mathcal{I} = T \setminus H$ (i.e., , reveal the bits of T outside of H)
3. Choose $\phi \leftarrow \Pi_n$, s.t. C is mapped to the cycle in H
4. Add all the entries in H corresponding to non edges in G (with respect to ϕ) to $\mathcal{I}$
5. Output $\pi = (\mathcal{I}, \phi)$

NIZK for Hamiltonicity in HBM cont.

Algorithm 5 (V)

Input: a graph G, index set $\mathcal{I} \subseteq [n^3] \times [n^3]$, ordered set $\{T_i\}_{i \in \mathcal{I}}$ and a mapping ϕ

1. If all the bits of T are revealed and T is not useful, accept. Otherwise,
2. Verify that $\exists$ $n \times n$ submatrix $H \subseteq T$ with all entries in $T \setminus H$ are zeros.
3. Verify that $\phi \in \Pi_n$, and that all the entries of H not corresponding (according to ϕ) to edges of G are zeros

NIZK for Hamiltonicity in HBM cont.

Algorithm 5 (V)

Input: a graph G, index set $\mathcal{I} \subseteq [n^3] \times [n^3]$, ordered set $\{T_i\}_{i \in \mathcal{I}}$ and a mapping ϕ

1. If all the bits of T are revealed and T is not useful, accept. Otherwise,
2. Verify that $\exists\ n \times n$ submatrix $H \subseteq T$ with all entries in $T \setminus H$ are zeros.
3. Verify that $\phi \in \Pi_n$, and that all the entries of H not corresponding (according to ϕ) to edges of G are zeros

Claim 6

The above protocol is a perfect NIZK for 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, according to the right distribution.
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
3. Set $\mathcal{I} = T \setminus H$
4. Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
5. Add the entries in H corresponding to non edges in G to $\mathcal{I}$
6. Output $\pi = (T, \mathcal{I}, \phi)$

Algorithm 7 (S)

Input: G

1. Choose T at random, according to the right distribution.
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
3. Set $\mathcal{I} = T \setminus H$
4. Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
5. Add the entries in H corresponding to non edges in G to $\mathcal{I}$
6. Output $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful T's.

Algorithm 7 (S)

Input: G

1. Choose T at random, according to the right distribution.
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
3. Set $\mathcal{I} = T \setminus H$
4. Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
5. Add the entries in H corresponding to non edges in G to $\mathcal{I}$
6. 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, according to the right distribution.
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
3. Set $\mathcal{I} = T \setminus H$
4. Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
5. Add the entries in H corresponding to non edges in G to $\mathcal{I}$
6. 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 is both cases

Algorithm 7 (S)

Input: G

1. Choose T at random, according to the right distribution.
2. If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
3. Set $\mathcal{I} = T \setminus H$
4. Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
5. Add the entries in H corresponding to non edges in G to $\mathcal{I}$
6. 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 is both cases
- Hence, the simulation is perfect

Section 2

From HBM to Standard NIZK

trapdoor permutations

Definition 8 (trapdoor permutations)

A triplet (G, f, Inv), where G is a PPT, and f and Inv are polynomial-time computable functions, is a family of trapdoor permutation (TDP), if:

1. On input 1^n, $G(1^n)$ 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 $pk \in \mathrm{Supp}(G(1^n)_2)$.
3. $\mathrm{Inv}(sk, \cdot) \equiv f_{pk}^{-1}$ for every $(sk, pk) \in \mathrm{Supp}(G(1^n))$
4. For any PPT A,
$\Pr_{x \leftarrow \{0,1\}^n, pk \leftarrow G(1^n)_2}[\mathrm{A}(pk, x) = f_{pk}^{-1}(x)] = \mathrm{neg}(n)$

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n.

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n.

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

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n.

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n] : \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$ (equals $(p-1)(q-1)$ for $n = pq$ with $p, q \in \mathrm{P}$)

example, RSA

In the following $n \in \mathbb{N}$ 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 $p, q \in \mathrm{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ is a permutation over $\mathbb{Z}_n^*$.

example, RSA

In the following $n \in \mathbb{N}$ 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 $p, q \in \mathrm{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ 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 $n \in \mathbb{N}$ 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 $p, q \in \mathrm{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ 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 9 (RSA)

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

example, RSA

In the following $n \in \mathbb{N}$ 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 $p, q \in \mathrm{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ 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 9 (RSA)

- $G(p, q)$ 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 $n \in \mathbb{N}$ 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 $p, q \in \mathrm{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ 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 9 (RSA)

- $G(p, q)$ 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. Other direction?

The transformation

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

The transformation

- Let $(\mathrm{P}_H, \mathrm{V}_H)$ be a HBM NIZK for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let (G, f, Inv) be a TDP and let b be an hardcore bit for f. For simplicity we assume $G(1^n)$ 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 $(\mathrm{P}_H, \mathrm{V}_H)$ be a HBM NIZK for $\mathcal{L}$, and let $\ell(n)$ be the length of the CRS used for $x \in \{0,1\}^n$.
- Let (G, f, Inv) be a TDP and let b be an hardcore bit for f. For simplicity we assume $G(1^n)$ 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 NIZK (P, V) for $\mathcal{L}$, with the same completeness and “not too large" soundness error.

The protocol

Algorithm 10 (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 $(sk, pk) \leftarrow G(sk)$ 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 10 (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 $(sk, pk) \leftarrow G(sk)$ 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 $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$

Algorithm 11 (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 $\mathrm{V}_H(x, \pi_H, \mathcal{I}, c^H)$, where $c_i^H = b(z_i)$ for every $i \in \mathcal{I}$.

Claim 12

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

Claim 12

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a NIZK for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a 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 12

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a NIZK for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a 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 12

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a NIZK for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a 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 12

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a NIZK for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a 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 12

Assuming that $(\mathsf{P}_H, \mathsf{V}_H)$ is a NIZK for $\mathcal{L}$ in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a 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 13 (S)

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

- Let $(\pi_H, \mathcal{I}, c^H) = \mathrm{S}_H(x)$, where S_H is the simulator of $(\mathrm{P}_H, \mathrm{V}_H)$
- Output $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
  - $pk \leftarrow G(U_n)$
  - 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 13 (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
  - $pk \leftarrow G(U_n)$
  - 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.

- Exists 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 13 (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
  - $pk \leftarrow G(U_n)$
  - 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.

- Exists efficient M s.t. $M(\mathsf{S}_H(x)) \equiv \mathsf{S}(x)$ and $M(\mathsf{P}_H(x, w_x)) \approx_c \mathsf{P}(x, w_x)$
- Distinguishing $\mathsf{P}(x, w_x)$ from $\mathsf{S}(x)$ is hard

Section 3

Adaptive NIZK

Adaptive NIZK

x is chosen *after* the CRS.

Adaptive NIZK

x is chosen *after* the CRS.

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

Adaptive NIZK

x is chosen *after* the CRS.

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

Adaptive NIZK

x is chosen *after* the CRS.

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

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

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 $(x, c, \mathsf{S}_2(x, c, s))$

Adaptive NIZK

x is chosen *after* the CRS.

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

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

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 $(x, c, \mathsf{S}_2(x, c, s))$

- Adaptive completeness and soundness are easy to achieve from any non-adaptive NIZK.

- Adaptive completeness and soundness are easy to achieve from any non-adaptive NIZK.
- Not every NIZK is adaptive (but the above protocol is).

- Adaptive completeness and soundness are easy to achieve from any non-adaptive NIZK.
- Not every NIZK is adaptive (but the above protocol is).

Theorem 14

Assume TDP exist, then every NP *language has an adaptive* NIZK *with perfect completeness and negligible soundness error.*

- Adaptive completeness and soundness are easy to achieve from any non-adaptive NIZK.
- Not every NIZK is adaptive (but the above protocol is).

Theorem 14

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

In the following, when saying adaptive NIZK, we mean negligible completeness and soundness error.

Section 4

Simulation Sound NIZK

Simulation Soundness

A NIZK system (P, V) for $\mathcal{L}$ has *(one-time) simulation soundness*, if $\exists$ a pair of PPT's $S = (S_1, S_2)$ satisfying the ZK property of P with respect to $\mathcal{L}$, such that the following holds $\forall$ pair of PPT's (P_1^*, P_2^*): let

Experiment 15 (Exp^n_{V,S,P^*})

1. $(c, s) \leftarrow S_1(1^n)$
2. $(x, p) \leftarrow P_1^*(1^n, c)$
3. $\pi \leftarrow S_2(x, c, s)$
4. $(x', \pi') \leftarrow P_2^*(p, \pi)$
5. Output (c, x, π, x', π')

Simulation Soundness

A NIZK system (P, V) for $\mathcal{L}$ has *(one-time) simulation soundness*, if $\exists$ a pair of PPT's $S = (S_1, S_2)$ satisfying the ZK property of P with respect to $\mathcal{L}$, such that the following holds $\forall$ pair of PPT's (P_1^*, P_2^*): let

Experiment 15 (Exp^n_{V,S,P^*})

1. $(c, s) \leftarrow S_1(1^n)$
2. $(x, p) \leftarrow P_1^*(1^n, c)$
3. $\pi \leftarrow S_2(x, c, s)$
4. $(x', \pi') \leftarrow P_2^*(p, \pi)$
5. Output (c, x, π, x', π')

We require $\Pr[(r, x, \pi, x', \pi') \leftarrow \mathrm{Exp}^n_{V,S,P^*} : x' \notin \mathcal{L} \wedge V(x', \pi', c) = 1 \wedge (x', \pi') \neq (x, \pi)] = \mathrm{neg}(n)$.

- Even for $x \notin \mathcal{L}$, hard to generate additional false proofs

- Even for $x \notin \mathcal{L}$, hard to generate additional false proofs
- Definition only considers efficient provers

- Even for $x \notin \mathcal{L}$, hard to generate additional false proofs
- Definition only considers efficient provers
- (P, V) might be adaptive or non-adaptive

- Even for $x \notin \mathcal{L}$, hard to generate additional false proofs
- Definition only considers efficient provers
- (P, V) might be adaptive or non-adaptive
- Adaptive NIZK guarantees weak type of simulation soundness

- Even for $x \notin \mathcal{L}$, hard to generate additional false proofs
- Definition only considers efficient provers
- (P, V) might be adaptive or non-adaptive
- Adaptive NIZK guarantees weak type of simulation soundness
- Does the adaptive NIZK we seen in class have simulation soundness?

Construction

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

Ingredients:

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

Construction

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

Ingredients:

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

Construction

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

Ingredients:

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

Construction

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

Ingredients:

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

Construction

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

Ingredients:

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

Construction

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

Ingredients:

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

Construction

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

Ingredients:

1. Strong signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ (one time suffice)
2. Non-interactive, perfectly-binding commitment Com
   - Pseudorandom range: for some $\ell \in$ poly $\{\mathsf{Com}(s, r \leftarrow \{0,1\}^{\ell(|s|)}\}_{s \in \{0,1\}^*} \approx_c \{u \leftarrow \{0,1\}^{\ell(|s|)}\}_{s \in \{0,1\}^*}$
     * implied by OWP (or TDP)
   - Negligible support: a random string is a valid commitment only with negligible probability.
     * achieved from any commitment scheme by committing to the same value many times

Construction

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

Ingredients:

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

Construction

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

Ingredients:

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

Algorithm 16 (P)

Input: $x \in \mathcal{L}$ and $w \in R_{\mathcal{L}}(x)$, and CRS $r = (r_1, r_2)$

1. $(sk, vk) \leftarrow \mathrm{Gen}(1^{|x|})$
2. $\pi_A \leftarrow \mathrm{P}_A((x, r_1, vk), w, r_2)$
3. $\sigma \leftarrow \mathrm{Sign}_{sk}(x, \pi_A)$
4. Output $\pi = (vk, \pi_A, \sigma)$

Algorithm 16 (P)

Input: $x \in \mathcal{L}$ and $w \in R_{\mathcal{L}}(x)$, and CRS $r = (r_1, r_2)$

1. $(sk, vk) \leftarrow \mathrm{Gen}(1^{|x|})$
2. $\pi_A \leftarrow \mathrm{P}_A((x, r_1, vk), w, r_2)$
3. $\sigma \leftarrow \mathrm{Sign}_{sk}(x, \pi_A)$
4. Output $\pi = (vk, \pi_A, \sigma)$

Algorithm 17 (V)

Input: $x \in \{0,1\}^*$, $\pi = (vk, \pi_A, \sigma)$ and a CRS $r = (r_1, r_2)$
Verify that $\mathrm{Vrfy}_{vk}((x, \pi), \sigma) = 1$ and $\mathrm{V}_A((x, r_1, vk), r_2, \pi_A) = 1$

Algorithm 16 (P)

Input: $x \in \mathcal{L}$ and $w \in R_{\mathcal{L}}(x)$, and CRS $r = (r_1, r_2)$

1. $(sk, vk) \leftarrow \mathrm{Gen}(1^{|x|})$
2. $\pi_A \leftarrow \mathrm{P}_A((x, r_1, vk), w, r_2)$
3. $\sigma \leftarrow \mathrm{Sign}_{sk}(x, \pi_A)$
4. Output $\pi = (vk, \pi_A, \sigma)$

Algorithm 17 (V)

Input: $x \in \{0,1\}^*$, $\pi = (vk, \pi_A, \sigma)$ and a CRS $r = (r_1, r_2)$
Verify that $\mathrm{Vrfy}_{vk}((x, \pi), \sigma) = 1$ and $\mathrm{V}_A((x, r_1, vk), r_2, \pi_A) = 1$

Claim 18

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

Proving Claim 18

- **Adaptive Completeness:** Clear

Proving Claim 18

- **Adaptive Completeness:** Clear
- **Adaptive ZK:**
  - $S_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathrm{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $r_1 = \mathrm{Com}(vk, z)$.
    2. Output $(r = (r_1, r_2), s = (z, sk, vk))$, where r_2 is chosen uniformly at random

Proving Claim 18

- **Adaptive Completeness:** Clear
- **Adaptive** ZK**:**
  - $S_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathrm{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $r_1 = \mathrm{Com}(vk, z)$.
    2. Output $(r = (r_1, r_2), s = (z, sk, vk))$, where r_2 is chosen uniformly at random
  - $S_2(x, r, s = (z, sk, vk))$:
    1. let $\pi_A \leftarrow \mathrm{P}_A((x, r_1, vk), z, r_2)$
    2. $\sigma \leftarrow \mathrm{Sign}_{sk}(x, \pi_A)$
    3. Output $\pi = (vk, \pi_A, \sigma)$

Proving Claim 18

- **Adaptive Completeness:** Clear
- **Adaptive** ZK**:**
  - $\mathsf{S}_1(1^n)$:
    1. Let $(sk, vk) \leftarrow \mathsf{Gen}(1^n)$, $z \leftarrow \{0,1\}^{\ell(n)}$ and $r_1 = \mathsf{Com}(vk, z)$.
    2. Output $(r = (r_1, r_2), s = (z, sk, vk))$, where r_2 is chosen uniformly at random
  - $\mathsf{S}_2(x, r, s = (z, sk, vk))$:
    1. let $\pi_A \leftarrow \mathsf{P}_A((x, r_1, vk), z, r_2)$
    2. $\sigma \leftarrow \mathsf{Sign}_{sk}(x, \pi_A)$
    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 below

Proving simulation soundness

Let $\mathrm{P}^* = (\mathrm{P}_1^*, \mathrm{P}_2^*)$ be a pair of PPT's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $r = (r_1, r_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathrm{Exp}^n_{\mathrm{V},\mathrm{S},\mathrm{P}^*}$.

Proving simulation soundness

Let $\mathrm{P}^* = (\mathrm{P}_1^*, \mathrm{P}_2^*)$ be a pair of PPT's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $r = (r_1, r_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathrm{Exp}^n_{\mathrm{V,S,P^*}}$.

Assuming $\mathrm{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$, then with save but negligible probability:

- vk' is not the signing key in π

Proving simulation soundness

Let $\mathsf{P}^* = (\mathsf{P}^*_1, \mathsf{P}^*_2)$ be a pair of PPT's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $r = (r_1, r_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathsf{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assuming $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$, then with save but negligible probability:

- vk' is not the signing key in π
- $\nexists z \in \{0,1\}^*$ s.t. $r_1 = \mathsf{Com}(vk', z)$

Proving simulation soundness

Let $\mathrm{P}^* = (\mathrm{P}_1^*, \mathrm{P}_2^*)$ be a pair of PPT's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $r = (r_1, r_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathrm{Exp}^n_{\mathrm{V},\mathrm{S},\mathrm{P}^*}$.

Assuming $\mathrm{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$, then with save but negligible probability:

- vk' is not the signing key in π
- $\nexists z \in \{0,1\}^*$ s.t. $r_1 = \mathrm{Com}(vk', z)$
- $x'_A = (x', r_1, vk') \notin \mathcal{L}_A$

Proving simulation soundness

Let $\mathsf{P}^* = (\mathsf{P}_1^*, \mathsf{P}_2^*)$ be a pair of PPT's attacking the simulation soundness of (V, S) with respect to $\mathcal{L}$, and let $r = (r_1, r_2)$, x , π, x' and $\pi' = (vk', \pi'_A, \sigma')$ be the values generated by a random execution of $\mathrm{Exp}^n_{\mathsf{V},\mathsf{S},\mathsf{P}^*}$.

Assuming $\mathsf{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$, $x' \notin \mathcal{L}$ and $(x', \pi') \neq (x, \pi)$, then with save but negligible probability:

- vk' is not the signing key in π
- $\nexists z \in \{0,1\}^*$ s.t. $r_1 = \mathrm{Com}(vk', z)$
- $x'_A = (x', r_1, vk') \notin \mathcal{L}_A$

Since r_2 was chosen at random by S_1, the adaptive soundness of $(\mathsf{P}_A, \mathsf{V}_A)$ yields that $\Pr[\mathsf{V}_A(x'_A, r_2, \pi'_A) = 1] = \mathrm{neg}(n)$.

Part II

Proof of Knowledge

Proof of Knowledge

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

Proof of Knowledge

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

Definition 19 (knowledge extractor)

Let (P,V) be an interactive proof $\mathcal{L} \in \mathrm{NP}$. A probabilistic machine E is a knowledge extractor for (P,V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in$ 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 $\delta(x) = \Pr[(\mathsf{P}^*,\mathsf{V})(x) = 1]$.

Proof of Knowledge

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

Definition 19 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \mathrm{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta\colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in$ 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 $\delta(x) = \Pr[(\mathsf{P}^*, \mathsf{V})(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

- A property of V

Proof of Knowledge

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

Definition 19 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \mathrm{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in$ 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 $\delta(x) = \Pr[(\mathsf{P}^*, \mathsf{V})(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

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

Proof of Knowledge

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

Definition 19 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \mathrm{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta \colon \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in$ 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 $\delta(x) = \Pr[(\mathsf{P}^*, \mathsf{V})(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

- A property of V
- Why do we need it? Proving that you know the password

Proof of Knowledge

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

Definition 19 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \mathrm{NP}$. A probabilistic machine 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 $\delta(x) = \Pr[(\mathsf{P}^*, \mathsf{V})(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

- A property of V
- Why do we need it? Proving that you know the password
- Relation to ZK

Examples

Claim 20

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

Examples

Claim 20

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

Proof: ?

Examples

Claim 20

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

Proof: ?

Claim 21

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

Examples

Claim 20

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

Proof: ?

Claim 21

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

Proof: ?