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

Handout Mode

Iftach Haitner, Tel Aviv University

Tel Aviv University.

April 15, 2019
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, then $\mathcal{L} \in \mathcal{BPP}$. \(^{a}\)

\(^{a}\)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

1.1 Witness Indistinguishability

$$\{ \langle (P(w^1_x), V^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}} \approx_c \{ \langle (P(w^2_x), V^*)(x) \rangle_{V^*} \}_{x \in \mathcal{L}},$$

for any $\{w^1_x \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$ and $\{w^2_x \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

1.2 Witness hiding

1.3 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 $L \in \mathcal{NP}$, if $\exists \ell \in \text{poly}$ s.t.

- **Completeness:** $\Pr_{c \leftarrow \{0, 1\}^\ell(x)} [V(x, c, P(x, w(x), c)) = 1] \geq \frac{2}{3}$, for any $x \in L$ and $w(x) \in R_L(x)$.

- **Soundness:** $\Pr_{c \leftarrow \{0, 1\}^\ell(x)} [V(x, c, P^*(x, c)) = 1] \leq \frac{1}{3}$, for any $P^*$ and $x \notin L$.

- **Zero knowledge:** $\exists$ PPTM $S$ s.t.
  $$\{(x, c, P(x, w(x), c))_{c \leftarrow \{0, 1\}^\ell(x)}\}_{x \in L} \approx_c \{x, S(x)\}_{x \in L}$$
  for any poly-bounded function $w$ with $w(x) \in R_L(x)$.

- $c$ – 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 $x \notin L$?
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.

Let $c^H$ be the “hidden” CRS:

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

Soundness, completeness and ZK are naturally defined.

- We give a $\mathsf{NIZK}$ for $HC$, Directed Graph Hamiltonicity, in the HBM, and then transfer it into a $\mathsf{NIZK}$ for $HC$ in the standard model.
- The latter implies a $\mathsf{NIZK}$ for all $NP$. 
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, $\Pr[T$ is useful$] \in \Omega(n^{-3/2})$. 
Proving $\Pr[T \text{ is useful}] \in \Omega(n^{-3/2})$

- The expected number of ones (entries) in $T$ is $n^6 \cdot n^{-5} = n$.
- By convergent to the Gaussian distribution, $T$ contains exactly $n$ ones w.p. $\theta(1/\sqrt{n})$.
- Each row/column 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 $G = ([n], E)$
- We assume w.l.o.g. 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 $\pi = \bot$.

2. Otherwise, let $H$ be the (generalized) $n \times n$ sub-matrix containing the hamiltonian cycle in $T$.
   2.1 Set $\mathcal{I} = T \setminus H$ (i.e., reveal the bits of $T$ outside of $H$).
   2.2 Choose $\phi \leftarrow \Pi_n$ s.t. $C$ is mapped to the cycle in $H$.
   2.3 Add the entries in $H$ corresponding to non edges in $G$ (wrt. $\phi$) to $\mathcal{I}$.

3. Output $\pi = \phi$ and $\mathcal{I}$. 

Iftach Haitner (TAU)  Foundation of Cryptography  April 15, 2019  10 / 36
Algorithm 5 (V)

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

Accept if $\phi = \bot$, 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 $\phi$, 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 NIZK for $HC$ in the HBM, with perfect completeness and soundness error $1 - \Omega(n^{-3/2})$. 
Proving Claim 6

- Completeness: Clear.

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

  Hence, $\phi^{-1}$ 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 $I = n^3 \times n^3$ and $\phi = \bot$.
3. Otherwise,
   3.1 Set $I = T \setminus H$ (where $H$ is the hamiltonian sub-matrix in $T$).
   3.2 Let $\phi \leftarrow \Pi_n$. Replace all entries of $H$ with zeros.
   3.3 Add the entries in $H$ corresponding to non edges in $G$ to $I$.
4. Output $\pi = (T, I, \phi)$.

- Perfect simulation for non-useful $T$’s.
- For useful $T$, the location of $H$ is uniform in the real and simulated case.
- $\phi$ is a random element in $\Pi_n$ in both (real and simulated) cases (?)
- Hence, the simulation is perfect!
Section 2

From HBM to Standard NIZK
Subsection 1

Trapdoor Permutations
Definition 8 (trapdoor permutations)

A triplet \((G, f, \text{Inv})\), where \(G\) is a PPTM, and \(f\) and \(\text{Inv}\) are poly-time computable, 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 \(pk \in \text{Supp}(G(1^n)_2)\).
3. \(\text{Inv}_{sk} = \text{Inv}(sk, \cdot) \equiv f_{pk}^{-1}\) for every \((sk, pk) \in \text{Supp}(G(1^n))\).
4. For any PPTM \(A\),
   \[\Pr_{x \leftarrow \{0,1\}^n, pk \leftarrow G(1^n)_2} \left[ A(pk, x) = f_{pk}^{-1}(x) \right] = \text{neg}(n)\]
Hardcore Predicates for Trapdoor Permutations

**Definition 9 (hardcore predicates for TDP)**

A polynomial-time computable \( b : \{0, 1\}^n \rightarrow \{0, 1\} \) is a **hardcore predicate** of a TDP \((G, f, \text{Inv})\), if

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

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 $N$.

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

Definition 10 (RSA)

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

Factoring is easy $\implies$ RSA is easy. The other direction?
Subsection 2

The Transformation
The transformation

Let $(P_H, V_H)$ 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 $(G, f, \text{Inv})$ be a TDP and let $b$ be an hardcore bit for it. For simplicity, assume that $G(1^n)$ chooses $(sk, pk)$ as follows:

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

where $PK: \{0, 1\}^n \mapsto \{0, 1\}^n$ 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 $(sk, pk) \leftarrow G(sk)$ and compute $c^H = (b(z_1 = f^{-1}_{pk}(c_1)), \ldots, b(z_{\ell(n)} = f^{-1}_{pk}(c_\ell)))$

2. Let $(\pi_H, \mathcal{I}) \leftarrow P_H(x, w, c^H)$ 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 $V_H(x, \pi_H, \mathcal{I}, c^H)$, where $c^H_i = b(z_i)$ for every $i \in \mathcal{I}$. 
Claim 13

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

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

For every \(pk \in \{0, 1\}^n\): \(\left(b(f^{-1}_{pk}(c_1)), \ldots, b(f^{-1}_{pk}(c_\ell))\right)_{c \leftarrow \{0, 1\}^{np}}\) 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:?
Algorithm 14 (S)

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

- Let $(\pi_H, I, c^H) = S_H(x)$, where $S_H$ is the simulator of $(P_H, V_H)$
- Output $(c, (\pi_H, I, pk, \{z_i\}_{i \in I}))$, 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 $i \in I$, and a random value in $\{0, 1\}^n$ otherwise.

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

- Direct solution for our $NIZK$
- An “adaptive” $NIZK$
Section 3

Adaptive NIZK
Adaptive NIZK

\( x \) is chosen after the CRS.

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

- **Soundness:** \( \forall f : \{0, 1\}^{\ell(n)} \rightarrow \{0, 1\}^n \) and \( P^* \)
  \[
  \Pr_{c \leftarrow \{0,1\}^{\ell(n)}; x=f(c)} [V(x, c, P^*(c)) = 1 \land x \notin \mathcal{L}] \leq 1/3
  \]

- **ZK:** \( \exists \) pair of PPTM’s \((S_1, S_2)\) s.t. \( \forall f : \{0, 1\}^{\ell(n)} \rightarrow \mathcal{L} \cap \{0, 1\}^n \)
  \[
  \{(c \leftarrow \{0, 1\}^{\ell(n)}, x = f(c), P(x, w(x)))\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}.
  \]

where \( S^f(n) \) is the output of the following process

1. \((c, s) \leftarrow S_1(1^n)\)
2. \(x = f(c)\)
3. Output \((c, x, S_2(x, c, s))\)

Why do we need \( s \)?
Adaptive \( \mathcal{NIZK} \), cont.

- Adaptive completeness and soundness are easy to achieve from any non-adaptive \( \mathcal{NIZK} \).(?)
- 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 \textit{NIZK} system \((P, V)\) for \(\mathcal{L}\) has (one-time) simulation soundness, if \(\exists\) a pair of PPTM’s \(S = (S_1, S_2)\) that satisfies the \textit{ZK} property of \(P\) with respect to \(\mathcal{L}\), and in addition

\[
\Pr \left[ (c, x, \pi, x', \pi') \leftarrow \text{Exp}_{V, S, P^*} \quad \left[ x' \notin \mathcal{L} \land V(x', \pi', c) = 1 \land (x', \pi') \neq (x, \pi) \right] = \text{neg}(n) \right]
\]

for any pair of PPTM’s \(P^* = (P_1^*, P_2^*)\).

**Experiment 16** \((\text{Exp}_{V, S, P^*}^n)\)

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 \((c, x, \pi, x', \pi')\)
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 NIZK guarantees weak type of simulation soundness (hard to fake proofs for simulated CRS and predefined \(x'\)) (?)
- Does the adaptive NIZK we seen have simulation soundness?
Construction

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

**Ingredients:**

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

2. Non-interactive, perfectly-binding commitment \( \text{Com} \).
   - Pseudorandom range: for some \( \ell \in \text{poly} \)
     \[
     \{ \text{Com}(w, r \leftarrow \{0, 1\}^{\ell(|w|)}) \}_{w \in \{0,1\}^*} \approx_c \{ u \leftarrow \{0, 1\}^{\ell(|w|)} \}_{w \in \{0,1\}^*}
     \]
     * 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 \( \mathcal{NIZK} (P_A, V_A) \) for \( L_A := \{(x, \text{com}, w) : x \in L \lor \exists r \in \{0, 1\}^* : \text{com} = \text{Com}(w, r) \} \in \mathcal{NP} \)
   * adaptive WI suffices
Construction, cont.

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

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. $\pi_A \leftarrow P_A((x, c_1, vk), w, c_2)$
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 $V_A((x, c_1, vk), c_2, \pi_A) = 1$

Claim 19

The proof system $(P, V)$ is an adaptive NIZK for $\mathcal{L}$, with one-time simulation soundness.
Proving Claim 19

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

- **Adaptive completeness**: Follows by the adaptive completeness of $(P_A, V_A)$.

- **Adaptive ZK**:
  
  - $S_1(1^n)$:
    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.

  - $S_2(x, c = (c_1, c_2), s = (z, sk, vk))$:
    1. Let $\pi_A \leftarrow P_A((x, c_1, vk), z, c_2)$
    2. $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
    3. Output $\pi = (vk, \pi_A, \sigma)$

  Proof follows by the adaptive WI of $(P_A, V_A)$ and the pseudorandomness of $\text{Com}$

- **Adaptive soundness**: Implicit in the proof of simulation soundness, given next slide.
Proving simulation soundness

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

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

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

Then with all but negligible probability:

- $vk'$ is not the verification key appeared in $\pi$ ($(\text{Gen, Sign, Vrfy})$ is a strong signature)
  
  $\implies \nexists r \in \{0, 1\}^* \text{ s.t. } c_1 = \text{Com}(vk', r)$ (Com is perfectly binding)

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

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

Adaptive soundness?
Part II

Proof of Knowledge
Proof of Knowledge

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

**Definition 20 (knowledge extractor)**

Let \((P, V)\) be an interactive proof for \(L \in \mathcal{NP}\). A probabilistic algorithm \(E\) is a knowledge extractor for \((P, V)\) and \(R_L\) with error \(\eta: \mathbb{N} \mapsto \mathbb{R}\), if \(\exists t \in \text{poly} \) s.t.

\[
\forall x \in L \text{ and deterministic algorithm } P^*, E^{P^*}(x) \text{ runs in expected time bounded by } \frac{t(|x|)}{\delta(x) - \eta(|x|)} \text{ and outputs } w \in R_L(x), \text{ where } \delta(x) = \Pr[(P^*, V)(x) = 1].
\]

\((P, V)\) is a proof of knowledge for \(L\) with error \(\eta\).

- A property of \(V\)
- Why do we need it? Authentication schemes
- Why only deterministic \(P^*\)?
Examples

Claim 21
The $\mathcal{ZK}$ proof we’ve seen in class for $GI$, 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: ?