Non-Interactive Zero-Knowledge

Ron Rothblum

Technion

Zero-Knowledge

- So far today: Zero-Knowledge is really awesome!
- ZK Crucially relies on a combination of interaction and randomness.
- Even more awesome – ZK with “no” interaction! Prover just sends a ZK proof and verifier is convinced (a la NP proof).
- Non-interactive proofs are very important in some domains.

  For example, can simply post proof on website (or blockchain).

Non-interactive Zero-knowledge?

Claim: If L has a ZK proof in which prover sends a single message then $L \in BPP$.

Proof: Decision procedure for L:

1. Given $x \in L$, run $Sim(x)$ to get a simulated proof π.
2. Output $V(x, \pi)$.

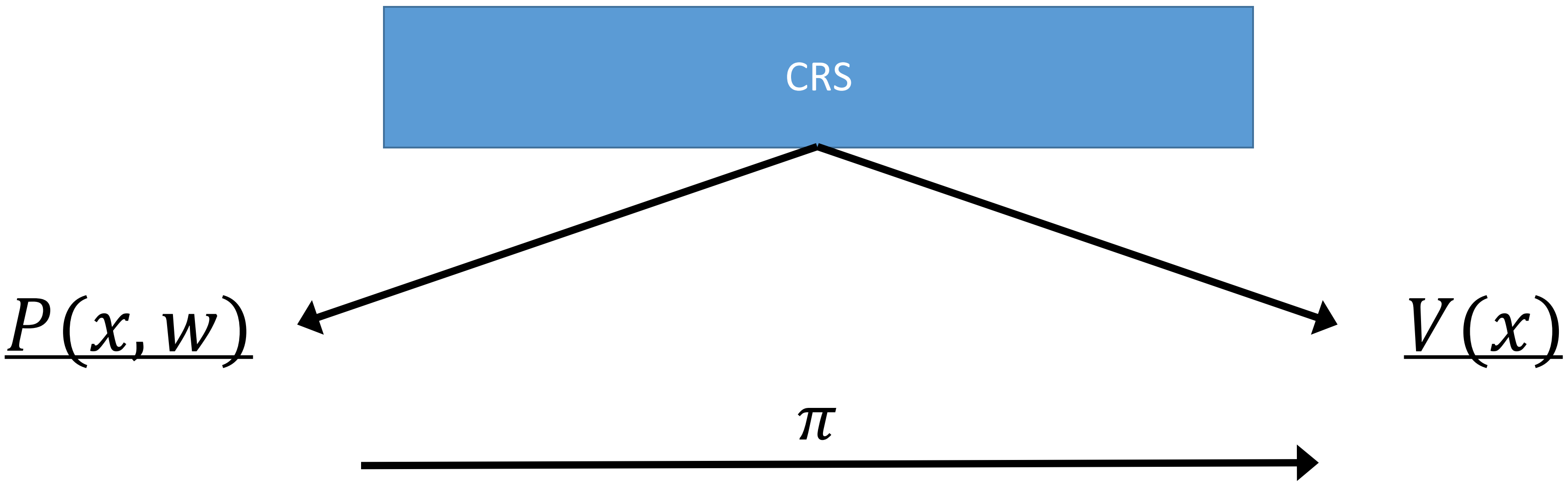
- Completeness: If $x \in L$ then simulated proof indis. from real proof $\Rightarrow V$ accepts.
- Soundness: If $x \notin L$ then V rejects all proofs (whp).

Thanks!

Non-Interactive Zero-Knowledge [BFM88]

- **Key idea:** *trusted setup*.
- Typically, the Common Reference String (CRS) model.
- A trusted party generates a CRS that all parties can see.
- Even Better: common uniform random string (CURS).

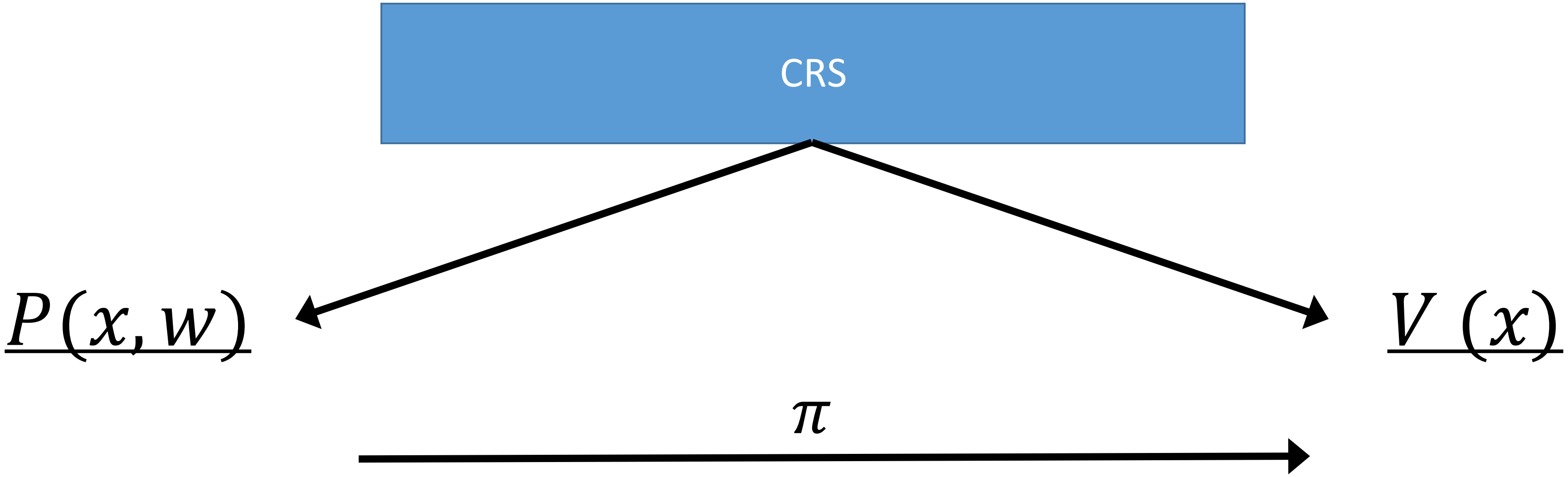
Definition: NIZK



Completeness: if $x \in L \Rightarrow \Pr[V\ accepts] = 1 - negl$

Soundness: if $x \notin L \Rightarrow \forall$PPT P^*, $\Pr[V\ accepts] = negl$

Definition: NIZK



Zero-Knowledge: "Can simulate view of the verifier"

$\exists Sim$ such that for $x \in L$

$$Sim(x) \approx^c (CRS, \pi)$$

Philosophical Detour: is NIZK actually ZK?

You can share an NIZK proof with your friends and convince them that $x \in L$!

Q: you've not learned only that $x \in L$ but also a convincing proof for that fact. How can this be ZK???

A: you've learned a proof for this specific CRS. Arguably did not learn directly about x.

Regardless of philosophical mumbo jumbo, very useful in applications!

Impossibility Results No Longer Applies!

False Claim: If L has an NIZK in CRS model then $L \in BPP$.

Wrong Proof: Decision procedure for L:

1. Given $x \in L$, run $Sim(x)$ to get (π, CRS).
2. Output $V(x, CRS, \pi)$.

- Completeness: If $x \in L$ then simulated proof indis. from real proof $\Rightarrow V$ accepts.
- Soundness: If $x \notin L$ then V rejects all proofs (whp).

NIZK Applications

- CCA secure encryption [NY90].
- Unique signatures [BG89].
- MPC with low round complexity [AJJTVW12].
- CS proofs [Micali94]
- Mechanism design [LMPS04]
- Cryptocurrencies zk-SNARGS, zk-STARKS [BCGGMTV14,…]
- …

Variants of NIZKs (aka the Boring Slide)

- **Multi theorem:** can-reuse CRS for many x's.
- **Adaptive soundness:** sound even if $x \notin L$ chosen after CRS.
- **Adaptive ZK:** ZK distinguisher can choose $x \in L$ after CRS.
- **Statistical soundness (proof):** sound against unbounded provers.
- **Statistical ZK:** ZK for unbounded distinguishers.

Feasibility Results [Circa 2018]

[FLS90]: NIZK for all of NP from Trapdoor Permutations*.

Corollary: NIZK based on hardness of factoring.

Other known results:

- Bilinear maps [GOS06].
- Random oracle model (tomorrow).
- Obfuscation [SW13,BP15].
- Optimal hardness assumptions [CCRR18,CCHLRR18].

New & Exciting Feasibility Results [2019]

- LWE + circular security [CLW19]
- *Last week:* LWE! [PS19]

Still Open:

1. From discrete log type assumptions (in standard group).
2. From less structured generic assumptions.
   - One way functions???

Feasibility Results [Circa 2018]

[FLS90]: NIZK for all of NP from Trapdoor Permutations*.

Corollary: NIZK based on hardness of factoring.

Other known results:

- Bilinear maps [GOS06].
- Random oracle model (tomorrow).
- Obfuscation [SW13,BP15].
- Optimal hardness assumptions [CCRR18,CCHLRR18].

Feasibility Results [Circa 2018]

[FLS90]: NIZK for all of NP from Trapdoor Permutations*.

Corollary: NIZK based on hardness of factoring.

Other known results:

- Bilinear maps [GOS06].
- Random oracle model (tomorrow).
- Obfuscation [SW13,BP15].
- Optimal hardness assumptions [CCRR18,CCHLRR18].

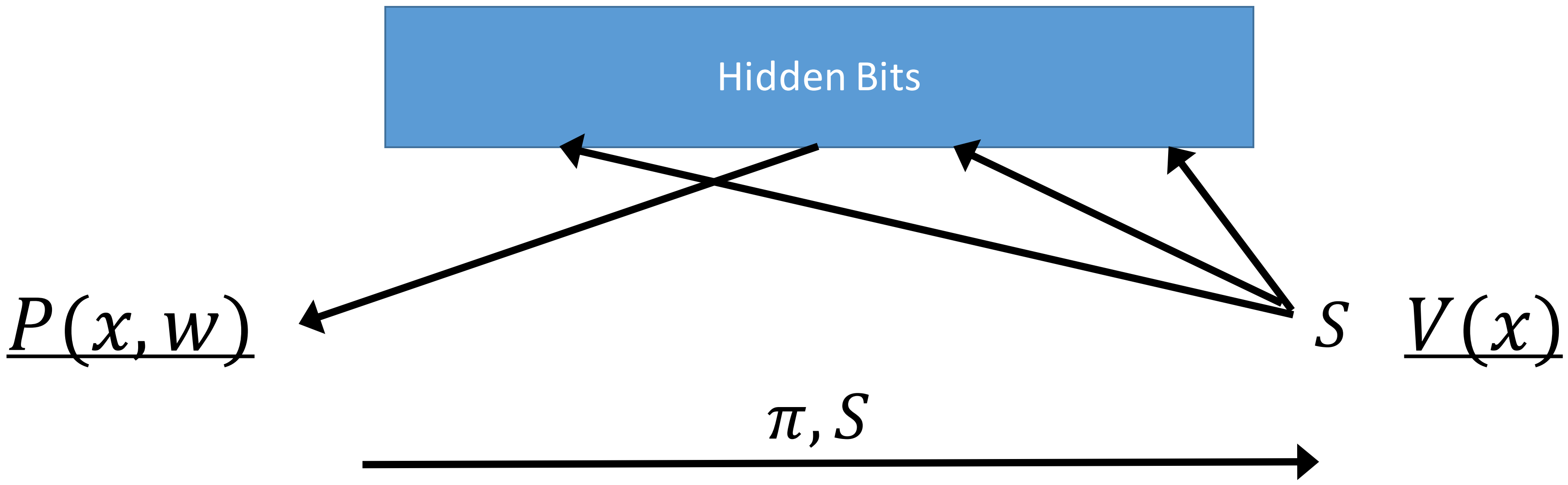
The FLS Paradigm

Construction has two main steps:

1. Construct NIZK in the “hidden bits” model.

2. Compile hidden bits NIZK to standard NIZK.

The Hidden Bits Model



Think of CRS model, except verifier only sees a part of the CRS determined by the prover.

The FLS Paradigm

Construction has two main steps:

1. Construct NIZK in the “hidden bits model”.

2. Compile any hidden bits NIZK to standard NIZK.

NIZK in the Hidden Bits Model

Construct hidden bits NIZK for *Hamiltonicity* – given a graph G, does it contain a Hamiltonian cycle?

Hamiltonicity is NP complete $\Rightarrow$ Hidden bits $NIZK$ for all of NP.

Construction is information theoretic.

- Prover is polynomial-time (given the cycle).
- Perfect completeness.
- Perfect* soundness even against unbounded prover!

Hidden Bits NIZK for Hamiltonicity

Common Input: A graph $G = (V, E)$

Auxiliary Prover Input: Hamiltonian cycle $H \subseteq E$.

CRS: random cycle graph C on $|V|$ vertices (represented by adjacency matrix).*

Hidden Bits NIZK for Hamiltonicity

Random cycle graph $C = (V_C, E_C)$

$\underline{P(G, H)}$ $\underline{V(G)}$

π, S →

Find injective mapping $\pi: V \to V_C$
that preserves cycle structure

Reveal $S \subseteq V_C \times V_C$ s.t.:
$S = \pi(V^2 \setminus E)$

Check that
1. π is injective
2. $\forall e \notin E$, the edge $\pi(e)$ was revealed (as a non-edge)

Completeness

Random cycle graph $C = (V_C, E_C)$

$\underline{P(G,H)}$ $\underline{V(G)}$

π, S →

Find injective mapping $\pi: V \to V_C$
that preserves cycle structure

Reveal $S \subseteq V_C \times V_C$ s.t.:
$S = \pi(V^2 \setminus E)$

Check that
1. π is injective
2. $\forall e \notin E$, the edge $\pi(e)$
was revealed (as a non-edge)

Soundness

Suppose V accepts.

1. π is injective.
2. All non-edges of E

Consider the inverse E'

1. $E' \subseteq E$ (i.e,. contains only a
2. E' forms a Hamiltonian c

$\Rightarrow G$ is Hamiltonian.

Perfect soundness!

Actually, for CURS (instead of CRS) pay exponentially small soundness error.

Hidden Bits NIZK for Hamiltonicity: Zero-Knowledge

Intuitively, all the verifier sees is a mapping $\pi: V \to V_C$ and that all the non-edges of G were revealed.

How to simulate? Given graph G:

- Choose random injective function $\pi \to [n]$.
- Output (π, S, CRS_S) where $S = \pi(V^2 \setminus E)$ and $CRS_S = 000 \ldots 0$.

Claim 1: for every fixed choice of π the simulated view is identical to the real.

Claim 2: mapping in real execution is a random injective function.

The FLS Paradigm

Construction has two main steps:

1. Construct NIZK in the “hidden bits model”.

2. Compile any hidden bits NIZK to standard NIZK.

From Hidden Bits to CRS

Hidden bits model is a fictitious abstraction.

Will use crypto to compile into standard CRS model.

Main tool: Trapdoor Permutations (TDP).

Trapdoor Permutations

- Will use an idealized definition.
- Actual candidates don’t satisfy this… ☹
- To make a long story short, it causes massive headaches.
- See: enhanced TDP [G04], doubly-enhanced TDP [G11,GR13], certifying TDP [BY96,CL18]…

Idealized Trapdoor Permutations

Definition: *a collection of efficiently computable permutations*

$\{p_\alpha : \{0,1\}^\lambda \rightarrow \{0,1\}^\lambda\}_{\alpha \in \{0,1\}^\lambda}$ *such that:*

1. $\exists PPT$ *algorithm that samples* α *together with a "trapdoor"* τ
2. $\alpha, p_\alpha(x) \nrightarrow x$.
3. $\tau, p_\alpha(x) \rightarrow x$.

Examples*: RSA, Rabin.

Hardcore bit of TDP: efficient $h: \{0,1\}^\lambda \rightarrow \{0,1\}$ s.t. $\alpha, p_\alpha(x) \nrightarrow h(x)$.

Implementing Hidden Bits Model – Bird's Eye

CRS consists of $y_1, \ldots, y_\ell \in \{0,1\}^\lambda$.

Prover chooses a TDP (α, τ).

Hidden bits are defined as $b_i = h(y_i)$.

To reveal a bit the prover sends x_i.

Implementing Hidden Bits – Frog's Eye

$y_1, \ldots y_\ell$

$\underline{P(x,w)}$ $\qquad\qquad$ $\underline{V(x)}$

$\alpha, \pi, \{x_i\}_{i \in S}$ →

$P(x,w)$:

1. Choose (α, τ).
2. Define $x_i = p_\alpha^{-1}(y_i)$.
3. Hidden bits are $b_i = h(x_i)$
4. Run HB prover on $(x, w, (b_1, \ldots, b_\ell))$
5. Get proof π and $S \subseteq [\ell]$.

$V(x)$:

1. Check α in collection
2. $\forall i \in S$, check $p_\alpha(x_i) = y_i$.
3. Define $b_i = h(x_i)$
4. Check that HB verifier accepts $(x, \pi, \{b_i\}_{\{i \in S\}})$

Completeness

$y_1, \dots y_\ell$

$\underline{P(x,w)}$ $\underline{V(x)}$

$\alpha, \pi, \{x_i\}_{i \in S}$

1. Choose (α, τ).
2. Define $x_i = p_\alpha^{-1}(y_i)$.
3. Hidden bits are $b_i = h(x_i)$
4. Run HB prover on $\left(x, w, (b_1, \dots, b_\ell)\right)$
5. Get proof π and $S \subseteq [\ell]$.

1. Check α in collection
2. $\forall i \in S$, check $p_\alpha(x_i) = y_i$.
3. Define $b_i = h(x_i)$
4. Check that HB verifier accepts $(x, \pi, \{b_i\}_{\{i \in S\}})$

From Hidden Bits to NIZK – Zero Knowledge

- Intuitively the bits $\{b_i\}_{i \in S}$ are revealed and by the hard-core property + hybrid argument the bits $\{b_i\}_{i \notin S}$ are hidden.
- Formally(ish) can construct a simulator $Sim(x)$ as follows:
  - Run $Sim_{HB}(x)$ to get $(\pi, S, \{b_i\}_{i \in S})$.
  - Sample (α, τ).
  - For every $i \in S$ sample x_i s.t. $h(x_i) = b_i$. Set $y_i = p_\alpha(x_i)$.
  - For every $i \notin S$ sample $y_i \in \{0,1\}^\lambda$.
  - Output $\big((\alpha, \pi, S), (y_1, \dots, y_\ell)\big)$.
- Exercise: show that $Sim(x) \approx_C Real$.

From Hidden Bits to NIZK: Soundness

Suppose α is fixed (Important!).

Then, the hidden bits are automatically defined as

$$b_i = h\left(f_\alpha^{-1}(y_i)\right)$$

Now soundness follows immediately from HB soundness.

Problem: cannot assume α is fixed – choice of α gives prover leverage in deciding the values of $b_1, \ldots, b_\ell$.

From Hidden Bits to NIZK: Soundness

Idea: repeat HB proof-system enough times so that the soundness is $2^{-2\lambda}$.

Now:

$\Pr[\exists \alpha \ on\ which\ Prover\ can\ cheat]$

From Hidden Bits to NIZK: Soundness

Idea: repeat HB proof-system enough times so that the soundness is $2^{-2\lambda}$.

Now:

$$\Pr[\exists\alpha\ on\ which\ Prover\ can\ cheat] \leq \sum_{\alpha} \Pr[Prover\ can\ cheat\ on\ \alpha]$$

From Hidden Bits to NIZK: Soundness

Idea: repeat HB proof-system enough times so that the soundness is $2^{-2\lambda}$.

Now:

$$\Pr[\exists\alpha \; on \; which \; Prover \; can \; cheat] \leq \sum_{\alpha} \Pr[Prover \; can \; cheat \; on \; \alpha]$$

$$\leq 2^{\lambda} \cdot 2^{-2\lambda}$$

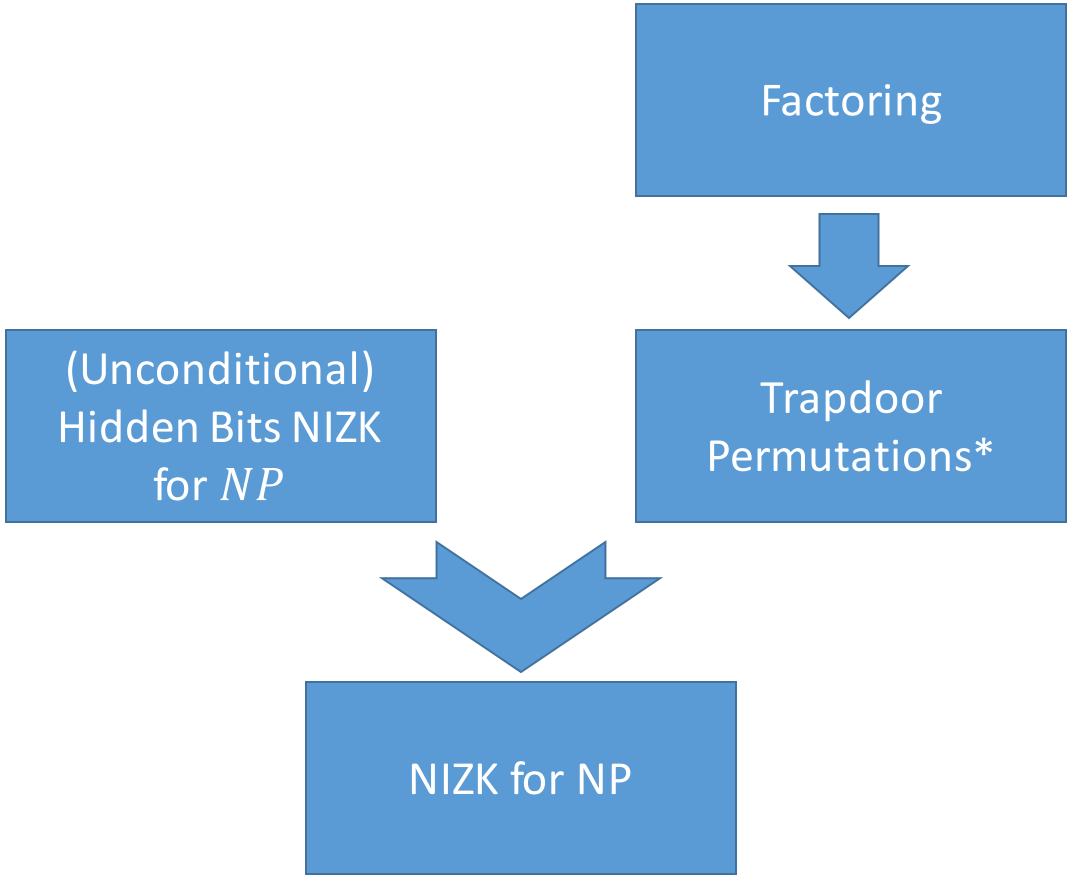
From Hidden Bits to NIZK: Soundness

Idea: repeat HB proof-system enough times so that the soundness is $2^{-2\lambda}$.

Now:

$$\Pr[\exists \alpha \text{ on which Prover can cheat}] \leq \sum_{\alpha} \Pr[\text{Prover can cheat on } \alpha]$$
$$\leq 2^{\lambda} \cdot 2^{-2\lambda}$$
$$= 2^{-\lambda}$$

Putting it all together



Thm: if factoring is hard, then $\exists NIZK$ for all of NP.

Thanks!