Theory and Practice of Succinct Zero Knowledge Proofs

Lecture 05:
Holographic PIOP for R1CS

Pratyush Mishra
UPenn
Fall 2023

A PIOP for R1CS

R1CS

An rank-1 constraint system (R1CS) is a generalization of arithmetic circuits

$$(F := (\mathbb{F}, n \in \mathbb{N}, A, B, C), x, w)$$

$$z := \begin{bmatrix} x \\ w \end{bmatrix} \qquad n\left\{\overbrace{\begin{bmatrix} \quad A \quad \end{bmatrix}}^{n}\right. \begin{bmatrix} z \end{bmatrix} \circ \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} z \end{bmatrix} = \begin{bmatrix} C \end{bmatrix} \begin{bmatrix} z \end{bmatrix}$$

What checks do we need?

Step 1: Correct Hadamard product

check that for each i, $z_A[i] \cdot z_B[i] = z_C[i]$

Step 2: Correct matrix-vector multiplication

check that $Mz = z_M \quad \forall M \in \{A, B, C\}$

PIOP for Hadamard Product

Prover(F, x, w)

1. Let $H \subseteq \mathbb{F}$ be a set of size n.
2. Interpolate z_A, z_B, z_C over H to get p_A, p_B, p_C.
3. Compute quotient $q = \dfrac{p_A \cdot p_B - p_C}{v_H}$.

p_A p_B p_C q

Verifier(F, x)

r ← QUERY

$$p_A(r) \cdot p_B(r) - p_C(r) \stackrel{?}{=} q(r) \cdot v_H(r)$$

PIOP for Matrix-vector products

Prover(M, z)

1. Compute $z_M := Mz$
2. Interpolate z_M over H to get $\hat{z}_M$
3. Interpolate $(\vec{r}, \vec{r}^\top M)$ to get $(\hat{r}, \hat{r}_M)$
4. Use sumcheck lemma to compute g, q such that

$$\hat{r}(X) \cdot \hat{z}_M(X) - \hat{r}_M(X) \cdot \hat{z}(X) = X \cdot g(X) + q(X) v_H(X)$$

Prover → Verifier: z, $\hat{z}_M$

Verifier → Prover: r

Prover → Verifier: g, q

Verifier(M)

1. $r \xleftarrow{\$} \mathbb{F}$
2. $\vec{r} := (1, r, \ldots, r^{n-1})$
3. Interpolate $(\vec{r}, \vec{r}^\top M)$ to get $(\hat{r}, \hat{r}_M)$
4. Invoke PIOP for ZC!

New tool: univariate sum check

Lemma:

$$\sum_{h \in H} p(h) = \sigma$$

$$\Updownarrow$$

$\exists\ q, g$ s.t.

$$p(X) = X g(X) + \frac{\sigma}{|H|} + q(X) \cdot v_H(X)$$

Reduce sumcheck to Zero check!

Why is the verifier slow?

$\mathbf{P}\text{rover}(M, z)$

1. Compute $z_M := Mz$
2. Interpolate z_M over H to get $\hat{z}_M$
3. Interpolate $(\vec{r}, \vec{r}^\top M)$ to get $(\hat{r}, \hat{r}_M)$
4. Use sumcheck lemma to compute g, q such that

$$\hat{r}(X) \cdot \hat{z}_M(X) - \hat{r}_M(X) \cdot \hat{z}(X) = X \cdot g(X) + q(X) v_H(X)$$

z $\hat{z}_M$

r

$\mathbf{V}\text{erifier}(M)$

1. $r \xleftarrow{\$} \mathbb{F}$
2. $\vec{r} := (1, r, \ldots, r^{n-1})$
3. Interpolate $(\vec{r}, \vec{r}^\top M)$ to get $(\hat{r}, \hat{r}_M)$

oke PIOP for ZC!

Must compute $\vec{r}^\top \cdot M$!

To check this, it must evaluate $\hat{r}(X)$ and $\hat{r}_M(X)$

Key tool: Lagrange polynonmials

Lagrange polynomials for set $H \subseteq \mathbb{F}$:

For each $i \in H$, $L_H^i(X)$ is 1 at i, and 0 for all $j \in H, j \neq i$

Polynomial Interpolation:

Given a list $A = (a_0, \ldots, a_d)$, and a set $H \subseteq \mathbb{F}$, the interpolation of A over H is

$$\hat{a}(X) := \sum_{i \in H} a_i \cdot L_H^i(X)$$

Relation to vanishing polynomial: $L_H^i(X) := c_i \cdot \frac{v_H(X)}{X - i}$

Step 1: Efficient $\hat{r}(X)$

Can write $\hat{r}(X)$ as $\sum_{i \in H} r^i \cdot L_H^i(X)$

Efficiently evaluating this at a random point β requires efficiently computing each r^i and $L_H^i(\beta)$

Let's interpret this as $\sum_{i \in H} Y^i \cdot L_H^i(X)$

Monomial basis

Lagrange basis

Step 1: Efficient $\hat{r}(X)$

1. Replace Monomial with Lagrange basis $\sum_{i \in H} L_H^i(Y) \cdot L_H^i(X)$
2. Can rewrite this as $\dfrac{v_H(Y)X - v_H(X)Y}{|H|(X - Y)}$

This can be evaluated in time $O(\log |H|)$!

Why is the verifier slow?

Prover(M, z)

1. Compute $z_M := Mz$
2. Interpolate z_M over H to get $\hat{z}_M$

z $\hat{z}_M$ →

← α

3. Use sumcheck lemma to compute g, q such that

$$\hat{r}(\alpha, X) \cdot \hat{z}_M(X) - \hat{r}_M(\alpha, X) \cdot \hat{z}(X)$$
$$=$$
$$X \cdot g(X) + q(X) v_H(X)$$

g q →

Verifier(M)

1. $\alpha \xleftarrow{\$} \mathbb{F}$
2. ???

Must compute $\mathbf{L}_H^i(\alpha)^\top \cdot M$!

3. Invoke PIOP for ZC!

How to use this?

Recall: We have to show

$$\hat{r}(\alpha,\beta)\cdot\hat{z}_M(\beta)-\hat{r}_M(\alpha,\beta)\cdot\hat{z}(\beta)$$
$$=$$
$$\beta\cdot g(\beta)+q(\beta)v_H(\beta)$$

Let's expand $\hat{r}_M(\alpha,\beta)=\sum_{i\in H}\hat{r}(\alpha,i)\cdot\hat{M}(i,\beta)$

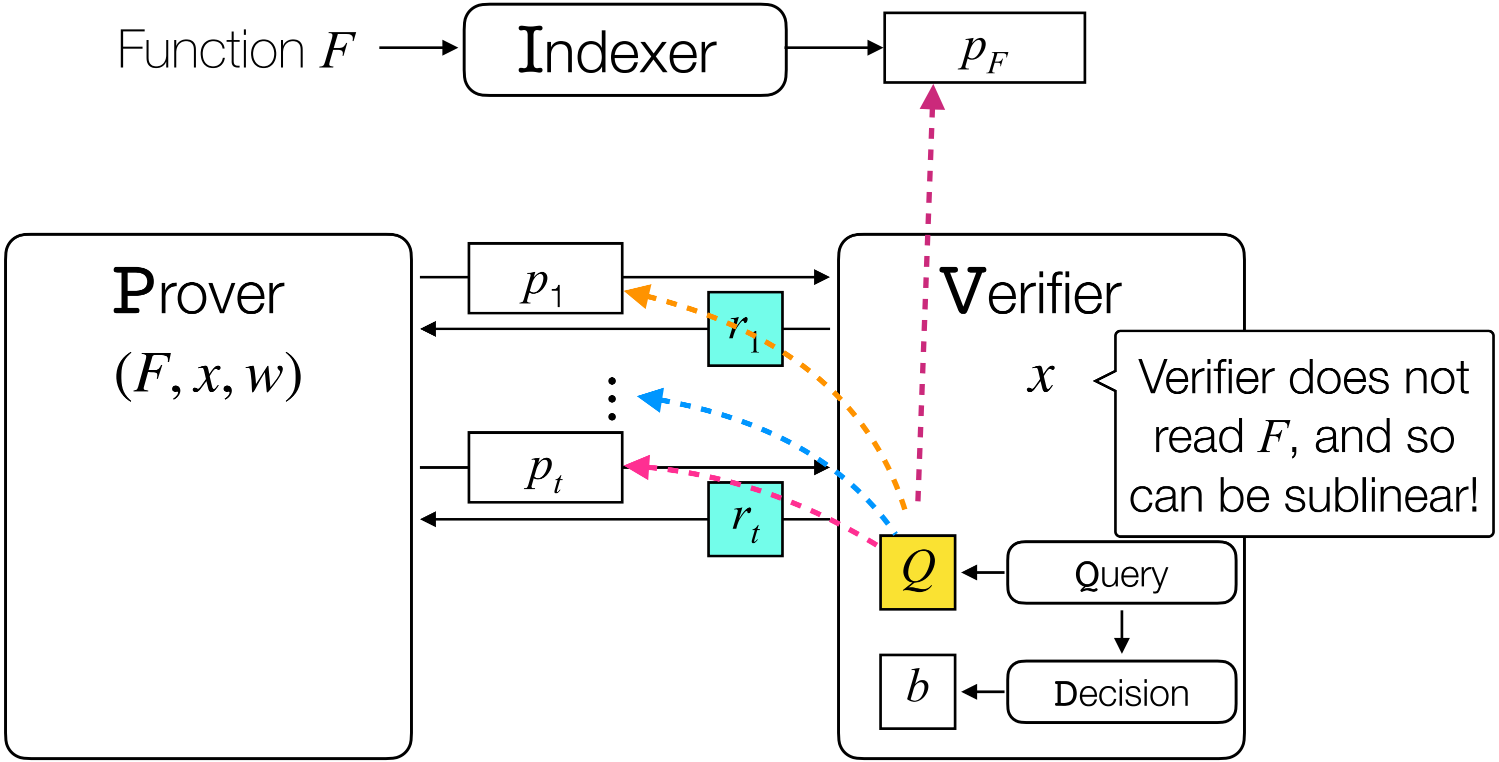
This is yet another sumcheck, so we engage in another sumcheck PIOP, which results in the following check: $\hat{r}(\alpha,\gamma)\cdot\hat{M}(\gamma,\beta)=\gamma\cdot g'(\gamma)+h'(\gamma)v_H(\gamma)$

How to evaluate $\hat{M}(\gamma,\beta)$?

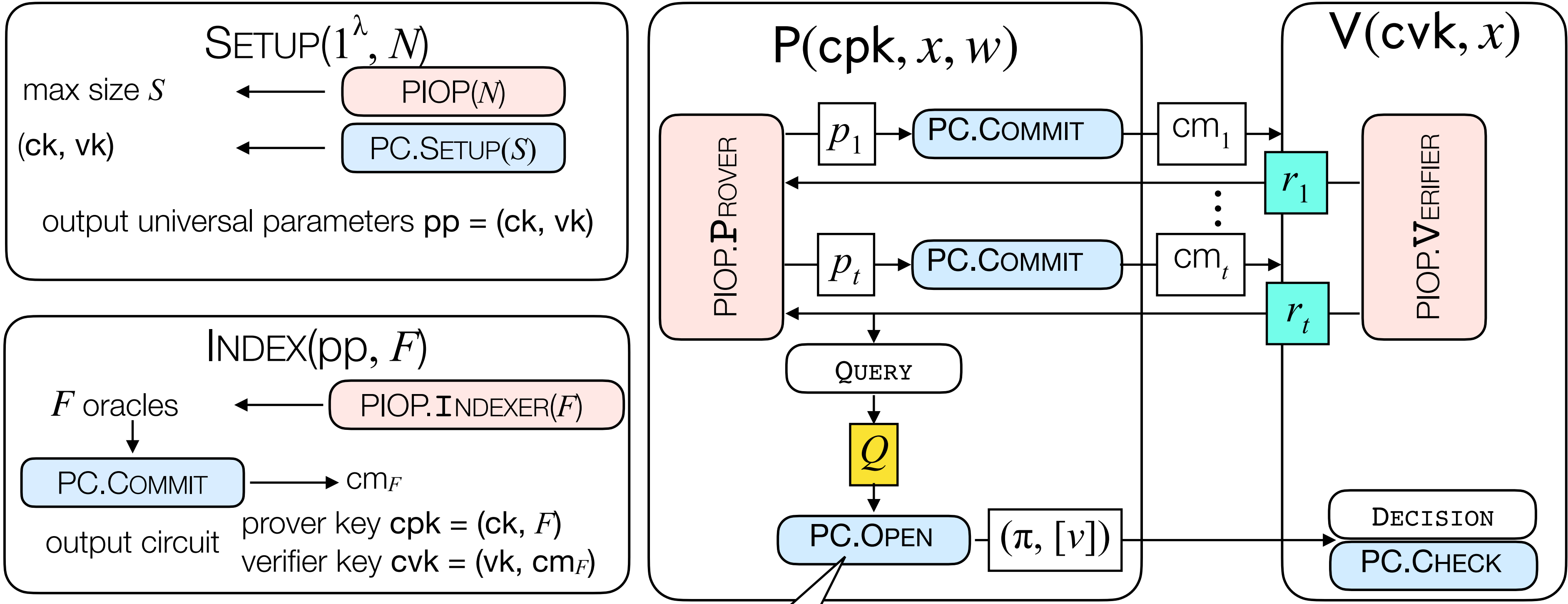
Sublinear verification for PIOP-based SNARKs

Holographic PIOPs [CHMMVW20, COS20]

Introduce a new algorithm to preprocess the matrices



Holographic PIOPs + PC Schemes → Preprocessing SNARKs



Prover answers queries to F oracles too

+ Fiat—Shamir to get non-interactivity

Verifier Complexity of Holographic PIOP-based SNARKs

$$T(\text{SNARK.V}) = T(\textsc{Check}) + T(\text{HIOP.}\mathbf{V})$$

Now sublinear!

Holography enables sublinear verification for arbitrary circuits computations!

How to encode matrix?

Polynomial Interpolation of Lists:

Given a list $A = (a_0, \ldots, a_d)$, and a set $H \subseteq \mathbb{F}$, the interpolation of A over H is

$$\hat{a}(X) := \sum_{i \in H} a_i \cdot L_H^i(X)$$

Polynomial Interpolation of Matrices?:

Given a list $M \in \mathbb{F}^{n \times n}$, and a set $H \subseteq \mathbb{F}$, the bivariate interpolation of A over H is

$$\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot L_H^i(X) \cdot L_H^j(Y)$$

Problem: computing this requires $O(|H|^2)$ work

Insight: The matrices are sparse!

Polynomial Interpolation of Matrices?:

Given a list $M \in \mathbb{F}^{n \times n}$, and a set $H \subseteq \mathbb{F}$, the bivariate interpolation of A over H is

$$\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot L_H^i(X) \cdot L_H^j(Y)$$

Most M_{ij} are zero!

Can rewrite as $\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot \frac{v_H(X)}{X - i} \cdot \frac{v_H(Y)}{Y - j}$,

Additionally, sum only over non-zero entries!

Final Matrix Encoding

Let m be the number of non-zero entries, and $K \subset \mathbb{F}$ be a subset of size m. Then, a *sparse* bivariate interpolation of A over K is

$$\hat{M}(X, Y) := \sum_{k \in K} \mathsf{v}(k) \cdot \frac{v_H(X)}{X - \mathsf{r}(k)} \cdot \frac{v_H(Y)}{Y - \mathsf{c}(k)}$$

Actually, we need polynomials, so we will replace $\mathsf{r}, \mathsf{c}, \mathsf{v}$ with their interpolations over K, i.e. $\hat{\mathsf{r}}, \hat{\mathsf{c}}, \hat{\mathsf{v}}$

$$\hat{M}(X, Y) := \sum_{k \in K} \hat{\mathsf{v}}(k) \cdot \frac{v_H(X)}{X - \hat{\mathsf{r}}(k)} \cdot \frac{v_H(Y)}{Y - \hat{\mathsf{c}}(k)}$$

Value of k-th non-zero entry

Row index of k-th non-zero entry

Col index of k-th non-zero entry