Lecture 11: Argument compilation and NIZK

Zero-knowledge proofs

263-4665-00L

Lecturer: Jonathan Bootle

Announcements

- Exam date 02/02/2023, 15:00-17:00.
- Continued office hours after New Year on 16/01, 23/01, 30/01.

- Guests visiting today’s lecture to provide feedback

- Extra (optional, non-examinable) video about reducing verification costs for polynomial commitments will be posted early next week.

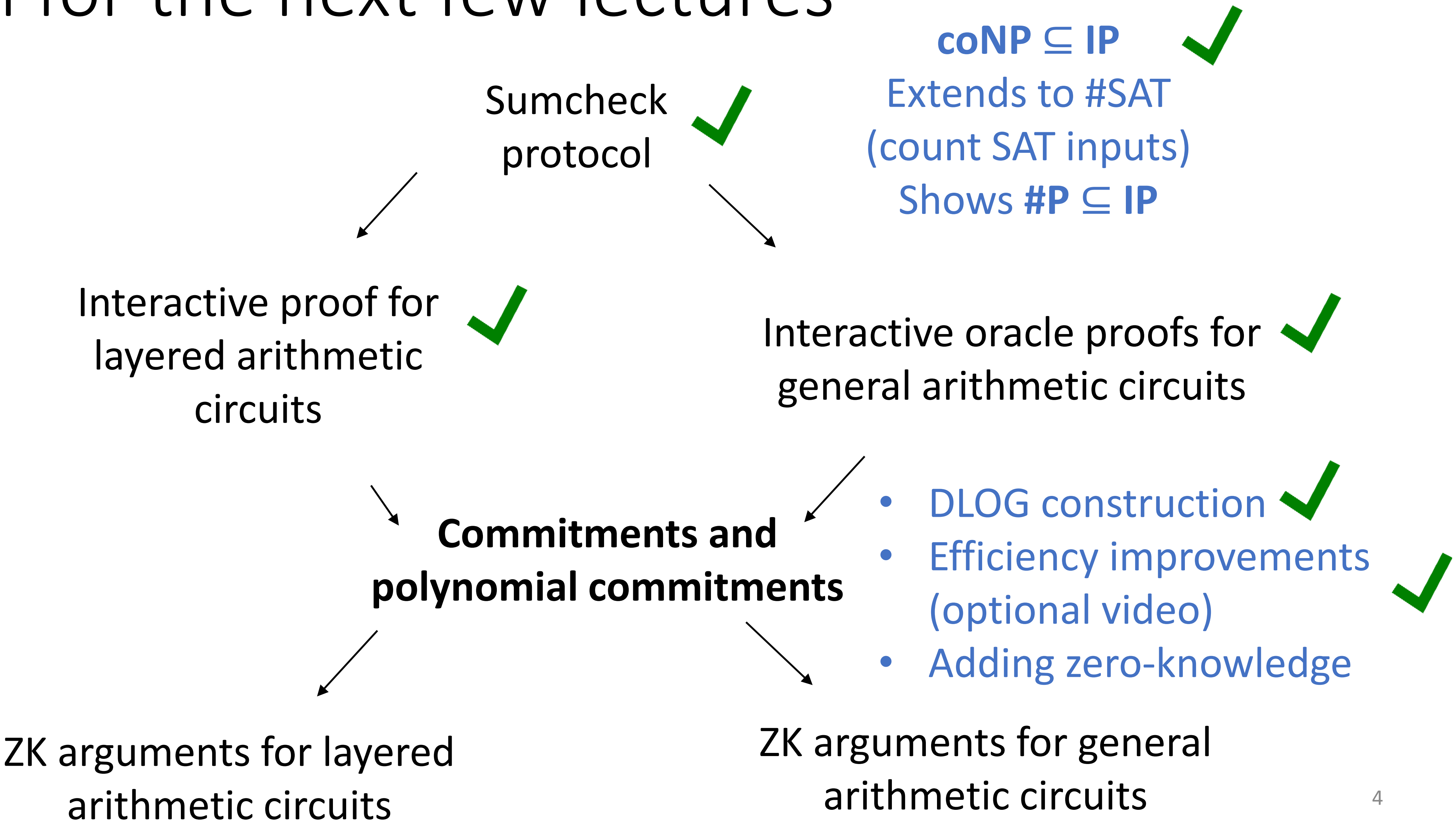
Pedersen Multicommitments Eval Summary

- Let $y_1, \dots, y_{\log N} \in \mathbb{Z}_p$ with $\tilde{f}\left(y_1, \dots, y_{\log N}\right) = z$.
- Let $\vec{Y} \coloneqq \mathrm{Expand}(\vec{y}) = \left(\widetilde{\mathrm{EQ}}(\vec{y}; \vec{\imath})\right)_{\vec{\imath} \in \{0,1\}^{\log N}}$ satisfying $f(\vec{y}) = \langle \vec{f}, \vec{Y} \rangle$.

- Write $C = \langle \vec{f}, \vec{G} \rangle + r \cdot H$ and $z = \langle \vec{f}, \vec{Y} \rangle$.
- $\mathcal{R}_{PedPC}(pp) \coloneqq \left\{ \left((C, z, y_1, \dots, y_\ell), \tilde{f} \right) : \begin{array}{c} \tilde{f} \in \mathbb{F}^{\leq 1}[X_1, \dots, X_\ell] \\ z = \langle \vec{f}, \vec{Y} \rangle, C = \langle \vec{f}, \vec{G} \rangle \end{array} \right\}$
- We constructed a protocol for Eval with prover complexity $O(N)$, proof size $O(\log N)$ and verifier complexity $O(N)$.

Set $r = 0$ initially
P can just send r to V.
Both remove $r \cdot H$ from C

Plan for the next few lectures



Sumcheck protocol ✓

coNP ⊆ IP ✓
Extends to #SAT
(count SAT inputs)
Shows **#P ⊆ IP**

Interactive proof for layered arithmetic circuits ✓

Interactive oracle proofs for general arithmetic circuits ✓

Commitments and polynomial commitments

- DLOG construction ✓
- Efficiency improvements (optional video) ✓
- Adding zero-knowledge

ZK arguments for layered arithmetic circuits

ZK arguments for general arithmetic circuits

Hiding polynomial evaluations

Evaluation relation for Pedersen multicommitments:

Given $y_1, \dots, y_{\log N}$, $z \in \mathbb{F}$, C, prove knowledge of openings $\{f_{\vec{i}}\}, r$ of C such that $\sum_{\vec{i} \in \{0,1\}^{log\ N}} f_{\vec{i}} \cdot \widetilde{\mathrm{EQ}}(\vec{y}; \vec{i}\,) = z$.

$$\mathcal{R}_{PedPC}(pp) := \left\{ \Big((C, z, y_1, \dots, y_\ell), (\tilde{f}, r) \Big) : \begin{array}{c} \tilde{f} \in \mathbb{F}^{\leq 1}[X_1, \dots, X_\ell] \\ \tilde{f}(y_1, \dots, y_\ell) = z, C = \langle \vec{f}, \vec{G} \rangle + r \cdot H \end{array} \right\}$$

$z, \vec{y}$ not hidden

Committed evaluation relation for Pedersen multicommitments:

$$\mathcal{R}_{ComPedPC}(pp, pp') := \left\{ \Big((C, C_z, y_1, \dots, y_\ell), (\tilde{f}, r, z, s) \Big) : \begin{array}{c} \tilde{f} \in \mathbb{F}^{\leq 1}[X_1, \dots, X_\ell] \\ \tilde{f}(y_1, \dots, y_\ell) = z, C = \langle \vec{f}, \vec{G} \rangle + r \cdot H \\ C_z = z \cdot G + s \cdot H \end{array} \right\}$$

z now hidden

pp for Pedersen multicommitments

pp' for plain Pedersen commitments

Committed evaluation protocol

Reduction completeness
$\Downarrow$
Completeness

Reduction 2-soundness
$\Downarrow$
$(2,4,\dots,4)$-soundness

Witness:

- vector $\vec{f} \in \mathbb{F}^N, r \in \mathbb{F}$
- $z, s \in \mathbb{F}$

Instance:

- commitment $C \in \mathbb{G}$, key $\vec{G} \in \mathbb{G}^N$
- vector $\vec{Y} \in \mathbb{F}^N$
- commitment $C_z \in \mathbb{G}$, key $G, H \in \mathbb{G}$

Language:

- $C = \left\langle \vec{f}, \vec{G} \right\rangle + r \cdot H \in \mathbb{G}$
- $z = \left\langle \vec{f}, \vec{Y} \right\rangle \in \mathbb{F}$
- $C_z = z \cdot G + s \cdot H$

Sample $\phi \leftarrow_\$ \mathbb{F}^{\leq 1}[X_1, \dots, X_\ell]$ with coefficient vector $\vec{\phi} \in \mathbb{F}^N$.

Sample $\rho, \sigma \leftarrow_\$ \mathbb{F}$.

Compute

$$\zeta = \phi(\vec{y})$$

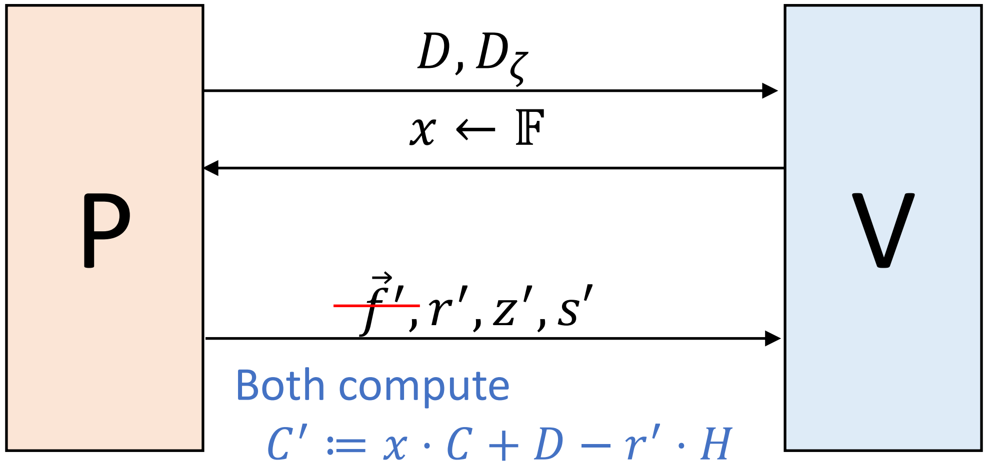
$$D = \left\langle \vec{\phi}, \vec{G} \right\rangle + \rho \cdot H$$

$$D_\zeta = \zeta \cdot G + \sigma \cdot H$$

Compute $f' \coloneqq xf + \phi$

$\vec{f}' \coloneqq x\vec{f} + \vec{\phi}, z' \coloneqq xz + \zeta$

$r' \coloneqq xr + \rho, s' \coloneqq xs + \sigma$



P

V

D, D_ζ

$x \leftarrow \mathbb{F}$

$\cancel{\vec{f}'}, r', z', s'$

Both compute

$C' \coloneqq x \cdot C + D - r' \cdot H$

Check

$x \cdot C + D == \left\langle \vec{f}', \vec{G} \right\rangle + r' \cdot H$

$f'(\vec{y}) == z'$

Check

$x \cdot C_z + D_\zeta == z' \cdot G + s' \cdot H$

New witness:

$\vec{f}' \coloneqq x\vec{f} + \vec{\phi} \in \mathbb{F}^N$

New instance:

- commitment $C' \in \mathbb{G}$, key $\vec{G} \in \mathbb{G}^N$
- vector $\vec{Y} \in \mathbb{F}^N$, target $z \in \mathbb{F}$

Run previous Eval protocol

New language:

- $C' = \left\langle \vec{f}', \vec{G} \right\rangle \in \mathbb{G}$
- $z' = \left\langle \vec{f}', \vec{Y} \right\rangle \in \mathbb{F}$

Committed evaluation protocol completeness

- $C = \left\langle \vec{f}, \vec{G} \right\rangle + r \cdot H, z = f(\vec{y}) = \left\langle \vec{f}, \vec{Y} \right\rangle, C_z = z \cdot G + s \cdot H.$

Left-multiply by x

- $D = \left\langle \vec{\phi}, \vec{G} \right\rangle + \rho \cdot H, \zeta = \phi(\vec{y}) = \left\langle \vec{\phi}, \vec{Y} \right\rangle, D_\zeta = \zeta \cdot G + \sigma \cdot H.$

Add

- $xC + D = \left\langle \vec{f}', \vec{G} \right\rangle + r' \cdot H, z' = f'(\vec{y}) = \left\langle \vec{f}', \vec{Y} \right\rangle,$
- $xC_z + D_\zeta = z' \cdot G + s' \cdot H.$
- In the optimized version using Eval, we have reduced to a true instance.
- Completeness follows from the completeness of the previous Eval protocol.

SHVZK analysis

Inefficient protocol

$S(pp, pp', C, C_z, p, x)$

1. $\vec{f}' \leftarrow_\$ \mathbb{F}^N. z' := f(\vec{y})$.
2. $r', s' \leftarrow_\$ \mathbb{F}$.
3. $D = x \cdot C - \left\langle \vec{f}', \vec{G} \right\rangle - r' \cdot H$.
4. $D_\zeta = x \cdot C_z - z' \cdot G - s' \cdot H$.
5. Output $\left(D, D_\zeta, x, \vec{f}', r', z', s'\right)$.

What is the verifier's view?

- $\left(D, D_\zeta, x, \vec{f}', r', z', s'\right)$ with
- $x \cdot C + D = \left\langle \vec{f}', \vec{G} \right\rangle + r' \cdot H,\ f'(\vec{y}) = z'$ and $x \cdot C_z + D_\zeta = z' \cdot G + s' \cdot H$.
- $\rho \leftarrow_\$ \mathbb{F}$ so $r' = xr + \rho$ is uniform in $\mathbb{F}$. Similarly for s'.
- $D = x \cdot C - \left\langle \vec{f}', \vec{G} \right\rangle - r' \cdot H$ is uniquely determined. Similarly for D_ζ.

Why is the simulator valid? (efficient, indistinguishable)

- Clearly, the simulator is efficient.
- $\vec{f}', z', r', s'$ have identical distributions to the real protocol.
- The other values are uniquely determined by the verifier checks.

SHVZK analysis

Efficient protocol

Eval protocol transcript

$S(pp, pp', C, C_z, p, x, x_1, \dots, x_\ell)$

1. $\vec{f}' \leftarrow_\$ \mathbb{F}^N$. $z' \coloneqq f(\vec{y})$. $r', s' \leftarrow_\$ \mathbb{F}$.
2. $D = x \cdot C - \left\langle \vec{f}', \vec{G} \right\rangle - r' \cdot H$
3. $D_\zeta = x \cdot C_z - z' \cdot G - s' \cdot H$
4. Get transcript π by running the honest prover algorithm for Eval on the new instance.
5. Output $(D, D_\zeta, x, r', z', s', \pi)$.

What is the verifier's view?

- $\left(D, D_\zeta, x, \cancel{\vec{f}'}, r', z', s', \pi\right)$ with
- $x \cdot C + D = \left\langle \vec{f}', \vec{G} \right\rangle + r' \cdot H,\ f'(\vec{y}) = z'$ and $x \cdot C_z + D_\zeta = z' \cdot G + s' \cdot H$.
- $\rho \leftarrow_\$ \mathbb{F}$ so $r' = xr + \rho$ is uniform in $\mathbb{F}$. Similarly for s'.
- $D = x \cdot C - \left\langle \vec{f}', \vec{G} \right\rangle - r' \cdot H$ is uniquely determined. Similarly for D_ζ.

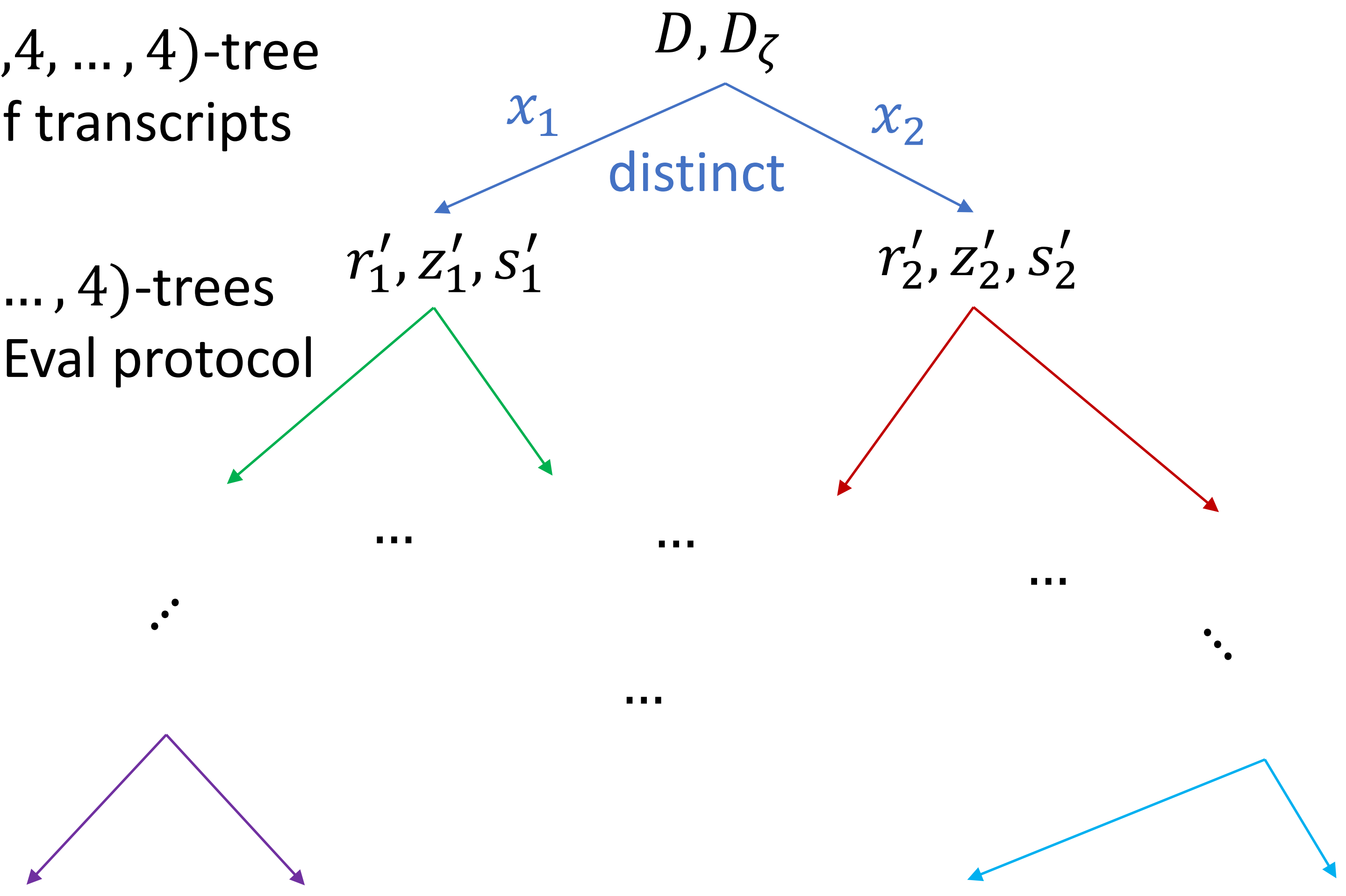
Why is the simulator valid? (efficient, indistinguishable)

- Clearly, the simulator is efficient.
- $\vec{f}', z', r', s'$ have identical distributions to the real protocol.
- The other values are uniquely determined by the verifier checks.
- The Eval prover algorithm is run on the same input distribution in both simulated and real protocol executions.

(2,4,..,4)-soundness from reduction 2-soundness

$(2,4,\dots,4)$-tree of transcripts

$(4,\dots,4)$-trees for Eval protocol x2



Witnesses for new instance

$$\langle \vec{f}'_1, \vec{G} \rangle + r'_1 \cdot H = x_1 \cdot C + D$$
$$\langle \vec{f}'_2, \vec{G} \rangle + r'_2 \cdot H = x_2 \cdot C + D$$

$$\uparrow$$

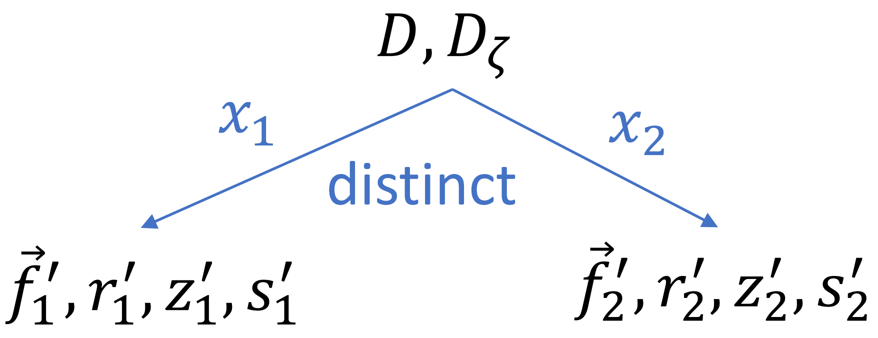
$$C'_1 := x_1 \cdot C + D - r'_1 \cdot H$$
$$C'_2 := x_2 \cdot C + D - r'_2 \cdot H$$

Eval protocol extractor produces $\vec{f}'_1$, $\vec{f}'_2$ such that:

$$C'_1 = \langle \vec{f}'_1, \vec{G} \rangle \qquad C'_2 = \langle \vec{f}'_2, \vec{G} \rangle$$
$$z'_2 = \langle \vec{f}'_1, \vec{Y} \rangle \qquad z'_2 = \langle \vec{f}'_2, \vec{Y} \rangle$$

2-soundness analysis of reduction

2-tree of transcripts



D, D_ζ

x_1 x_2

distinct

$\vec{f}_1', r_1', z_1', s_1'$ $\vec{f}_2', r_2', z_2', s_2'$

Satisfying

$$\langle \vec{f}_1', \vec{G}\rangle + r_1' \cdot H = x_1 \cdot C + D$$
$$\langle \vec{f}_2', \vec{G}\rangle + r_2' \cdot H = x_2 \cdot C + D$$
$$z_1' = \langle \vec{f}_1', \vec{Y}\rangle$$
$$z_2' = \langle \vec{f}_2', \vec{Y}\rangle$$
$$x_1 \cdot C_z + D_\zeta = z_1' \cdot G + s_1' \cdot H$$
$$x_2 \cdot C_z + D_\zeta = z_2' \cdot G + s_2' \cdot H$$

$$\begin{pmatrix}1 & x_1\\ 1 & x_2\end{pmatrix}\begin{pmatrix}D\\ C\end{pmatrix} = \begin{pmatrix}\vec{f}_1'\\ \vec{f}_2'\end{pmatrix} \cdot \vec{G} + \begin{pmatrix}r_1'\\ r_2'\end{pmatrix} \cdot H$$

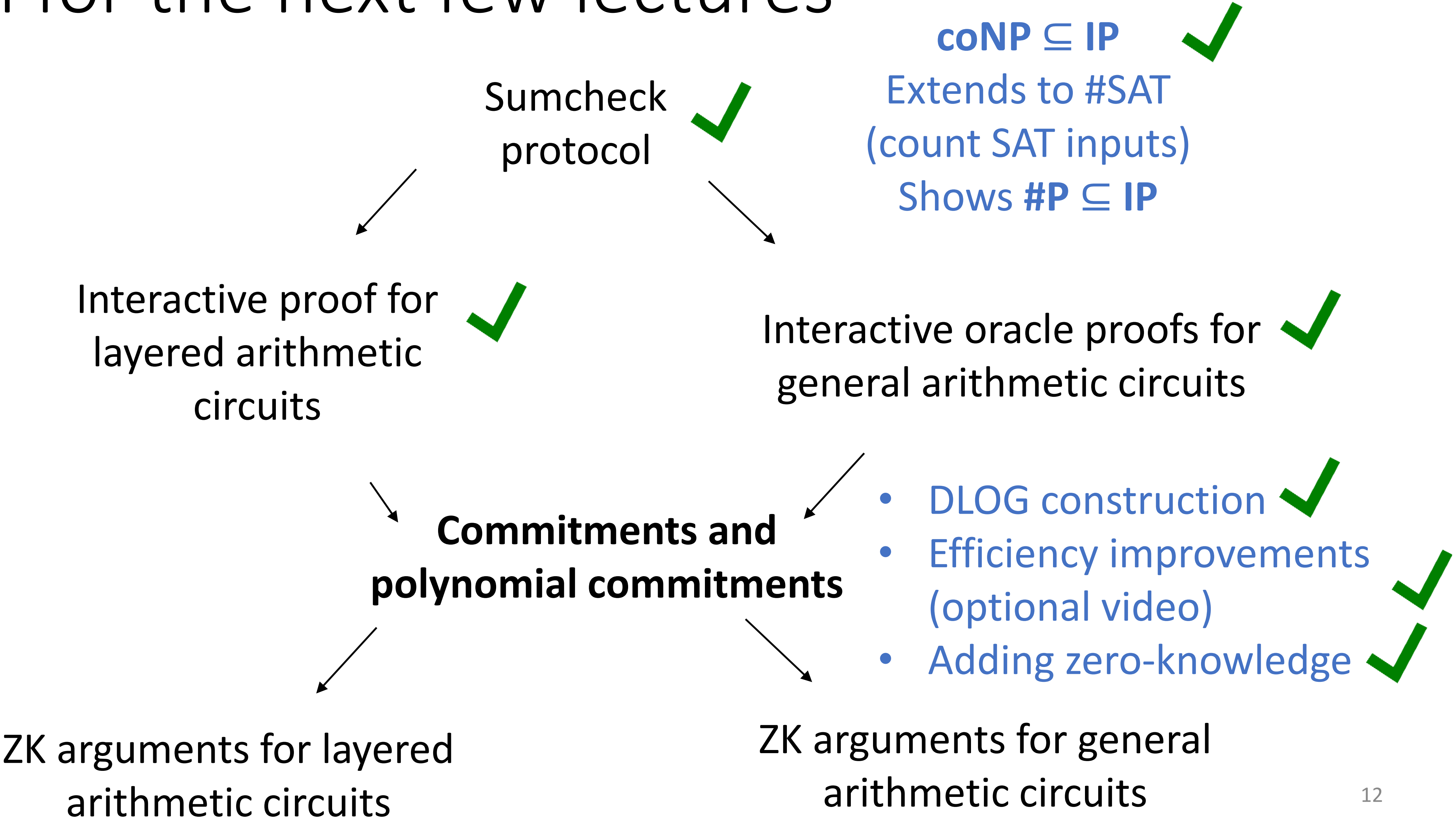
$$\begin{pmatrix}1 & x_1\\ 1 & x_2\end{pmatrix}\begin{pmatrix}D_\zeta\\ C_z\end{pmatrix} = \begin{pmatrix}z_1'\\ z_2'\end{pmatrix} \cdot G + \begin{pmatrix}s_1'\\ s_2'\end{pmatrix} \cdot H = \begin{pmatrix}\vec{f}_1'\\ \vec{f}_2'\end{pmatrix} \cdot (\vec{Y} \cdot G) + \begin{pmatrix}s_1'\\ s_2'\end{pmatrix} \cdot H$$

Inverting the linear system, we get $\vec{f}, \vec{\phi}, r, s, \rho, \sigma$ satisfying

$$\begin{pmatrix}D\\ C\end{pmatrix} = \begin{pmatrix}\vec{\phi}\\ \vec{f}\end{pmatrix} \cdot \vec{G} + \begin{pmatrix}\rho\\ r\end{pmatrix} \cdot H, \qquad \begin{pmatrix}D_\zeta\\ C_z\end{pmatrix} = \begin{pmatrix}\vec{\phi}\\ \vec{f}\end{pmatrix} \cdot \left(\vec{Y} \cdot G\right) + \begin{pmatrix}\sigma\\ s\end{pmatrix} \cdot H = \begin{pmatrix}\langle\vec{\phi}, \vec{Y}\rangle\\ \langle\vec{f}, \vec{Y}\rangle\end{pmatrix} \cdot G + \begin{pmatrix}\sigma\\ s\end{pmatrix} \cdot H$$

$$= f(\vec{y})$$

Plan for the next few lectures



Sumcheck protocol ✓

coNP ⊆ IP ✓
Extends to #SAT
(count SAT inputs)
Shows **#P ⊆ IP**

Interactive proof for layered arithmetic circuits ✓

Interactive oracle proofs for general arithmetic circuits ✓

Commitments and polynomial commitments

- DLOG construction ✓
- Efficiency improvements (optional video) ✓
- Adding zero-knowledge ✓

ZK arguments for layered arithmetic circuits

ZK arguments for general arithmetic circuits

How to make protocols succinct and ZK

Succinctness:

- Replace each large/oracle message with a polynomial commitment.
- Whenever V would have made a polynomial evaluation query to perform a check, P sends the evaluation, and they run the Eval protocol together.

Zero-knowledge:

- Also commit to each small message with a plain Pedersen commitment.
- Whenever V would have made a polynomial evaluation query to perform a check, P sends a commitment to the evaluation, and they run the hidden Eval protocol together.
- P and V run Σ-protocols on committed values to check that each verification equation would have been satisfied.

Compiling GKR to a ZK argument

- GKR language: $\left\{\left(\{add_i, mult_i\}_{i=0}^{D-1}, \vec{x}, \vec{y}\right) : C(\vec{x}) = \vec{y}\right\}.$

P language

- Compiled GKR relation:

Easily generalizes to different lengths

Instance

Witness

$$\left\{\left(\left(\{add_i, mult_i\}_{i=0}^{D-1}, C_{\vec{x}}, C_{\vec{y}}\right), (\vec{x}, r, \vec{y}, s)\right) : \begin{array}{c} C_{\vec{x}} = \langle \vec{x}, \vec{G} \rangle + r \cdot H \\ C_{\vec{y}} = \langle \vec{y}, \vec{G} \rangle + s \cdot H \\ C(\vec{x}) = \vec{y} \end{array}\right\}.$$

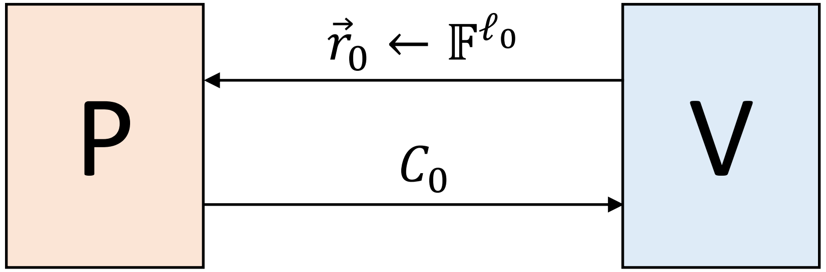
(polynomial) commitments

NP relation

Example: initial reduction

w_0 computed from $\vec{x}$

Initial claim: $C_{\vec{y}}$ contains $\vec{y}$
satisfying $\widetilde{w}_0 \equiv \tilde{y}$



Compute $v_0 := \tilde{y}(\vec{r}_0)$.
Sample $s \leftarrow_{\$} \mathbb{F}$.
Compute $C_0 = v_0 \cdot G + s \cdot H$.

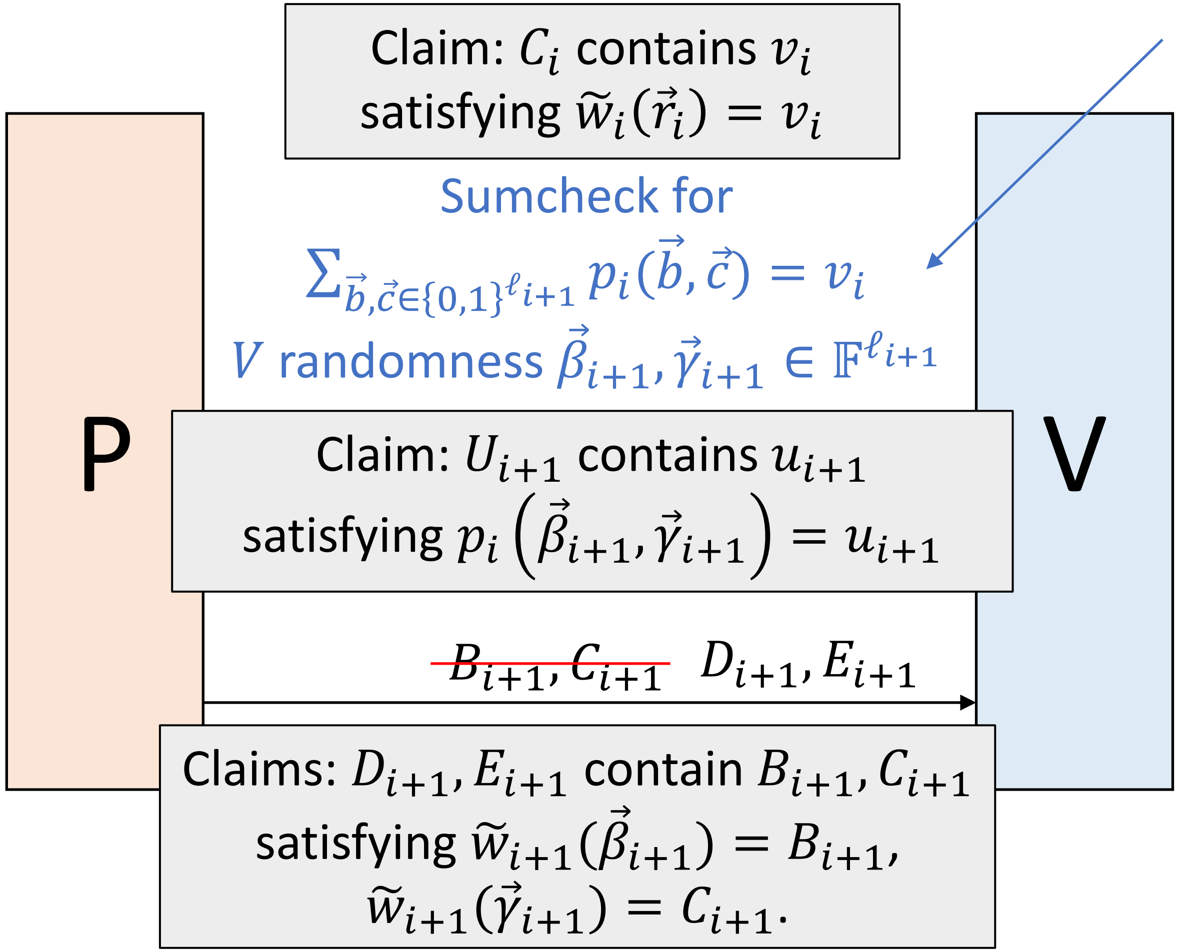
~~Both compute $v_0 := \tilde{y}(\vec{r}_0)$~~

Committed eval protocol proves:
$C_{\vec{y}}, C_0$ contain $\vec{y}, v_0$
satisfying $\tilde{y}(\vec{r}_0) = v_0$

Claim: C_0 contains v_0
satisfying $\widetilde{w}_0(\vec{r}_0) = v_0$

Example: sumcheck level i to level i+1 reduction

- P sends *commitments* to sumcheck messages.
- V uses Σ-protocols to perform sumcheck checks but on committed values.
- V computes commitment U_{i+1} to final target evaluation using homomorphism.



Claim: C_i contains v_i satisfying $\widetilde{w}_i(\vec{r}_i) = v_i$

Sumcheck for

$$\sum_{\vec{b},\vec{c}\in\{0,1\}^{\ell_{i+1}}} p_i(\vec{b},\vec{c}) = v_i$$

V randomness $\vec{\beta}_{i+1}, \vec{\gamma}_{i+1} \in \mathbb{F}^{\ell_{i+1}}$

P

V

Claim: U_{i+1} contains u_{i+1} satisfying $p_i\left(\vec{\beta}_{i+1}, \vec{\gamma}_{i+1}\right) = u_{i+1}$

~~B_{i+1}, C_{i+1}~~ D_{i+1}, E_{i+1}

Claims: D_{i+1}, E_{i+1} contain B_{i+1}, C_{i+1} satisfying $\widetilde{w}_{i+1}(\vec{\beta}_{i+1}) = B_{i+1}$, $\widetilde{w}_{i+1}(\vec{\gamma}_{i+1}) = C_{i+1}$.

Sample $r_i, s_i \leftarrow \mathbb{F}$.

$$B_{i+1} \coloneqq \widetilde{w}_{i+1}(\vec{\beta}_{i+1})$$
$$C_{i+1} \coloneqq \widetilde{w}_{i+1}(\vec{\gamma}_{i+1})$$

$$D_{i+1} = B_{i+1} \cdot G + r_i \cdot H$$
$$E_{i+1} = C_{i+1} \cdot G + s_i \cdot H$$

Both P, V run a Σ-protocol to check that $D_{i+1}, E_{i+1}, U_{i+1}$ contain $B_{i+1}, C_{i+1}, u_{i+1}$ satisfying

$$\widetilde{\mathrm{add}}_i\left(\vec{r}_i, \vec{\beta}_{i+1}, \vec{\gamma}_{i+1}\right)[B_{i+1} + C_{i+1}]$$
$$+ \widetilde{\mathrm{mult}}_i\left(\vec{r}_i, \vec{\beta}_{i+1}, \vec{\gamma}_{i+1}\right)[B_{i+1} \cdot C_{i+1}]$$
$$== u_{i+1}$$

Compiling R1CS IOP to a ZK argument

$$\mathcal{R}_{R1CS} = \left\{ \left((\mathbb{F}, A, B, C, \vec{x}), \vec{w} \right) : \begin{array}{c} A, B, C \in \mathbb{F}^{N_r \times N_c}, \vec{x} \in \mathbb{F}^k \\ \vec{w} \in \mathbb{F}^{N_c - k}, \vec{z} := \vec{x} || \vec{w} \\ A\vec{z} \circ B\vec{z} = C\vec{z} \end{array} \right\}.$$

NP relation

- Compiled R1CS relation:

Instance

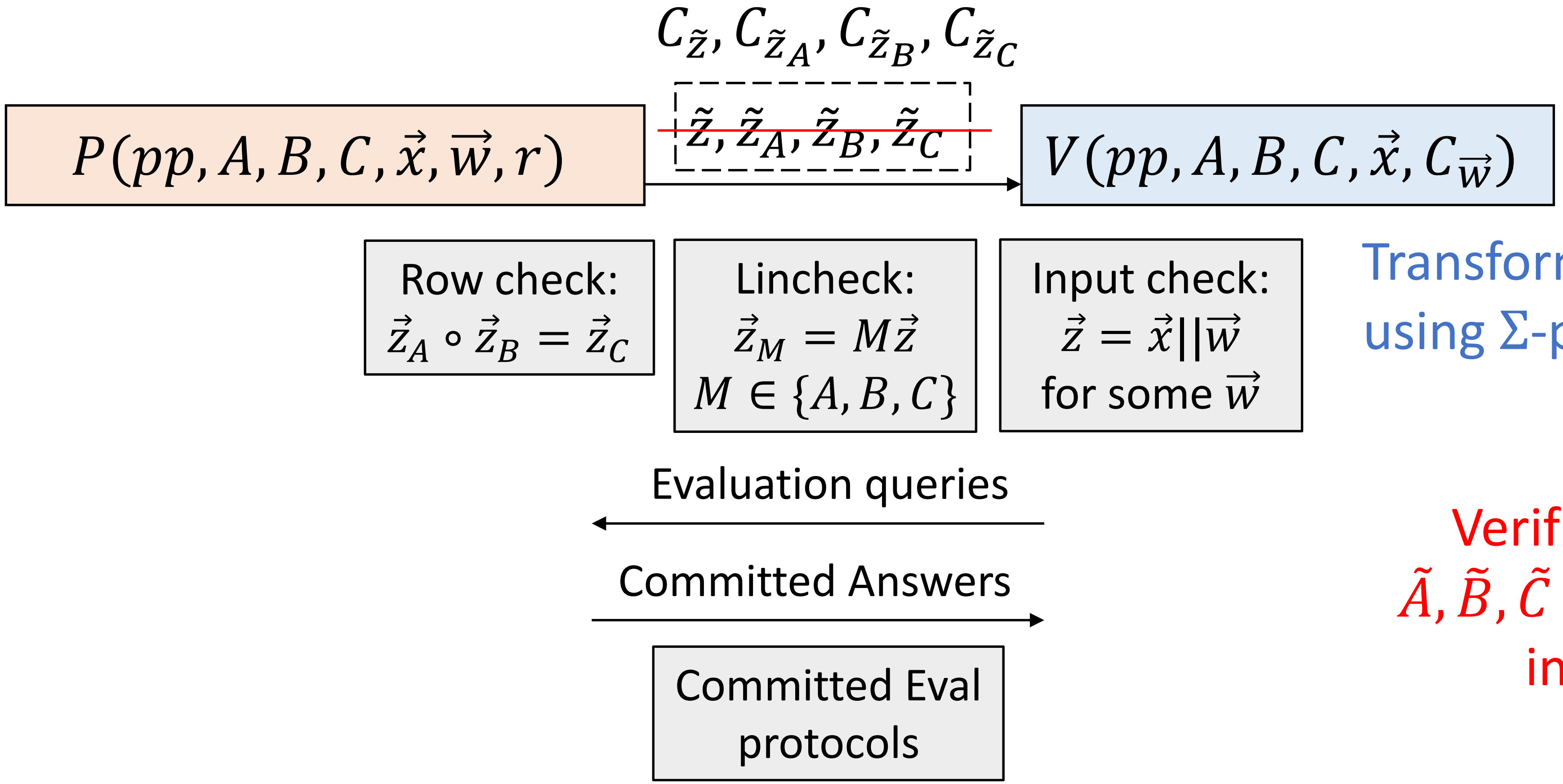
$$\left\{ \left((\mathbb{F}, A, B, C, \vec{x}, C_{\vec{w}}), (\vec{w}, r) \right) : \begin{array}{c} A, B, C \in \mathbb{F}^{N \times N}, \vec{x} \in \mathbb{F}^k \\ C_{\vec{w}} = \langle \vec{w}, \vec{G} \rangle + r \cdot H \\ \vec{w} \in \mathbb{F}^{N-k}, \vec{z} := \vec{x} || \vec{w} \\ A\vec{z} \circ B\vec{z} = C\vec{z} \end{array} \right\}.$$

Witness

Alternatively, send $C_{\vec{w}}$ in first message

NP relation

ZK argument for R1CS



$C_{\tilde{z}}, C_{\tilde{z}_A}, C_{\tilde{z}_B}, C_{\tilde{z}_C}$

~~$\tilde{z}, \tilde{z}_A, \tilde{z}_B, \tilde{z}_C$~~

$P(pp, A, B, C, \vec{x}, \vec{w}, r)$

$V(pp, A, B, C, \vec{x}, C_{\vec{w}})$

Row check:
$\vec{z}_A \circ \vec{z}_B = \vec{z}_C$

Lincheck:
$\vec{z}_M = M\vec{z}$
$M \in \{A, B, C\}$

Input check:
$\vec{z} = \vec{x}||\vec{w}$
for some $\vec{w}$

Evaluation queries

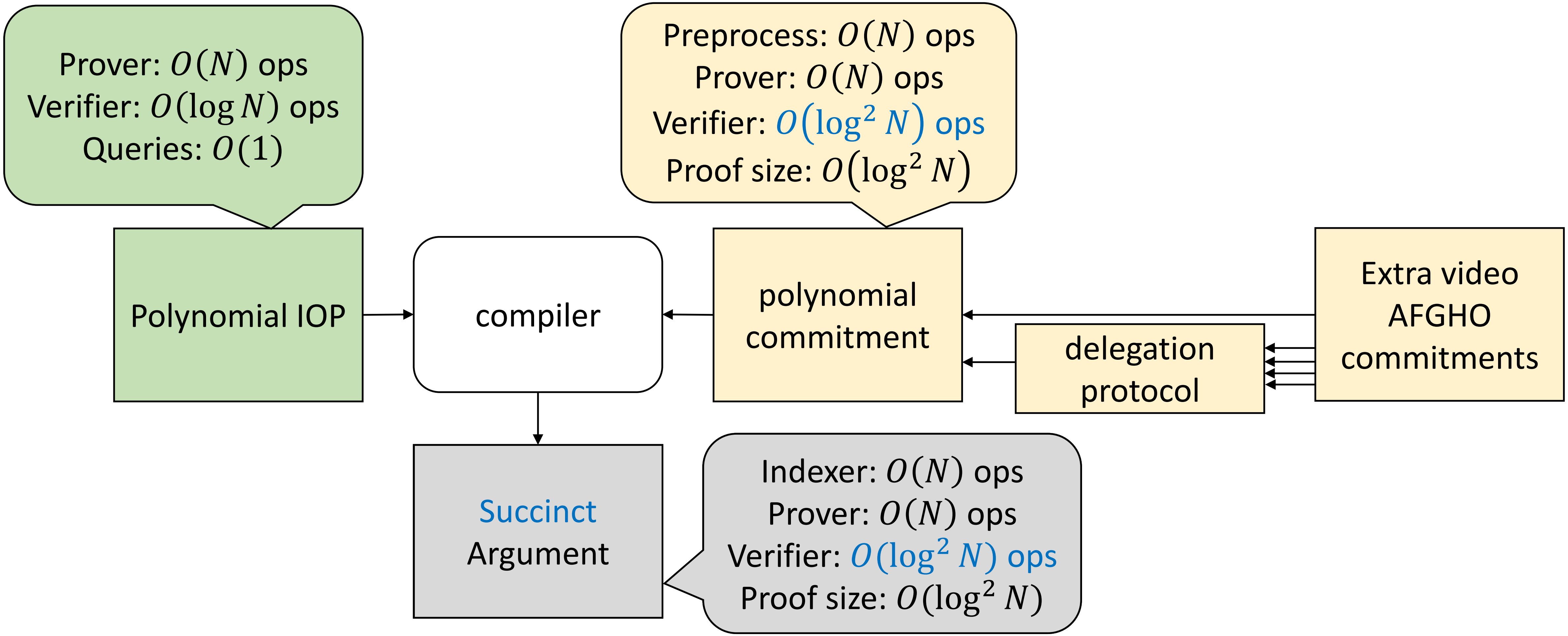
Committed Answers

Committed Eval protocols

Transform verifier checks using Σ-protocols as with GKR

Verifier evaluates $\tilde{A}, \tilde{B}, \tilde{C}$ for themselves in Lincheck

Summary of CSAT argument result



Prover: $O(N)$ ops
Verifier: $O(\log N)$ ops
Queries: $O(1)$

Polynomial IOP

compiler

Preprocess: $O(N)$ ops
Prover: $O(N)$ ops
Verifier: $O(\log^2 N)$ ops
Proof size: $O(\log^2 N)$

polynomial commitment

delegation protocol

Extra video AFGHO commitments

Succinct Argument

Indexer: $O(N)$ ops
Prover: $O(N)$ ops
Verifier: $O(\log^2 N)$ ops
Proof size: $O(\log^2 N)$

Security sketch

Completeness:

- Inherited from the underlying IP/IOP, hidden Eval and Σ-protocols

SHVZK:

- Sample uniformly random commitments C and C_z for each polynomial f and evaluation z.
- Sample uniformly random commitments for each small message.
- Simulate the Σ-protocols and committed Eval protocols.

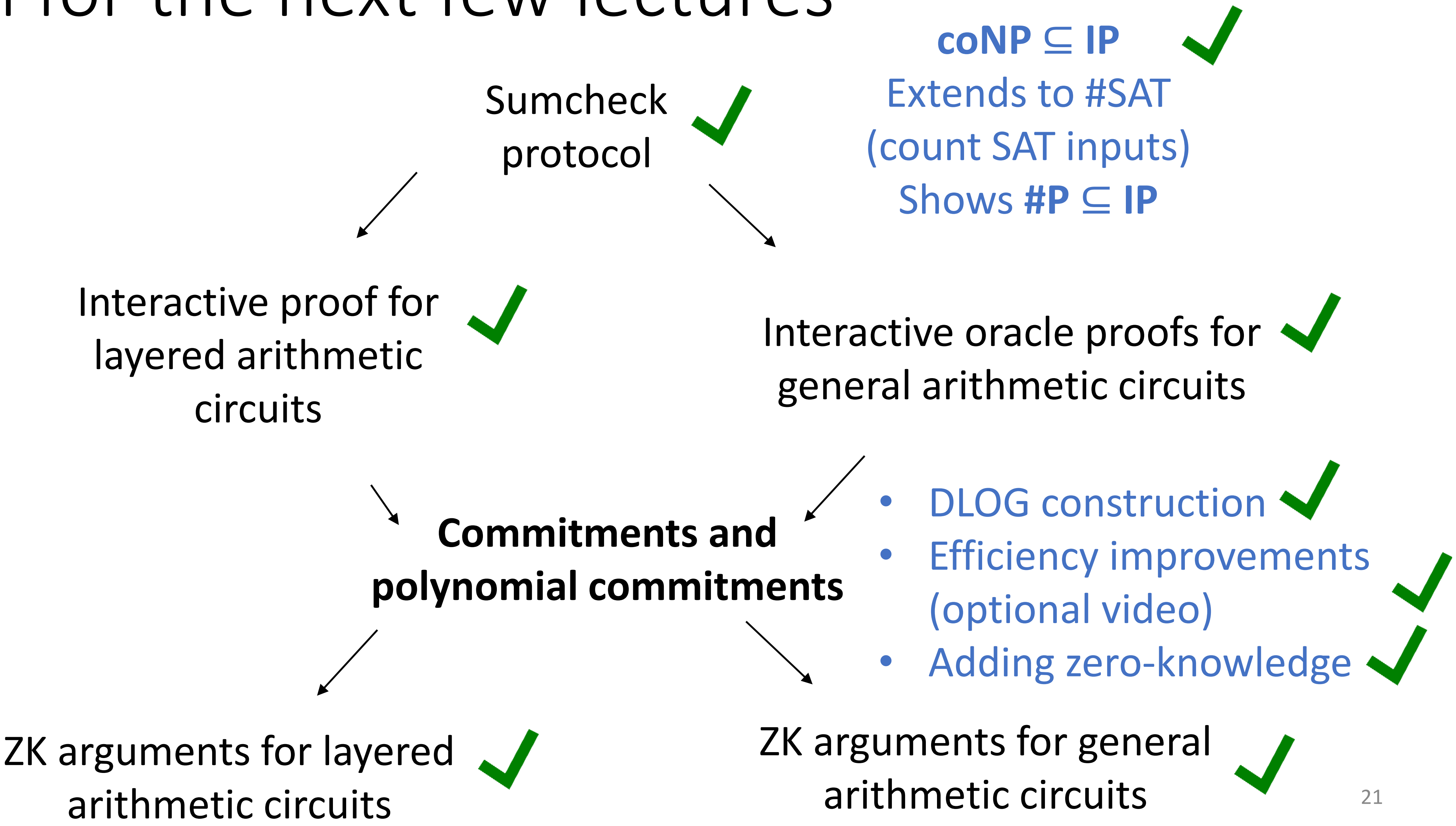
Soundness:

$$\text{ZK argument adversary} \xrightarrow{\text{Hidden eval, } \Sigma\text{-protocol extractors}} \text{IP/IOP argument adversary}$$

Knowledge error bounds in terms of

- IOP/IP soundness error
- Comitted Eval knowledge error
- Σ-protocol knowledge error

Plan for the next few lectures



Sumcheck protocol ✓

coNP $\subseteq$ IP ✓

Extends to #SAT (count SAT inputs)

Shows **#P $\subseteq$ IP**

Interactive proof for layered arithmetic circuits ✓

Interactive oracle proofs for general arithmetic circuits ✓

Commitments and polynomial commitments

- DLOG construction ✓
- Efficiency improvements (optional video) ✓
- Adding zero-knowledge ✓

ZK arguments for layered arithmetic circuits ✓

ZK arguments for general arithmetic circuits ✓

Course Outline (13 lectures)

1. **Introduction and definitions** ~2 lectures

2. **Sigma protocols** ~3 lectures

3. **ZK arguments with short proofs** ~4 lectures

4. **Non-interactive zero-knowledge** ~3 lectures

5. **Bonus material?** ~1 lecture

Non-interactive zero-knowledge

- Non-interactive zero-knowledge (NIZK) definitions

Pairing-based constructions of NIZK

- From reasonable cryptographic assumptions

$O(N)$ proof size for
Boolean circuits

- From strong cryptographic assumptions

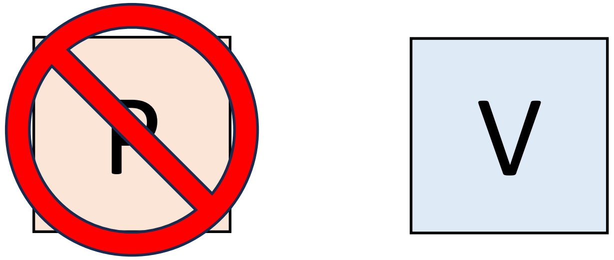
$O(1)$ proof size for
Arithmetic circuits

Non-interactive proofs

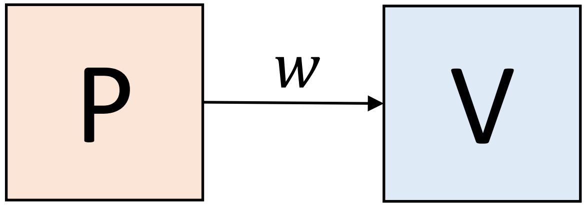
- A non-interactive proof consists of a single message from P to V.

Examples:

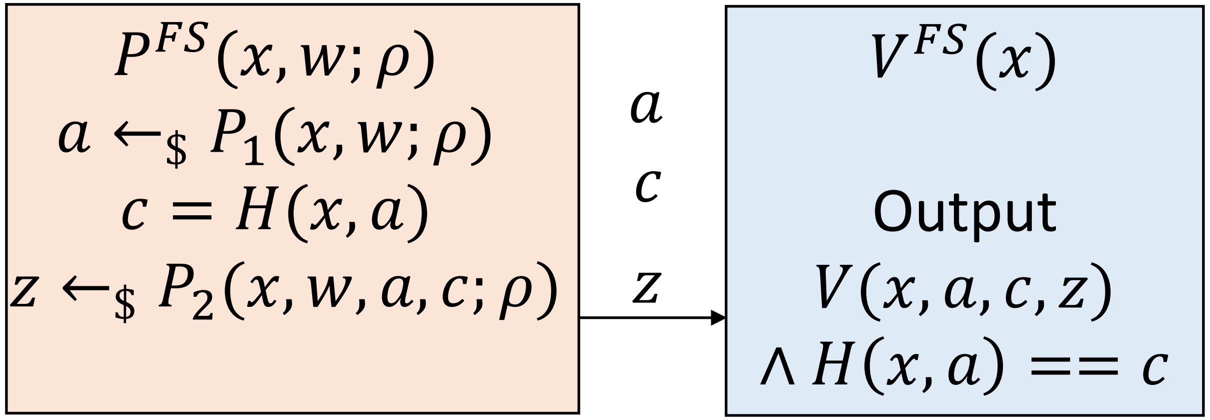
$\mathcal{L} \in$ **P** ⟷ $x \in \mathcal{L}$ is obvious and needs no proof!



$\mathcal{L} \in$ **NP** ⟷ $x \in \mathcal{L}$ has a proof which is easy to check



Σ-protocol for $\mathcal{R}$ + Fiat-Shamir Heuristic



$P^{FS}(x, w; \rho)$

$a \leftarrow_{\$} P_1(x, w; \rho)$

$c = H(x, a)$

$z \leftarrow_{\$} P_2(x, w, a, c; \rho)$

$V^{FS}(x)$

Output

$V(x, a, c, z)$

$\wedge H(x, a) == c$

The complexity class **BPP**

Definition:

$\mathcal{L} \in$ **BPP** if $\exists$ efficient decision algorithm M (polynomial time in $|x|$) satisfying:

- $\forall x \in \mathcal{L}, \Pr_r[M(x,r) = 1] \geq {}^{3}/_{4}$

"True statements usually accepted"

- $\forall x \notin \mathcal{L}, \Pr_r[M(x,r) = 1] \leq {}^{1}/_{2}$

"False statements usually rejected"

Any constants with a gap define the same complexity class

Why doesn't this apply to interactive proofs or the examples on the previous slide?

Impossibility of interesting NIZK?

Theorem:

- Let (P, V) be a NIZK for $\mathcal{L}$. Then $\mathcal{L} \in$ **BPP**.

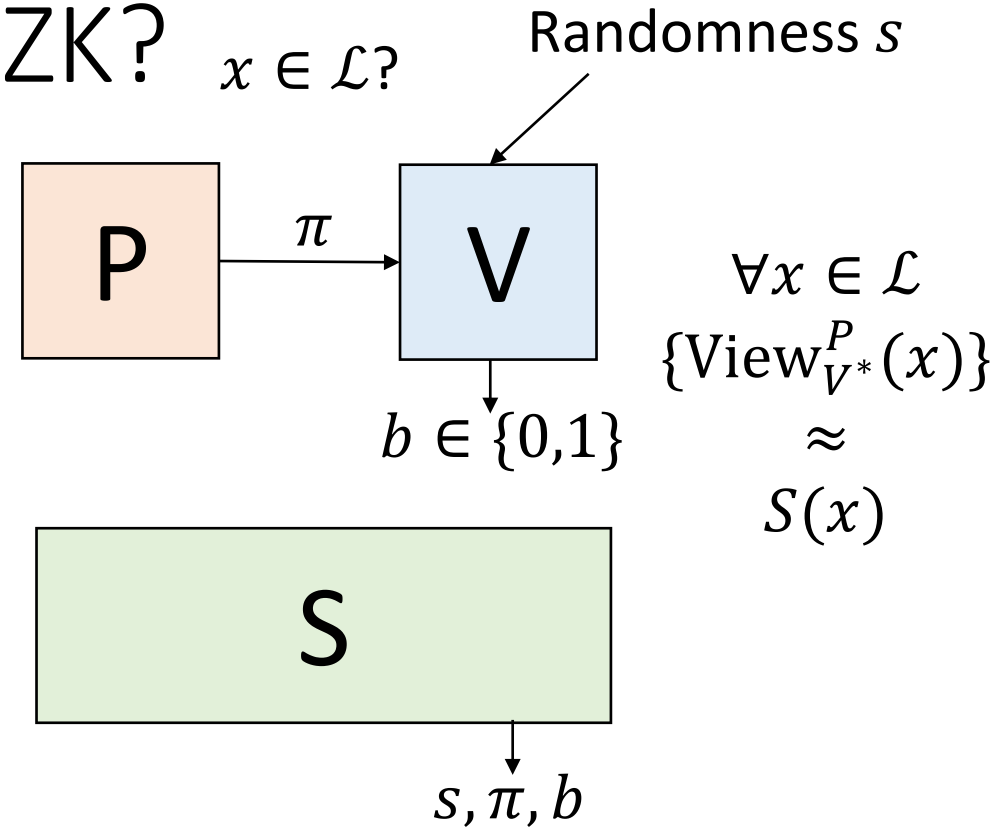
Proof: **BPP** decider M for $\mathcal{L}$:

- $x \in \mathcal{L} \Rightarrow S$ output indistinguishable from P output (by ZK of (P, V)).

$\Rightarrow M$ outputs 1 with probability $\geq \frac{3}{4}$

(by completeness of (P, V)).

- $x \notin \mathcal{L} \Rightarrow M$ outputs 1 with probability $\leq \frac{1}{2}$ (by soundness of (P, V)).
- M is efficient because S and V are.

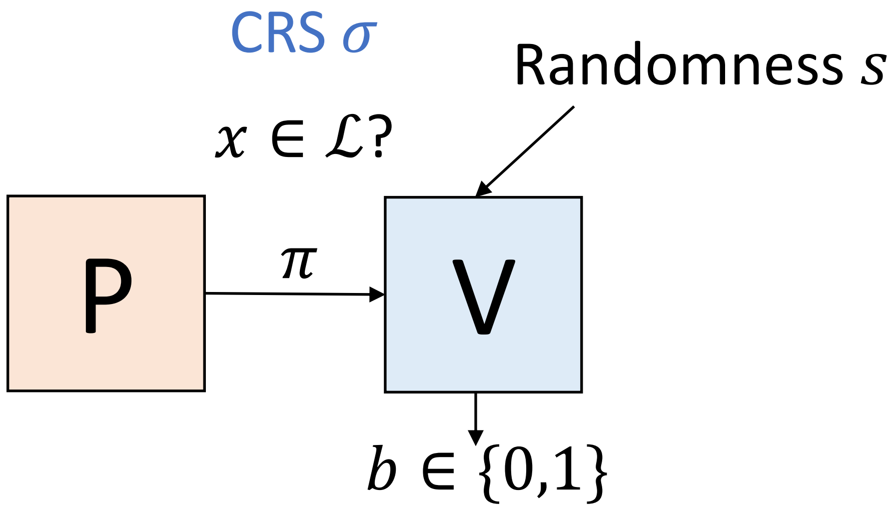


$M(x)$

1. $s, \pi, b \leftarrow S(x)$.
2. Sample s'.
3. Output $V(x, \pi; s') \in \{0,1\}$.

Syntax for non-interactive zero-knowledge

- Include a common reference string (CRS) containing ingredients used in the proof.


CRS σ

Randomness s

$x \in \mathcal{L}$?

P

π

V

$b \in \{0,1\}$

Definition:

A *non-interactive proof system* for an NP relation $\mathcal{R}$ consists of three efficient algorithms (K, P, V) which are

- the CRS generator $K\left(1^{\lambda}\right) \rightarrow \sigma$,
- the prover $P(\sigma, x, w) \rightarrow \pi$,
- the verifier $V(\sigma, x, \pi) \rightarrow b$.

Suppressing random inputs

K may take $|x|$ or even x as input

Ideally σ is uniformly random but may be structured

Security of non-interactive proofs

Easy to modify to get computational and statistical security notions

- **Completeness:** $\forall (x, w) \in \mathcal{R}$,

$$\Pr[b = 1 |\, \sigma \leftarrow K(1^\lambda), \pi \leftarrow P(\sigma, x, w), b \leftarrow V(\sigma, x, \pi)] = 1$$

Adaptive

- **Soundness:** $\forall P^*$,

$$\Pr[x \notin \mathcal{L}_\mathcal{R}, b = 1 |\, \sigma \leftarrow K(1^\lambda), (x, \pi) \leftarrow P^*(\sigma), b \leftarrow V(\sigma, x, \pi)] \approx 0$$

Adaptive

- **Zero-knowledge:** $\exists$ efficient simulators (S_1, S_2) such that $\forall A$ producing $(x, w) \in \mathcal{R}$,

$$\{(\sigma, \pi) : \sigma \leftarrow K(1^\lambda), (x, w) \leftarrow A(\sigma), \pi \leftarrow P(\sigma, x, w)\}$$
$$\approx \{(\sigma, \pi) : (\sigma, \tau) \leftarrow S_1(1^\lambda), (x, w) \leftarrow A(\sigma), \pi \leftarrow S_2(\sigma, x, \tau)\}$$

Simulated σ indistinguishable from normal σ

Simulation trapdoor τ (replaces oracle access to V^*)

In non-adaptive definitions, x is not chosen based on σ

These are *single-theorem* definitions. No security guarantees reusing σ for many x.

Knowledge soundness

Definition:

(K, P, V) is a *proof of knowledge* for a relation $\mathcal{R}$ if ∃ efficient extractors E_1, E_2 such that for all P^*,

Extractor's σ indistinguishable from normal σ

Extraction trapdoor ξ
(replaces oracle access to P^*)

- $\{\sigma : (\sigma, \xi) \leftarrow E_1(1^\lambda)\} \approx \{\sigma : \sigma \leftarrow K(1^\lambda)\}$, and
- $\Pr\left[\begin{array}{c} V(\sigma, x, \pi) = 0 \\ \vee\ (x, w) \in \mathcal{R} \end{array} : \begin{array}{c} (\sigma, \xi) \leftarrow E_1(1^\lambda), (x, \pi) \leftarrow P^*(\sigma) \\ w \leftarrow E_2(\sigma, \xi, x, pi) \end{array}\right] \approx 1$

$V(\sigma, x, \pi) \Rightarrow (x, w) \in \mathcal{R}$

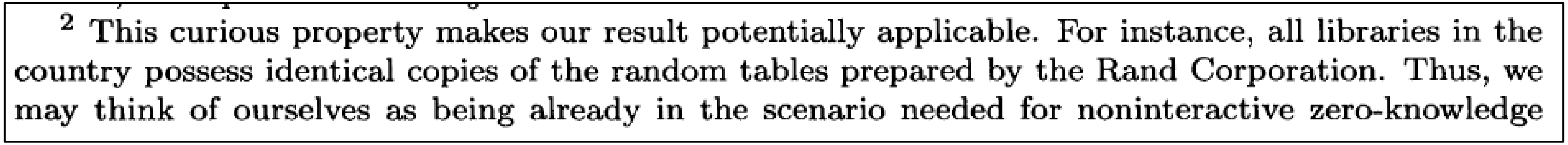
How can we trust the CRS?

- Simulation trapdoors let us produce proofs without knowing witnesses (breaking soundness)
- Extraction trapdoors let us extract witnesses from proofs (breaking ZK)
- Trapdoor σ are indistinguishable from normal σ.

Mitigate risks using

- “Subversion resistant” NIZK constructions
- “Updatable CRS” NIZK constructions
- “Verifiable CRS” NIZK constructions
- MPC protocols to generate σ

How can we trust the CRS?



[2] This curious property makes our result potentially applicable. For instance, all libraries in the country possess identical copies of the random tables prepared by the Rand Corporation. Thus, we may think of ourselves as being already in the scenario needed for noninteractive zero-knowledge

Tech

Edward Snowden Played Key Role in Zcash Privacy Coin's Creation

The NSA whistleblower and privacy advocate was one of six participants in the cryptocurrency's fabled 2016 "trusted setup" ceremony, using a pseudonym.

By Naomi Brockwell Apr 27, 2022 at 10:17 p.m. Updated Apr 28, 2022 at 7:13 p.m.

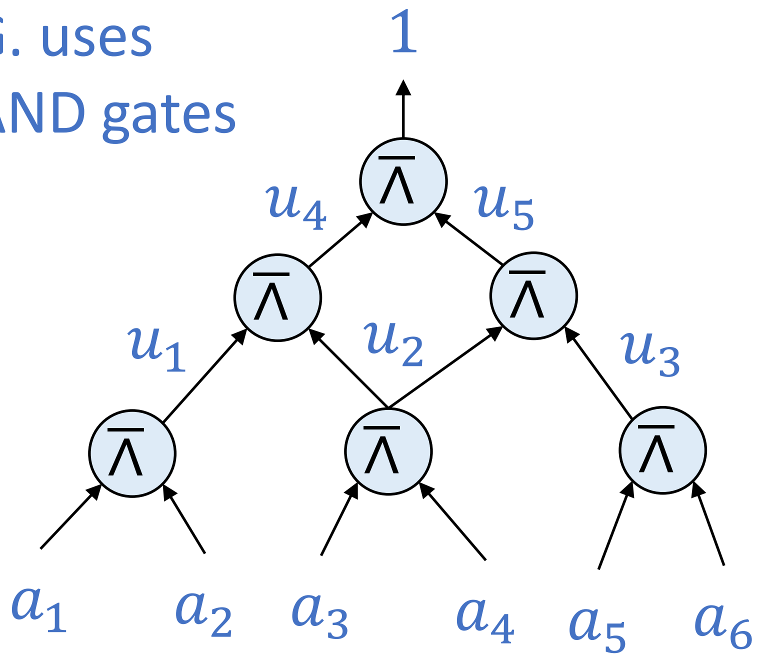
Boolean circuit NIZK idea

Instance: circuit over $\mathbb{Z}_2$ with output.

Witness: input wire values giving correct output.

a	b	c	$\overline{a \wedge b} == c$	$a + b + 2c - 2$
0	0	0	0	-2
0	0	1	1	0
0	1	0	0	-1
0	1	1	1	1
1	0	0	0	-1
1	0	1	1	1
1	1	0	1	0
1	1	1	0	2

W.L.O.G. uses only NAND gates



$$\overline{a \wedge b} = c \Leftrightarrow a + b + 2c - 2 \in \{0,1\}$$

Proof idea:

- Commit to each wire value.
- Prove each wire value $\in \{0,1\}$.
- Prove $a + b + 2c - 2 \in \{0,1\}$ for wires around each gate.

Need a commitment scheme with NI bit proofs

Composite-order symmetric pairings

Definition:

A *symmetric bilinear group* is a triple of two groups of order $n = pq$ (where p, q are distinct primes) and a *bilinear map* $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ satisfying

$$\forall a, b \in \mathbb{Z}_p, \forall G_1, G_2 \in \mathbb{G},$$
$$e(a \cdot G_1, b \cdot G_2) = ab \cdot e(G_1, G_2)$$

Pairing maps ‘multiply DLOGs’

which is non-degenerate i.e.

If $\mathbb{G} = \langle G_1 \rangle = \langle G_2 \rangle$, then $\mathbb{G}_T = \langle e(G_1, G_2) \rangle$

Changes from last time:

- Order n instead of p
- “Type 1” setting with $\mathbb{G}_1 \cong \mathbb{G}_2 \cong \mathbb{G}$

Facts

Claim: (symmetry)

$\forall G_1, G_2 \in \mathbb{G}, e(G_1, G_2) = e(G_2, G_1)$

Proof:

Let $\mathbb{G} = \langle G \rangle$. Write $G_1 = a \cdot G$ and $G_2 = b \cdot G$. Then

$e(G_1, G_2) = ab \cdot e(G, G) = ba \cdot e(G, G) = e(G_2, G_1)$ by bilinearity.

Both claims together imply homomorphism in the right input

Claim:

$\forall G_1, G_2, H \in \mathbb{G}, e(G_1 + G_2, H) = e(G_1, H) + e(G_2, H)$.

Proof: exercise

The Boneh-Goh-Nissim Cryptosystem

Large enough that n is difficult to factor e.g. $O(\lambda^3)$ bits

Setup: on input $\lambda \in \mathbb{N}$, sample distinct primes p, q and composite order symmetric bilinear group $e, \mathbb{G}, \mathbb{G}_T$ of order $n = pq$, and $G, H \in \mathbb{G}$.

Sample $B \in \mathbb{N}$. Output $pp \coloneqq (e, \mathbb{G}, \mathbb{G}_T, G, H, n, B)$.

$B \leq \mathrm{poly}(\lambda)$

Exact sampling method to be discussed

Commit: given $m \in \{0, \ldots, B-1\}, pp$, sample $r \leftarrow \mathbb{Z}_n$.

Compute $C = m \cdot G + r \cdot H$. Output (C, r).

Verify: check $m \in \{0, \ldots, B-1\}$ and $C == m \cdot G + r \cdot H$.

Homomorphic (looks like Pedersen)

For us, a commitment scheme (but can be used as an encryption scheme)

Dual-mode parameter generation

- $\mathbb{G}$ has order $n = pq$.
- $\text{Setup}_{binding}: G \leftarrow \mathbb{G},\ s \leftarrow \mathbb{Z}_n^*,\ H = ps \cdot G.$

random generator of order q subgroup

a generator

- $\text{Setup}_{hiding}: G \leftarrow \mathbb{G},\ s \leftarrow \mathbb{Z}_n^*,\ H = s \cdot G.$

random generator of whole group $\mathbb{G}$

Definition:

The subgroup hiding assumption holds if

$$\{\text{Setup}_{binding}(1^\lambda)\} \approx_c \{\text{Setup}_{hiding}(1^\lambda)\}$$

The BGN cryptosystem is hiding

Proof:

random generator of whole group $\mathbb{G}$

- Using Setup_{hiding}: $G \leftarrow_\$ \mathbb{G},\ s \leftarrow_\$ \mathbb{Z}_n^*,\ H = s \cdot G.$
- For $r \leftarrow_\$ \mathbb{Z}_n, r \cdot H$ is uniformly random in $\mathbb{G}$.
- Hence $C = m \cdot G + r \cdot H$ is uniformly random in $\mathbb{G}$.
- Therefore Setup_{hiding} gives *perfect* hiding.

  With Setup_{hiding}, BGN is *equivocable* with equivocation key $s.$

  $C = m \cdot G + r \cdot H = m' \cdot G + r' \cdot H$ where $a = \frac{m - m\prime}{s} mod\ n.$

- The output of $\mathrm{Setup}_{binding}$ is computationally indistinguishable from Setup_{hiding} under the subgroup hiding assumption.
- Therefore $\mathrm{Setup}_{binding}$ still gives *computational* hiding.

The BGN cryptosystem is perfectly binding

Proof:

random generator of order q subgroup

- Using Setup$_{binding}$: $G \leftarrow_\$ \mathbb{G},\ s \leftarrow_\$ \mathbb{Z}_n^*,\ H = ps \cdot G$.
- Suppose $C = m \cdot G + r \cdot H = m' \cdot G + r' \cdot H$ for distinct $m, m' \in \{0, \dots, B-1\}$.
- Then $e(C, q \cdot G) = e(m \cdot G + r \cdot H, q \cdot G) = e(m \cdot G + rps \cdot G, q \cdot G)$
  $= qm \cdot e(G,G) + rspq \cdot e(G,G) = qm \cdot e(G,G)$
- Similarly, $e(C, q \cdot G) = qm' \cdot e(G,G)$. Hence $q(m - m') \cdot e(G,G) = 0$.
- By non-degeneracy, $e(G,G)$ has order n so $n \mid q(m - m')$.
- Hence $q(m - m') = kn$, so $(m - m') = kp$.
- $m \equiv m' mod\ p$ but $B \ll p$ so $m = m'$.

With Setup$_{binding}$, BGN is *extractable* with extraction key s.

Compute $e(C, q \cdot G)$, check whether it is equal to $qm \cdot e(G,G)$ for each $m \in \{0, \dots, B-1\}$.

- The output of Setup$_{hiding}$ is computationally indistinguishable from Setup$_{binding}$ under subgroup hiding, so Setup$_{hiding}$ gives *computational* binding.

ETH Zurich, Department of Computer Science — Dr. Jonathan Bootle
HS 2023 — Karen Klein, Varun Maram, Antonio Merino Gallardo

Zero-Knowledge Proofs

Exercise 11

11.1 AFGHO Commitments

Recall the definition of AFGHO commitments [AFG$^+$10] presented in the lectures. Given a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where the groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are of prime order p, we have generators g, h of $\mathbb{G}_2$ to be the commitment keys. Then a commitment to message $m \in \mathbb{G}_1$ using randomness $r \in \mathbb{G}_1$ is computed as $C = e(m, g) + e(r, h) \in \mathbb{G}_T$ (verification recomputes the commitment).

a) Show that the above AFGHO commitment scheme is perfectly hiding and computationally binding under the DPAIR assumption[1].

Consider the following Σ-protocol to prove knowledge of an opening for an AFGHO commitment $C = e(m, g) + e(r, h) \in \mathbb{G}_T$.

$\underline{P}$		$\underline{V}$
$y, s \leftarrow_{\$} \mathbb{G}_1$		
$D = e(y, g) + e(s, h) \in \mathbb{G}_T$	$\xrightarrow{D}$	
	$\xleftarrow{c}$	$c \leftarrow_{\$} \mathbb{Z}_p$
$z = c \cdot m + y$		
$t = c \cdot r + s$	$\xrightarrow{z, t}$	Accepts if and only if
		$e(z, g) + e(t, h) = c \cdot C + D.$

b) Prove that the above protocol is complete, 2-special-sound and special honest-verifier zero-knowledge (SHVZK).

11.2 Sumchecks and Discrete-Logarithm-Based Polynomial Commitments

Let $\mathbb{G}$ be a prime-order group ($p \coloneqq |G|$) and consider $n = 2^\ell$ group elements $g_0, \ldots, g_{n-1}$. Given a Pedersen commitment $C = \langle \mathbf{a}, \mathbf{g} \rangle = \sum_{i=0}^{n-1} a_i \cdot g_i$, knowledge of an opening can be proved with logarithmic prover communication complexity using split-and-fold techniques as seen in the lectures.

In this exercise, we want to show that similar split-and-fold based proofs of knowledge of Pedersen commitment openings can be abstracted by sumcheck protocols.

a) Given $\mathbf{a} \in \mathbb{Z}_p^n$ and $\mathbf{g} \in \mathbb{G}^n$, define multi-linear extension polynomials $\tilde{\mathbf{a}} : \mathbb{Z}_p^\ell \to \mathbb{Z}_p$ and $\tilde{\mathbf{g}} : \mathbb{Z}_p^\ell \to \mathbb{G}$ corresponding to $\mathbf{a}$ and $\mathbf{g}$ respectively.

---

[1] Roughly speaking, the DPAIR assumption states that given a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and generators g, h of $\mathbb{G}_2$, it is computationally hard to come up with a non-trivial pair $m, r \in \mathbb{G}_1$ (i.e., $(m, r) \neq (0, 0)$) such that $e(m, g) + e(r, h) = 0 \in \mathbb{G}_T$.

Now consider the polynomial $p : \mathbb{Z}_p^\ell \to \mathbb{G}$ defined as the product of the polynomials $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{g}}$; namely, $p(X_0, \ldots, X_{\ell-1}) = \tilde{\mathbf{a}}(X_0, \ldots, X_{\ell-1}) \cdot \tilde{\mathbf{g}}(X_0, \ldots, X_{\ell-1})$. Note that the statement "$C = \langle \mathbf{a}, \mathbf{g} \rangle = \sum_{i=0}^{n-1} a_i \cdot g_i$" related to the opening of Pedersen commitment C is equivalent to the following instance of the sumcheck protocol with respect to the polynomial p:

$$\sum_{\omega_0, \ldots, \omega_{\ell-1} \in \{0,1\}} p(\omega_0, \ldots, \omega_{\ell-1}) = C$$

b) Consider the following variant of the sumcheck protocol on polynomial p:

$$\begin{array}{ll}
\underline{P} & \underline{V} \\
& \\
\xrightarrow{\qquad q_0(X_0) \qquad} & \\
\xleftarrow{\qquad\quad r_0 \quad\qquad} & r_0 \leftarrow_{\$} \mathbb{Z}_p \\
\xrightarrow{\tilde{\mathbf{a}}(r_0, X_1, \ldots, X_{\ell-1})} & \text{Accepts if and only if } \sum_{\omega_0 \in \{0,1\}} q_0(\omega_0) = C, \text{ and} \\
& \sum_{\omega_1, \ldots, \omega_{\ell-1} \in \{0,1\}} \tilde{\mathbf{a}}(r_0, \omega_1, \ldots, \omega_{\ell-1}) \cdot \tilde{\mathbf{g}}(r_0, \omega_1, \ldots, \omega_{\ell-1}) = q_0(r_0).
\end{array}$$

where the prover computes the polynomial $q_0(X_0) = \sum_{\omega_1, \ldots, \omega_{\ell-1} \in \{0,1\}} p(X_0, \omega_1, \ldots, \omega_{\ell-1})$ in the first round; also note that in the third round, the polynomial $\tilde{\mathbf{g}}$ corresponding to the "key" of Pedersen commitments above is public and known to the verifier beforehand, in contrast to the "opening" polynomial $\tilde{\mathbf{a}}$.

Show that the above protocol satisfies 3-special-soundness.

HINT: You might want to describe the polynomial $q_0(X_0)$ in the $(X_0^2, X_0(1-X_0), (1-X_0)^2)$-basis; i.e., $q_0(X_0) = X_0^2 \cdot C_0 + X_0(1-X_0) \cdot C_1 + (1-X_0)^2 \cdot C_2$ for $C_0, C_1, C_2 \in \mathbb{G}$.

References

[AFG$^+$10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 209–236. Springer, 2010.

ETH Zurich, Department of Computer Science Dr. Jonathan Bootle
HS 2023 Karen Klein, Varun Maram, Antonio Merino Gallardo

Zero-Knowledge Proofs

Exercise 11

11.1 AFGHO Commitments

Recall the definition of AFGHO commitments [AFG$^+$10] presented in the lectures. Given a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where the groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are of prime order p, we have generators g, h of $\mathbb{G}_2$ to be the commitment keys. Then a commitment to message $m \in \mathbb{G}_1$ using randomness $r \in \mathbb{G}_1$ is computed as $C = e(m, g) + e(r, h) \in \mathbb{G}_T$ (verification recomputes the commitment).

a) Show that the above AFGHO commitment scheme is perfectly hiding and computationally binding under the DPAIR assumption[1].

Solution: To prove perfect hiding, note that for *any* message $m \in \mathbb{G}_1$, we have the corresponding (honestly generated) commitment $C = e(m, g) + e(r, h)$ to be a uniformly random element in $\mathbb{G}_T$ which is *independent* of m. Because for uniform and independent randomness $r = s \cdot g' \leftarrow_\$ \mathbb{G}_1$ (equivalently, $s \leftarrow_\$ \mathbb{Z}_p$) where g' is a generator of $\mathbb{G}_1$, we have $e(r, h) = e(s \cdot g', h) = s \cdot e(g', h)$, and since $e(g', h)$ is a generator of $\mathbb{G}_T$ from the non-degeneracy property, we also have $s \cdot e(g', h)$ to be a uniformly random element in $\mathbb{G}_T$ which is independent of $e(m, g)$. Hence, the above distribution of C follows from a "one-time pad"-style argument.

To prove computational binding under the DPAIR assumption, assume there is an efficient adversary $\mathcal{A}$ which breaks the binding property of AFGHO commitments by providing a tuple (C, r, r', m, m') where we have the messages $m \neq m'$ such that $C = e(m, g) + e(r, h) = e(m', g) + e(r', h)$. However, using the linearity of e, it's not hard to obtain $e(m - m', g) + e(r - r', h) = 0 \in \mathbb{G}_T$ which in-turn allows us to break the DPAIR assumption using the pair $(m - m', r - r')$; note that our DPAIR "solution" is non trivial since $m - m' \neq 0$.

Consider the following Σ-protocol to prove knowledge of an opening for an AFGHO commitment $C = e(m, g) + e(r, h) \in \mathbb{G}_T$.

b) Prove that the above protocol is complete, 2-special-sound and special honest-verifier zero-knowledge (SHVZK).

Solution: To show completeness, note that the verifier's check always passes since

$$\begin{aligned} e(z, g) + e(t, h) &= e(c \cdot m + y, g) + e(c \cdot r + s, h) \\ &= \big(e(c \cdot m, g) + e(y, g)\big) + \big(e(c \cdot r, h) + e(s, h)\big) \\ &= c \cdot \big(e(m, g) + e(r, h)\big) + \big(e(y, g) + e(s, h)\big) = c \cdot C + D. \end{aligned}$$

---

[1] Roughly speaking, the DPAIR assumption states that given a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and generators g, h of $\mathbb{G}_2$, it is computationally hard to come up with a non-trivial pair $m, r \in \mathbb{G}_1$ (i.e., $(m, r) \neq (0, 0)$) such that $e(m, g) + e(r, h) = 0 \in \mathbb{G}_T$.

$$\begin{array}{lcl}
\underline{P} & & \underline{V} \\
 & & \\
y, s \leftarrow_{\$} \mathbb{G}_1 & & \\
D = e(y,g) + e(s,h) \in \mathbb{G}_T & \xrightarrow{\quad\quad D \quad\quad} & \\
 & \xleftarrow{\quad\quad c \quad\quad} & c \leftarrow_{\$} \mathbb{Z}_p \\
z = c \cdot m + y & & \\
t = c \cdot r + s & \xrightarrow{\quad\quad z, t \quad\quad} & \text{Accepts if and only if} \\
 & & e(z,g) + e(t,h) = c \cdot C + D.
\end{array}$$

To show 2-special-soundness, given 2 accepting transcripts $(D, c, (z, t))$ and $(D, c', (z', t'))$ – namely,

$$e(z,g) + e(t,h) = c \cdot C + D \tag{1}$$

$$e(z',g) + e(t',h) = c' \cdot C + D \tag{2}$$

with $c' \neq c$, we can efficiently extract an opening for commitment C as follows. Subtracting (1) and (2), and using linearity of e, we get

$$\begin{aligned}
(c - c') \cdot C &= e(z - z', g) + e(t - t', h) \\
\implies C &= \frac{1}{c - c'} \cdot e(z - z', g) + \frac{1}{c - c'} \cdot e(t - t', h) \\
\implies C &= e\Big(\frac{1}{c - c'} \cdot (z - z'), g\Big) + e\Big(\frac{1}{c - c'} \cdot (t - t'), h\Big),
\end{aligned}$$

where "$\frac{1}{c-c'}$" represents the inverse of $(c - c')$ in $\mathbb{Z}_p$ (it is guaranteed to exist since $c \neq c'$). Hence, we have extracted an opening of C with the message being $m = \frac{1}{c-c'} \cdot (z - z') \in \mathbb{G}_1$ and randomness $r = \frac{1}{c-c'} \cdot (t - t') \in \mathbb{G}_1$.
To prove the SHVZK property, note that the corresponding simulator – on input a challenge $c \in \mathbb{Z}_p$ – samples $z, t \leftarrow_{\$} \mathbb{G}_1$ uniformly at random and computes $D = e(z, g) + e(t, h) - c \cdot C$. To see why the distribution of simulator's output $(D, c, (z, t))$ is indistinguishable from the honest verifier's view on challenge c, note that in an honest execution of the above protocol, the values $z = c \cdot m + y$, $t = c \cdot r + s$ are independent and uniformly random values in $\mathbb{G}_1$ (by applying a "one-time pad"-style argument, since $y, s \leftarrow_{\$} \mathbb{G}_1$ are sampled independently).

11.2 Sumchecks and Discrete-Logarithm-Based Polynomial Commitments

Let $\mathbb{G}$ be a prime-order group ($p := |\mathbb{G}|$) and consider $n = 2^\ell$ group elements $g_0, \ldots, g_{n-1}$. Given a Pedersen commitment $C = \langle \mathbf{a}, \mathbf{g} \rangle = \sum_{i=0}^{n-1} a_i \cdot g_i$, knowledge of an opening can be proved with logarithmic prover communication complexity using split-and-fold techniques as seen in the lectures.

In this exercise, we want to show that similar split-and-fold based proofs of knowledge of Pedersen commitment openings can be abstracted by sumcheck protocols.

a) Given $\mathbf{a} \in \mathbb{Z}_p^n$ and $\mathbf{g} \in \mathbb{G}^n$, define multi-linear extension polynomials $\tilde{\mathbf{a}} : \mathbb{Z}_p^\ell \to \mathbb{Z}_p$ and $\tilde{\mathbf{g}} : \mathbb{Z}_p^\ell \to \mathbb{G}$ corresponding to $\mathbf{a}$ and $\mathbf{g}$ respectively.

Solution: We can simply define

$$\tilde{\mathbf{a}}(X_0, \ldots, X_{\ell-1}) = \sum_{i=0}^{n-1} \Big(\prod_{j=0}^{\ell-1} (X_j i_j + (1 - X_j)(1 - i_j)) \Big) a_i, \text{ and}$$

$$\tilde{\mathbf{g}}(X_0, \ldots, X_{\ell-1}) = \sum_{i=0}^{n-1} \Big(\prod_{j=0}^{\ell-1} (X_j i_j + (1 - X_j)(1 - i_j)) \Big) \cdot g_i,$$

where $(i_0, i_1, \dots, i_{\ell-1})$ is the ℓ-bit binary decomposition of integer i.

Now consider the polynomial $p : \mathbb{Z}_p^\ell \to \mathbb{G}$ defined as the product of the polynomials $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{g}}$; namely, $p(X_0, \dots, X_{\ell-1}) = \tilde{\mathbf{a}}(X_0, \dots, X_{\ell-1}) \cdot \tilde{\mathbf{g}}(X_0, \dots, X_{\ell-1})$. Note that the statement "$C = \langle \mathbf{a}, \mathbf{g} \rangle = \sum_{i=0}^{n-1} a_i \cdot g_i$" related to the opening of Pedersen commitment C is equivalent to the following instance of the sumcheck protocol with respect to the polynomial p:

$$\sum_{\omega_0, \dots, \omega_{\ell-1} \in \{0,1\}} p(\omega_0, \dots, \omega_{\ell-1}) = C$$

b) Consider the following variant of the sumcheck protocol on polynomial p:

$$\begin{array}{ll}
\underline{P} & \underline{V} \\
\xrightarrow{\quad q_0(X_0) \quad} & \\
\xleftarrow{\quad r_0 \quad} & r_0 \leftarrow_{\$} \mathbb{Z}_p \\
\xrightarrow{\tilde{\mathbf{a}}(r_0, X_1, \dots, X_{\ell-1})} & \text{Accepts if and only if } \sum_{\omega_0 \in \{0,1\}} q_0(\omega_0) = C, \text{ and} \\
 & \sum_{\omega_1, \dots, \omega_{\ell-1} \in \{0,1\}} \tilde{\mathbf{a}}(r_0, \omega_1, \dots, \omega_{\ell-1}) \cdot \tilde{\mathbf{g}}(r_0, \omega_1, \dots, \omega_{\ell-1}) = q_0(r_0).
\end{array}$$

where the prover computes the polynomial $q_0(X_0) = \sum_{\omega_1, \dots, \omega_{\ell-1} \in \{0,1\}} p(X_0, \omega_1, \dots, \omega_{\ell-1})$ in the first round; also note that in the third round, the polynomial $\tilde{\mathbf{g}}$ corresponding to the "key" of Pedersen commitments above is public and known to the verifier beforehand, in contrast to the "opening" polynomial $\tilde{\mathbf{a}}$.

Show that the above protocol satisfies 3-special-soundness.

HINT: You might want to describe the polynomial $q_0(X_0)$ in the $(X_0^2, X_0(1-X_0), (1-X_0)^2)$-basis; i.e., $q_0(X_0) = X_0^2 \cdot C_0 + X_0(1-X_0) \cdot C_1 + (1-X_0)^2 \cdot C_2$ for $C_0, C_1, C_2 \in \mathbb{G}$.

Solution: Consider 3 accepting transcripts $(q_0(X_0), r_0^{(i)}, \tilde{\mathbf{a}}^{(i)}(X_1, \dots, X_{\ell-1}))_{i \in \{1,2,3\}}$, where each $r_0^{(i)} \in \mathbb{Z}_p$ is pairwise distinct and each $\tilde{\mathbf{a}}^{(i)} : \mathbb{Z}_p^{\ell-1} \to \mathbb{Z}_p$ is a polynomial. Because the verifier accepts these transcripts, we have $q_0(0) + q_0(1) = C$. Since $q_0(X_0)$ is supposed to be a quadratic polynomial, following the hint, we can write $q_0(X_0) = X_0^2 \cdot C_0 + X_0(1-X_0) \cdot C_1 + (1-X_0)^2 \cdot C_2$ for $C_0, C_1, C_2 \in \mathbb{G}$. Also from the verifier's second check, we have

$$\sum_{\omega_1, \dots, \omega_{\ell-1} \in \{0,1\}} \tilde{\mathbf{a}}^{(i)}(\omega_1, \dots, \omega_{\ell-1}) \cdot \tilde{\mathbf{g}}(r_0^{(i)}, \omega_1, \dots, \omega_{\ell-1}) = q_0(r_0^{(i)})$$

for $i \in \{1, 2, 3\}$. From our definition of $\tilde{\mathbf{g}}$ in the previous subtask, it's not hard to see that we have $\tilde{\mathbf{g}}(r_0^{(i)}, \omega_1, \dots, \omega_{\ell-1}) = (1 - r_0^{(i)}) \cdot g_{0\omega} + r_0^{(i)} \cdot g_{1\omega}$, where "$b\omega$"is the integer represented by the binary decomposition $(b, \omega_1, \dots, \omega_{\ell-1})$ (with b being the most significant bit).

Hence writing the verifier's second check as a system of linear equations, we have

$$\begin{aligned}
&\begin{bmatrix} (r_0^{(1)})^2 & (r_0^{(1)})(1 - r_0^{(1)}) & (1 - r_0^{(1)})^2 \\ (r_0^{(2)})^2 & (r_0^{(2)})(1 - r_0^{(2)}) & (1 - r_0^{(2)})^2 \\ (r_0^{(3)})^2 & (r_0^{(3)})(1 - r_0^{(3)}) & (1 - r_0^{(3)})^2 \end{bmatrix} \cdot \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} \\
= &\sum_{\omega_1, \dots, \omega_{\ell-1} \in \{0,1\}} \begin{bmatrix} \tilde{\mathbf{a}}^{(1)}(\omega_1, \dots, \omega_{\ell-1})(1 - r_0^{(1)}) & \tilde{\mathbf{a}}^{(1)}(\omega_1, \dots, \omega_{\ell-1}) r_0^{(1)} \\ \tilde{\mathbf{a}}^{(2)}(\omega_1, \dots, \omega_{\ell-1})(1 - r_0^{(2)}) & \tilde{\mathbf{a}}^{(2)}(\omega_1, \dots, \omega_{\ell-1}) r_0^{(2)} \\ \tilde{\mathbf{a}}^{(3)}(\omega_1, \dots, \omega_{\ell-1})(1 - r_0^{(3)}) & \tilde{\mathbf{a}}^{(3)}(\omega_1, \dots, \omega_{\ell-1}) r_0^{(3)} \end{bmatrix} \cdot \begin{bmatrix} g_{0\omega} \\ g_{1\omega} \end{bmatrix} \qquad (3)
\end{aligned}$$

Now note that the matrix

$$\begin{bmatrix} (r_0^{(1)})^2 & (r_0^{(1)})(1-r_0^{(1)}) & (1-r_0^{(1)})^2 \\ (r_0^{(2)})^2 & (r_0^{(2)})(1-r_0^{(2)}) & (1-r_0^{(2)})^2 \\ (r_0^{(3)})^2 & (r_0^{(3)})(1-r_0^{(3)}) & (1-r_0^{(3)})^2 \end{bmatrix} = \begin{bmatrix} 1 & r_0^{(1)} & (r_0^{(1)})^2 \\ 1 & r_0^{(2)} & (r_0^{(2)})^2 \\ 1 & r_0^{(3)} & (r_0^{(3)})^2 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -2 \\ 1 & -1 & 1 \end{bmatrix} \tag{4}$$

is invertible since the Vandermonde matrix on the left side of the product is invertible, and so is the matrix on the right (non-zero determinant). Hence, by left-multiplying equation (1) with the inverse of matrix described in (2), we get

$$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} = \sum_{\omega_1,\ldots,\omega_{\ell-1}\in\{0,1\}} \mathbf{A}_\omega \cdot \begin{bmatrix} g_{0\omega} \\ g_{1\omega} \end{bmatrix}$$

where we have the matrices $\mathbf{A}_\omega \in \mathbb{Z}_p^{3\times 2}$ indexed by integer $0 \leq \omega \leq n/2 - 1$ represented by the binary decomposition $(\omega_1, \ldots, \omega_{\ell-1})$. Now note that since we are effectively "summing" over all entries of vector $\mathbf{g} \in \mathbb{G}^n$ on the right hand side of the above equation, we can derive vectors $\mathbf{a_0}, \mathbf{a_1}, \mathbf{a_2} \in \mathbb{Z}_p^n$ from matrices $\mathbf{A}_\omega$ such that $C_i = \langle \mathbf{a_i}, \mathbf{g} \rangle$ for $i \in \{0, 1, 2\}$. Now from the verifier's first check – namely, $q_0(0) + q_0(1) = C$ – we have $C_0 + C_2 = C = \langle \mathbf{a_0} + \mathbf{a_2}, \mathbf{g} \rangle$, which leads to the extractor obtaining the opening of commitment C as $\mathbf{a} = \mathbf{a_0} + \mathbf{a_2}$.

Note: If we describe the polynomial $q_0(X_0)$ in the $(1, X_0, X_0^2)$-basis instead – i.e., $q_0(X_0) = C_0 + X_0 \cdot C_1 + X_0^2 \cdot C_2$ – then the solution should be easier. That is, instead of working with the matrix $\begin{bmatrix} (r_0^{(1)})^2 & (r_0^{(1)})(1-r_0^{(1)}) & (1-r_0^{(1)})^2 \\ (r_0^{(2)})^2 & (r_0^{(2)})(1-r_0^{(2)}) & (1-r_0^{(2)})^2 \\ (r_0^{(3)})^2 & (r_0^{(3)})(1-r_0^{(3)}) & (1-r_0^{(3)})^2 \end{bmatrix}$ above, we would *directly* work with the (invertible) Vandermonde matrix $\begin{bmatrix} 1 & r_0^{(1)} & (r_0^{(1)})^2 \\ 1 & r_0^{(2)} & (r_0^{(2)})^2 \\ 1 & r_0^{(3)} & (r_0^{(3)})^2 \end{bmatrix}$. The only difference in the solution with this choice of basis would be: $q_0(0) + q_0(1) = C = C_0 + (C_0 + C_1 + C_2) = 2 \cdot C_0 + C_1 + C_2 = \langle 2\mathbf{a_0} + \mathbf{a_1} + \mathbf{a_2}, \mathbf{g} \rangle$. Hence, the opening of commitment C would instead be $\mathbf{a} = 2\mathbf{a_0} + \mathbf{a_1} + \mathbf{a_2}$.

References

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 209–236. Springer, 2010.