Benoît Libert
Zama

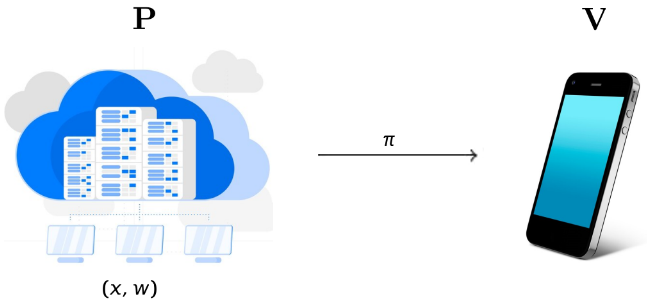
April 16, 2024

Simulation-Extractable KZG Polynomial Commitments and Applications to HyperPlonk

PKC 2024 - Sydney

ZAMA

Succinct Non-Interactive Arguments



- **Succinctness:** $|\pi| \ll |\mathcal{C}|, |w|$
- **Knowledge-soundness:** a convincing **P** must "know" a witness w such that $R(x, w) = 1$
- **Zero-knowledge:** π leaks nothing about w

SNARKs from PCS and PIOPs

Polynomial Interactive Oracle Proofs (PIOPs)
(Ben-Sasson *et al.*; TCC'16-B)

+

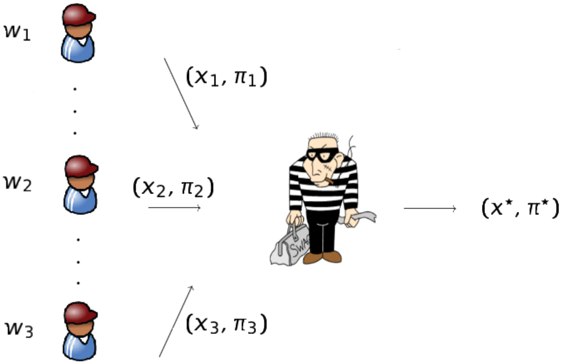
Polynomial Commitment Schemes (PCS)
(Kate-Zaverucha-Goldberg; Asiacrypt'10)

$\overset{\text{FS}}{\Longrightarrow}$ SNARKs

- PIOP: multi-round protocol where **P** sends oracles to polynomials at each round
- PCS: **P** commits to polynomial $f[X] \in \mathbb{F}$ and succinctly proves $y = f(z)$ for any $z \in \mathbb{F}$;
  Evaluation proofs of size $|\pi| \ll \deg(f)$; verification cost $\ll \deg(f)$

Simulation-Extractability: Motivation

- **Knowledge-soundness:** given oracle access to $\mathbf{P}^*$ that outputs a verifying pair $(x^\star, \pi^\star)$, an efficient extractor $\mathcal{E}$ can reconstruct $w^\star$ such that $R(x^\star, w^\star) = 1$
- Adversary observing legitimate proofs may be able to maul them and fake a proof without knowing a witness



Simulation-Extractability

Definition (Sahai, FOCS'99; De Santis *et al.*, Crypto'01):

No PPT attacker can defeat knowledge-extraction after having seen simulated proofs:

$(\mathbf{crs}, \mathbf{tk}) \leftarrow \mathrm{CRS\text{-}Gen}(\lambda, pp)$

$\mathrm{Sim}(\mathbf{tk}, \cdot)$

$\mathbf{crs}$ →

⋮

← x_i

π_i →

⋮

← $(x^\star, \pi^\star)$

Adversary wins if:

- $\mathrm{Verify}_{\mathbf{srs}}(x^\star, \pi^\star) = 1$ and $(x^\star, \pi^\star) \neq (x_i, \pi_i)$ for all queries x_i to $\mathrm{Sim}(\mathbf{tk}, \cdot)$
- $R(x^\star, w^\star) = 0$ where $w^\star \leftarrow \mathcal{E}(\mathbf{tk}, x^\star, \pi^\star)$

Simulation-Extractability

Definition (Sahai, FOCS'99; De Santis *et al.*, Crypto'01):

No PPT attacker can defeat knowledge-extraction after having seen simulated proofs:

$(\mathbf{crs}, \mathbf{tk}) \leftarrow \mathbf{CRS\text{-}Gen}(\lambda, pp)$

$\mathbf{Sim}(\mathbf{tk}, \cdot)$

$\mathbf{crs}$ →

⋮

← x_i

π_i →

⋮

← $(x^\star, \pi^\star)$

Adversary wins if:

- $\mathbf{Verify}_{\mathbf{srs}}(x^\star, \pi^\star) = 1$ and $(x^\star, \pi^\star) \neq (x_i, \pi_i)$ for all queries x_i to $\mathbf{Sim}(\mathbf{tk}, \cdot)$
- $R(x^\star, w^\star) = 0$ where $w^\star \leftarrow \mathcal{E}(\mathbf{tk}, x^\star, \pi^\star)$

SIM-EXT SNARKs: Prior Work

- Scheme-specific results
  - (Variants of) Groth16 in the AGM (Baghery *et al.*, CANS'20; FC'01)
  - Plonk, Sonic, Marlin in the AGM+ROM using **trapdoor-less** simulators (Ganesh *et al.*, SCN'22)
  - BulletProofs and Spartan in the ROM (Dao-Grubbs, Eurocrypt'23; Ganesh *et al.*, ePrint 2023/147)
- General compilers with black-box straight-line extractors
  - Without witnesss succinctness (Abdolmaleki *et al.*; ACM-CCS'20, CSF'24)
  - UC security with witness succinctness (Ganesh *et al.*, Eurocrypt'23)
- Compilers applying to existing univariate PIOPs (Marlin, Lunar, Plonk, ...)
  - Based on arguments with **trapdoor-less simulators** and **weak unique responses** (Kohlweiss *et al.*, TCC'23)
  - From PCS with **trapdoor-based** simulators and satisfying a **relaxed** notion of **SIM-EXT** (Faonio *et al.*, TCC'23)

Contributions

Building a SIM-EXT SNARK from a multilinear PIOP:

- Use **strongly** SIM-EXT PCS in the AGM+ROM with a **simple** trapdoor-less simulator
- Non-generic, but can be applied to multilinear PIOPs

- Two constructions of KZG-based PCS with straight-line SIM-EXT in the AGM+ROM:
  - Multivariate PST commitments
    (based on `Papamanthou-Shi-Tamassia, TCC'13`; `Zhang` *et al.*, `ePrint 2017/1146`):

    $O(1)$-size commitments to μ-variate polynomials, proofs live in $\mathbb{G}^{\mu+1} \times \mathbb{Z}_p$
  - Univariate (i.e., $\mu = 1$) randomized KZG: proof in $\mathbb{G} \times \mathbb{Z}_p^2$
- **Application** to HyperPlonk (`Chen` *et al.*, `Eurocrypt'23`):
  - Instantiation with straight-line SIM-EXT in the AGM+ROM
    (retains linear-time prover and large-degree custom gates)

Agenda

SNARKs

Simulation-Extractable SNARKs: Motivation and prior work

Simulation-Extractable PCS in the AGM+ROM

- Reminder on KZG and PST Polynomial Commitments
- A Simulation-Extractable Variant of Multivariate KZG/PST Commitments
- Proof Intuition

Application: Simulation-Extractable instantiation of HyperPlonk

KZG Polynomial Commitments

- Use pairings $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ and a CRS of size $O(\lambda \cdot d)$, where $d = \max(\deg(f))$:

$$srs = \left(g,\ \{g^{(\alpha^i)}\}_{i \in [d]},\ (\hat{g}, \hat{g}^{\alpha}) \right)$$

- Commitment to polynomial $f[X]$ consists of $C = g^{f(\alpha)}$

Key idea (Kate-Zaverucha-Goldberg; Asiacrypt'10):

- $y = f(z) \iff \exists q[X]$ s.t.

$$f[X] - y = q[X] \cdot (X - z)$$

- Proof that $y = f(z)$ is $\pi = g^{q(\alpha)} \in \mathbb{G}$ and satisfies

$$e(C \cdot g^{-y}, \hat{g}) = e(\pi, \hat{g}^{\alpha} \cdot \hat{g}^{-z})$$

- **Evaluation-binding** under the d-SDH assumption; **Knowledge-sound** in the AGM
- **Malleable** since homomorphic, but still satisfies a form of (policy-based) SIM-EXT (Faonio *et al.*; TCC'23)

Multivariate KZG/PST Commitments

- μ-variate polynomials of variable-degree d require a CRS of size $O(\lambda \cdot d^\mu)$:

$$srs = \left(\left\{ g^{\alpha_1^{i_1} \cdots \alpha_\mu^{i_\mu}} \right\}_{(i_1,\ldots,i_\mu)\in[0,d]^\mu}, \left(\hat{g}, \{\hat{g}^{\alpha_i}\}_{i=1}^{\mu}\right) \right)$$

- Commitment to polynomial $f[X_1, \ldots, X_\mu]$ consists of $C = g^{f(\alpha_1,\ldots,\alpha_\mu)}$

Key idea (Papamanthou-Shi-Tamassia; TCC'13):

- $y = f(z_1, \ldots, z_\mu) \iff \exists\, q_i[X_1, \ldots, X_\mu]$ for $i \in [\mu]$ s.t.

$$f[X_1, \ldots, X_\mu] - y = \sum_{i=1}^{\mu} q_i[X_1, \ldots, X_\mu] \cdot (X_i - z_i)$$

- Proof that $y = f(z_1, \ldots, z_\mu)$ is $\left\{\pi_i = g^{q_i(\alpha_1,\ldots,\alpha_\mu)}\right\}_{i=1}^{\mu}$ satisfying

$$e(C \cdot g^{-y}, \hat{g}) = \prod_{i=1}^{\mu} e(\pi_i, \hat{g}^{\alpha_i} \cdot \hat{g}^{-z_i})$$

Randomized PST Commitments

Zhang *et al.*'s randomized PST commitments (`ePrint 2017/1146`):

- μ-variate polynomials of variable-degree d require a CRS of size $O(\lambda \cdot d^{\mu})$:

$$srs = \left(\left\{ g^{\alpha_1^{i_1} \cdots \alpha_\mu^{i_\mu}} \right\}_{(i_1,\dots,i_\mu)\in[0,d]^\mu},\ g^{\alpha_r},\ \left(\hat{g},\ \{\hat{g}^{\alpha_i}\}_{i=1}^{\mu},\ \hat{g}^{\alpha_r}\right) \right)$$

- Commitment to $f[X_1, \dots, X_\mu]$ consists of $C = g^{f(\alpha_1,\dots,\alpha_\mu)+\alpha_r \cdot r}$ with $r \xleftarrow{R} \mathbb{Z}_p$

- Evaluation proof is $(\pi_1, \dots, \pi_\mu, \pi_r)$ with $\pi_i = g^{q_i(\alpha_1,\dots,\alpha_\mu)+\alpha_r \cdot s_i}$ for $s_i \xleftarrow{R} \mathbb{Z}_p$

- Verification equation is

$$e(C \cdot g^{-y}, \hat{g}) = \prod_{i=1}^{\mu} e(\pi_i,\ \hat{g}^{\alpha_i} \cdot \hat{g}^{-z_i}) \cdot e(\pi_r, \hat{g}^{\alpha_r})$$

- **Knowledge-sound** in the AGM under the $(d \cdot \mu, d \cdot \mu)$-DLOG assumption, but **malleable**

Simulation-Extractable Variant of rPST

- μ-variate polynomials of variable-degree d require a CRS of size $O(\lambda \cdot d^{\mu})$:

$$srs = \left(\left\{ g^{\alpha_1^{i_1} \cdots \alpha_\mu^{i_\mu}} \right\}_{(i_1,\ldots,i_\mu)\in[0,d]^\mu}, \; g^{\alpha_r}, \; \left(\hat{g}, \{\hat{g}^{\alpha_i}\}_{i=1}^{\mu}, \hat{g}^{\alpha_r}\right) \right)$$

- Commitment to $f[X_1, \ldots, X_\mu]$ consists of $C = g^{f(\alpha_1,\ldots,\alpha_\mu)+\alpha_r \cdot r}$ with $r \xleftarrow{R} \mathbb{Z}_p$

Our non-malleable evaluation proofs

- **P** proves $y = f(\mathbf{z})$ by revealing $(\pi_1, \ldots, \pi_\mu) \in \mathbb{G}^{\mu}$ and proving knowledge of π_r s.t.

$$e(C \cdot g^{-y}, \hat{g}) \Big/ \prod_{i=1}^{\mu} e(\pi_i, \hat{g}^{\alpha_i} \cdot \hat{g}^{-z_i}) = e(\boxed{\pi_r}, \hat{g}^{\alpha_r})$$

- Σ-protocol proof is $\big((\pi_1, \ldots, \pi_\mu), (c, S_\pi)\big)$ with $c = H\big(\mathbf{z}, y, C, (\pi_i)_{i=1}^{\mu}, R_\pi, \text{label}\big)$

Simulation-Extractable Variant of rPST

- Given $\big((\pi_1, \ldots, \pi_\mu), (c, S_\pi)\big)$, verifier **V** accepts if $c = H\big(\mathbf{z}, y, C, (\pi_i)_{i=1}^{\mu}, R_\pi, \text{label}\big)$ where

$$R_\pi = e(S_\pi, \hat{g}^{\alpha_r}) \cdot \left(\frac{e(C \cdot g^{-y}, \hat{g})}{\prod_{i=1}^{\mu} e(\pi_i, \hat{g}^{\alpha_i} \cdot \hat{g}^{-z_i})} \right)^{-c} \quad (1)$$

Theorem

The scheme is **SIM-EXT** in the AGM+ROM under the $(d \cdot \mu, d \cdot \mu)$-**DLOG** assumption:
i.e., computing $\alpha \in \mathbb{Z}_p$ is hard given $\left(g, \ \{g^{(\alpha^i)}\}_{i \in [d \cdot \mu]}, \ \{\hat{g}^{(\alpha^i)}\}_{i \in [d \cdot \mu]}\right)$

Proof idea:

- In $\mathcal{A}$'s forgery, element R_π of (1) must have been queried to H and $\mathcal{A}$ must have supplied an AGM representation defining $R[X_1, \ldots, X_\mu, X_r]$ s.t. $R_\pi = e(g, \hat{g})^{R(\alpha_1, \ldots, \alpha_\mu, \alpha_r)}$
- AGM representations of $(C, S_\pi, \{\pi_i\}_{i=1}^{\mu})$ define $T[X_1, \ldots, X_\mu, X_r]$ s.t. $T(\alpha_1, \ldots, \alpha_\mu, \alpha_r) = 0$
- Statistical argument shows that $T[X_1, \ldots, X_\mu, X_r] \not\equiv 0$ w.h.p. unless AGM representation of C provide a witness $f[X_1, \ldots, X_\mu]$ s.t. $y = f(\mathbf{z})$

Simulation-Extractable Variant of rPST

- Given $\big((\pi_1, \ldots, \pi_\mu), (c, S_\pi)\big)$, verifier **V** accepts if $c = H\big(\mathbf{z}, y, C, (\pi_i)_{i=1}^{\mu}, R_\pi, \mathsf{label}\big)$ where

$$R_\pi = e(S_\pi, \hat{g}^{\alpha_r}) \cdot \left(\frac{e(C \cdot g^{-y}, \hat{g})}{\prod_{i=1}^{\mu} e(\pi_i, \hat{g}^{\alpha_i} \cdot \hat{g}^{-z_i})} \right)^{-c} \quad (1)$$

Theorem

The scheme is **SIM-EXT** in the AGM+ROM under the $(d \cdot \mu, d \cdot \mu)$-**DLOG** assumption:

i.e., computing $\alpha \in \mathbb{Z}_p$ is hard given $\big(g, \{g^{(\alpha^i)}\}_{i \in [d \cdot \mu]}, \{\hat{g}^{(\alpha^i)}\}_{i \in [d \cdot \mu]}\big)$

Proof idea:

- In $\mathcal{A}$'s forgery, element R_π of (1) must have been queried to H and $\mathcal{A}$ must have supplied an AGM representation defining $R[X_1, \ldots, X_\mu, X_r]$ s.t. $R_\pi = e(g, \hat{g})^{R(\alpha_1, \ldots, \alpha_\mu, \alpha_r)}$
- AGM representations of $(C, S_\pi, \{\pi_i\}_{i=1}^{\mu})$ define $T[X_1, \ldots, X_\mu, X_r]$ s.t. $T(\alpha_1, \ldots, \alpha_\mu, \alpha_r) = 0$
- Statistical argument shows that $T[X_1, \ldots, X_\mu, X_r] \not\equiv 0$ w.h.p. unless AGM representation of C provide a witness $f[X_1, \ldots, X_\mu]$ s.t. $y = f(\mathbf{z})$

Simulation-Extractable Variant of rPST

- Given $\left((\pi_1, \ldots, \pi_\mu), (c, S_\pi)\right)$, verifier **V** accepts if $c = H\left(\mathbf{z}, y, C, (\pi_i)_{i=1}^{\mu}, R_\pi, \mathsf{label}\right)$ where

$$R_\pi = e(S_\pi, \hat{g}^{\alpha_r}) \cdot \left(\frac{e(C \cdot g^{-y}, \hat{g})}{\prod_{i=1}^{\mu} e(\pi_i, \hat{g}^{\alpha_i} \cdot \hat{g}^{-z_i})} \right)^{-c} \quad (1)$$

Theorem

The scheme is **SIM-EXT** in the AGM+ROM under the $(d \cdot \mu, d \cdot \mu)$-**DLOG** assumption:
i.e., computing $\alpha \in \mathbb{Z}_p$ is hard given $\left(g, \{g^{(\alpha^i)}\}_{i \in [d \cdot \mu]}, \{\hat{g}^{(\alpha^i)}\}_{i \in [d \cdot \mu]}\right)$

Proof idea:

- In $\mathcal{A}$'s forgery, element R_π of (1) must have been queried to H and $\mathcal{A}$ must have supplied an AGM representation defining $R[X_1, \ldots, X_\mu, X_r]$ s.t. $R_\pi = e(g, \hat{g})^{R(\alpha_1, \ldots, \alpha_\mu, \alpha_r)}$
- AGM representations of $(C, S_\pi, \{\pi_i\}_{i=1}^{\mu})$ define $T[X_1, \ldots, X_\mu, X_r]$ s.t. $T(\alpha_1, \ldots, \alpha_\mu, \alpha_r) = 0$
- Statistical argument shows that $T[X_1, \ldots, X_\mu, X_r] \not\equiv 0$ w.h.p. unless AGM representation of C provide a witness $f[X_1, \ldots, X_\mu]$ s.t. $y = f(\mathbf{z})$

Simulation-Extractable Variant of rPST

- Given $\left((\pi_1, \ldots, \pi_\mu), (c, S_\pi)\right)$, verifier **V** accepts if $c = H\left(\mathbf{z}, y, C, (\pi_i)_{i=1}^{\mu}, R_\pi, \mathsf{label}\right)$ where

$$R_\pi = e(\, S_\pi \,, \hat{g}^{\alpha_r}) \cdot \left(\frac{e(C \cdot g^{-y}, \hat{g})}{\prod_{i=1}^{\mu} e(\pi_i \,,\ \hat{g}^{\alpha_i} \cdot \hat{g}^{-z_i})} \right)^{-c} \tag{1}$$

Theorem

The scheme is **SIM-EXT** in the AGM+ROM under the $(d \cdot \mu, d \cdot \mu)$-**DLOG** assumption:

i.e., computing $\alpha \in \mathbb{Z}_p$ is hard given $\left(g,\ \{g^{(\alpha^i)}\}_{i \in [d \cdot \mu]},\ \{\hat{g}^{(\alpha^i)}\}_{i \in [d \cdot \mu]}\right)$

Proof idea:

- In $\mathcal{A}$'s forgery, element R_π of (1) must have been queried to H and $\mathcal{A}$ must have supplied an AGM representation defining $R[X_1, \ldots, X_\mu, X_r]$ s.t. $R_\pi = e(g, \hat{g})^{R(\alpha_1, \ldots, \alpha_\mu, \alpha_r)}$
- AGM representations of $(C, S_\pi, \{\pi_i\}_{i=1}^{\mu})$ define $T[X_1, \ldots, X_\mu, X_r]$ s.t. $T(\alpha_1, \ldots, \alpha_\mu, \alpha_r) = 0$
- Statistical argument shows that $T[X_1, \ldots, X_\mu, X_r] \not\equiv 0$ w.h.p. unless AGM representation of C provide a witness $f[X_1, \ldots, X_\mu]$ s.t. $y = f(\mathbf{z})$

Agenda

SNARKs

Simulation-Extractable SNARKs: Motivation and prior work

Simulation-Extractable PCS in the AGM+ROM

- Reminder on KZG and PST Polynomial Commitments
- A Simulation-Extractable Variant of Multivariate KZG/PST Commitments
- Proof Intuition

Application: Simulation-Extractable instantiation of HyperPlonk

Application to HyperPlonk

HyperPlonk at a high level:

- Prover encodes computation trace in matrix $\mathbf{M} = \{(L_i, R_i, O_i)\}_{i=1}^{N}$ where $N = 2^\mu$
- Commits to multilinear $\{M[X_1, ..., X_\mu, \text{bin}(i)]\}_{i=0}^{2}$ evaluating to $\mathbf{M}$'s columns over $\{0,1\}^\mu$
- Prove that $\{M[X_1, ..., X_\mu, \text{bin}(i)]\}_{i=0}^{2}$ satisfies a **gate identity** by showing that

$$\forall \mathbf{x} \in \{0,1\}^\mu : f(\mathbf{x}) = 0$$

  for some $f[X_1, ..., X_\mu]$ depending on $\{M[X_1, ..., X_\mu, \text{bin}(i)]\}_{i=0}^{2}$, input-encoding polynomial $I[X_1, ..., X_\mu]$, and selector polynomials $\{S_j[X_1, ..., X_\mu]\}_{j=1,2,3}$

- Prove that $\{M[\mathbf{X}, \text{bin}(i)]\}_{i=0,1,2}$ satisfies a **wiring identity**

$$M[\mathbf{x}, \text{bin}(i)] = M[\sigma(\mathbf{x}, \text{bin}(i))] \qquad \forall \mathbf{x} \in \{0,1\}^\mu,\ i \in \{0,1,2\}$$

  for a public permutation σ

Application to HyperPlonk

HyperPlonk at a high level:

- Prover encodes computation trace in matrix $\mathbf{M} = \{(L_i, R_i, O_i)\}_{i=1}^{N}$ where $N = 2^\mu$
- Commits to multilinear $\{M[X_1, \ldots, X_\mu, \mathrm{bin}(i)]\}_{i=0}^{2}$ evaluating to $\mathbf{M}$'s columns over $\{0,1\}^\mu$
- Prove that $\{M[X_1, \ldots, X_\mu, \mathrm{bin}(i)]\}_{i=0}^{2}$ satisfies a **gate identity** by showing that

$$\forall \mathbf{x} \in \{0,1\}^\mu : f(\mathbf{x}) = 0$$

  for some $f[X_1, \ldots, X_\mu]$ depending on $\{M[X_1, \ldots, X_\mu, \mathrm{bin}(i)]\}_{i=0}^{2}$, input-encoding polynomial $I[X_1, \ldots, X_\mu]$, and selector polynomials $\{S_j[X_1, \ldots, X_\mu]\}_{j=1,2,3}$

- Prove that $\{M[\mathbf{X}, \mathrm{bin}(i)]\}_{i=0,1,2}$ satisfies a **wiring identity**

$$M[\mathbf{x}, \mathrm{bin}(i)] = M[\sigma(\mathbf{x}, \mathrm{bin}(i))] \qquad \forall \mathbf{x} \in \{0,1\}^\mu,\ i \in \{0,1,2\}$$

  for a public permutation σ

Application to HyperPlonk

HyperPlonk at a high level:

- Prover encodes computation trace in matrix $\mathbf{M} = \{(L_i, R_i, O_i)\}_{i=1}^{N}$ where $N = 2^\mu$
- Commits to multilinear $\{M[X_1, \ldots, X_\mu, \text{bin}(i)]\}_{i=0}^{2}$ evaluating to $\mathbf{M}$'s columns over $\{0,1\}^\mu$
- Prove that $\{M[X_1, \ldots, X_\mu, \text{bin}(i)]\}_{i=0}^{2}$ satisfies a **gate identity** by showing that

  $$\forall \mathbf{x} \in \{0,1\}^\mu : f(\mathbf{x}) = 0$$

  for some $f[X_1, \ldots, X_\mu]$ depending on $\{M[X_1, \ldots, X_\mu, \text{bin}(i)]\}_{i=0}^{2}$, input-encoding polynomial $I[X_1, \ldots, X_\mu]$, and selector polynomials $\{S_j[X_1, \ldots, X_\mu]\}_{j=1,2,3}$
- Prove that $\{M[\mathbf{X}, \text{bin}(i)]\}_{i=0,1,2}$ satisfies a **wiring identity**

  $$M[\mathbf{x}, \text{bin}(i)] = M[\sigma(\mathbf{x}, \text{bin}(i))] \qquad \forall \mathbf{x} \in \{0,1\}^\mu,\ i \in \{0,1,2\}$$

  for a public permutation σ

SIM-EXT Instantiation of HyperPlonk

- **Our trapdoor-less simulator:**
  - Computes *fake* witnesses $\{\hat{M}[X_1, \ldots, X_\mu, \mathrm{bin}(i)]\}_{i=0}^{2}$ satisfying the **gate identity**

$$\forall \mathbf{x} \in \{0,1\}^\mu : f(\mathbf{x}) = 0$$

  ...but not the **wiring identity**

$$\hat{M}[\mathbf{x}, \mathrm{bin}(i)] = \hat{M}[\sigma(\mathbf{x}, \mathrm{bin}(i))] \qquad \forall \mathbf{x} \in \{0,1\}^\mu,\ i \in \{0,1,2\} \tag{2}$$

  (easy by computing $\hat{M}[X_1, \ldots, X_\mu, X_{\mu+1}, X_{\mu+2}]$ as a multilinear extension)

  - Simulates proof for (2) via a simulated PCS proof that some polynomial $\tilde{v}[X_1, \ldots, X_{\mu+1}]$ satisfies $\tilde{v}(1, 1, \ldots, 1, 0) = 1$

- Earlier prover messages are embedded in label of each PCS evaluation proof (for non-malleability)

SIM-EXT Instantiation of HyperPlonk

- **Our trapdoor-less simulator:**
  - Computes *fake* witnesses $\{\hat{M}[X_1, \ldots, X_\mu, \mathrm{bin}(i)]\}_{i=0}^{2}$ satisfying the **gate identity**

    $$\forall \mathbf{x} \in \{0,1\}^\mu : f(\mathbf{x}) = 0$$

    ...but not the **wiring identity**

    $$\hat{M}[\mathbf{x}, \mathrm{bin}(i)] = \hat{M}[\sigma(\mathbf{x}, \mathrm{bin}(i))] \qquad \forall \mathbf{x} \in \{0,1\}^\mu,\ i \in \{0,1,2\} \tag{2}$$

    (easy by computing $\hat{M}[X_1, \ldots, X_\mu, X_{\mu+1}, X_{\mu+2}]$ as a multilinear extension)

  - Simulates proof for (2) via a simulated PCS proof that some polynomial $\tilde{v}[X_1, \ldots, X_{\mu+1}]$ satisfies $\tilde{v}(1, 1, \ldots, 1, 0) = 1$

- Earlier prover messages are embedded in label of each PCS evaluation proof (for non-malleability)

Summary

- Constructions of SIM-EXT PCS (with straight-line extractability) in the AGM+ROM; almost as efficient as the underlying malleable schemes
  - $\mu + 2$ pairings to verify in μ-variate PCS
  - 2 pairings for a variant of rKZG
  - Randomness of only one field element in both cases (no need for a large masking polynomial)
  - Simple trapdoor-less simulator via Fiat-Shamir and Σ-protocols
- Provide a SIM-EXT variant of HyperPlonk in the AGM+ROM
- Possible optimization using Zeromorph (`Kohrita-Towa; ePrint 2023/917`) to get $O(1)$ pairings **V** at the cost of a 2.5x overhead at **P**

Questions?