

Polynomial IOPs for Linear Algebra

Alan Szepieniec

alan@nervos.org

Yuncong Zhang

shjdzhangyuncong@sjtu.edu.cn

Nervos
Shanghai Jiao Tong University

PKC 2022

PART I

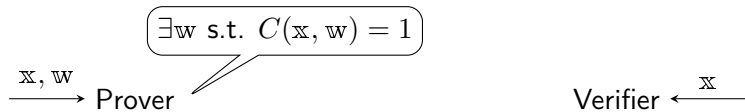
Background

SNARK

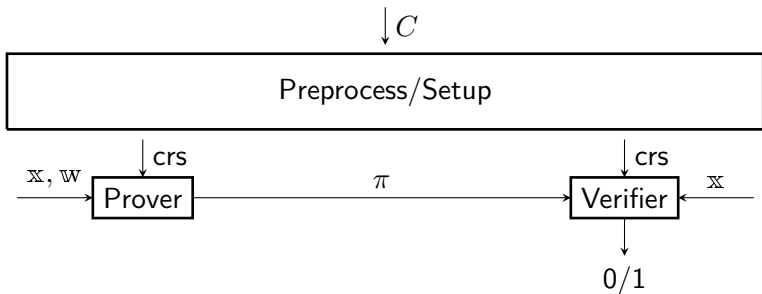
x, w
→ Prover

Verifier ← x

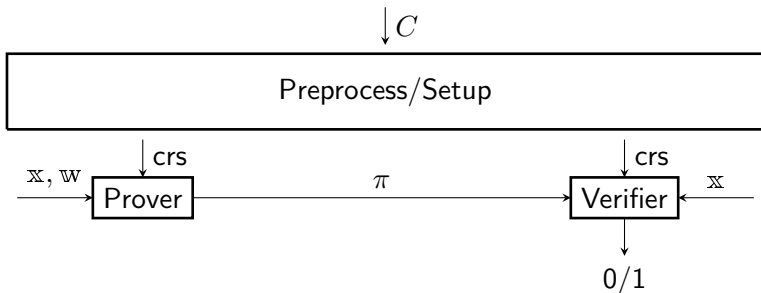
SNARK



SNARK

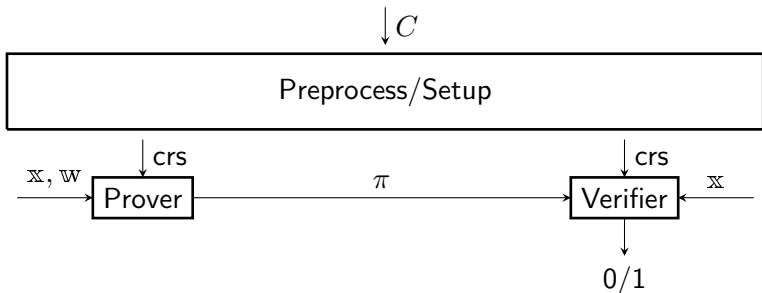


SNARK



S N A R K

SNARK



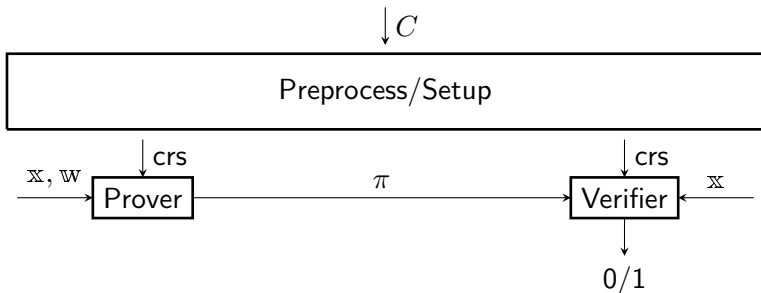
S N A R K

Succinct

$$|\pi| = O(\text{polylog}(|C| + |x|))$$

$$|\text{Verifier}| = O(\text{polylog}(|C|) + |x|)$$

SNARK



S N A R K

Succinct

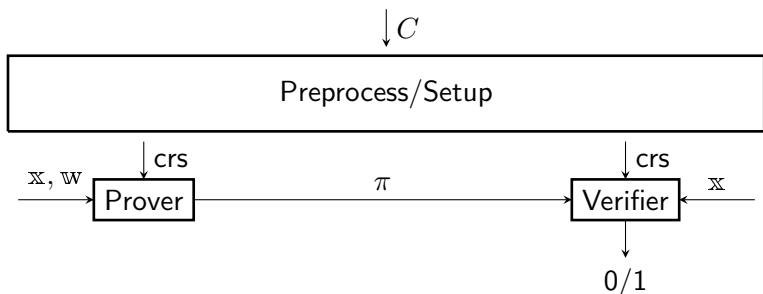
Non-interactive

$$|\pi| = O(\text{polylog}(|C| + |x|))$$

$$\#\text{rounds}=1$$

$$|\text{Verifier}| = O(\text{polylog}(|C|) + |x|)$$

SNARK



SNARK

Succinct

$$|\pi| = O(\text{polylog}(|C| + |x|))$$
$$|\text{Verifier}| = O(\text{polylog}(|C|) + |x|)$$

Non-interactive

$$\#\text{rounds}=1$$

ARgument-of-Knowledge

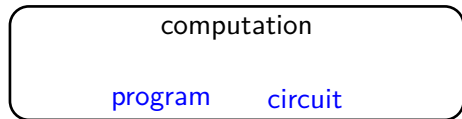
I "know" w , s.t.

$\exists w, \text{s.t.}$

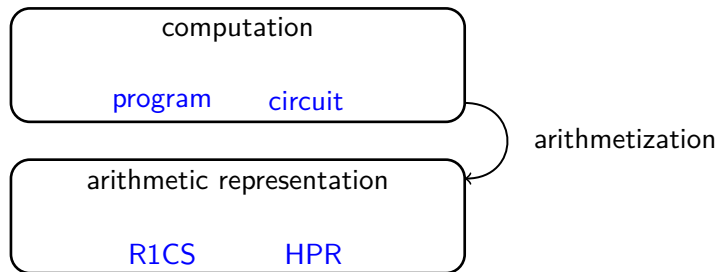
$$C(x, w) = 0$$

Construction Pipeline

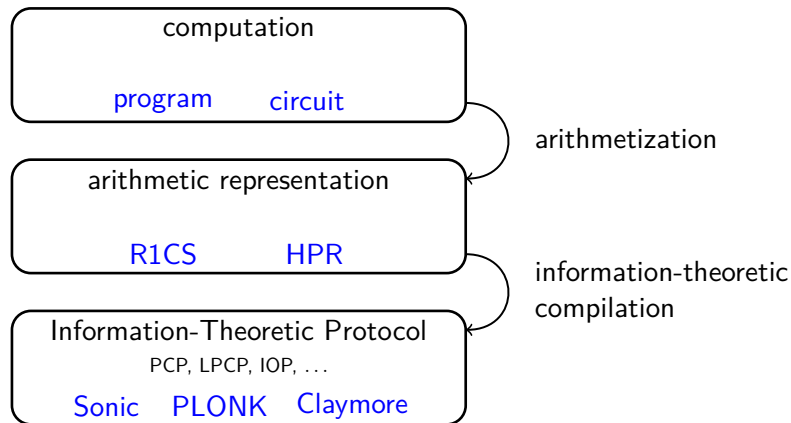
Construction Pipeline



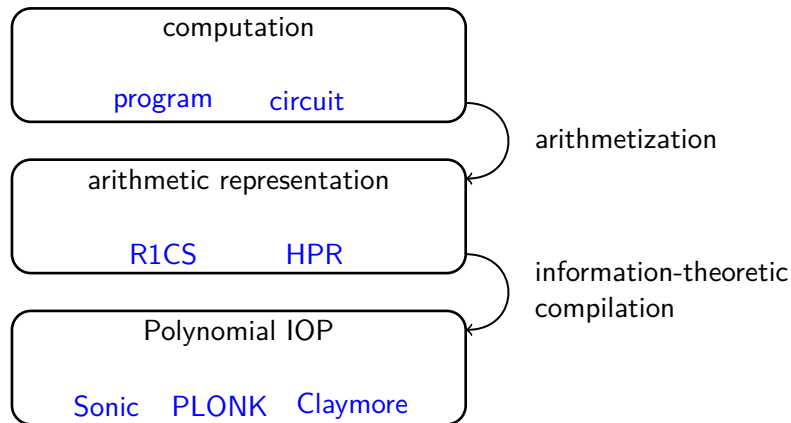
Construction Pipeline



Construction Pipeline

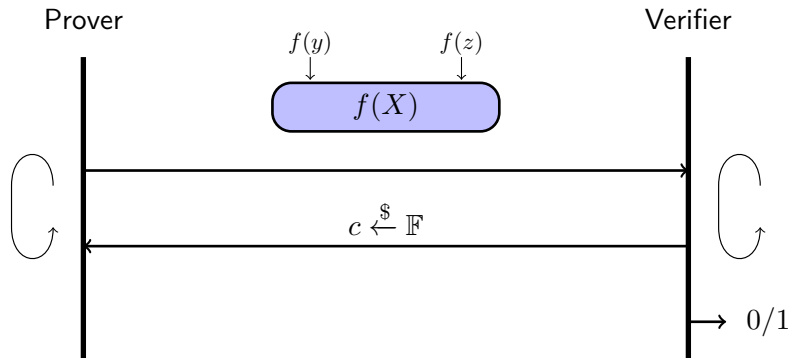


Construction Pipeline

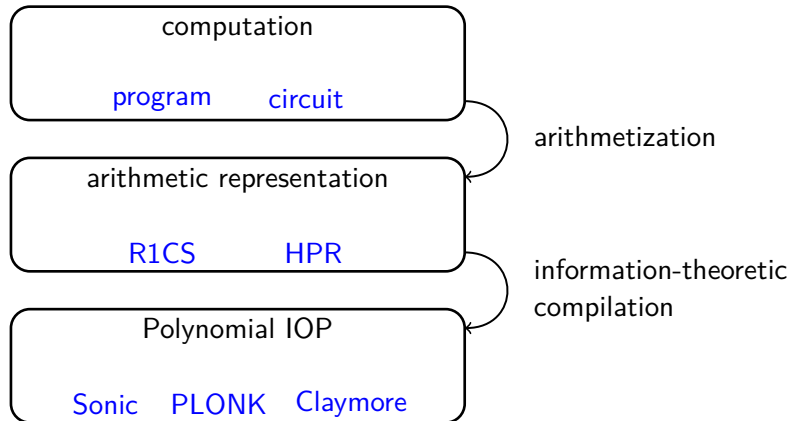


[1] B. Bünz, B. Fisch, and A. Szepieniec, “Transparent SNARKs from DARK Compilers,” in *Advances in Cryptology – EUROCRYPT 2020*, Cham, 2020, pp. 677–706. doi: 10.1007/978-3-030-45721-1_24.

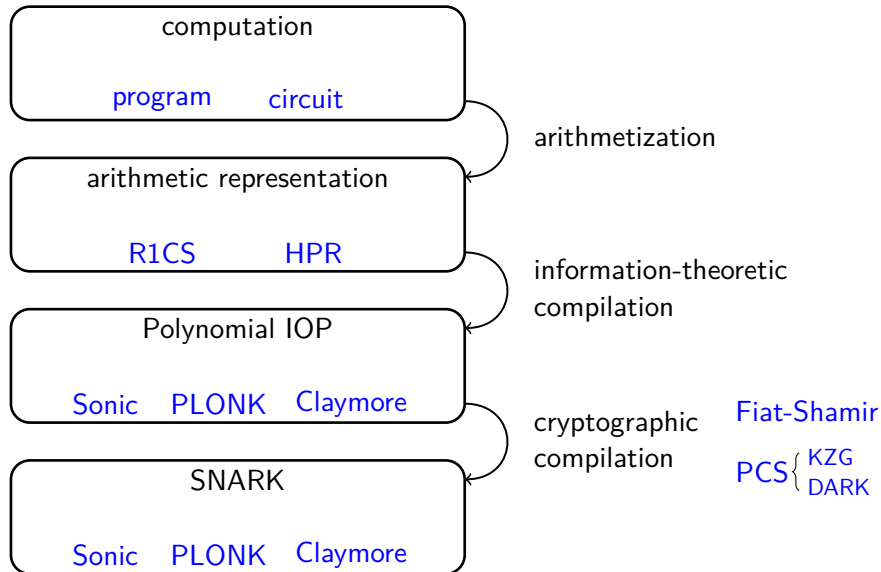
Polynomial IOP



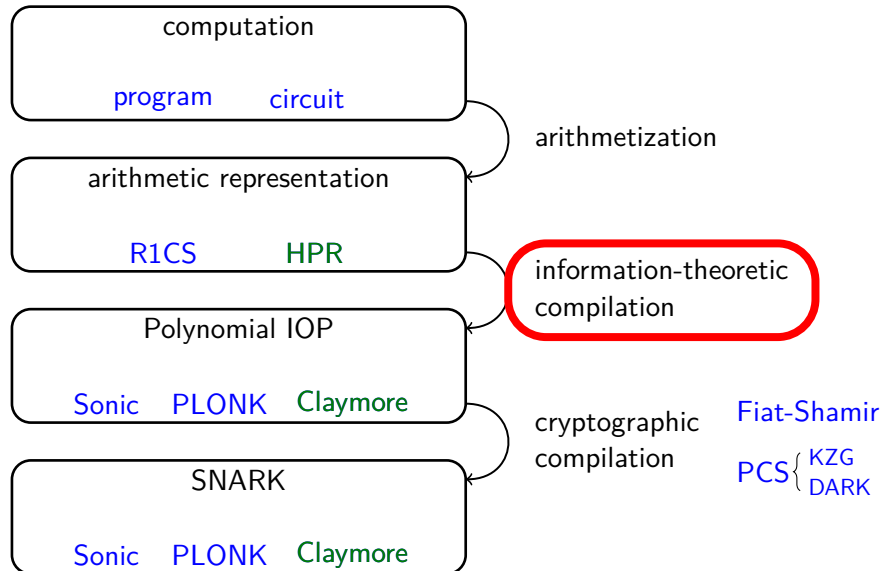
Construction Pipeline



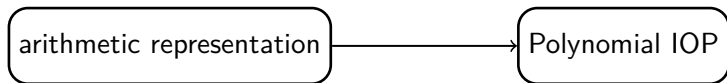
Construction Pipeline



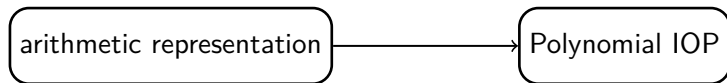
Construction Pipeline



Information-Theoretic Compilation



Information-Theoretic Compilation

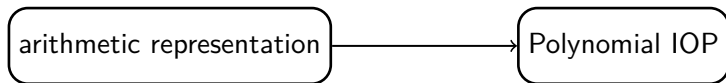


$$\mathbf{y} = M\mathbf{x}$$

$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

Information-Theoretic Compilation



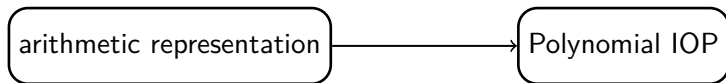
$$\mathbf{y} = M\mathbf{x}$$
$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

$$f(X^{-1}) = g(\alpha) \cdot h(X)$$
$$\deg(\{f, g, h\}) \leq d$$

polynomial algebra

Information-Theoretic Compilation



$$\mathbf{y} = M\mathbf{x}$$

$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

???

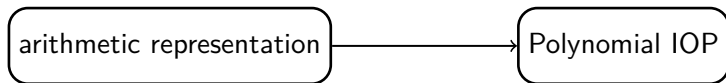


$$f(X^{-1}) = g(\alpha) \cdot h(X)$$

$$\deg(\{f, g, h\}) \leq d$$

polynomial algebra

Information-Theoretic Compilation



$$\mathbf{y} = M\mathbf{x}$$

$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

???



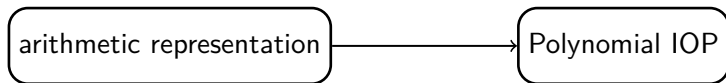
$$f(X^{-1}) = g(\alpha) \cdot h(X)$$

$$\deg(\{f, g, h\}) \leq d$$

polynomial algebra

representation

Information-Theoretic Compilation



$$\mathbf{y} = M\mathbf{x}$$

$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

???



$$f(X^{-1}) = g(\alpha) \cdot h(X)$$

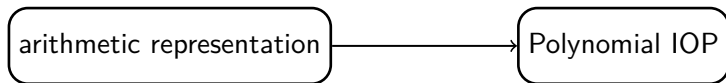
$$\deg(\{f, g, h\}) \leq d$$

polynomial algebra

representation

- Reed-Solomon Codes - $f(D)$

Information-Theoretic Compilation



$$\mathbf{y} = M\mathbf{x}$$

$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

???



$$f(X^{-1}) = g(\alpha) \cdot h(X)$$

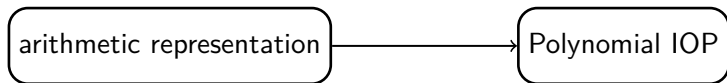
$$\deg(\{f, g, h\}) \leq d$$

polynomial algebra

representation

- Reed-Solomon Codes - $f(D)$
- Monomial Coefficients - $\sum c_i X^i$

Information-Theoretic Compilation



$$\mathbf{y} = M\mathbf{x}$$

$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

???



$$f(X^{-1}) = g(\alpha) \cdot h(X)$$

$$\deg(\{f, g, h\}) \leq d$$

polynomial algebra

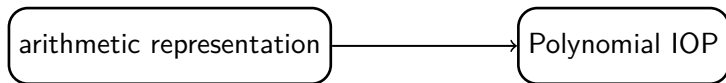
representation

- Reed-Solomon Codes - $f(D)$
- Monomial Coefficients - $\sum c_i X^i$

operation

?

Information-Theoretic Compilation



$$\mathbf{y} = M\mathbf{x}$$

$$\mathbf{a} \circ \mathbf{b} = \mathbf{c}$$

linear algebra

???



$$f(X^{-1}) = g(\alpha) \cdot h(X)$$

$$\deg(\{f, g, h\}) \leq d$$

polynomial algebra

representation

- Reed-Solomon Codes - $f(D)$

- Monomial Coefficients - $\sum c_i X^i$

operation Claymore

PART II

Claymore

Hadamard Product Relation

- ▶ $\mathbb{i} = M \in \mathbb{F}^{m \times (3n+1)}$ — linear transform
- ▶ $\mathbb{x} = \mathbf{x} \in \mathbb{F}^m$ — circuit outputs and inputs
- ▶ $\mathbb{w} = (\mathbf{w}_l, \mathbf{w}_r, \mathbf{w}_o) \in \mathbb{F}^{3n}$ — multiplication gate wires

- ▶ satisfied iff $\mathbf{w}_l \circ \mathbf{w}_r = \mathbf{w}_o$ and $\mathbf{x} = M \cdot \begin{pmatrix} 1 \\ \mathbf{w}_l \\ \mathbf{w}_r \\ \mathbf{w}_o \end{pmatrix}$

[1] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit, "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting," in *Advances in Cryptology - EUROCRYPT 2016, Proceedings, Part II*, 2016, vol. 9666, pp. 327–357. doi: 10.1007/978-3-662-49896-5_12.

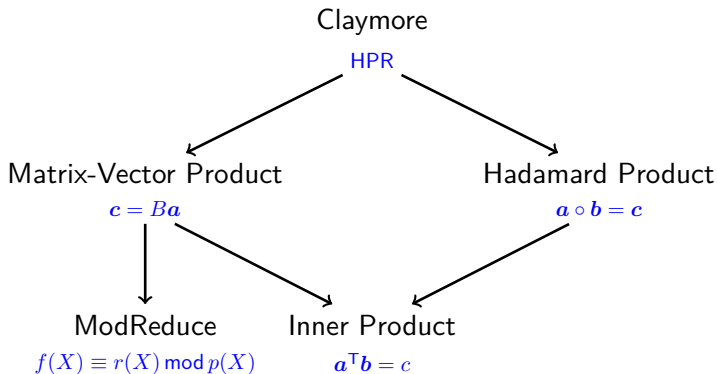
Representation

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \longleftrightarrow f_{\mathbf{a}}(X) = a_0 + a_1X + \cdots + a_{n-1}X^{n-1}$$

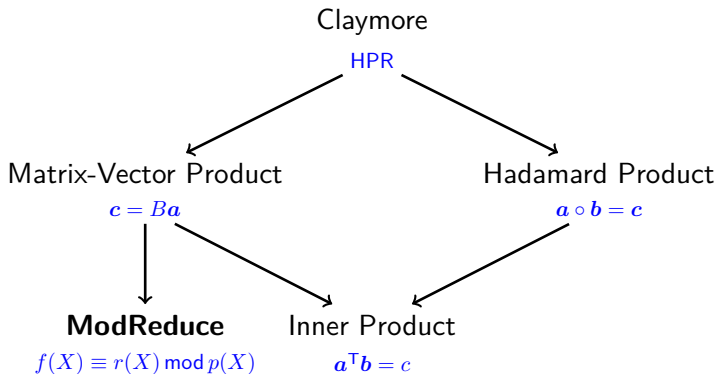
$$B = \begin{pmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,n-1} \\ b_{1,0} & b_{1,1} & \cdots & b_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m-1,0} & b_{m-1,1} & \cdots & b_{m-1,n-1} \end{pmatrix}$$

$$f_B(X) = b_{0,0} + b_{0,1}X + \cdots + b_{m-1,n-1}X^{(m-1)n+n-1}$$

Subprotocols



Subprotocols



Mod Reduce

Verifier has oracles $[f(X)], [r(X)]$

Verifier knows $p(X)$

claim: $f(X) \equiv r(X) \pmod{p(X)}$

Mod Reduce

Verifier has oracles $[f(X)], [r(X)]$

Verifier knows $p(X)$

claim: $f(X) \equiv r(X) \pmod{p(X)}$

Prover

$$q(X), r(X) \leftarrow f(X)/p(X)$$

$$\text{st. } q(X) \cdot p(X) + r(X) = f(X)$$

$$\text{and } \deg(r) < \deg(p)$$

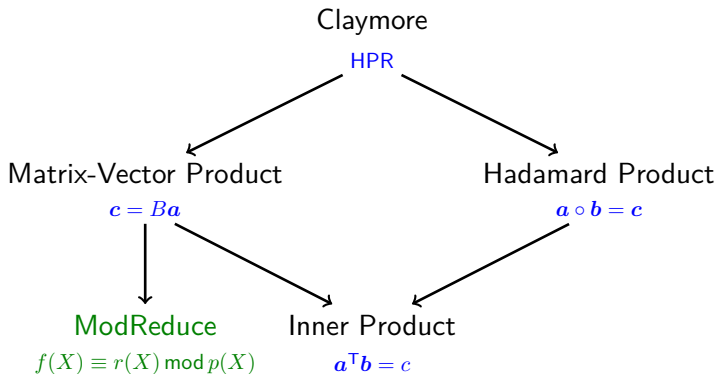
$q(X)$

Verifier

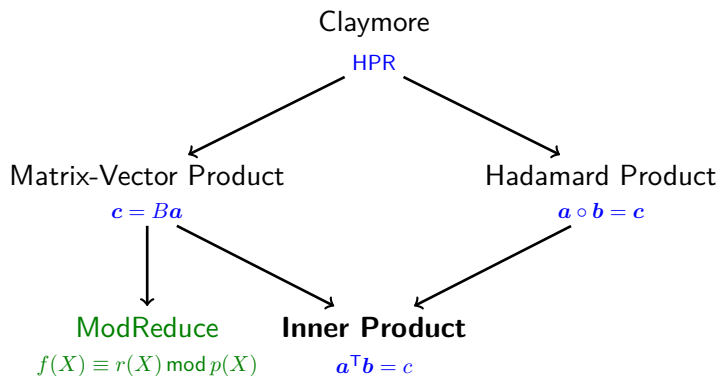
$$z \xleftarrow{\$} \mathbb{F}$$

$$f(z) \stackrel{?}{=} q(z)p(z) + r(z)$$

Subprotocols



Subprotocols



Inner Product

Verifier has oracles $[f_{\mathbf{a}}(X)], [f_{\mathbf{b}}(X)]$

Verifier knows $c \in \mathbb{F}$

claim: $\mathbf{a}^T \mathbf{b} = c$

Inner Product

Verifier has oracles $[f_a(X)], [f_b(X)]$

Verifier knows $c \in \mathbb{F}$

claim: $\mathbf{a}^T \mathbf{b} = c$

Prover



Verifier



Inner Product

Verifier has oracles $[f_a(X)], [f_b(X)]$

Verifier knows $c \in \mathbb{F}$

claim: $\mathbf{a}^\top \mathbf{b} = c$

Prover

$$h(X) \leftarrow X^d f_a(X^{-1}) \cdot f_b(X)$$

Verifier

Inner Product

Verifier has oracles $[f_a(X)], [f_b(X)]$

Verifier knows $c \in \mathbb{F}$

claim: $\mathbf{a}^\top \mathbf{b} = c$

Prover

Verifier

$$h(X) \leftarrow X^d f_a(X^{-1}) \cdot f_b(X)$$

$$\text{Find } \bar{h}(X) \text{ s.t. } h(X) = \bar{h}(X) \cdot \gamma^d - \bar{h}(X \cdot \gamma) + c \cdot X^d$$

$$\bar{h}(X)$$



Inner Product

Verifier has oracles $[f_a(X)], [f_b(X)]$

Verifier knows $c \in \mathbb{F}$

claim: $\mathbf{a}^\top \mathbf{b} = c$

Prover

Verifier

$$h(X) \leftarrow X^d f_a(X^{-1}) \cdot f_b(X)$$

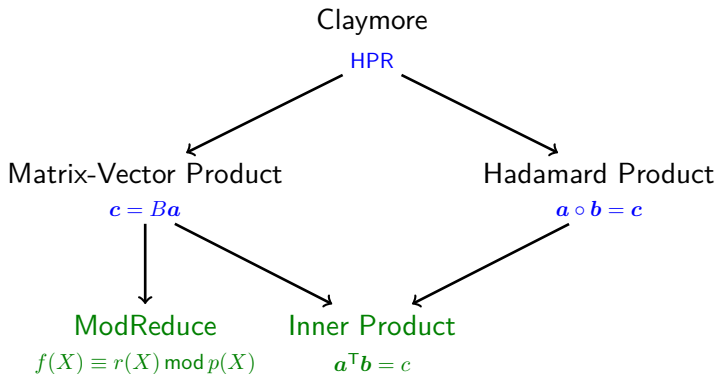
$$\text{Find } \bar{h}(X) \text{ s.t. } h(X) = \bar{h}(X) \cdot \gamma^d - \bar{h}(X \cdot \gamma) + c \cdot X^d$$

$$\bar{h}(X)$$

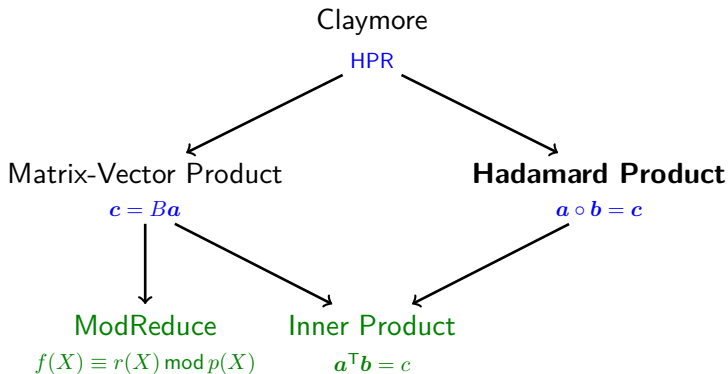
$$f_a(z^{-1}) \cdot f_b(z) \cdot z^d \stackrel{?}{=} \bar{h}(z) \cdot \gamma^d - \bar{h}(z \cdot \gamma) + c \cdot z^d$$

$$z \xleftarrow{\$} \mathbb{F}$$

Subprotocols



Subprotocols



Hadamard

Verifier has oracles $[f_a(X)], [f_b(X)], [f_c(X)]$

claim: $a \circ b = c$

Hadamard

Verifier has oracles $[f_a(X)], [f_b(X)], [f_c(X)]$

claim: $a \circ b = c$

idea: $f_{a \circ b}(\alpha) = f_c(\alpha)$

Hadamard

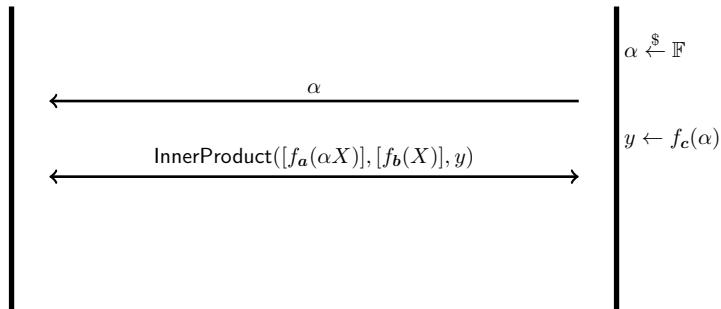
Verifier has oracles $[f_a(X)], [f_b(X)], [f_c(X)]$

claim: $\mathbf{a} \circ \mathbf{b} = \mathbf{c}$

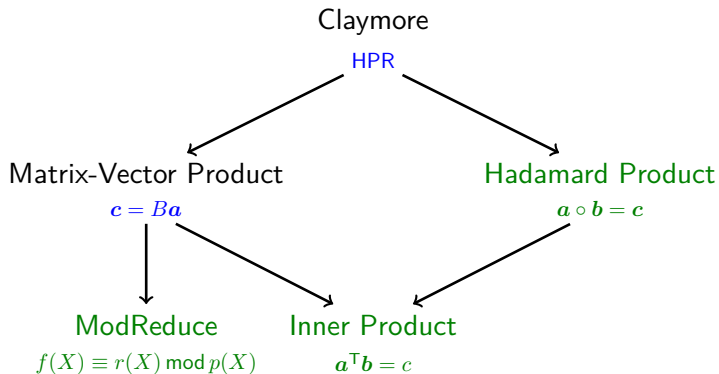
idea: $f_{\mathbf{a} \circ \mathbf{b}}(\alpha) = f_c(\alpha)$

Prover

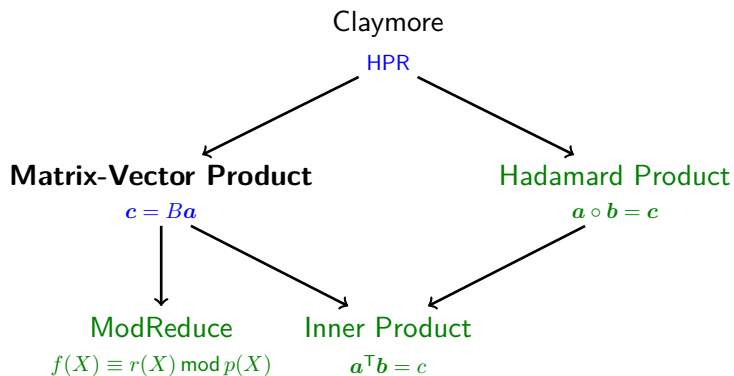
Verifier



Subprotocols



Subprotocols



Dense MVP

$$\begin{aligned} f_B(X) = & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\ & b_{1,0}X^n + b_{1,1}X^{n+1} + \cdots + b_{1,n-1}X^{2n-1} \\ & \vdots \\ & b_{m-1,0}X^{(m-1)n} + \cdots + b_{m-1,n-1}X^{mn-1} \end{aligned}$$

Dense MVP

$$\begin{aligned} f_B(X) = & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\ & b_{1,0}X^n + b_{1,1}X^{n+1} + \cdots + b_{1,n-1}X^{2n-1} \\ & \vdots \\ & b_{m-1,0}X^{(m-1)n} + \cdots + b_{m-1,n-1}X^{mn-1} \end{aligned} \begin{array}{l} \curvearrowright \times \frac{\alpha}{X^n} \\ \vdots \\ \curvearrowright \times \frac{\alpha}{X^n} \end{array}$$

reduce modulo $X^n - \alpha$

Dense MVP

$$\begin{aligned} f_B(X) = & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\ & b_{1,0}X^n + b_{1,1}X^{n+1} + \cdots + b_{1,n-1}X^{2n-1} \\ & \vdots \\ & b_{m-1,0}X^{(m-1)n} + \cdots + b_{m-1,n-1}X^{mn-1} \end{aligned} \begin{array}{l} \curvearrowright \times \frac{\alpha}{X^n} \\ \vdots \\ \curvearrowright \times \frac{\alpha}{X^n} \end{array}$$

reduce modulo $X^n - \alpha$

$$\begin{aligned} r(X) = & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\ & (b_{1,0} + b_{1,1}X + b_{1,2}X^2 + \cdots + b_{1,n-1}X^{n-1}) \cdot \alpha \\ & \vdots \\ & (b_{m-1,0} + b_{m-1,1}X + \cdots + b_{m-1,n-1}X^{n-1}) \cdot \alpha^{m-1} \end{aligned}$$

Dense MVP

$$\begin{aligned} f_B(X) = & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\ & b_{1,0}X^n + b_{1,1}X^{n+1} + \cdots + b_{1,n-1}X^{2n-1} \\ & \vdots \\ & b_{m-1,0}X^{(m-1)n} + \cdots + b_{m-1,n-1}X^{mn-1} \end{aligned} \begin{array}{l} \curvearrowright \times \frac{\alpha}{X^n} \\ \vdots \\ \curvearrowright \times \frac{\alpha}{X^n} \end{array}$$

reduce modulo $X^n - \alpha$

$$\begin{aligned} r(X) = & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\ & (b_{1,0} + b_{1,1}X + b_{1,2}X^2 + \cdots + b_{1,n-1}X^{n-1}) \cdot \alpha \\ & \vdots \\ & (b_{m-1,0} + b_{m-1,1}X + \cdots + b_{m-1,n-1}X^{n-1}) \cdot \alpha^{m-1} \end{aligned}$$

$$r = \alpha^\top B, \text{ where } \alpha = (1, \alpha, \alpha^2, \dots, \alpha^{m-1})$$

Dense MVP

$$\mathbf{c} = B \mathbf{a}$$

Dense MVP

$$\begin{array}{c} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \mathbf{c} \end{array} = \begin{array}{c} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \mathbf{B} \\ \mathbf{a} \end{array}$$

Dense MVP

$$\underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \left| \mathbf{c} \right. \end{matrix}}_{f_c(\alpha)} = \underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \boxed{B} \\ \left| \mathbf{a} \right. \end{matrix}}_{r(X) = f_B(X) \bmod X^n - \alpha}$$

$\text{InnerProduct}(r(X), f_a(X), f_c(\alpha))$

Dense MVP

Verifier has oracles $[f_c(X)], [f_B(X)], [f_a(X)]$

claim: $c = Ba$

Dense MVP

Verifier has oracles $[f_c(X)], [f_B(X)], [f_a(X)]$

claim: $c = Ba$

Prover



Verifier



Dense MVP

Verifier has oracles $[f_c(X)], [f_B(X)], [f_a(X)]$

claim: $c = Ba$

Prover



α

Verifier



$\alpha \xleftarrow{\$} \mathbb{F} \setminus \{0\}$

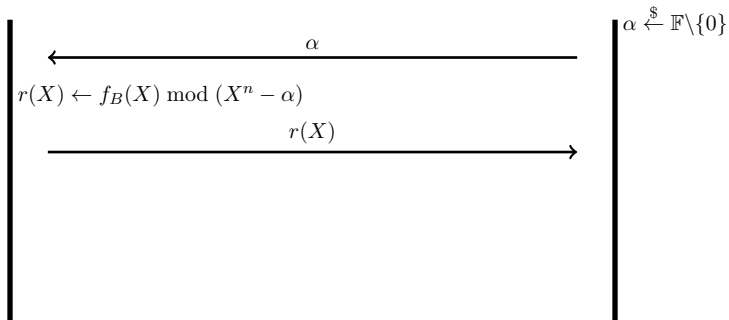
Dense MVP

Verifier has oracles $[f_c(X)], [f_B(X)], [f_a(X)]$

claim: $c = Ba$

Prover

Verifier



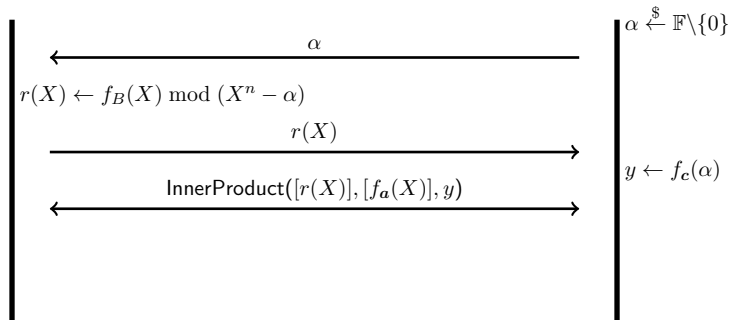
Dense MVP

Verifier has oracles $[f_c(X)], [f_B(X)], [f_a(X)]$

claim: $c = Ba$

Prover

Verifier



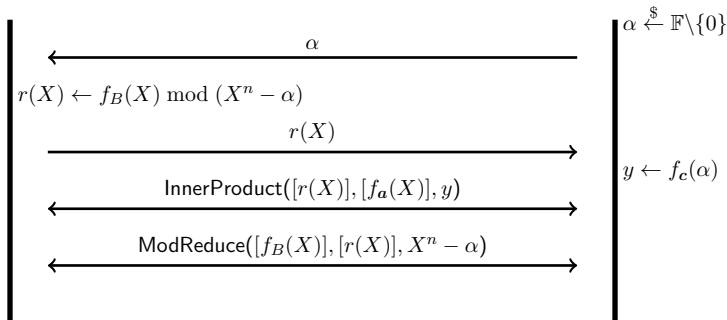
Dense MVP

Verifier has oracles $[f_c(X)], [f_B(X)], [f_a(X)]$

claim: $c = Ba$

Prover

Verifier

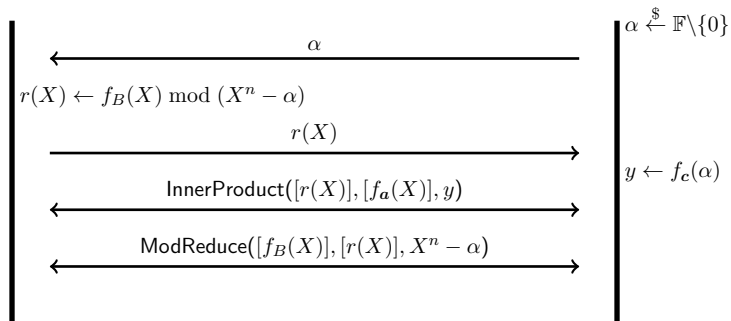


Dense MVP

Verifier has oracles $[f_c(X)], [f_B(X)], [f_a(X)]$
claim: $c = Ba$

Prover

Verifier



Prover complexity = $O(mn)$, even for sparse matrix!

Sparse MVP

$$\mathbf{c} = B \mathbf{a}$$

Sparse MVP

$$\begin{array}{c} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \mathbf{c} \end{array} = \begin{array}{c} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \mathbf{B} \\ \mathbf{a} \end{array}$$

Sparse MVP

$$\underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \mathbf{c} \end{matrix}}_{f_c(\alpha)} = \underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} & \boxed{B} & \boxed{\mathbf{a}} \end{matrix}}_{\mathbf{r} = \alpha^T B} \\ \text{InnerProduct}(\mathbf{r}(X), f_a(X), f_c(\alpha))$$

Sparse MVP

$$\mathbf{r} = \begin{bmatrix} 1 & \alpha^1 & \dots & \alpha^{m-1} \end{bmatrix} \mathbf{B}$$

Sparse MVP

$$\mathbf{r} = \begin{bmatrix} 1 \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{bmatrix} = \begin{bmatrix} 1 & \alpha^1 & \dots & \alpha^{m-1} \end{bmatrix} \mathbf{B} \begin{bmatrix} 1 \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{bmatrix}$$

Sparse MVP

$$\underbrace{\begin{matrix} \boxed{\mathbf{r}} \\ \vdots \\ \boxed{\beta^{n-1}} \end{matrix}}_{r(\beta)} = \underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} & \boxed{B} & \begin{matrix} \boxed{1} \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{matrix} \end{matrix}}_{???$$

The diagram illustrates a sparse matrix-vector product. On the left, a vector \mathbf{r} is shown in a rounded rectangle, with a vertical list of elements $1, \beta^1, \dots, \beta^{n-1}$ in a green box. A blue bracket below this is labeled $r(\beta)$. An equals sign follows. On the right, a blue rounded rectangle contains the sequence $1 \ \alpha^1 \ \dots \ \alpha^{m-1}$. To its right is a square matrix B , and further right is a green box containing the vertical list $1, \beta^1, \dots, \beta^{n-1}$. A purple bracket below these three components is labeled $???$.

Sparse MVP

$$\underbrace{\left(\begin{array}{c} \mathbf{r} \end{array} \right)}_{r(\beta)} = \underbrace{\left(\begin{array}{c} 1 \ \alpha^1 \ \dots \ \alpha^{m-1} \end{array} \right)}_{F_B(\alpha, \beta) := \sum_{k=1}^K c_k \cdot \alpha^{a_k} \cdot \beta^{b_k}} \cdot \left(\begin{array}{c} 1 \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{array} \right)$$

The diagram illustrates the Sparse MVP equation. On the left, a rounded rectangle containing the vector \mathbf{r} is underlined with a blue bracket labeled $r(\beta)$. This is followed by an equals sign and a blue rounded rectangle containing the vector $1 \ \alpha^1 \ \dots \ \alpha^{m-1}$, which is underlined with a purple bracket labeled $F_B(\alpha, \beta) := \sum_{k=1}^K c_k \cdot \alpha^{a_k} \cdot \beta^{b_k}$. To the right of the purple bracket is a large empty square labeled B . Finally, a green rounded rectangle contains the vector $1 \ \beta^1 \ \vdots \ \beta^{n-1}$.

Sparse MVP

$$\underbrace{\begin{matrix} \boxed{\mathbf{r}} \\ \vdots \\ \boxed{\beta^{n-1}} \end{matrix}}_{r(\beta)} = \underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} & \boxed{B} & \begin{matrix} \boxed{1} \\ \boxed{\beta^1} \\ \vdots \\ \boxed{\beta^{n-1}} \end{matrix} \end{matrix}}_{F_B(\alpha, \beta) := \sum_{k=1}^K c_k \cdot \alpha^{a_k} \cdot \beta^{b_k} = \mathbf{v}^\top(\mathbf{x} \circ \mathbf{y})}$$

Sparse MVP

$$\underbrace{\begin{matrix} \boxed{\mathbf{r}} \\ \vdots \\ \boxed{\beta^{n-1}} \end{matrix}}_{r(\beta)} = \underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} & \boxed{B} & \begin{matrix} \boxed{1} \\ \boxed{\beta^1} \\ \vdots \\ \boxed{\beta^{n-1}} \end{matrix} \end{matrix}}_{F_B(\alpha, \beta) := \sum_{k=1}^K c_k \cdot \alpha^{a_k} \cdot \beta^{b_k} = \mathbf{v}^\top(\mathbf{x} \circ \mathbf{y})}$$

$\text{InnerProduct}([f_v(X)], \text{Hadamard}([f_x(X)], [f_y(X)]), r(\beta))$

Sparse MVP

$$\underbrace{\begin{matrix} \boxed{\mathbf{r}} \\ \vdots \\ \boxed{\beta^{n-1}} \end{matrix}}_{r(\beta)} = \underbrace{\begin{matrix} \boxed{1 \ \alpha^1 \ \dots \ \alpha^{m-1}} \\ \boxed{B} \\ \boxed{\beta^{n-1}} \end{matrix}}_{F_B(\alpha, \beta) := \sum_{k=1}^K c_k \cdot \alpha^{a_k} \cdot \beta^{b_k} = \mathbf{v}^T(\mathbf{x} \circ \mathbf{y})}$$

$\text{InnerProduct}(\underbrace{[f_v(X)]}_{\text{Preprocessed}}, \text{Hadamard}(\underbrace{[f_x(X)]}_{???}, [f_y(X)]), r(\beta))$

Sparse MVP

$$\underbrace{\left(\mathbf{r} \quad \begin{bmatrix} 1 \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{bmatrix} \right)}_{r(\beta)} = \underbrace{\left(\begin{bmatrix} 1 & \alpha^1 & \dots & \alpha^{m-1} \end{bmatrix} \quad B \quad \begin{bmatrix} 1 \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{bmatrix} \right)}_{F_B(\alpha, \beta) := \sum_{k=1}^K c_k \cdot \alpha^{a_k} \cdot \beta^{b_k} = \mathbf{v}^\top(\mathbf{x} \circ \mathbf{y})}$$

$$\text{InnerProduct}(\underbrace{f_v(X)}_{\text{Preprocessed}}, \text{Hadamard}([f_x(X)], [f_y(X)]), r(\beta))$$

Preprocessed

SparseMonomialVector

Sparse MVP

Verifier has oracles $[f_c(X)]$, $[f_v(X)]$, $[f_a(X)]$

where v are the nonzero values of B

claim: $c = B\mathbf{a}$

Sparse MVP

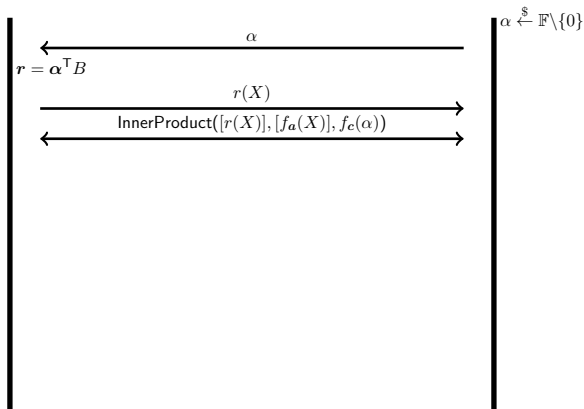
Verifier has oracles $[f_c(X)]$, $[f_v(X)]$, $[f_a(X)]$

where v are the nonzero values of B

claim: $c = B\mathbf{a}$

Prover

Verifier

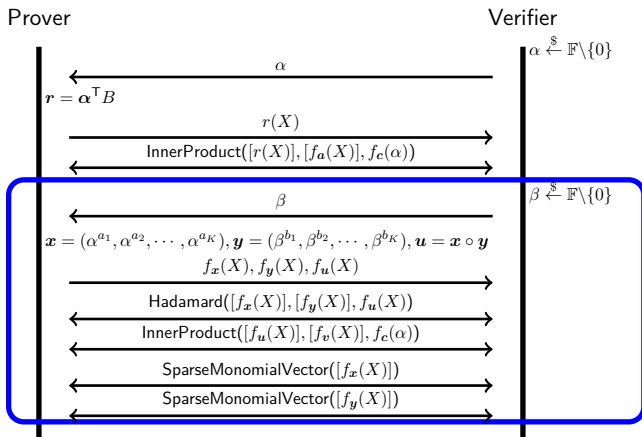


Sparse MVP

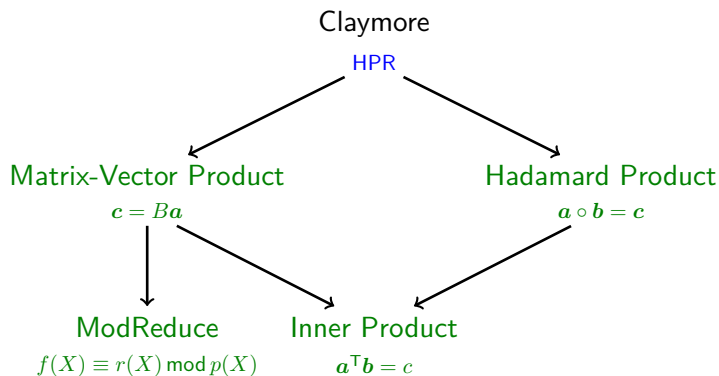
Verifier has oracles $[f_c(X)]$, $[f_v(X)]$, $[f_a(X)]$

where v are the nonzero values of B

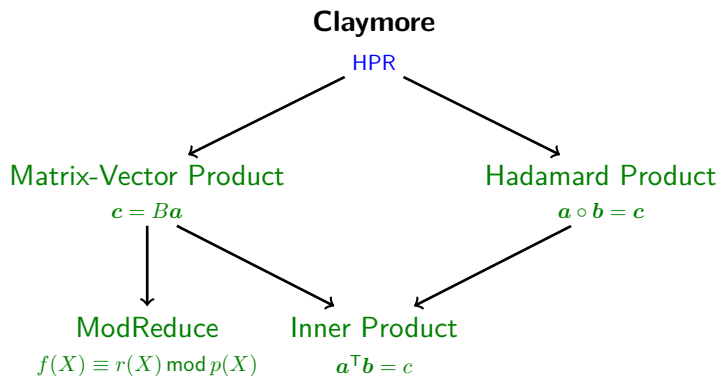
claim: $c = Ba$



Subprotocols



Subprotocols



Claymore

Verifier has oracle $[f_M(X)]$

Verifier knows x

claim: Prover knows $(\mathbf{w}_l, \mathbf{w}_r, \mathbf{w}_o) \in \mathbb{F}^{3n}$.

$$x = M \cdot (1|\mathbf{w}_l^\top|\mathbf{w}_r^\top|\mathbf{w}_o^\top)^\top \wedge \mathbf{w}_l \circ \mathbf{w}_r = \mathbf{w}_o$$

Claymore

Verifier has oracle $[f_M(X)]$

Verifier knows x

claim: Prover knows $(w_l, w_r, w_o) \in \mathbb{F}^{3n}$.

$$x = M \cdot (1|w_l^\top|w_r^\top|w_o^\top)^\top \wedge w_l \circ w_r = w_o$$

Prover



Verifier



Claymore

Verifier has oracle $[f_M(X)]$

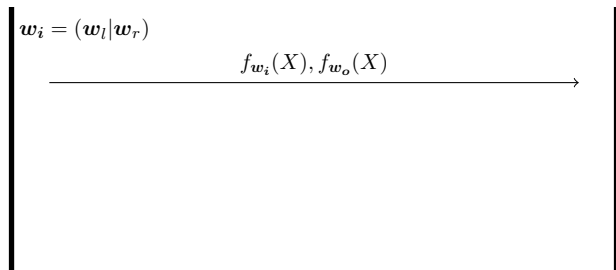
Verifier knows x

claim: Prover knows $(w_l, w_r, w_o) \in \mathbb{F}^{3n}$.

$$x = M \cdot (1|w_l^\top|w_r^\top|w_o^\top)^\top \wedge w_l \circ w_r = w_o$$

Prover

Verifier



Claymore

Verifier has oracle $[f_M(X)]$

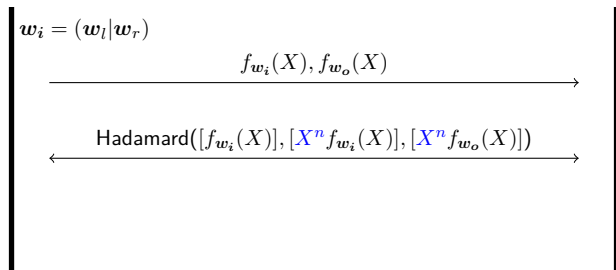
Verifier knows x

claim: Prover knows $(w_l, w_r, w_o) \in \mathbb{F}^{3n}$.

$$x = M \cdot (1|w_l^\top|w_r^\top|w_o^\top)^\top \wedge w_l \circ w_r = w_o$$

Prover

Verifier



Claymore

Verifier has oracle $[f_M(X)]$

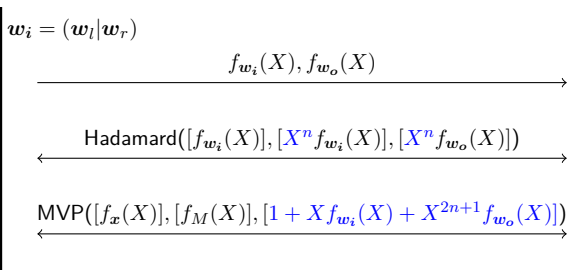
Verifier knows x

claim: Prover knows $(\mathbf{w}_l, \mathbf{w}_r, \mathbf{w}_o) \in \mathbb{F}^{3n}$.

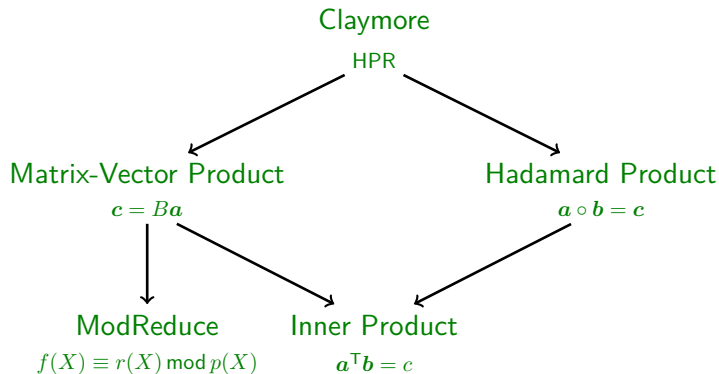
$$\mathbf{x} = M \cdot (1|\mathbf{w}_l^\top|\mathbf{w}_r^\top|\mathbf{w}_o^\top)^\top \wedge \mathbf{w}_l \circ \mathbf{w}_r = \mathbf{w}_o$$

Prover

Verifier



Subprotocols



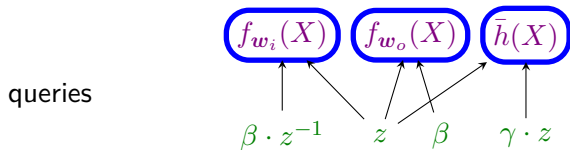
Zero-Knowledge

Witness-related polynomials

$$f_{w_i}(X) \quad f_{w_o}(X) \quad \bar{h}(X)$$

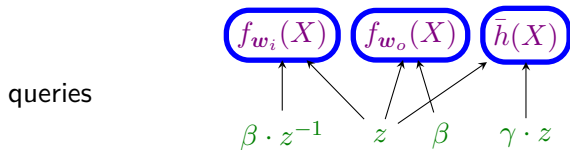
Zero-Knowledge

Witness-related polynomials

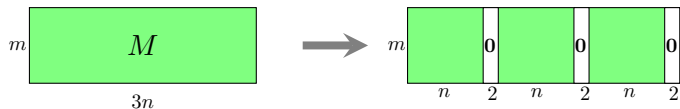


Zero-Knowledge

Witness-related polynomials

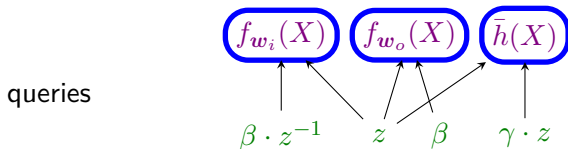


padding

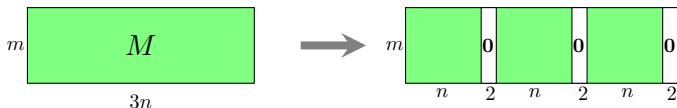


Zero-Knowledge

Witness-related polynomials



padding

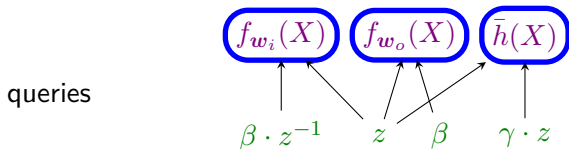


randomizing

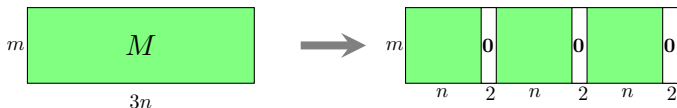


Zero-Knowledge

Witness-related polynomials



padding



randomizing



Performance

	# polys offline / online	# evals	# points	max. degree
Sonic				
PLONK				
Marlin				
DenseClaymore				
SparseClaymore				

Performance

	# polys offline / online	# evals	# points	max. degree
Sonic	$\frac{12f}{3f+7}$	$11f+3$	$9f+2$	$O(n)$
PLONK	8 / 6	7	2	$O(n)$
Marlin	9 / 12	18	3	$O(K)$
DenseClaymore	1 / 4	10	6	$O(mn)$
SparseClaymore	8 / 10	30	10	$O(K)$

Conclusion

- ▶ Polynomial IOPs for linear operations in monomial-basis
 - ▶ Inner Product
 - ▶ Hadamard Product
 - ▶ Dense/Sparse MVP
- ▶ Claymore: a Polynomial IOP/SNARK for circuit

Thank you!

<https://eprint.iacr.org/2020/1022>

Alan Szepieniec

alan@nervos.org

Yuncong Zhang

shjdzhangyuncong@sjtu.edu.cn