Polynomial IOPs for Linear Algebra

Alan Szepieniec

alan@nervos.org

Yuncong Zhang

shjdzhangyuncong@sjtu.edu.cn

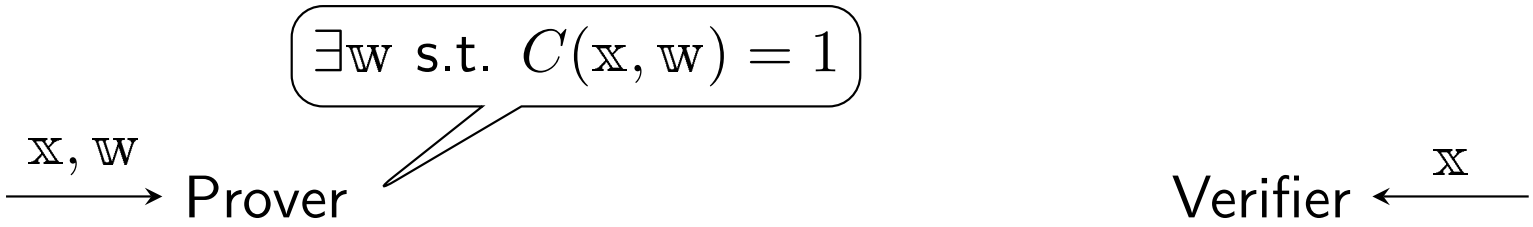
Nervos
Shanghai Jiao Tong University

PKC 2022

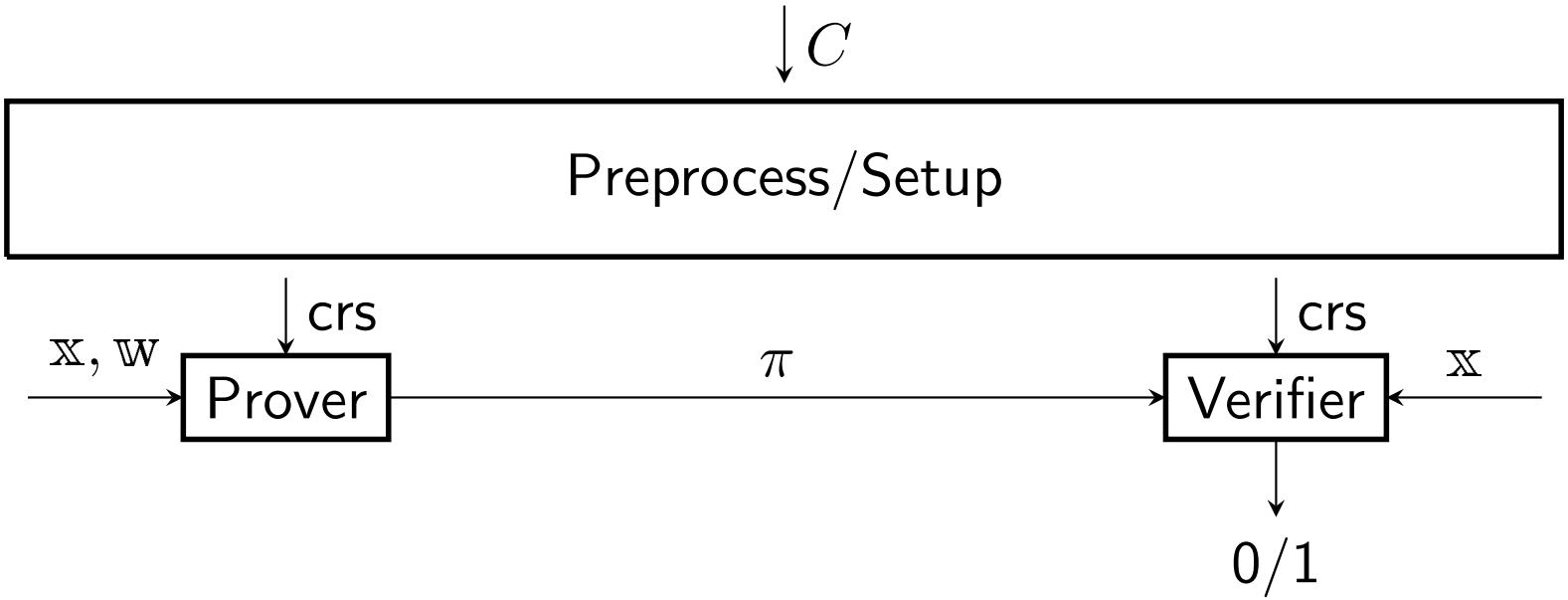
PART I
Background

SNARK

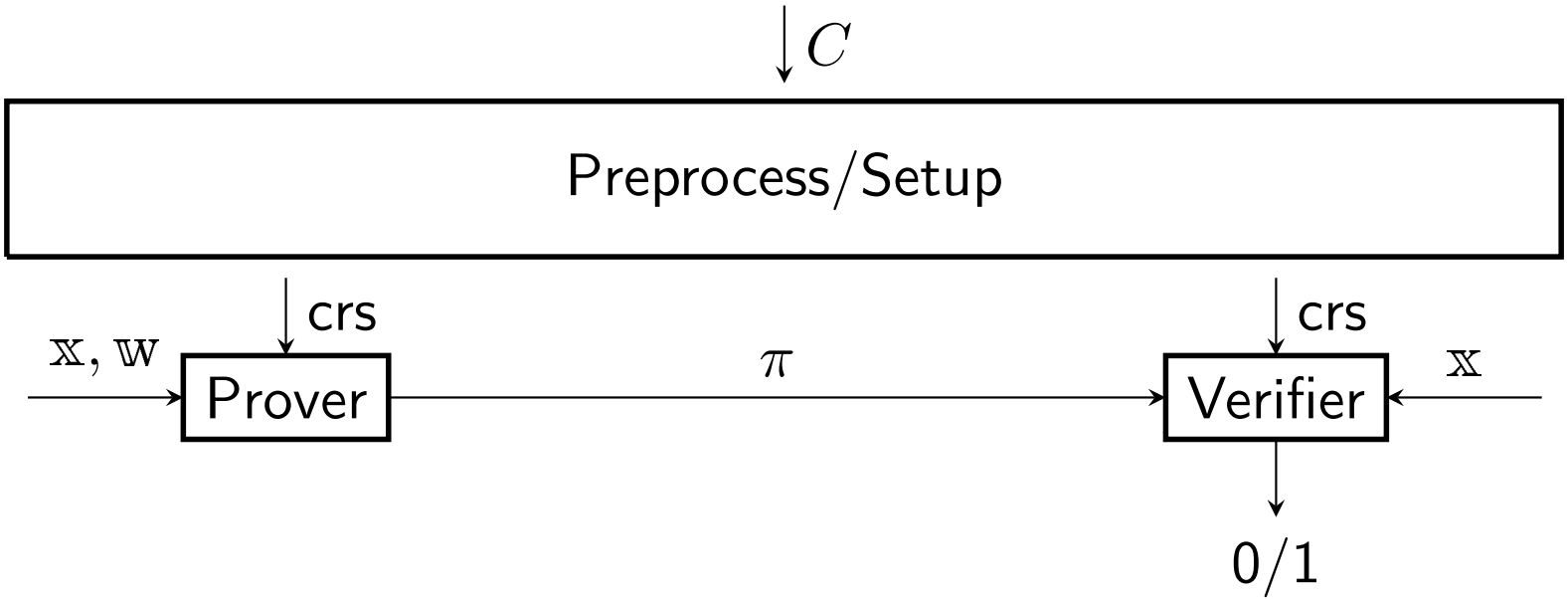
SNARK


$\exists \mathbb{w}$ s.t. $C(\mathbb{x}, \mathbb{w}) = 1$
$\mathbb{x}, \mathbb{w}$
Prover
Verifier
$\mathbb{x}$

SNARK

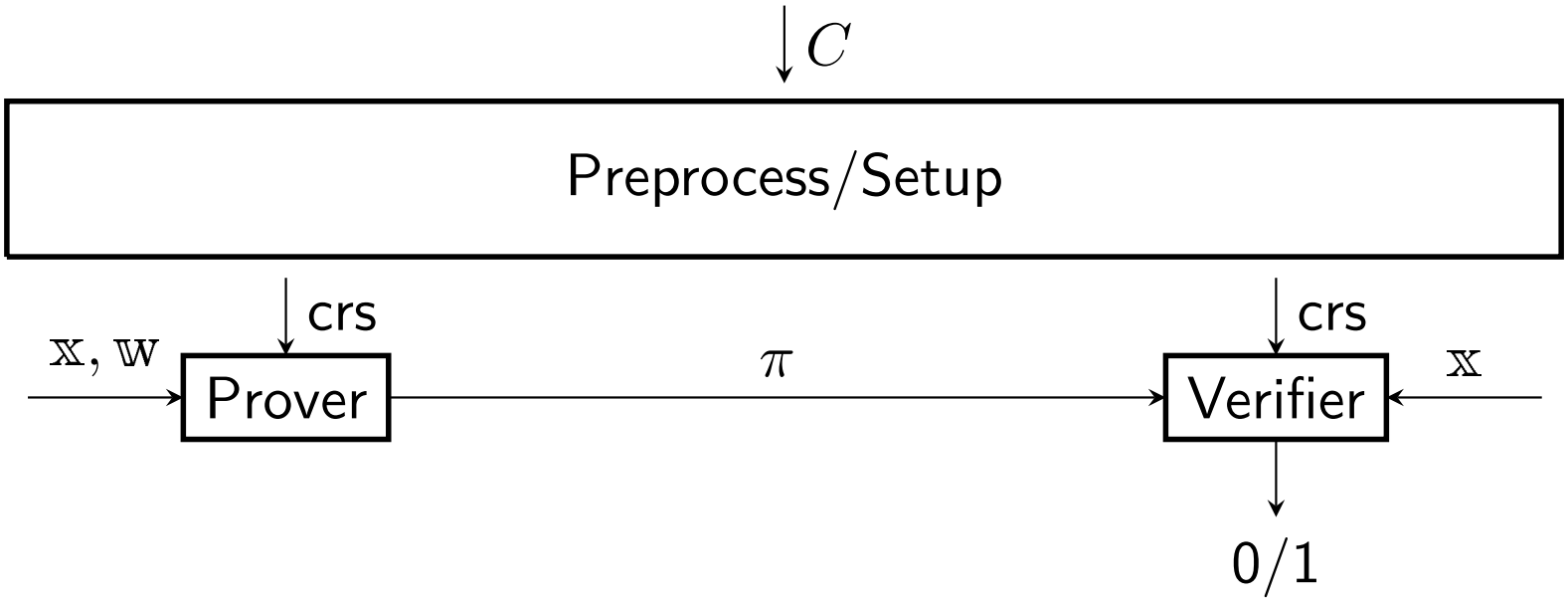

C
Preprocess/Setup
crs
$\mathbb{x}, \mathbb{w}$
Prover
π
crs
Verifier
$\mathbb{x}$
0/1

SNARK


C
Preprocess/Setup
crs
$\mathbb{x}, \mathbb{w}$
Prover
π
crs
Verifier
$\mathbb{x}$
0/1

S N ARK

SNARK



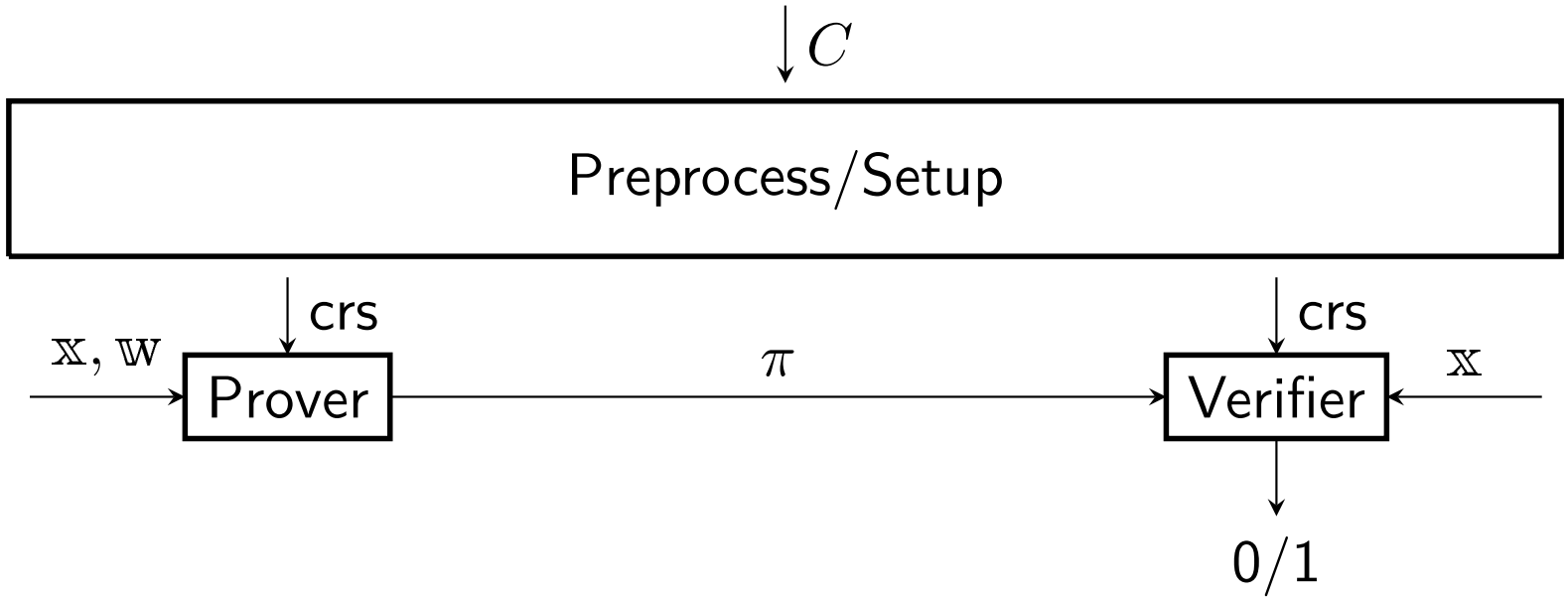
S N ARK

Succinct

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

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

SNARK



S N ARK

Succinct

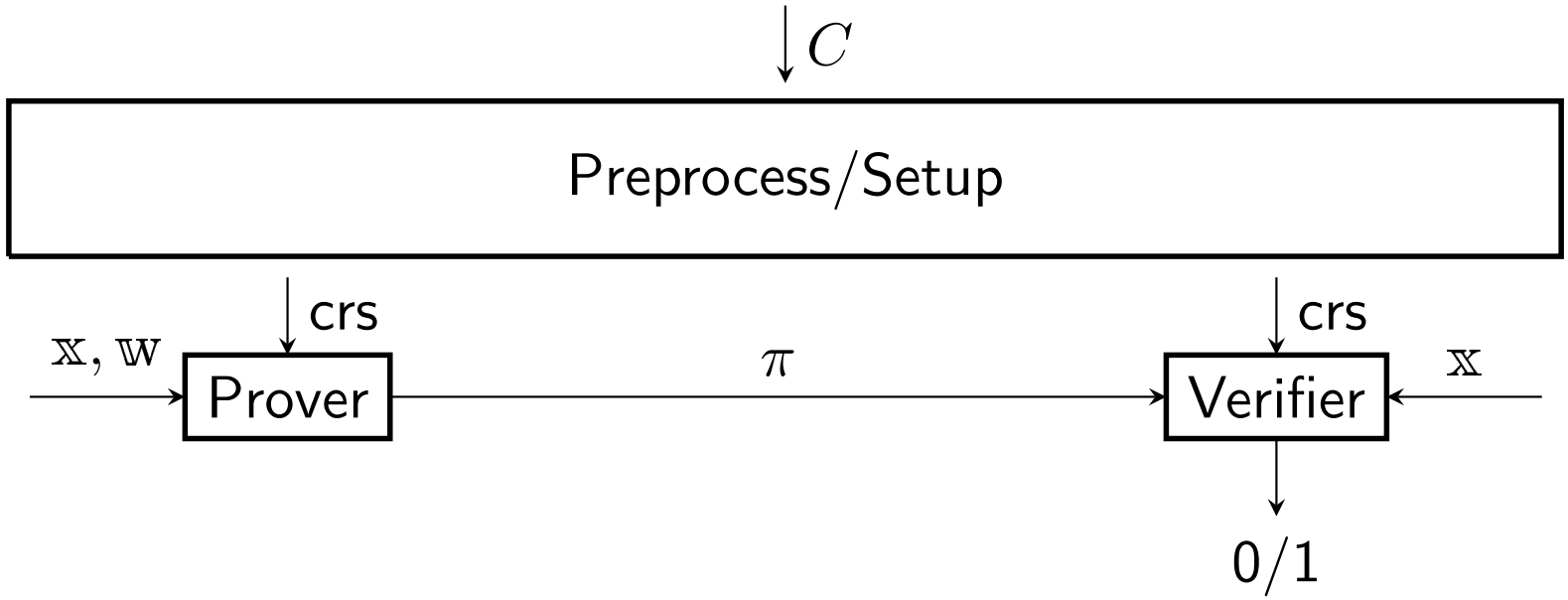
$|\pi| = O(polylog(|C| + |\mathbb{x}|))$

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

Non-interactive

#rounds=1

SNARK



S N ARK

Succinct

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

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

Non-interactive

#rounds=1

ARgument-of-Knowledge

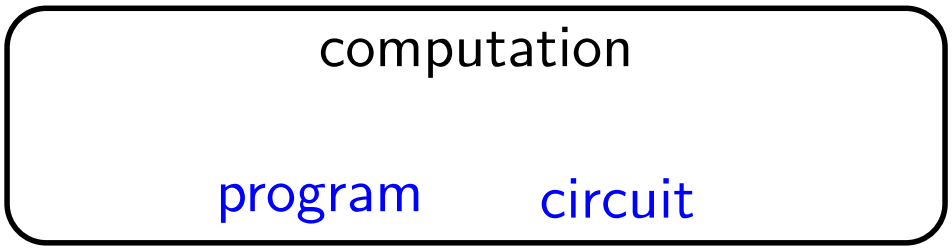
I "know" $\mathbb{w}$, s.t.

~~$\exists \mathbb{w}$, s.t.~~

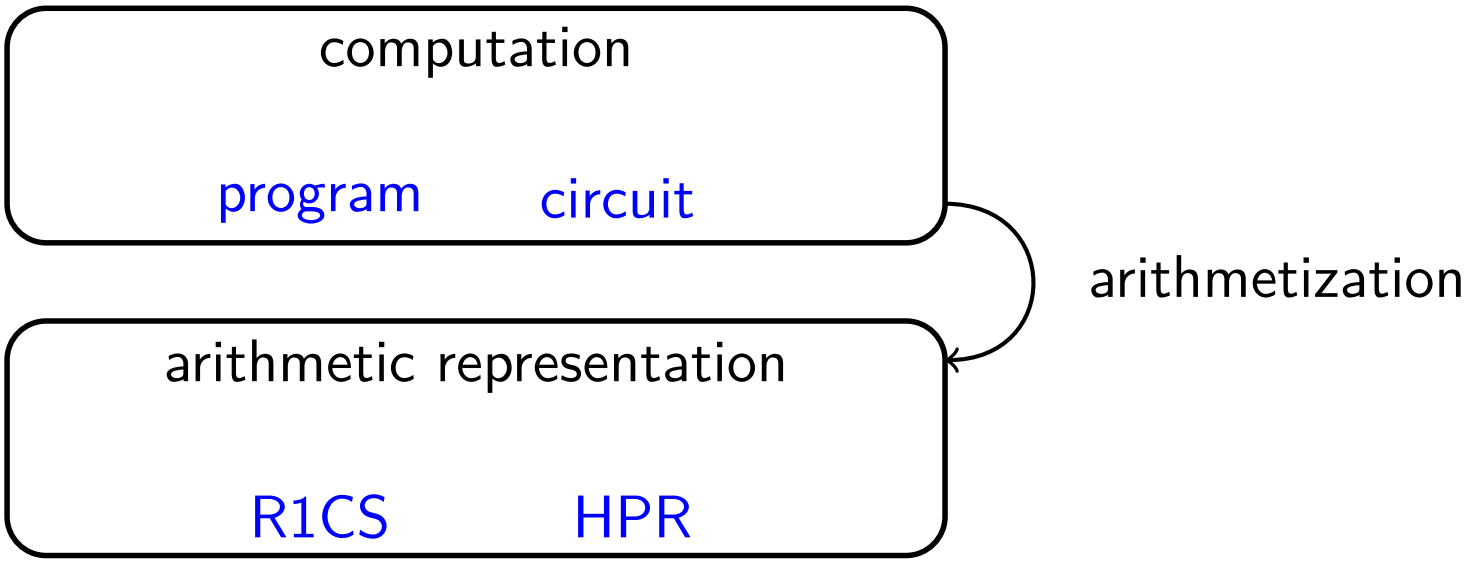
$C(\mathbb{x}, \mathbb{w}) = 0$

Construction Pipeline

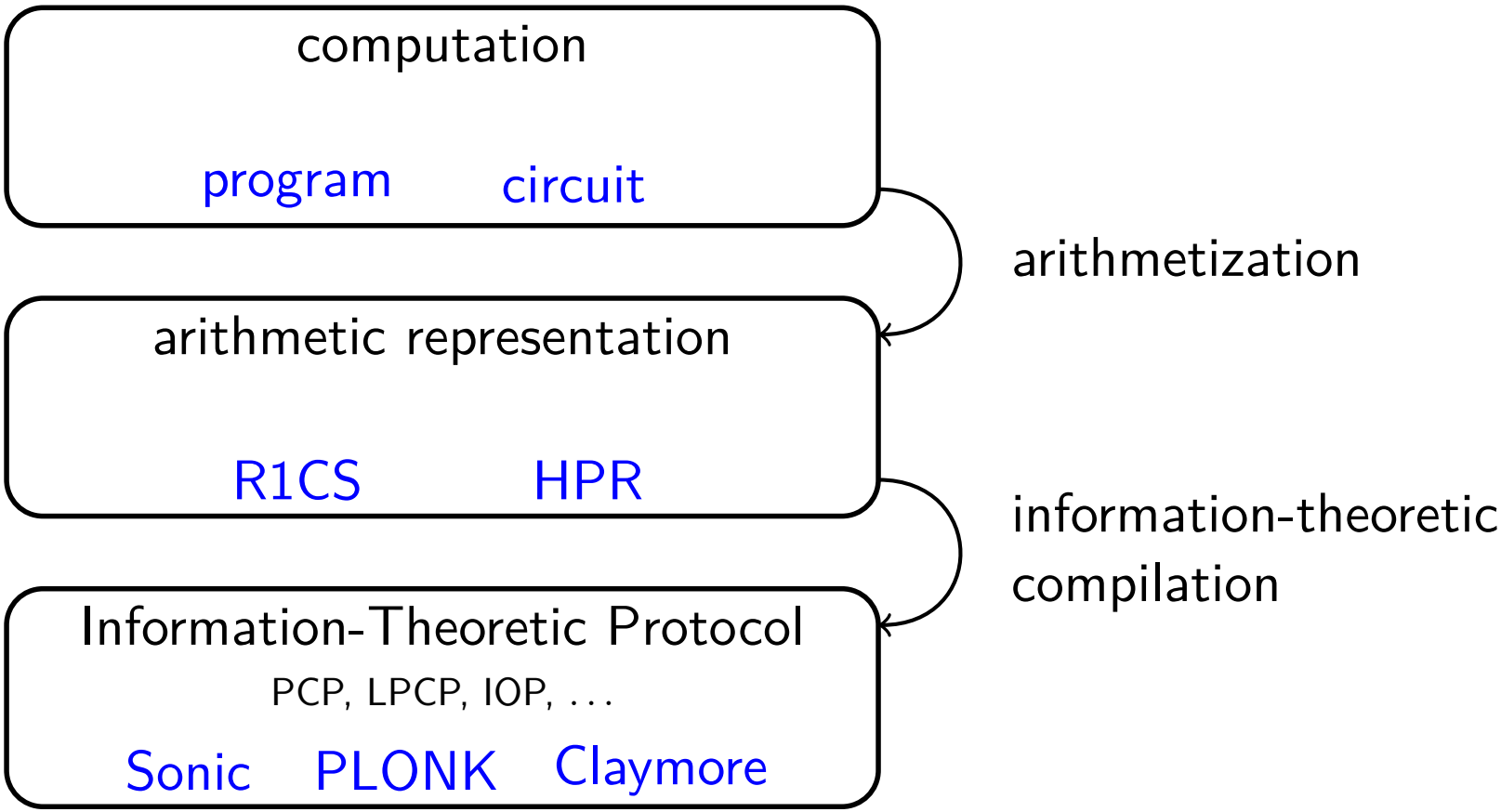
Construction Pipeline



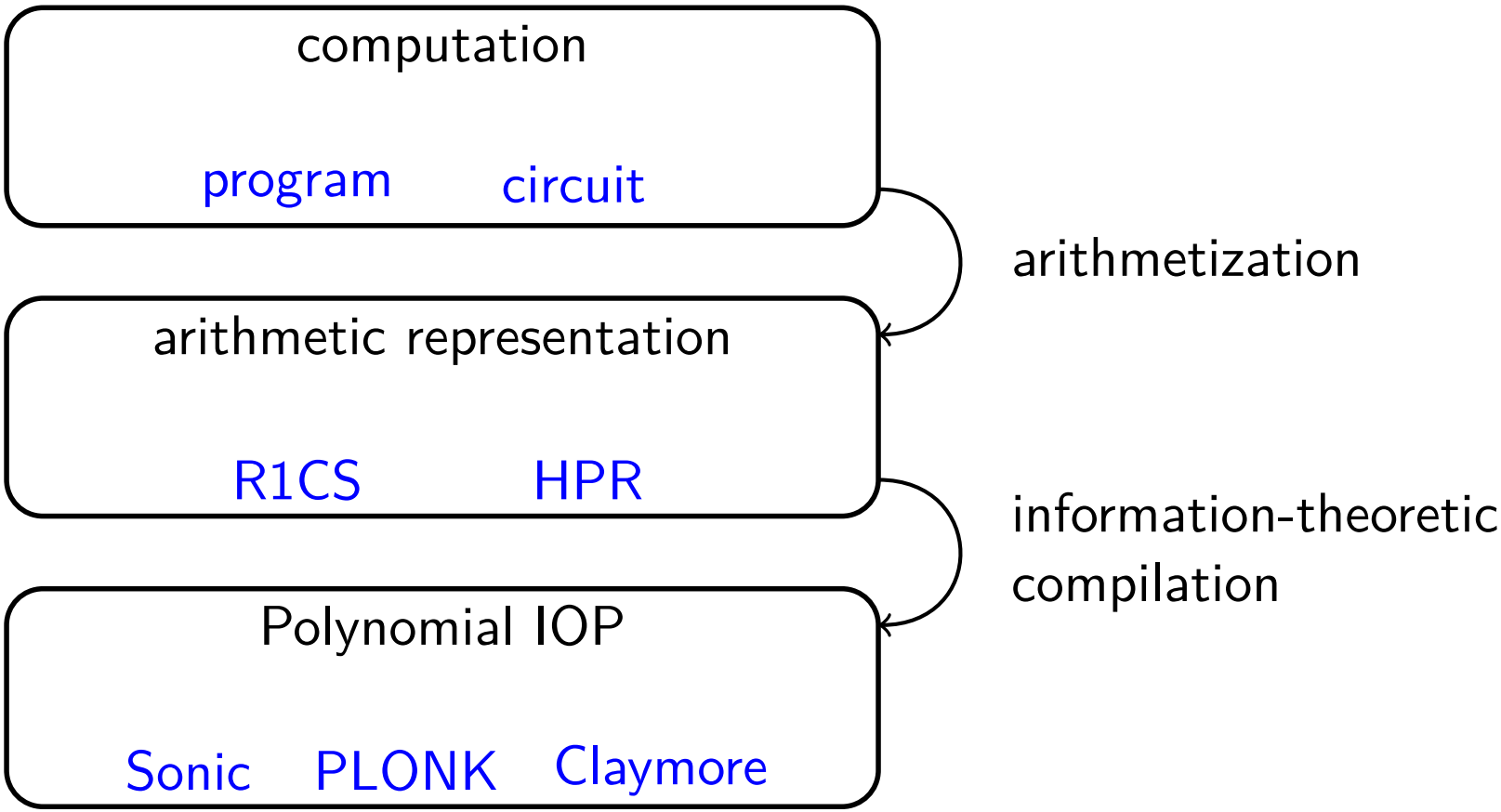
Construction Pipeline



Construction Pipeline

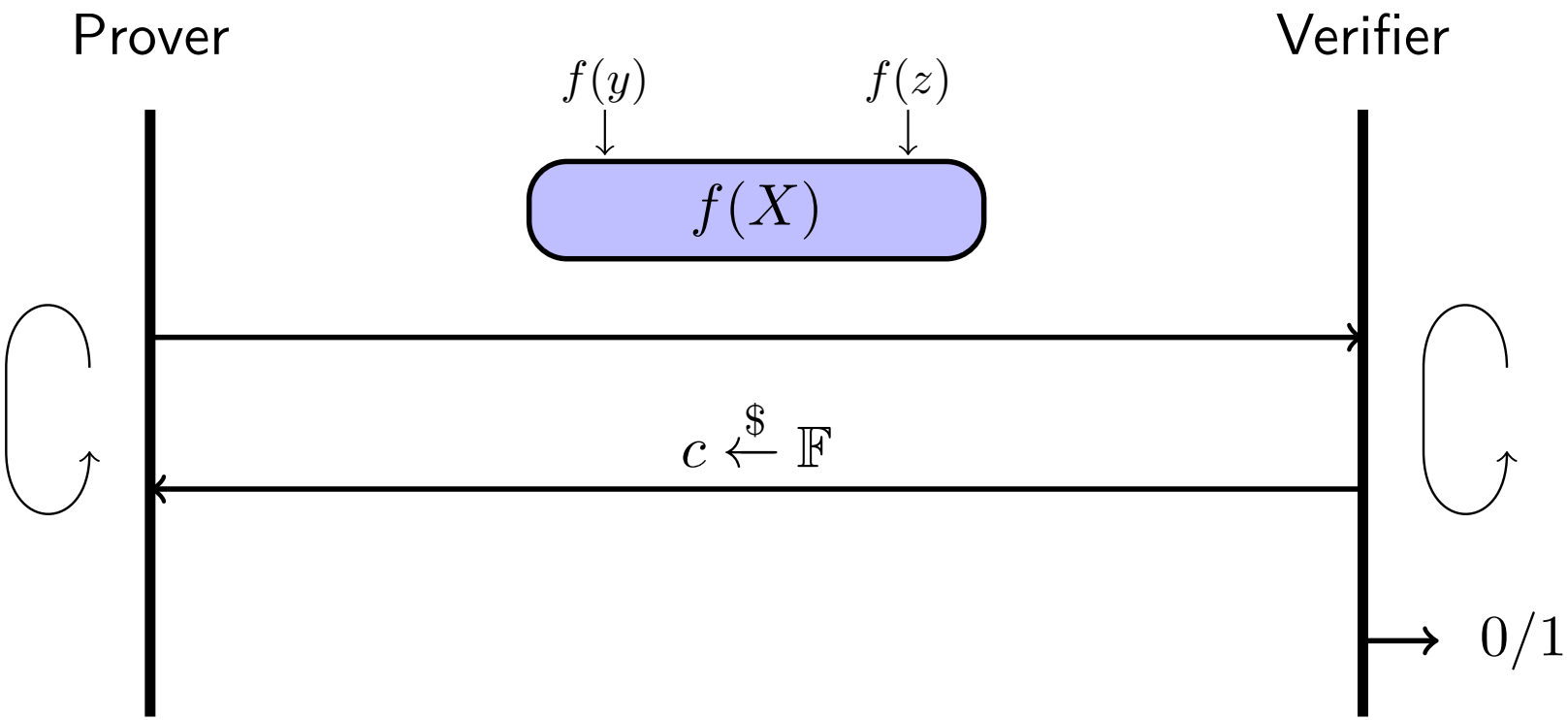


Construction Pipeline

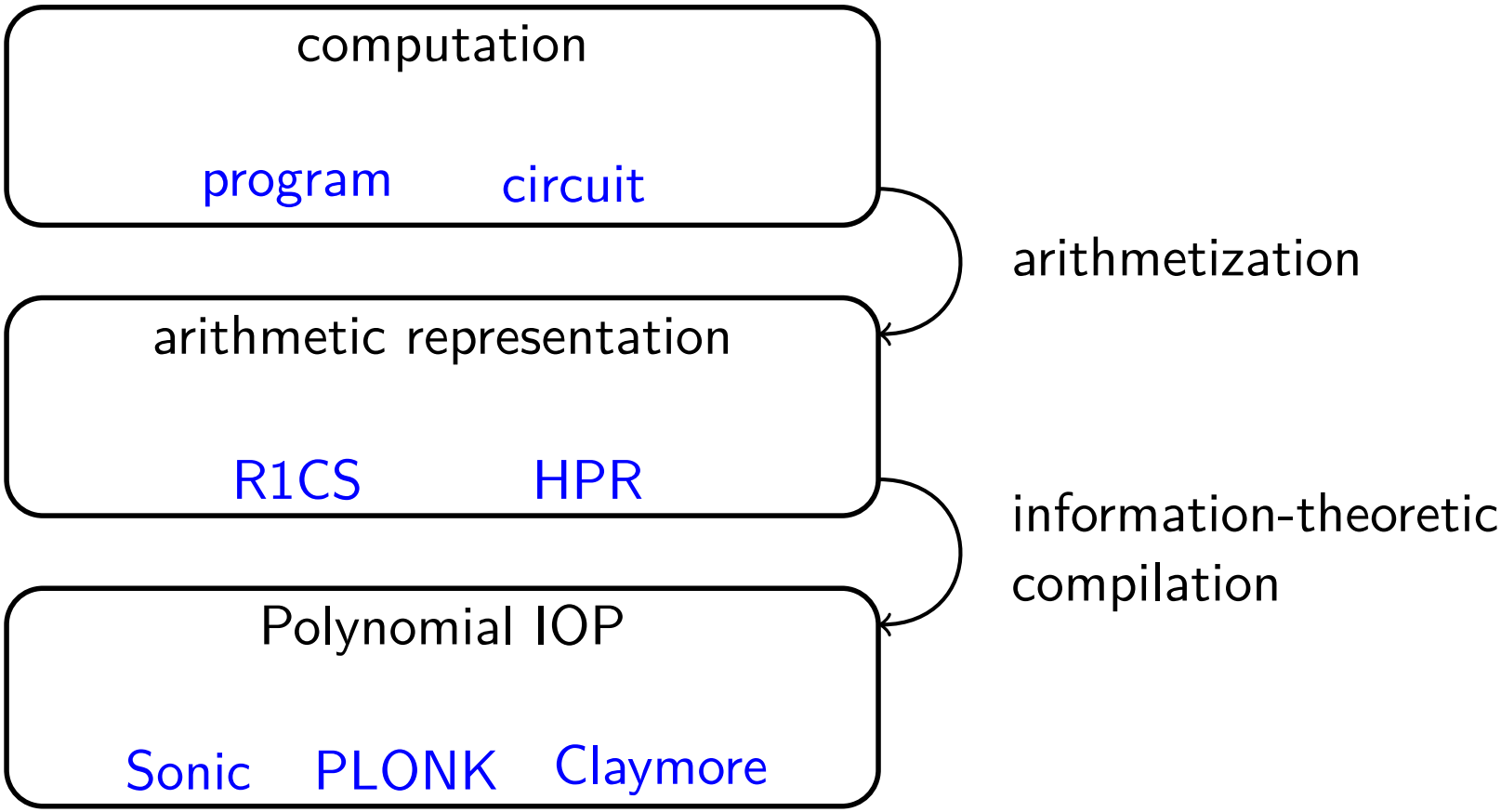


[1] B. Bünz, B. Fisch, and A. Szepieniec, "Transparent SNARKs from DARK Compilers," in Advances in Cryptology – EUROCRYPT 2020, Cham, 2020, pp. 677–706. doi: 10.1007/978-3-030-45721-1_24.

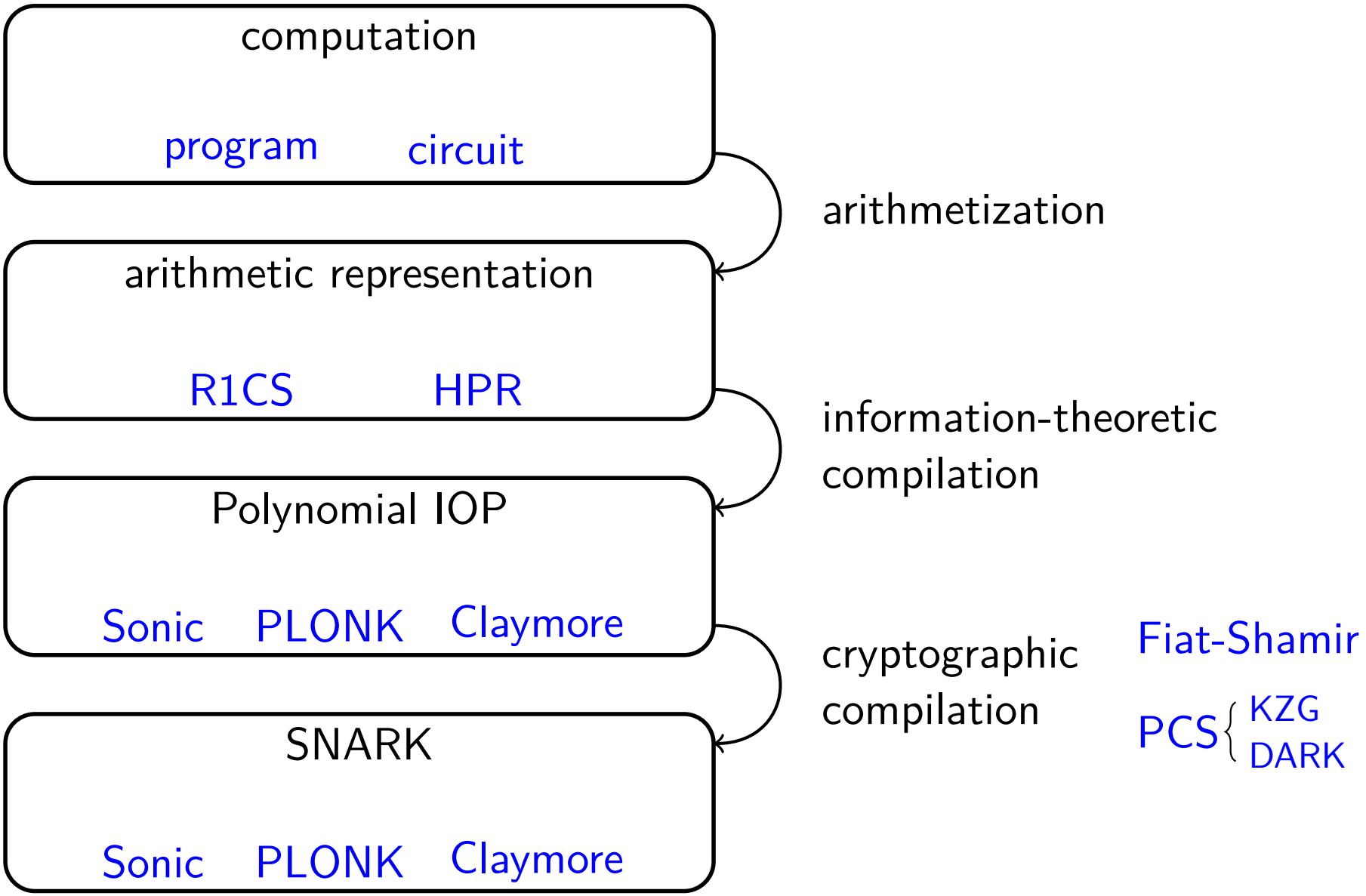
Polynomial IOP



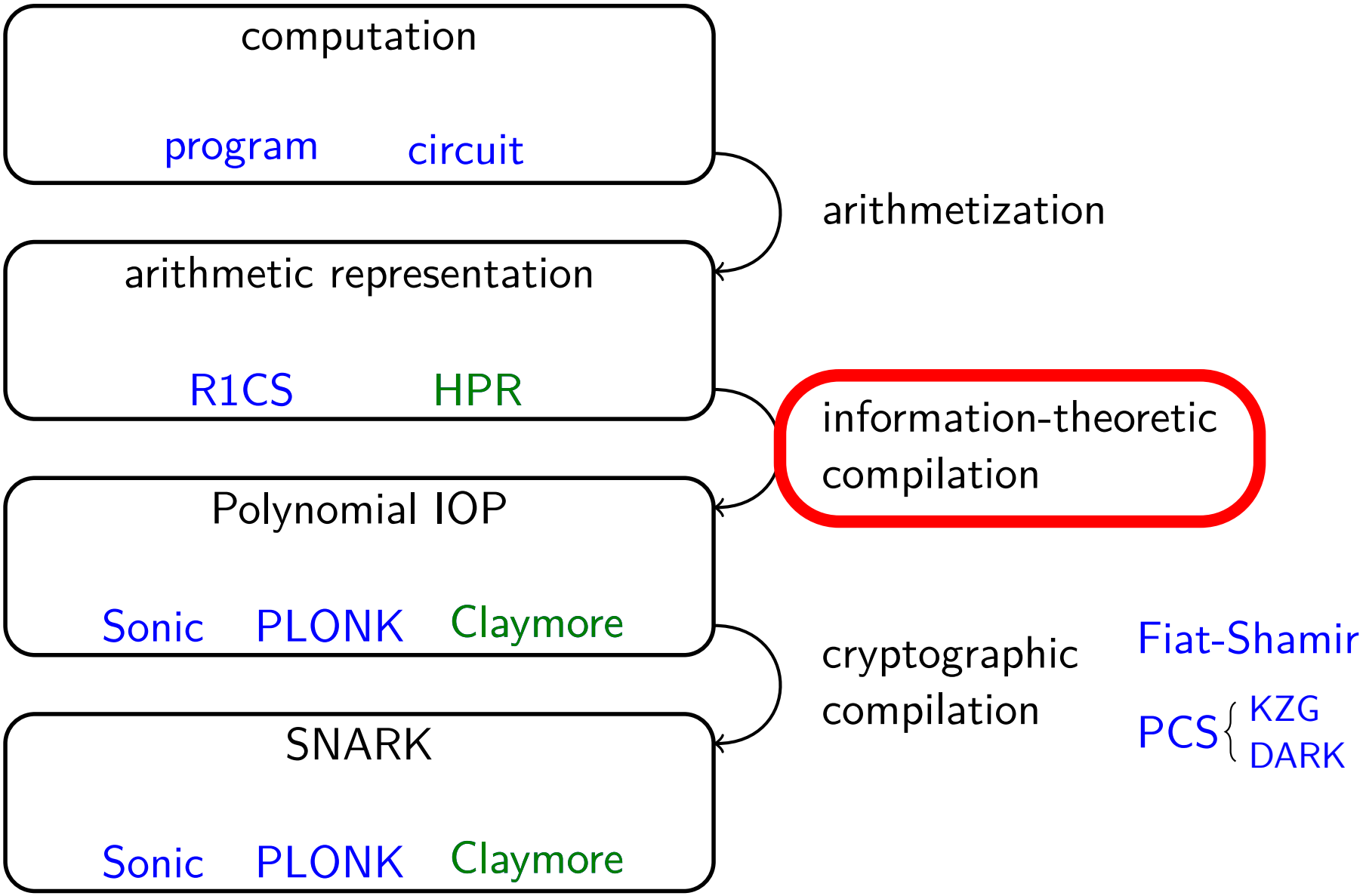
Construction Pipeline



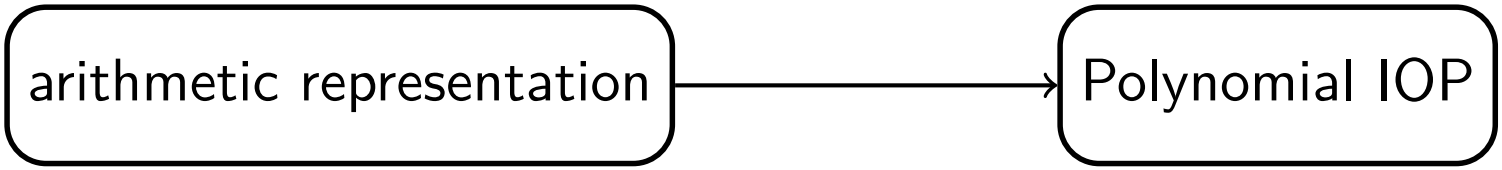
Construction Pipeline



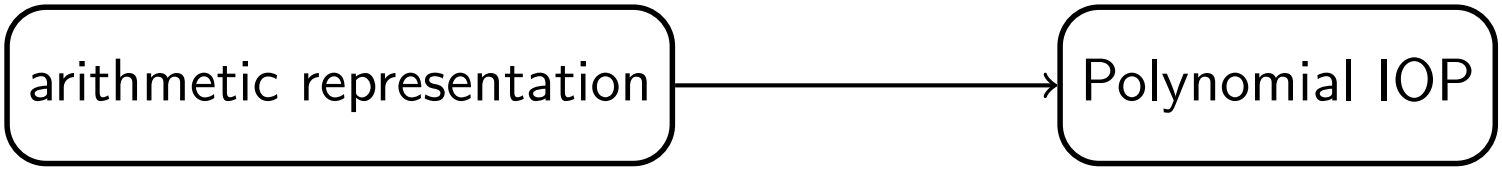
Construction Pipeline



Information-Theoretic Compilation



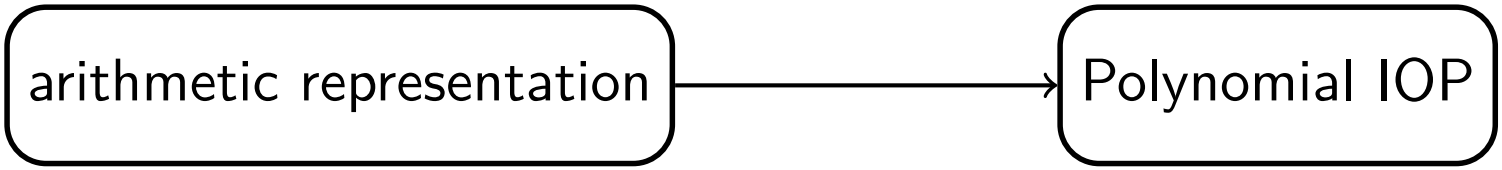
Information-Theoretic Compilation



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

linear algebra

Information-Theoretic Compilation



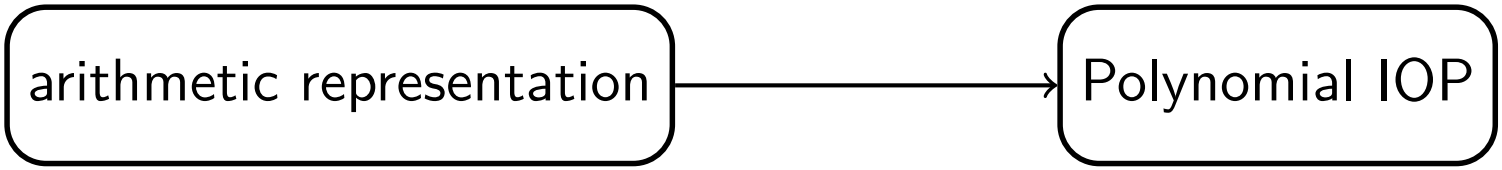
$$\boldsymbol{y} = M\boldsymbol{x}$$
$$\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c}$$

linear algebra

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

polynomial algebra

Information-Theoretic Compilation



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

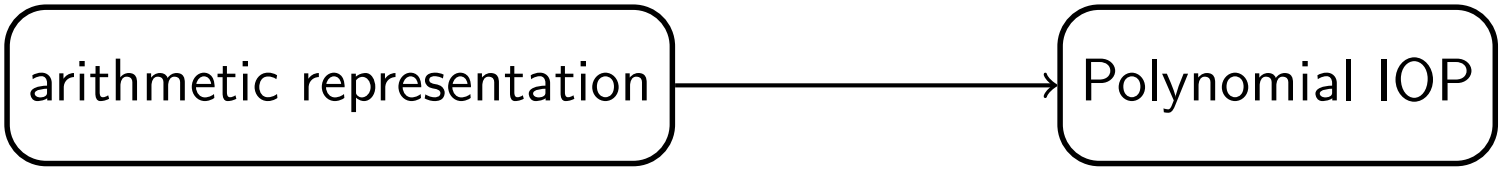
???

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

linear algebra

polynomial algebra

Information-Theoretic Compilation



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

???

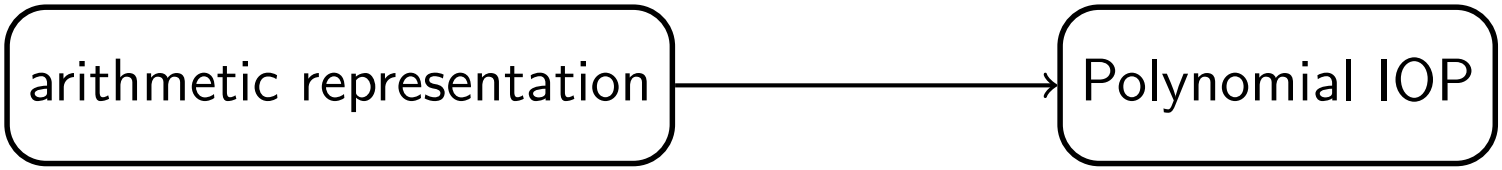
$f(X^{-1}) = g(\alpha) \cdot h(X)$
$\deg(\{f, g, h\}) \leq \mathsf{d}$

linear algebra

polynomial algebra

representation

Information-Theoretic Compilation



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

??? ⟶

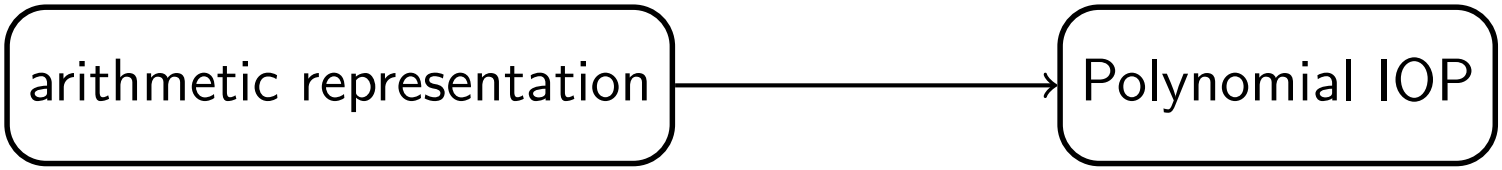
$f(X^{-1}) = g(\alpha) \cdot h(X)$
$\deg(\{f, g, h\}) \leq \mathsf{d}$

linear algebra — polynomial algebra

representation

- Reed-Solomon Codes — $f(D)$

Information-Theoretic Compilation



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

??? ⟶

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

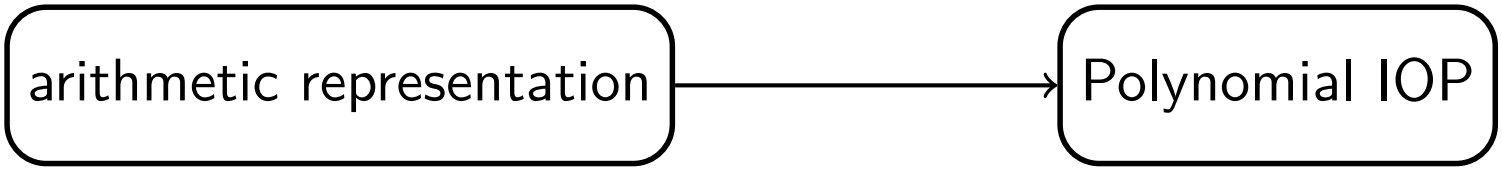
linear algebra

polynomial algebra

representation

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

Information-Theoretic Compilation



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

??? ⟶

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

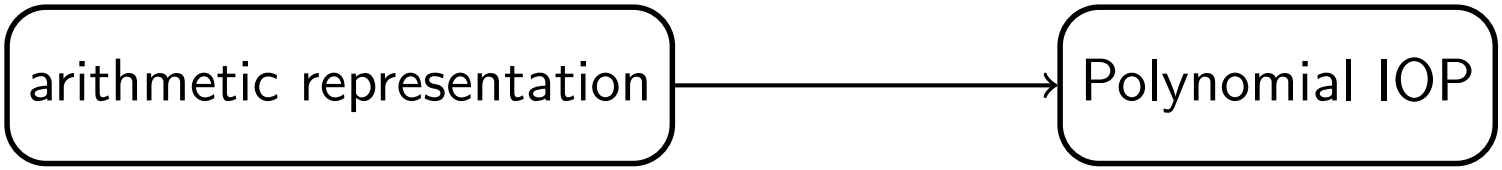
linear algebra — polynomial algebra

representation

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

operation ?

Information-Theoretic Compilation



$$\boldsymbol{y} = M\boldsymbol{x} \qquad \xrightarrow{\quad ??? \quad} \qquad f(X^{-1}) = g(\alpha) \cdot h(X)$$
$$\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c} \qquad\qquad \deg(\{f, g, h\}) \leq \mathsf{d}$$

linear algebra — 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
- $\mathbb{x} = \boldsymbol{x} \in \mathbb{F}^{m}$ — circuit outputs and inputs
- $\mathbb{w} = (\boldsymbol{w}_l, \boldsymbol{w}_r, \boldsymbol{w}_o) \in \mathbb{F}^{3n}$ — multiplication gate wires
- satisfied iff $\boldsymbol{w}_l \circ \boldsymbol{w}_r = \boldsymbol{w}_o$ and $\boldsymbol{x} = M \cdot \begin{pmatrix} 1 \\ \hline \boldsymbol{w}_l \\ \hline \boldsymbol{w}_r \\ \hline \boldsymbol{w}_o \end{pmatrix}$

[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.

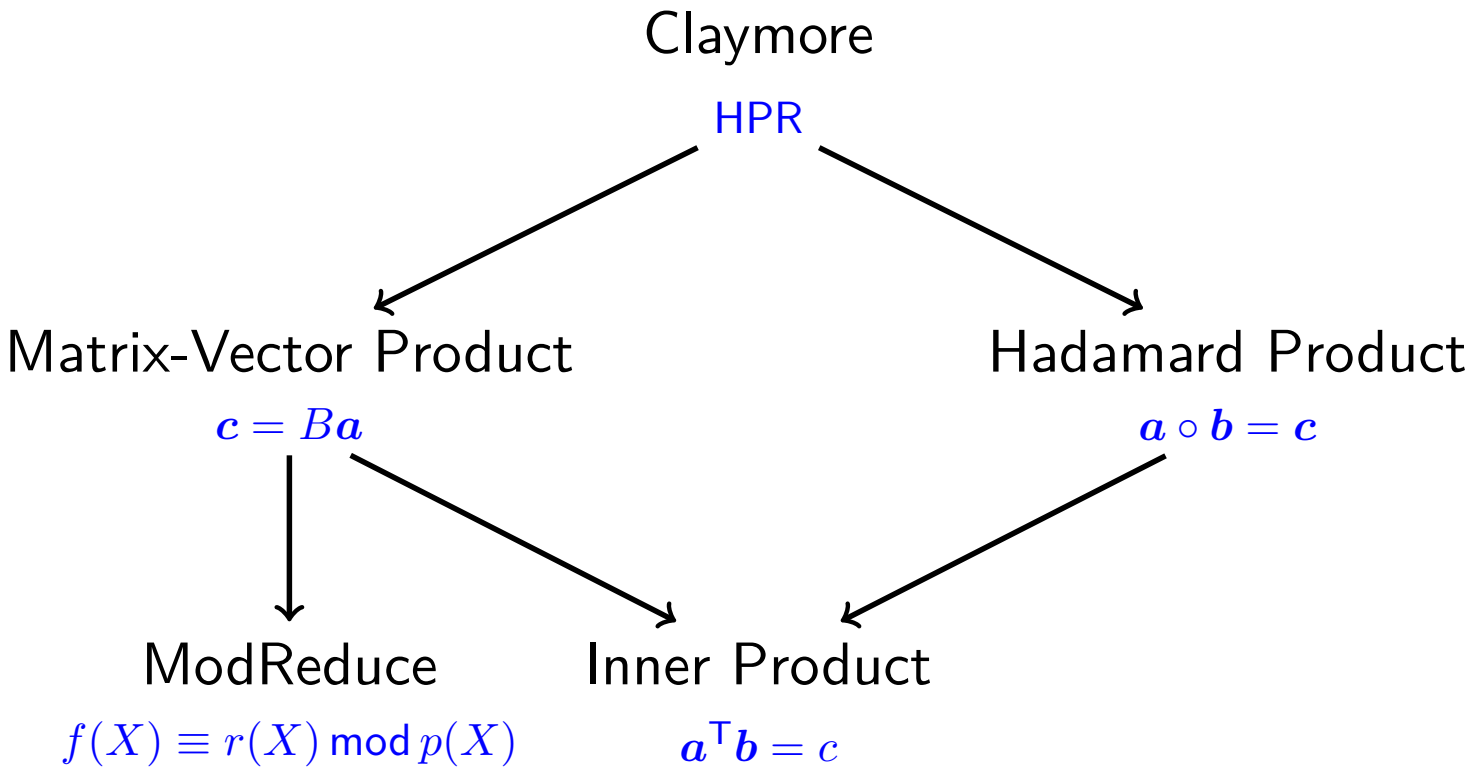
Representation

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

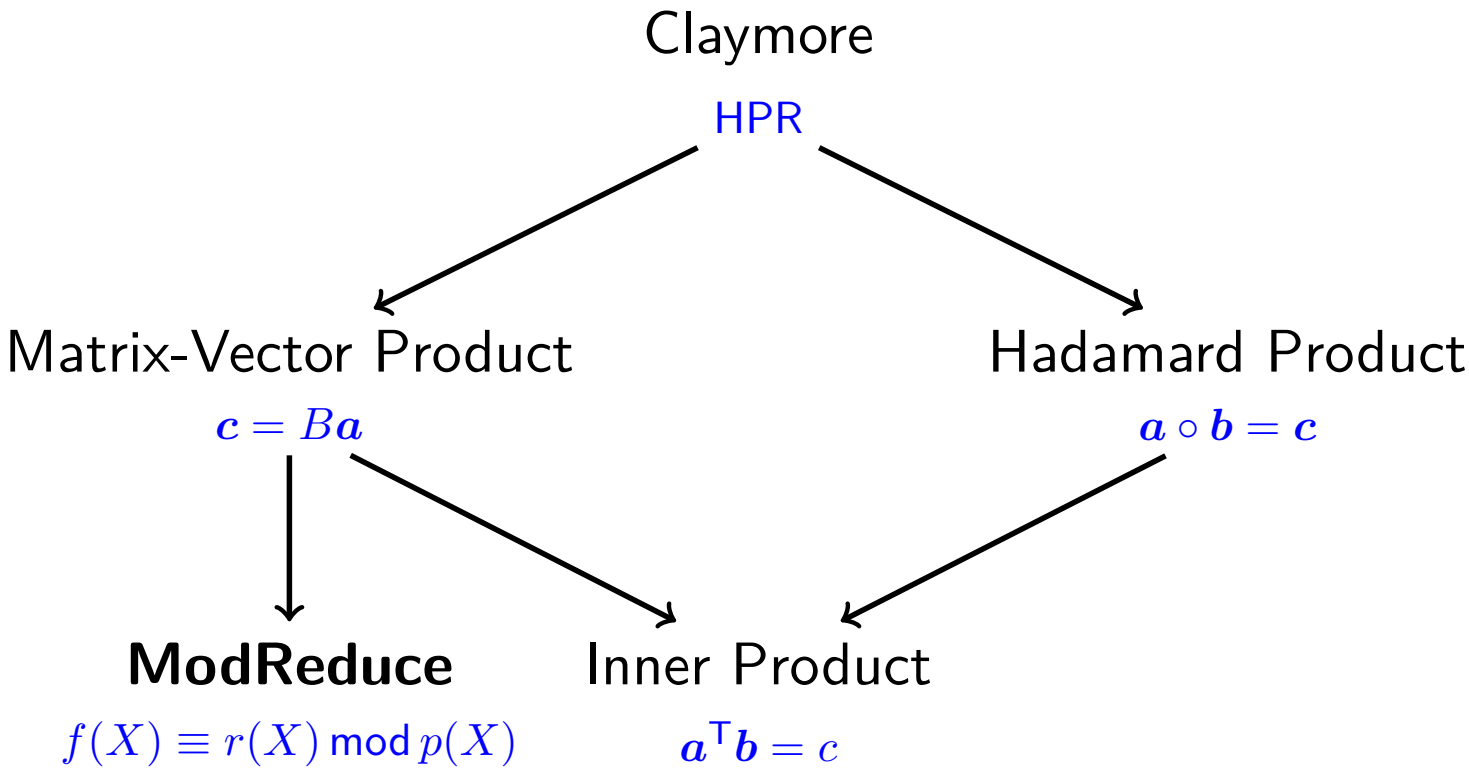
$$B = \begin{pmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,n-1} \\ b_{1,0} & b_{1,1} & \cdots & b_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m-1,0} & b_{m-1,1} & \cdots & b_{m-1,n-1} \end{pmatrix}$$

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

Subprotocols



Subprotocols



Mod Reduce

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

Verifier knows $p(X)$

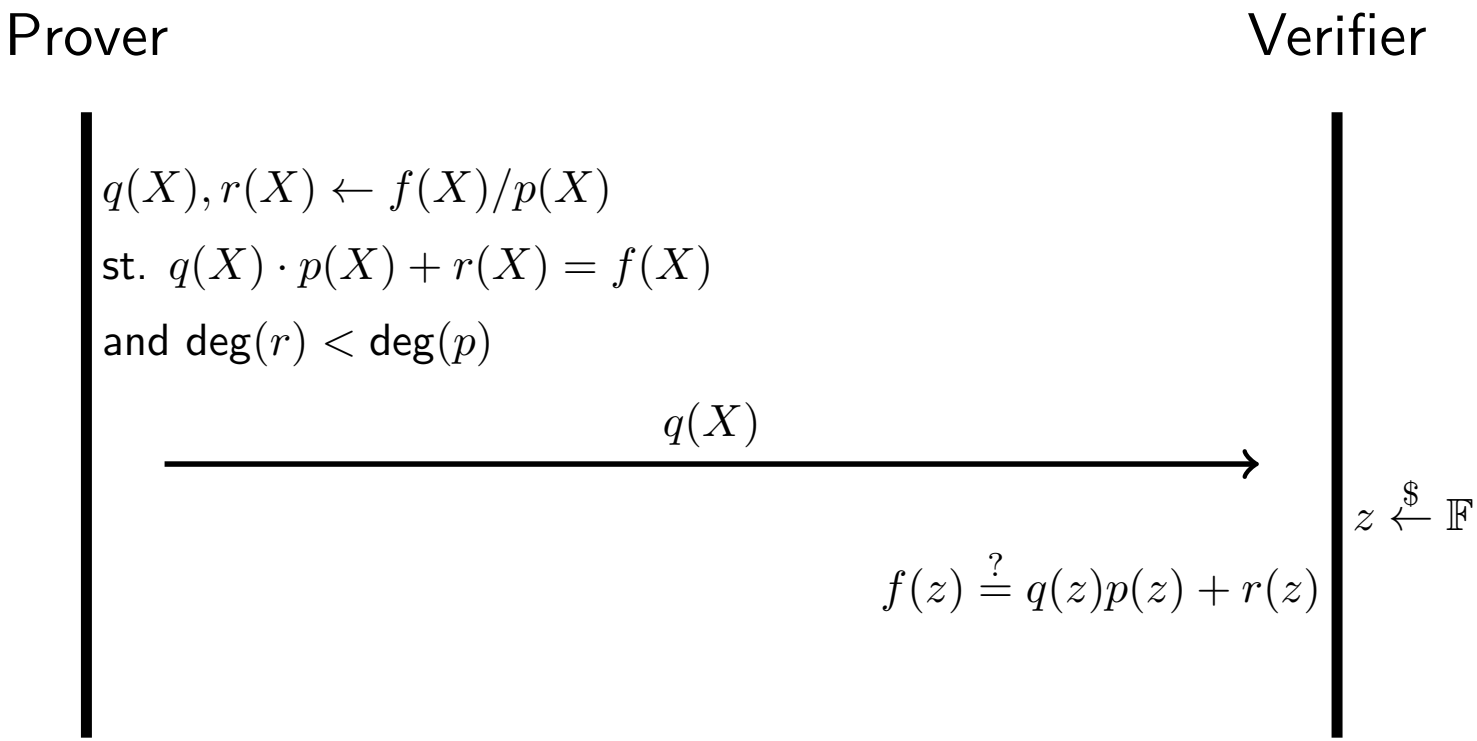
claim: $f(X) \equiv r(X) \bmod p(X)$

Mod Reduce

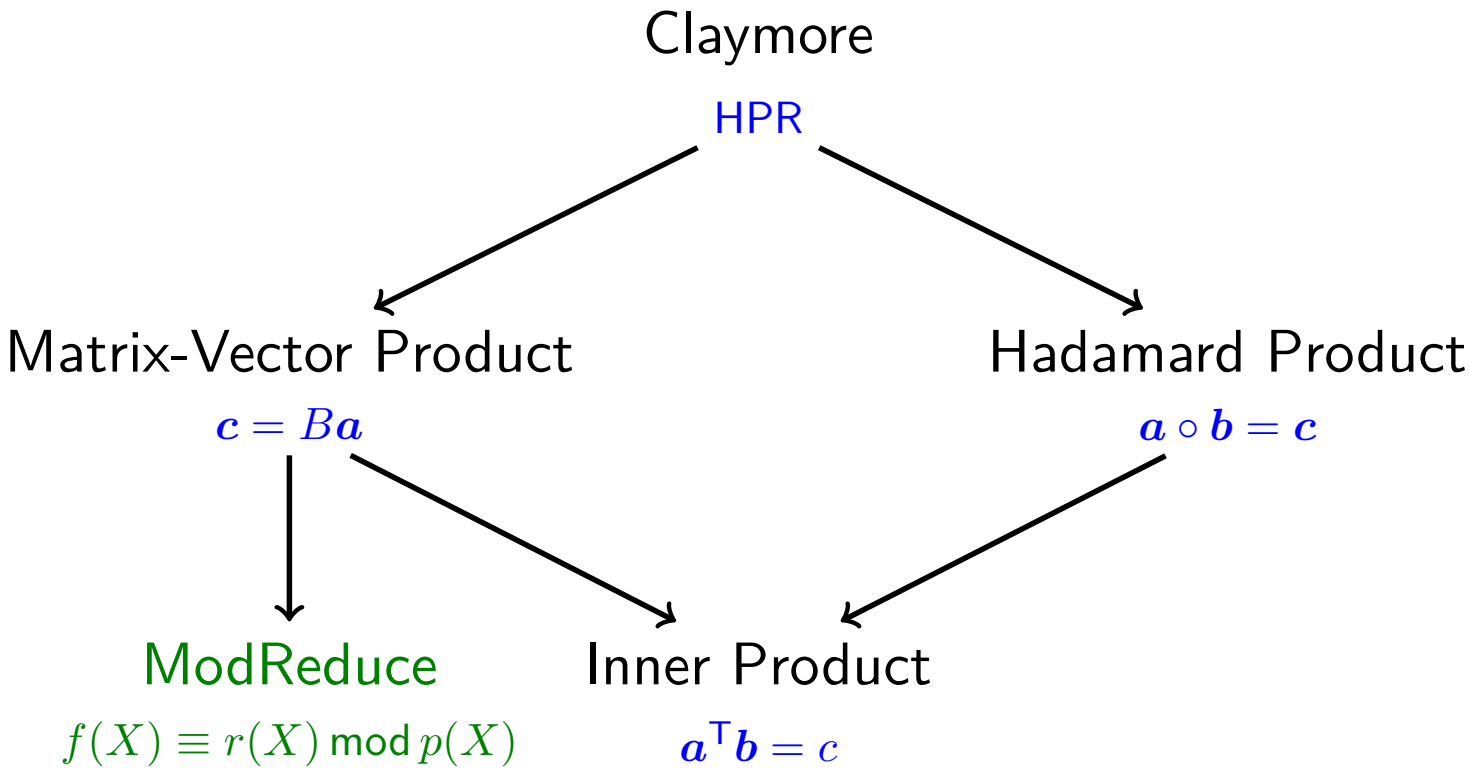
Verifier has oracles $[f(X)], [r(X)]$

Verifier knows $p(X)$

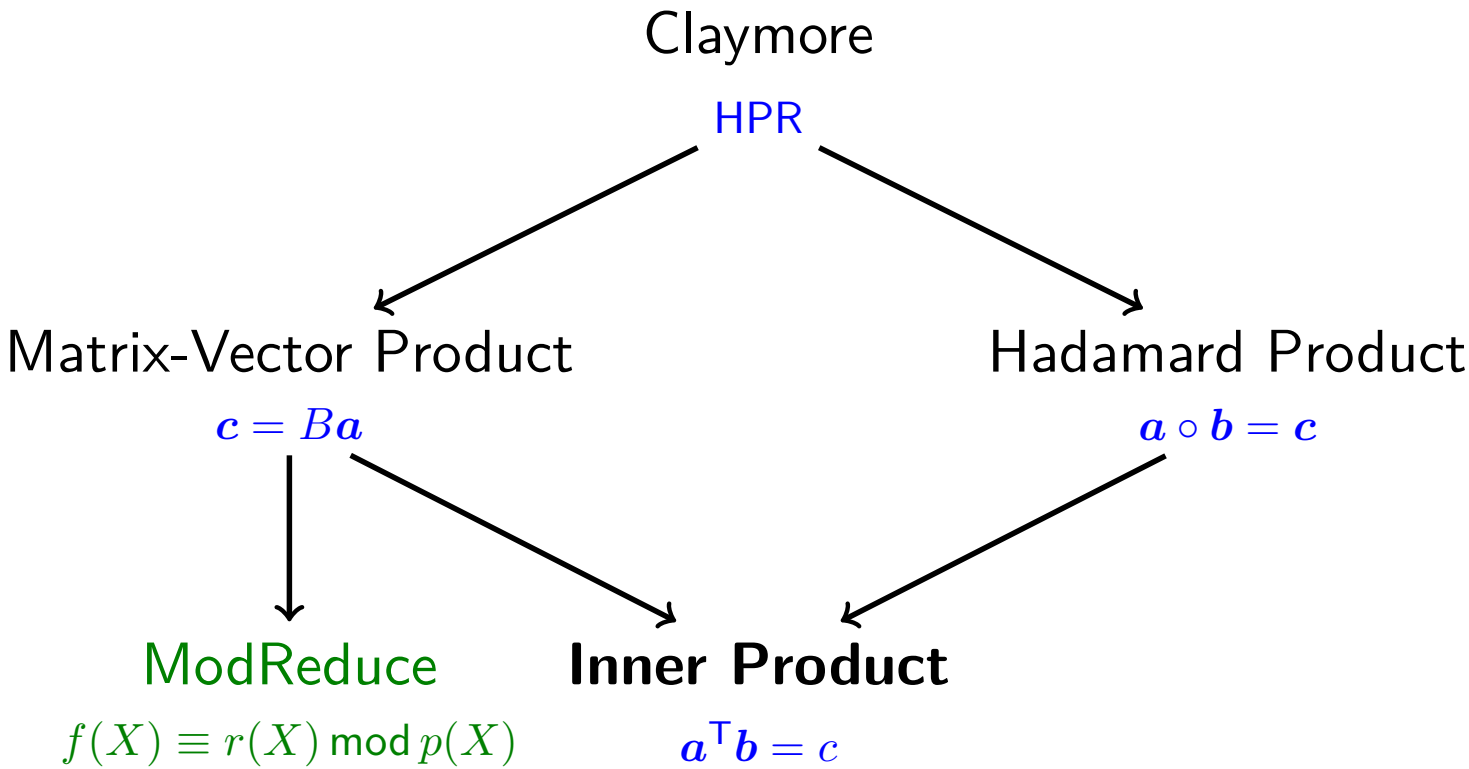
claim: $f(X) \equiv r(X) \bmod p(X)$



Subprotocols



Subprotocols



Inner Product

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

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

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

Inner Product

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

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

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

Inner Product

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

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

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

Prover

Verifier

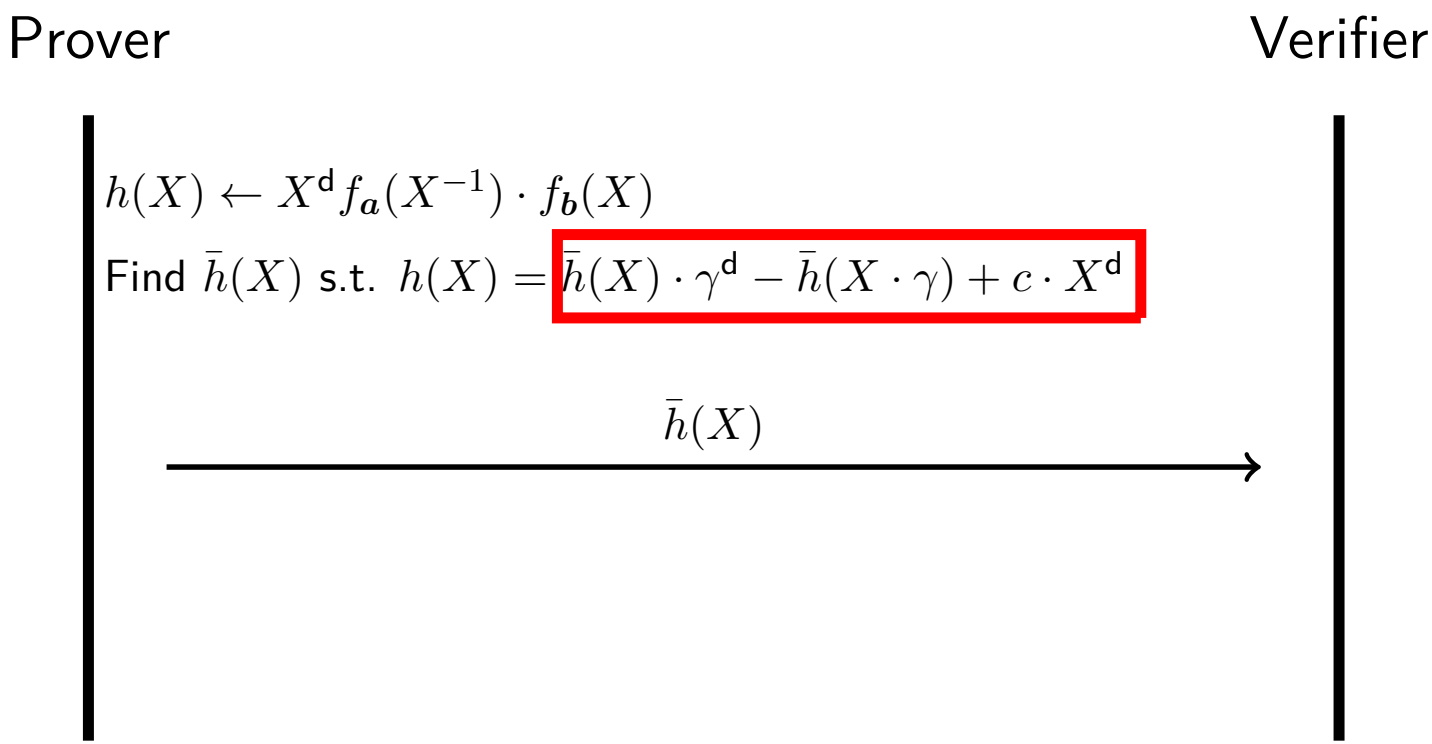
$h(X) \leftarrow X^{\mathsf{d}} f_{\boldsymbol{a}}(X^{-1}) \cdot f_{\boldsymbol{b}}(X)$

Inner Product

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

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

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

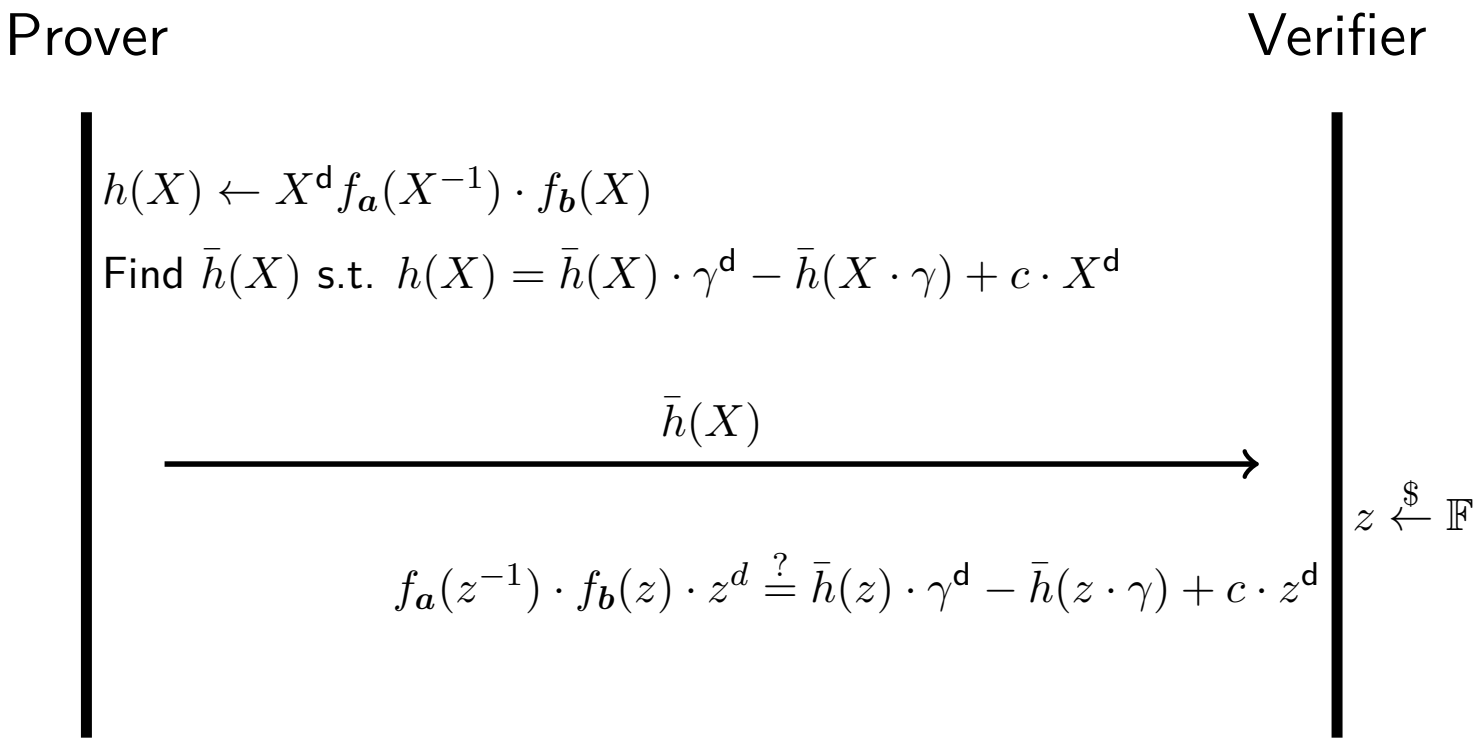


Inner Product

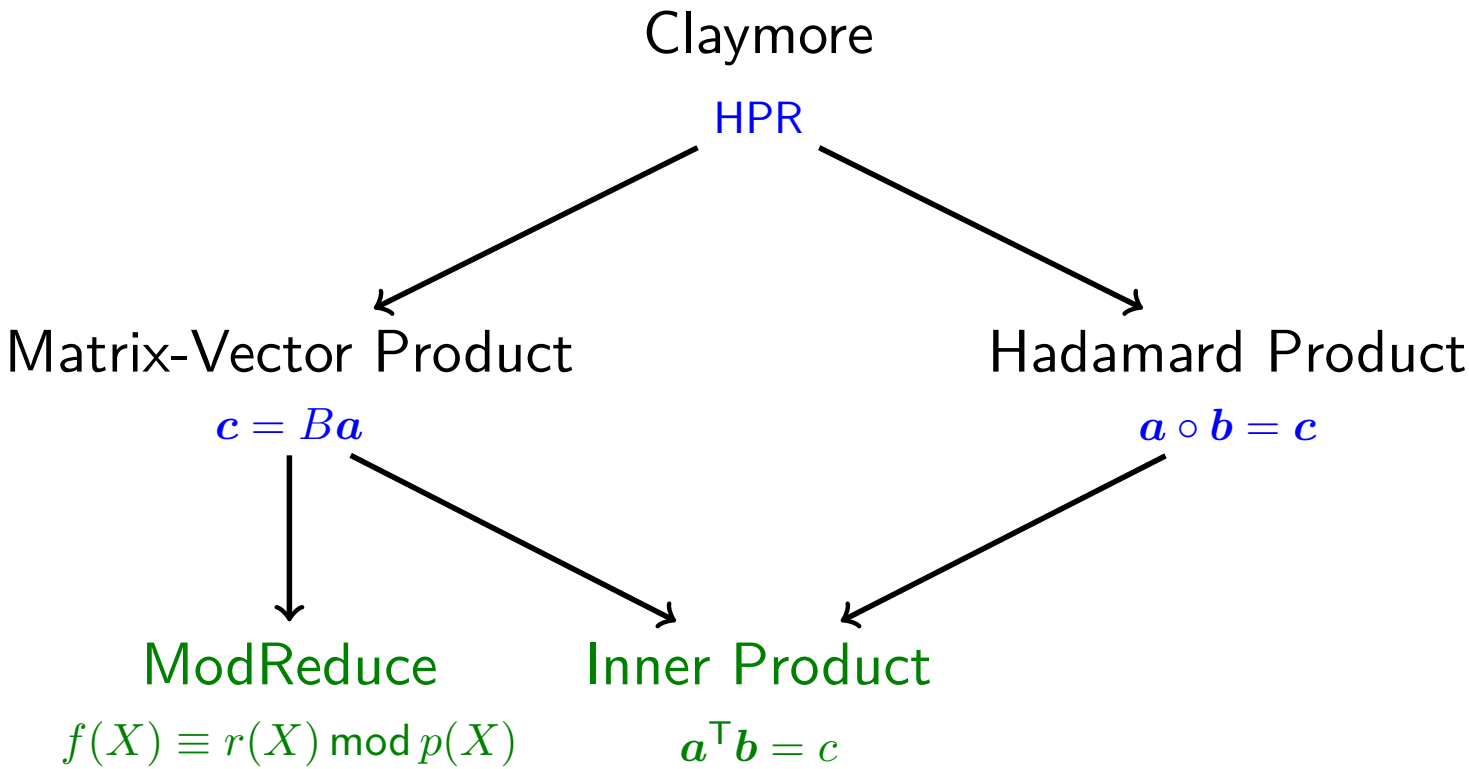
Verifier has oracles $[f_{\boldsymbol{a}}(X)], [f_{\boldsymbol{b}}(X)]$

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

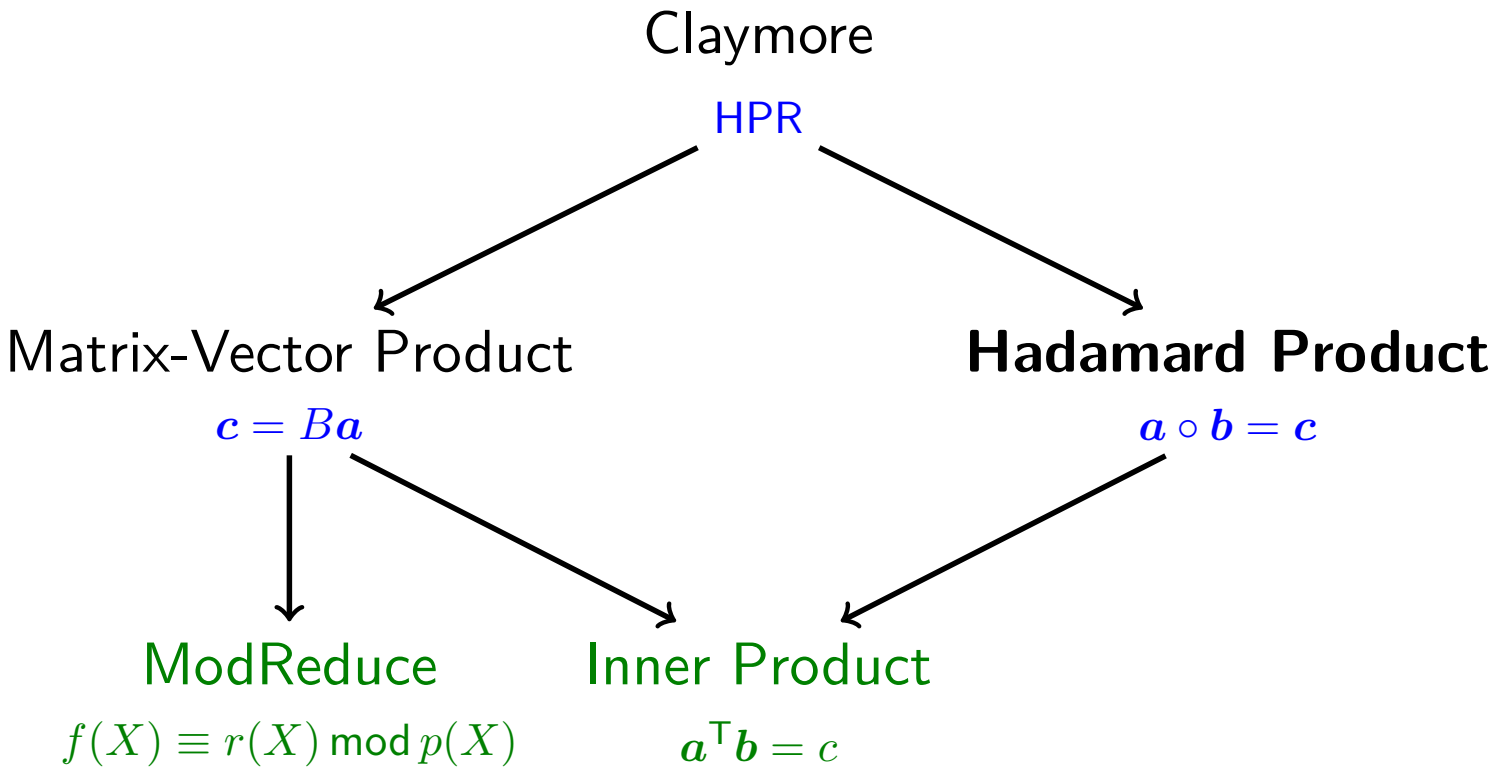
claim: $\boldsymbol{a}^{\mathsf{T}}\boldsymbol{b} = c$



Subprotocols



Subprotocols



Hadamard

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

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

Hadamard

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

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

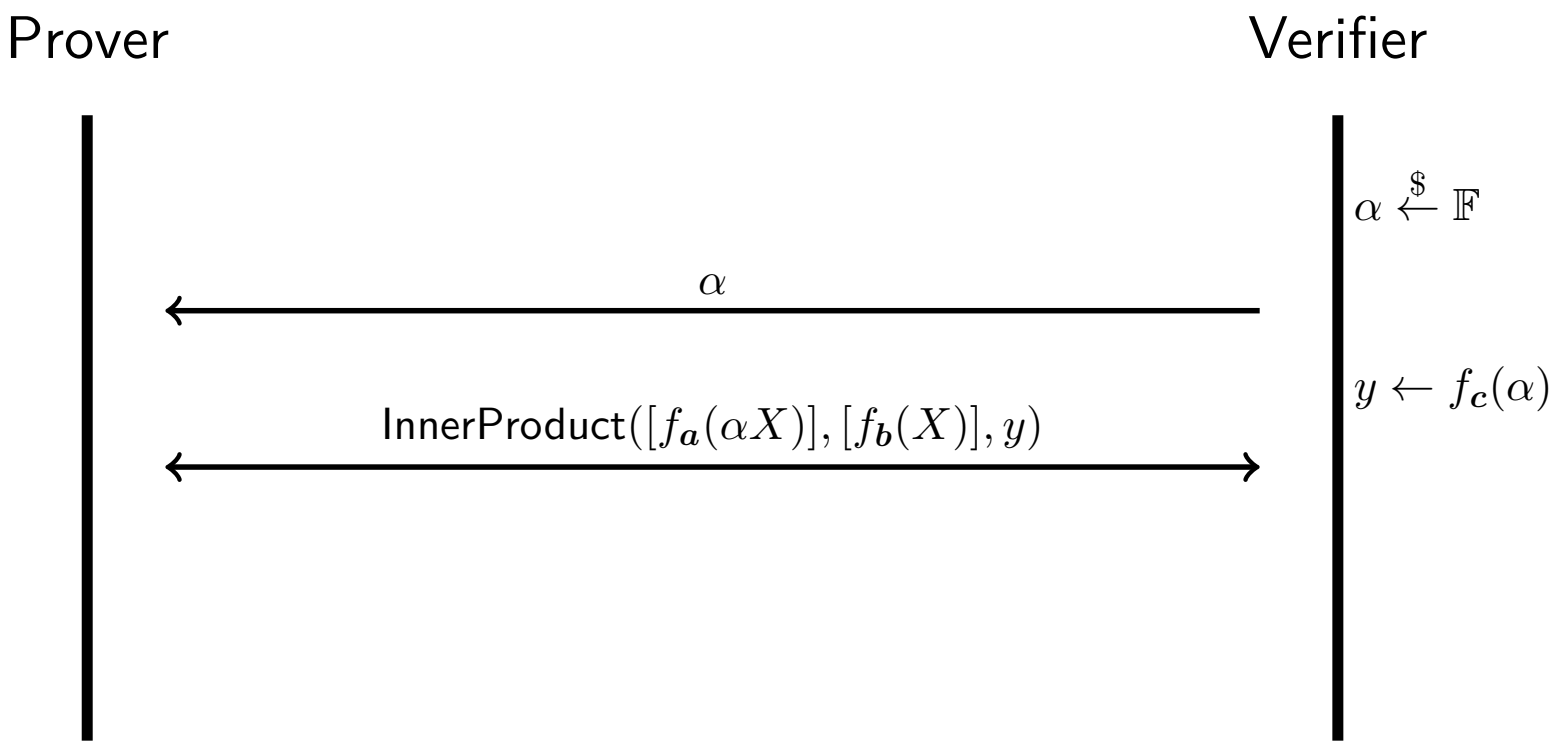
idea: $f_{\boldsymbol{a} \circ \boldsymbol{b}}(\alpha) = f_{\boldsymbol{c}}(\alpha)$

Hadamard

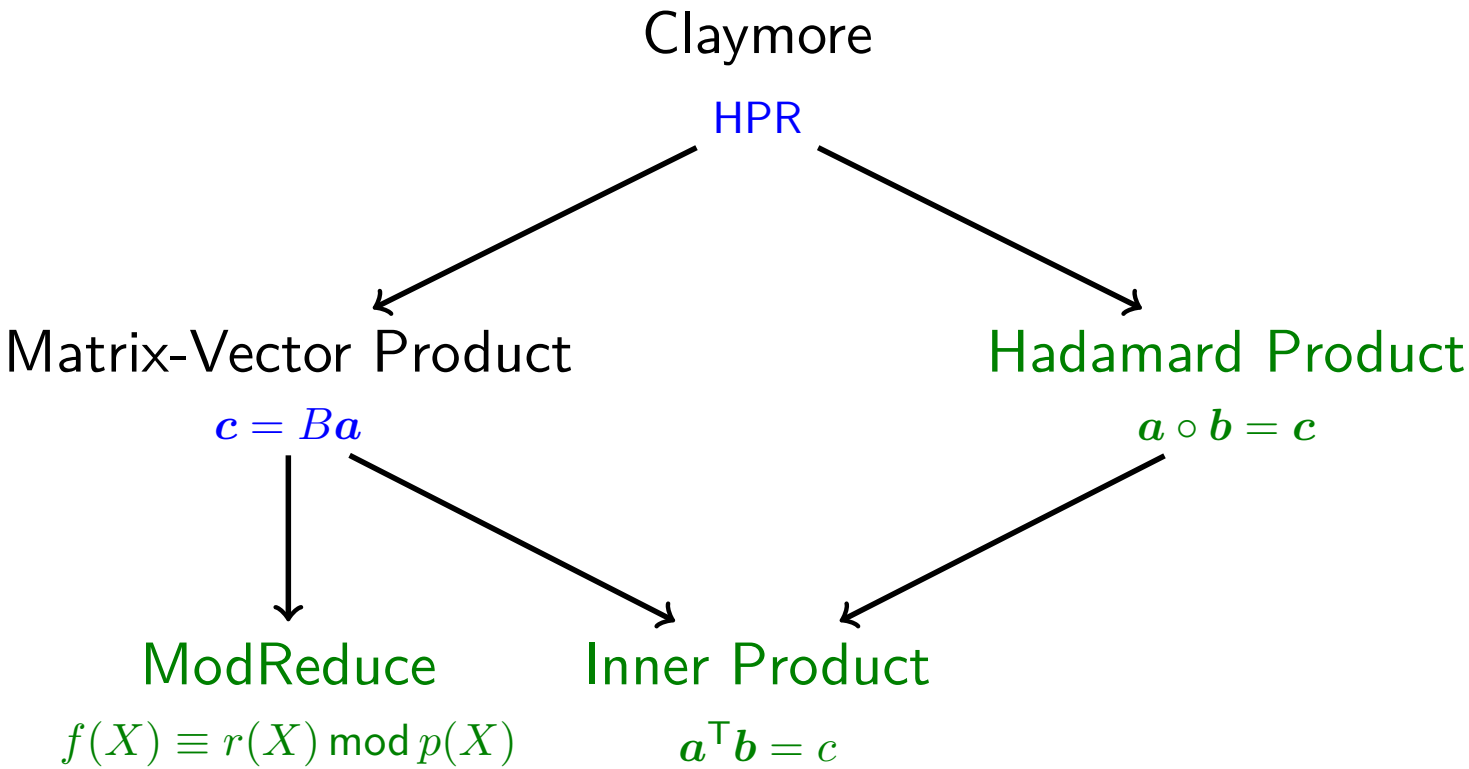
Verifier has oracles $[f_{\boldsymbol{a}}(X)], [f_{\boldsymbol{b}}(X)], [f_{\boldsymbol{c}}(X)]$

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

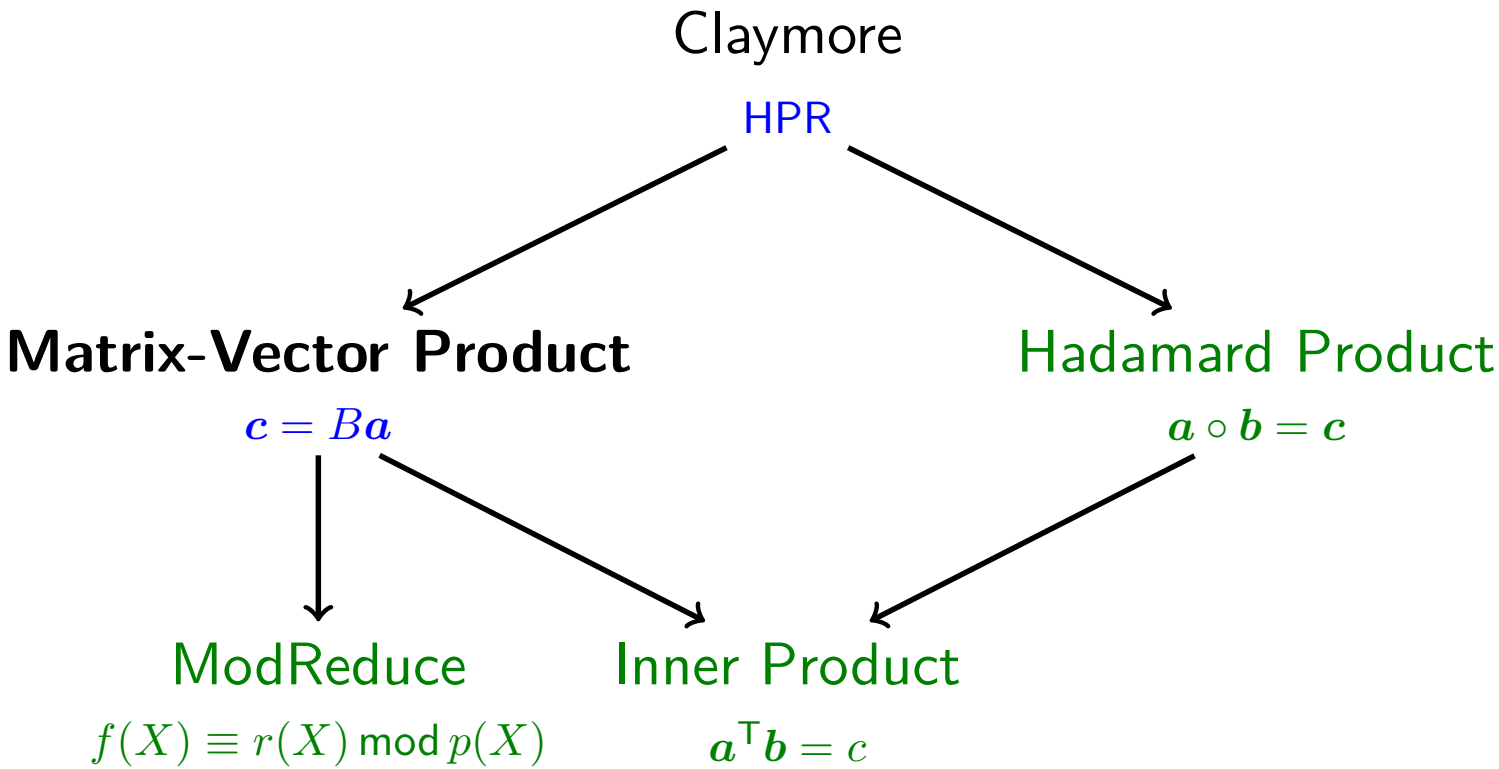
idea: $f_{\boldsymbol{a}\circ\boldsymbol{b}}(\alpha) = f_{\boldsymbol{c}}(\alpha)$



Subprotocols



Subprotocols



Dense MVP

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

Dense MVP

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

reduce modulo $X^n - \alpha$

Dense MVP

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

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

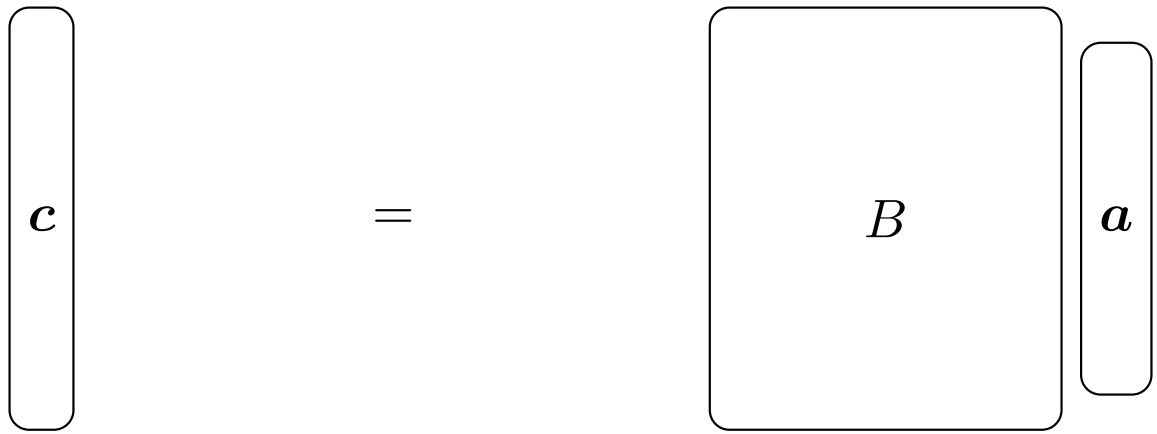
$$
\begin{aligned}
f_B(X) = \quad & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\
& b_{1,0}X^n + b_{1,1}X^{n+1} + \cdots + b_{1,n-1}X^{2n-1} \qquad \curvearrowleft \times \tfrac{\alpha}{X^n} \\
& \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots \\
& b_{m-1,0}X^{(m-1)n}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

reduce modulo $X^n - \alpha$

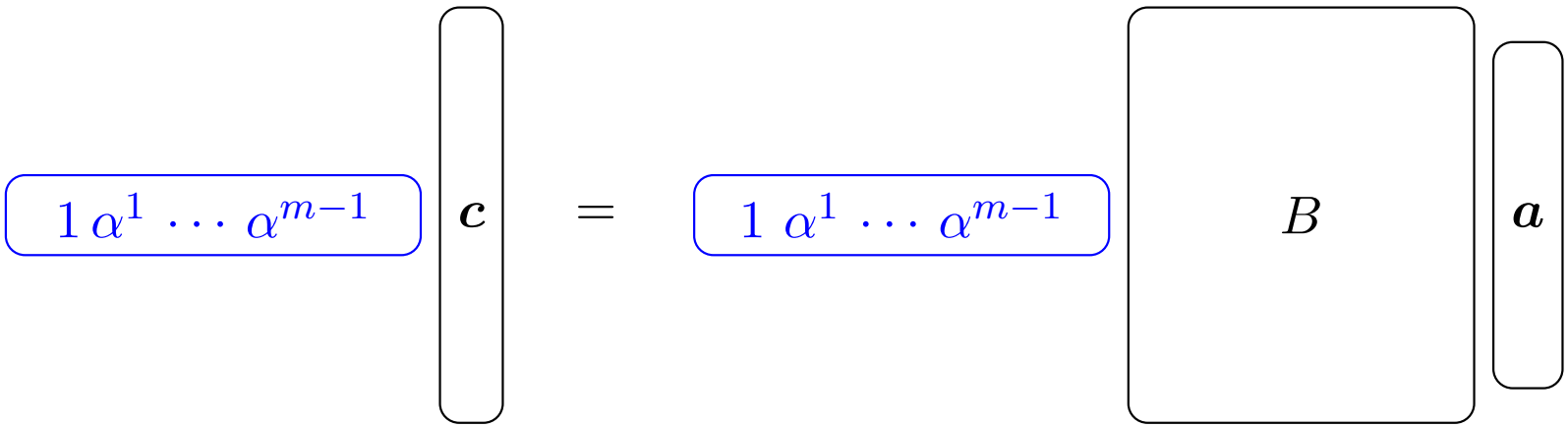
$$
\begin{aligned}
r(X) = \quad & b_{0,0} + b_{0,1}X + b_{0,2}X^2 + \cdots + b_{0,n-1}X^{n-1} \\
& (b_{1,0} + b_{1,1}X + b_{1,2}X^2 + \cdots + b_{1,n-1}X^{n-1}) \cdot \alpha \\
& \vdots \\
& (b_{m-1,0} + b_{m-1,1}X + \cdots + b_{m-1,n-1}X^{n-1}) \cdot \alpha^{m-1}
\end{aligned}
$$

$\boldsymbol{r} = \boldsymbol{\alpha}^{\mathsf{T}} B$, where $\boldsymbol{\alpha} = (1, \alpha, \alpha^2, \cdots, \alpha^{m-1})$

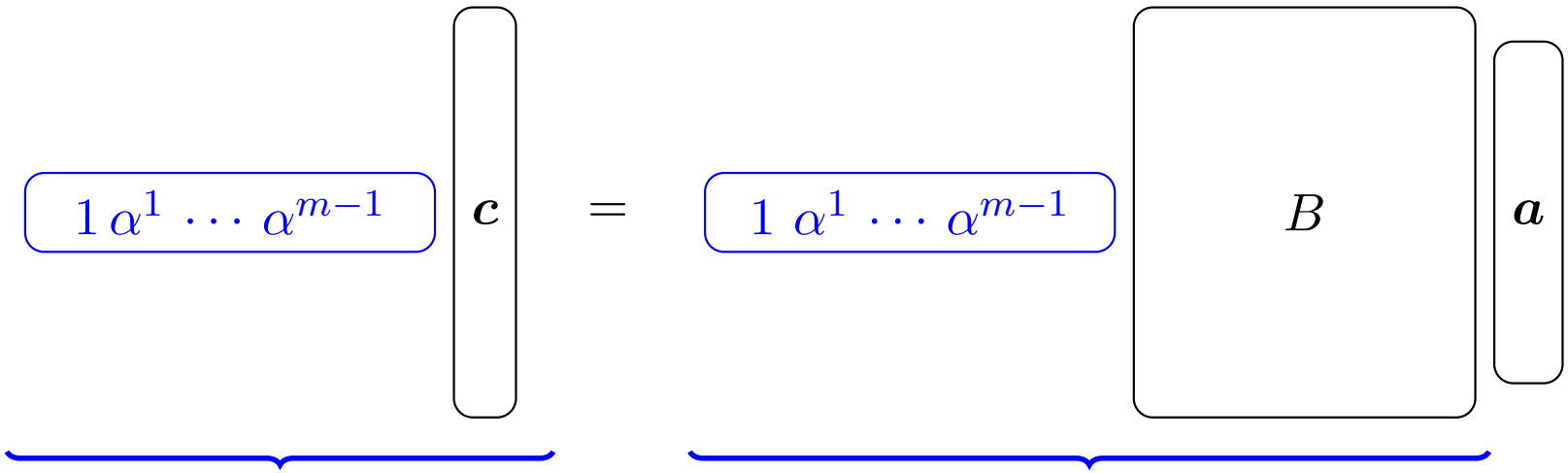
Dense MVP



Dense MVP


$1\,\alpha^1\,\cdots\,\alpha^{m-1}$
$\boldsymbol{c}$
$=$
$1\,\alpha^1\,\cdots\,\alpha^{m-1}$
B
$\boldsymbol{a}$

Dense MVP


$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
$\boldsymbol{c}$
$=$
$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
B
$\boldsymbol{a}$
$f_{\boldsymbol{c}}(\alpha)$
$r(X) = \ f_B(X) \bmod X^n - \alpha$
$\mathsf{InnerProduct}(r(X),\ f_{\boldsymbol{a}}(X), f_{\boldsymbol{c}}(\alpha))$

Dense MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)], [f_B(X)], [f_{\boldsymbol{a}}(X)]$

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

Dense MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)], [f_B(X)], [f_{\boldsymbol{a}}(X)]$

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

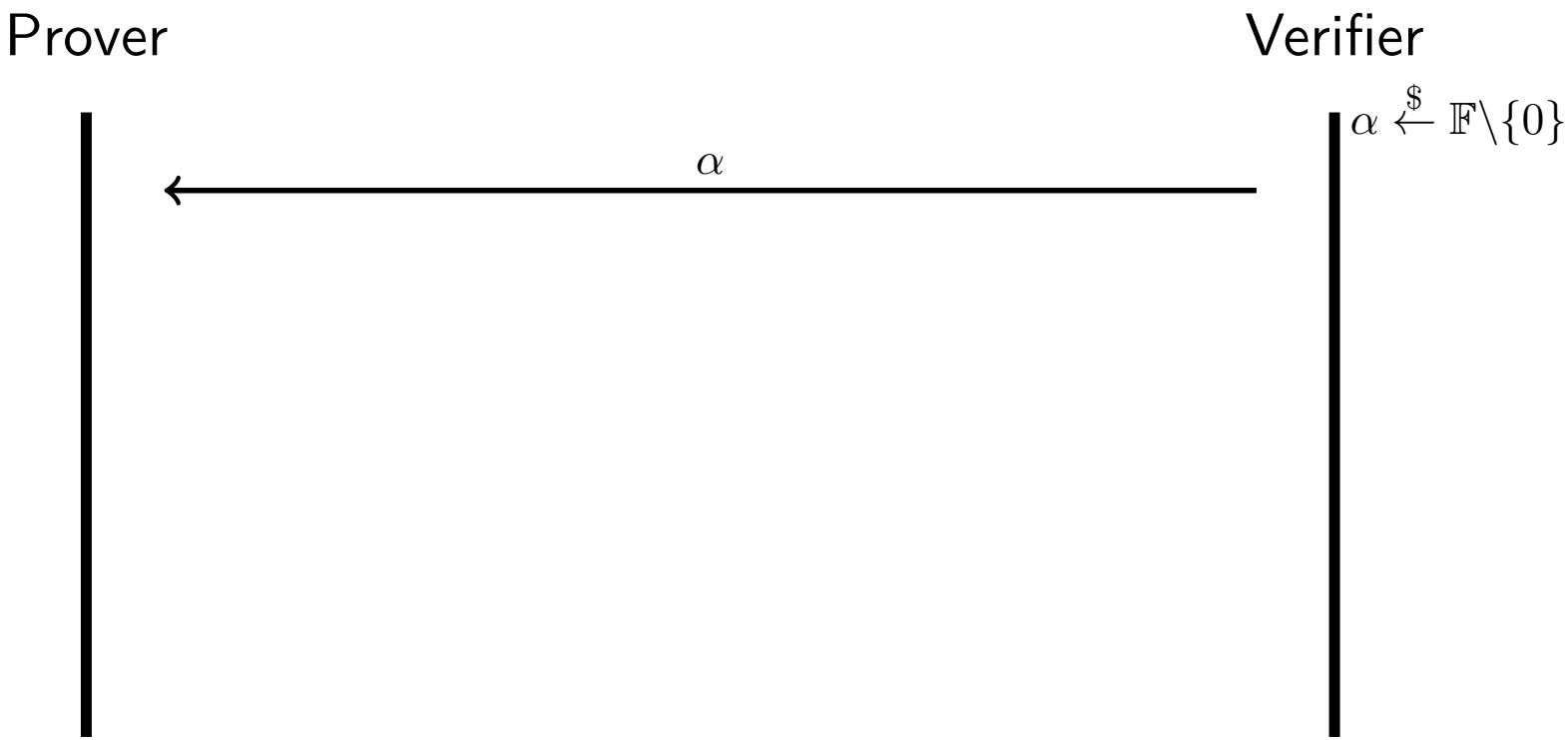
Prover

Verifier

Dense MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)], [f_B(X)], [f_{\boldsymbol{a}}(X)]$

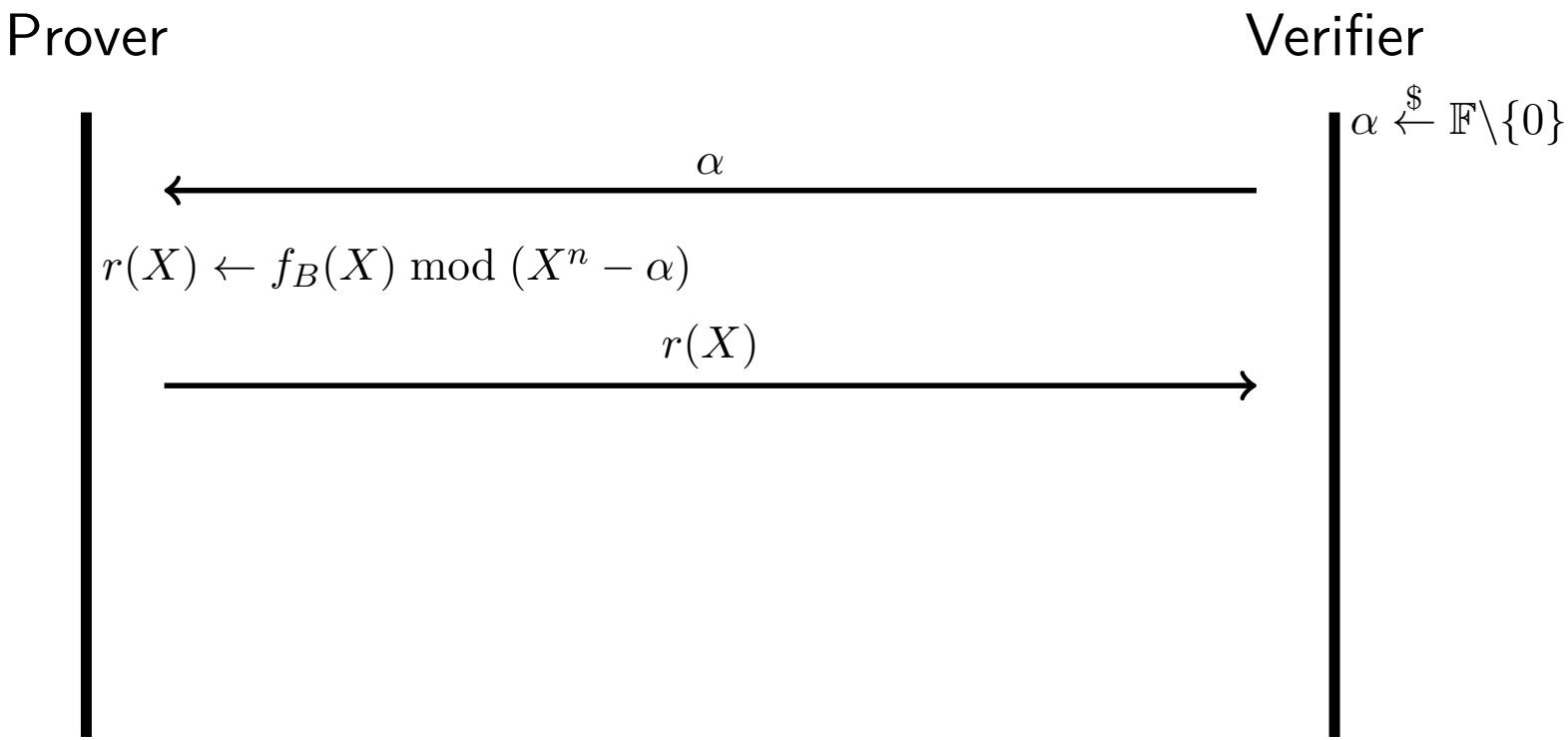
claim: $\boldsymbol{c} = B\boldsymbol{a}$



Dense MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)], [f_B(X)], [f_{\boldsymbol{a}}(X)]$

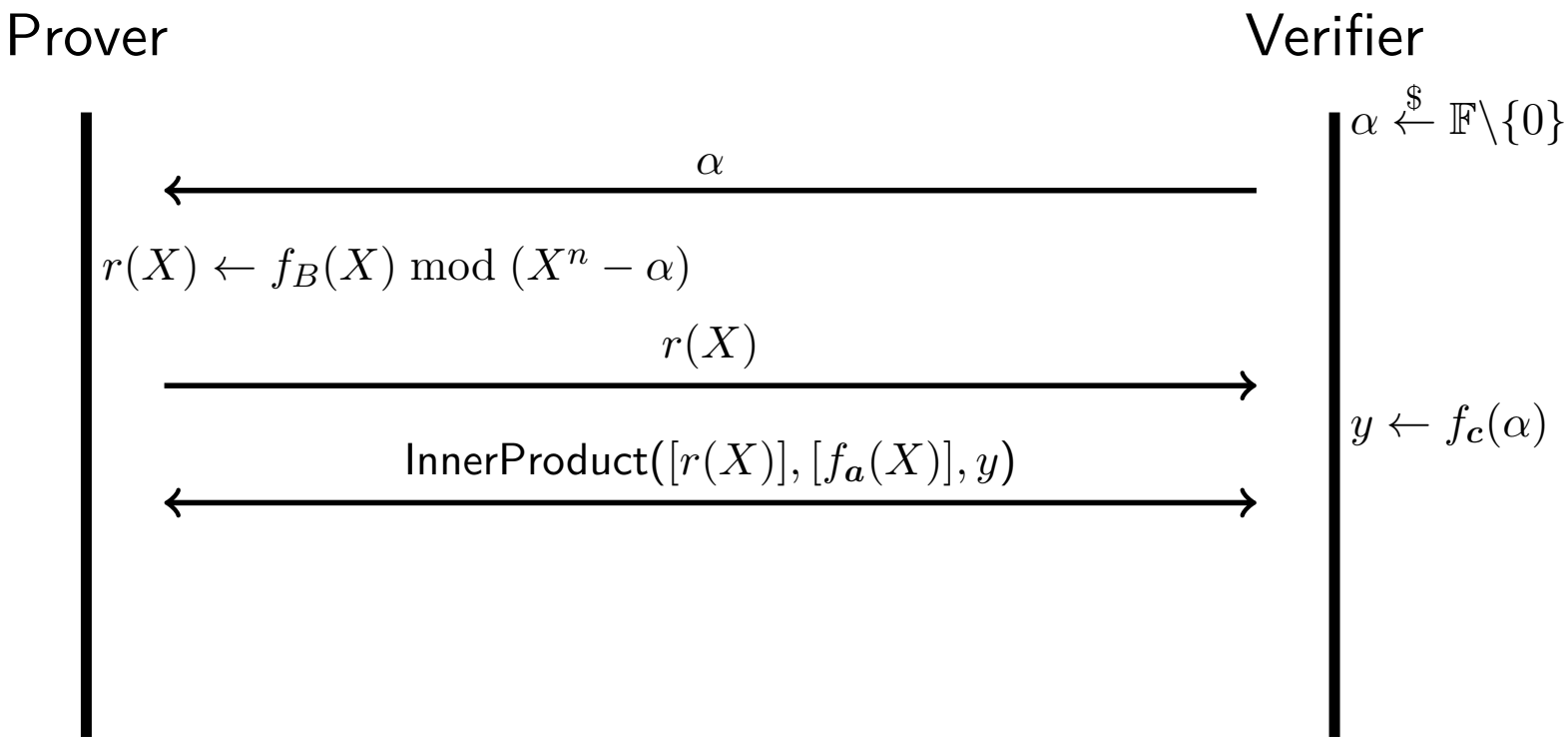
claim: $\boldsymbol{c} = B\boldsymbol{a}$



Dense MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)], [f_B(X)], [f_{\boldsymbol{a}}(X)]$

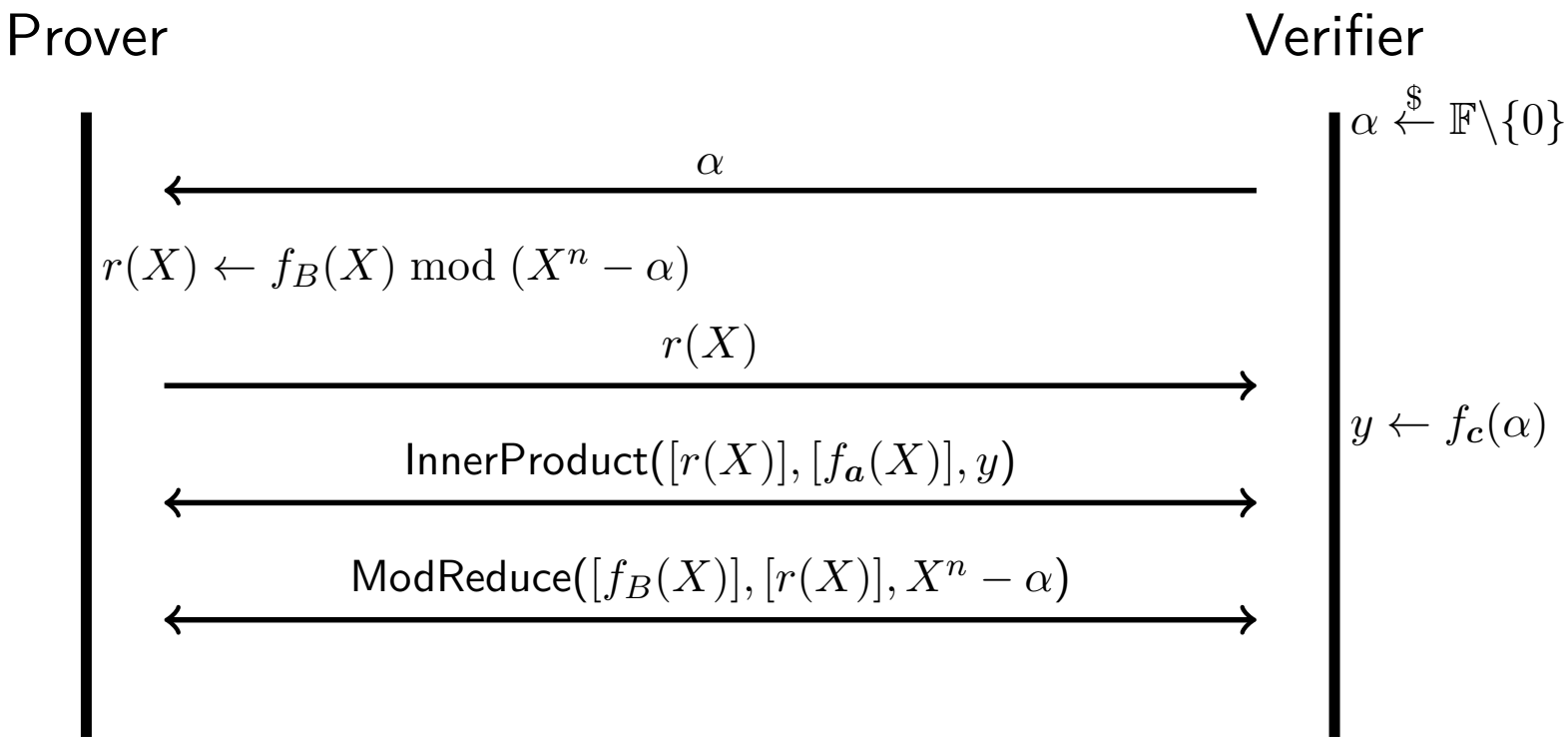
claim: $\boldsymbol{c} = B\boldsymbol{a}$



Dense MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)], [f_B(X)], [f_{\boldsymbol{a}}(X)]$

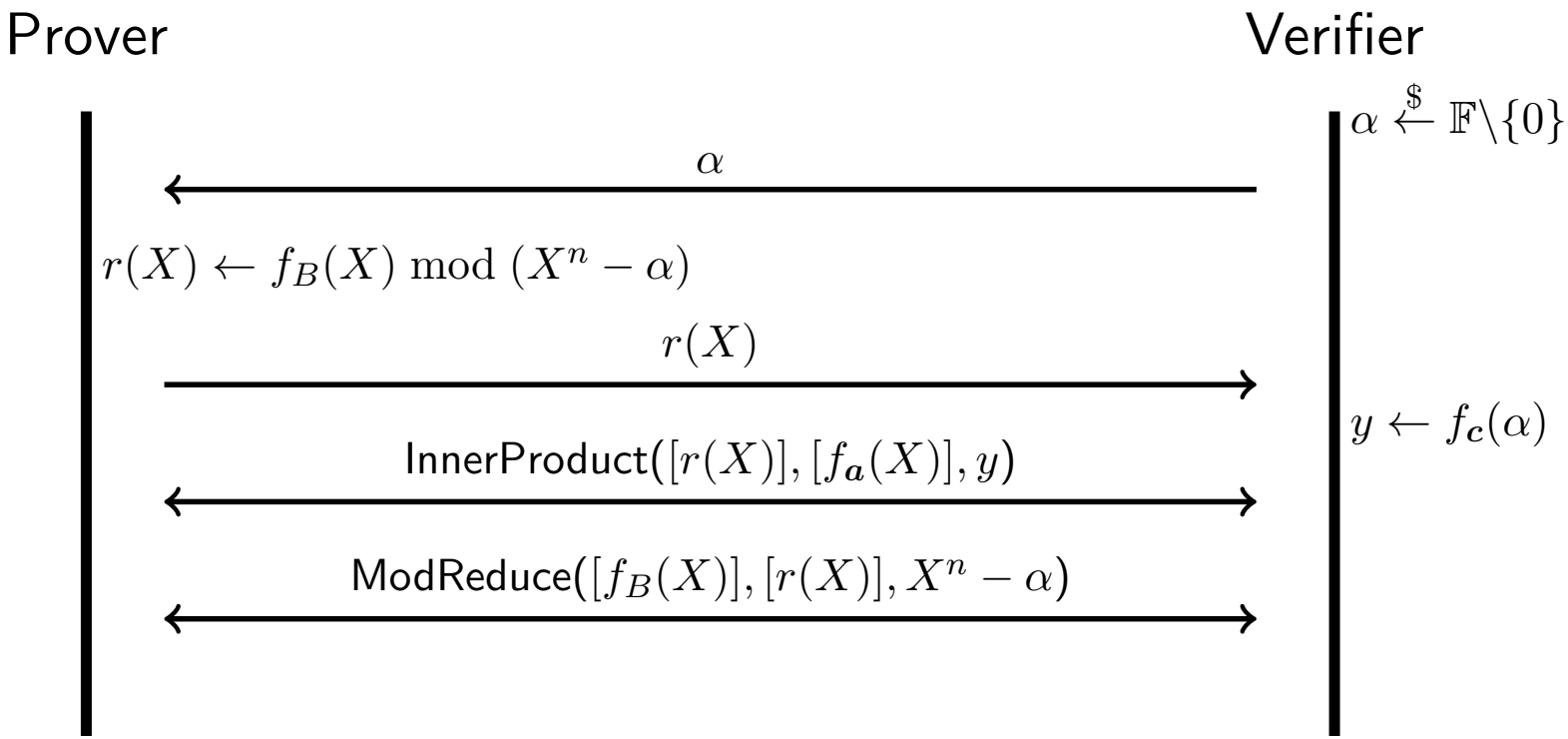
claim: $\boldsymbol{c} = B\boldsymbol{a}$



Dense MVP

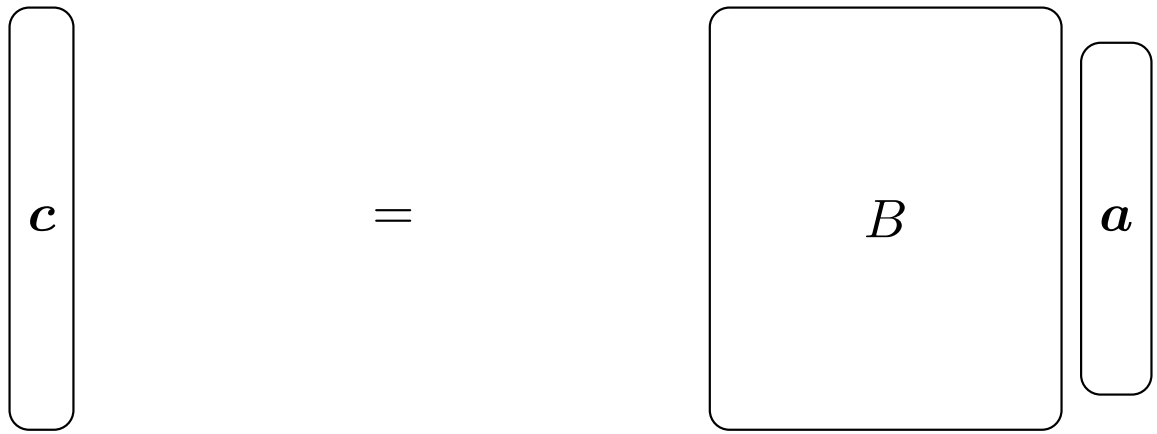
Verifier has oracles $[f_{\boldsymbol{c}}(X)], [f_B(X)], [f_{\boldsymbol{a}}(X)]$

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

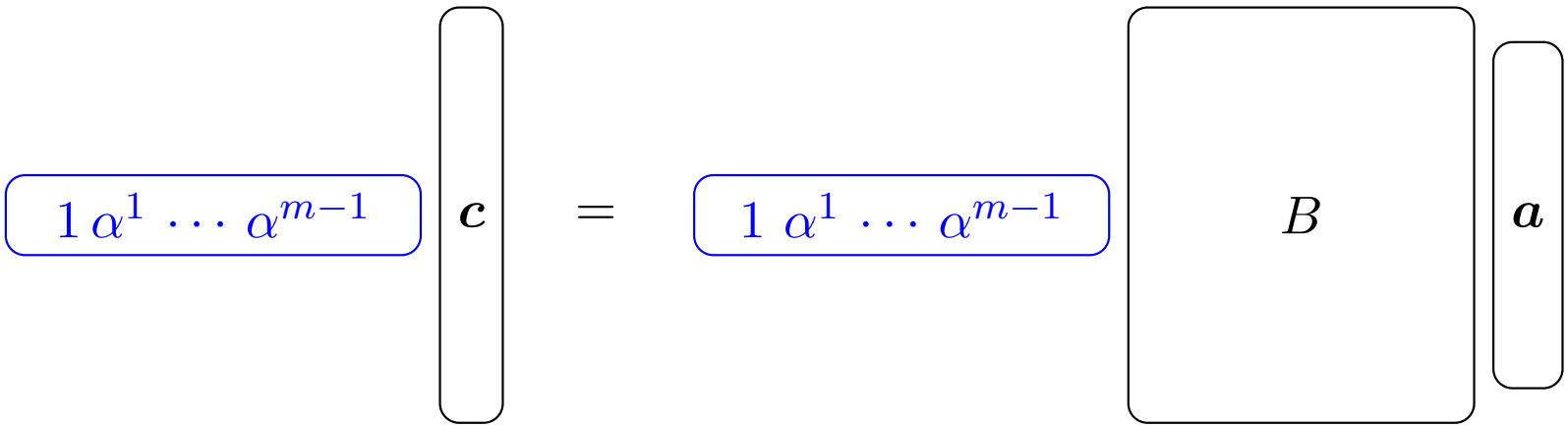


Prover complexity $= O(mn)$, even for sparse matrix!

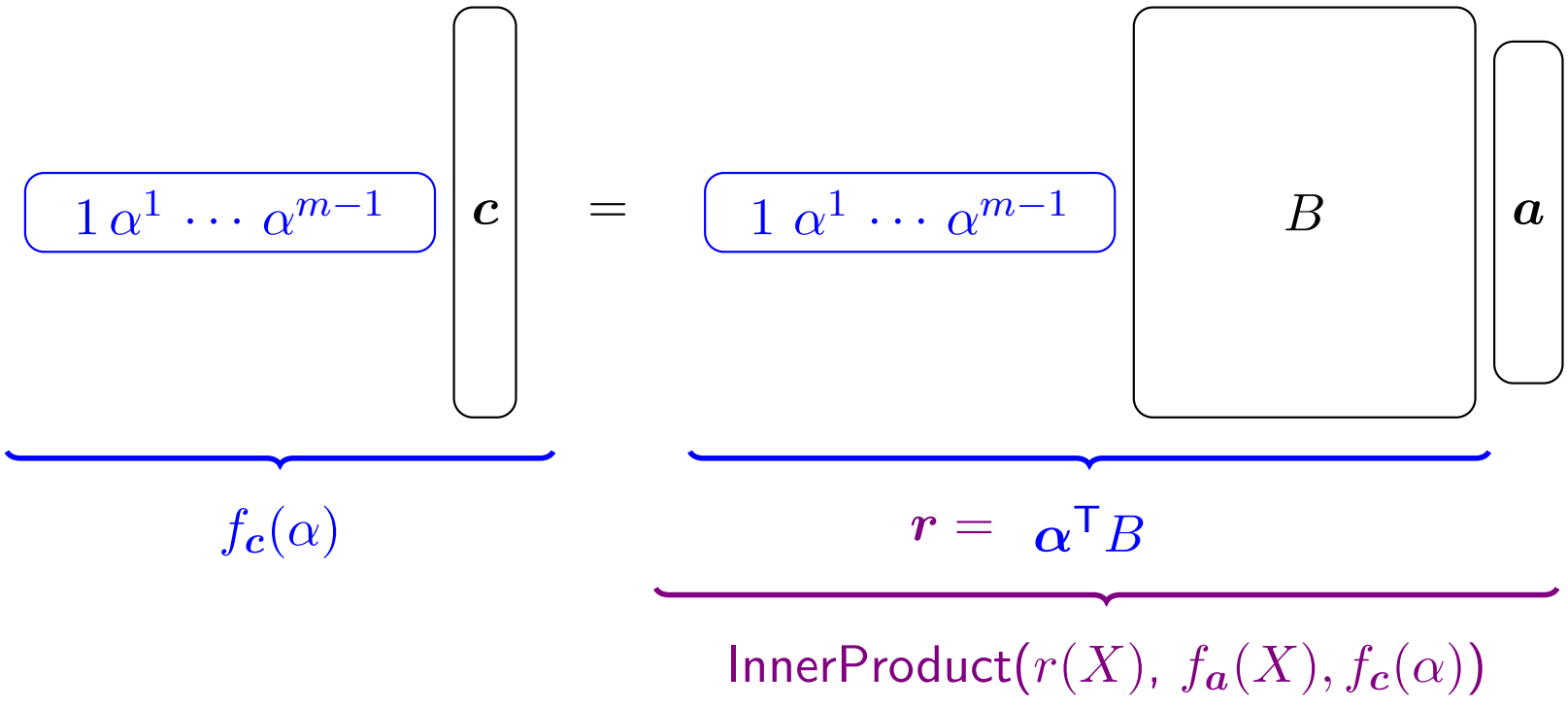
Sparse MVP



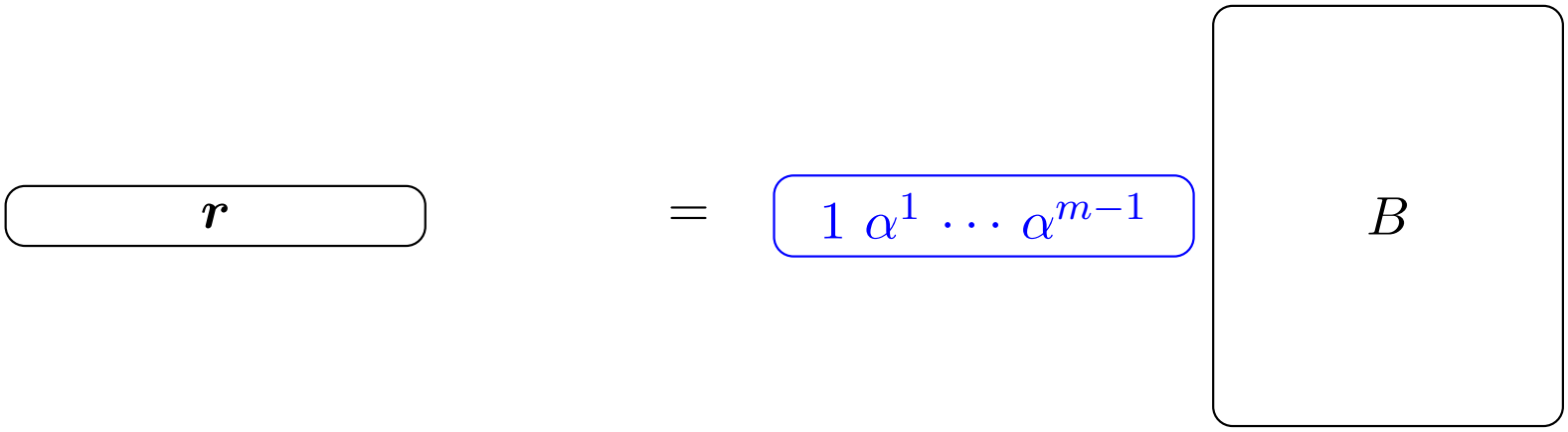
Sparse MVP


$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
$\boldsymbol{c}$
$=$
$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
B
$\boldsymbol{a}$

Sparse MVP


$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
$\boldsymbol{c}$
$=$
$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
B
$\boldsymbol{a}$
$f_{\boldsymbol{c}}(\alpha)$
$\boldsymbol{r} = \boldsymbol{\alpha}^{\mathsf{T}} B$
$\mathsf{InnerProduct}(r(X),\ f_{\boldsymbol{a}}(X), f_{\boldsymbol{c}}(\alpha))$

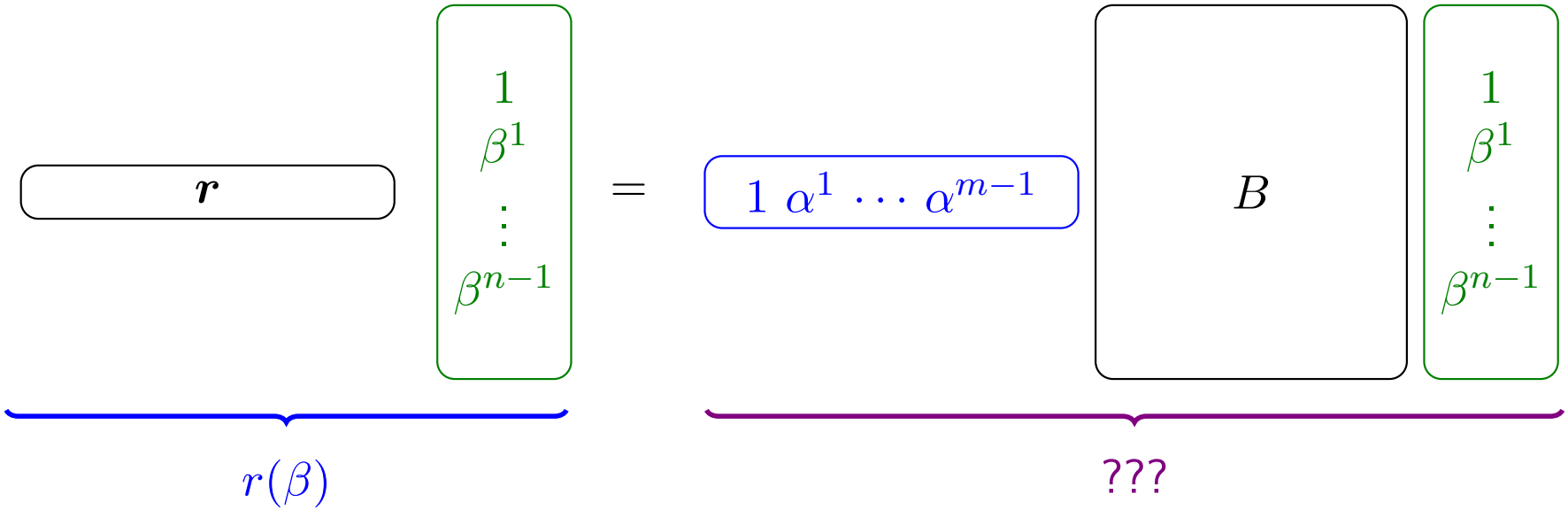
Sparse MVP



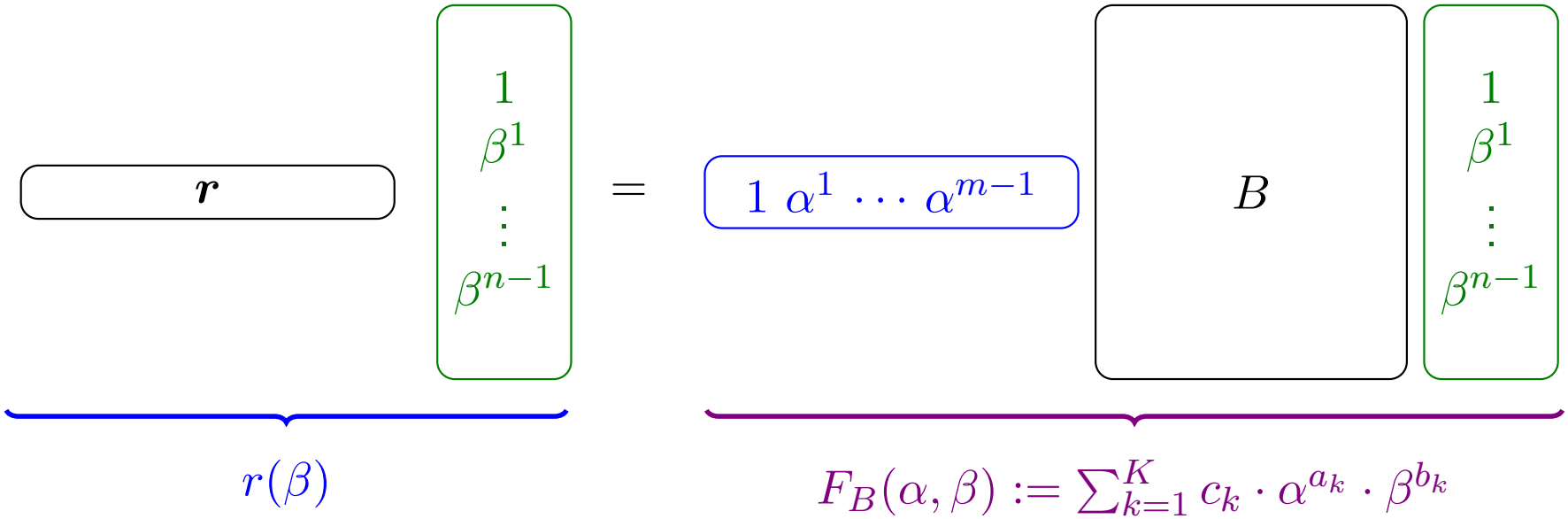
Sparse MVP

$$\boxed{\boldsymbol{r}} \begin{bmatrix} 1 \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{bmatrix} = \boxed{1\ \alpha^1\ \cdots\ \alpha^{m-1}} \boxed{B} \begin{bmatrix} 1 \\ \beta^1 \\ \vdots \\ \beta^{n-1} \end{bmatrix}$$

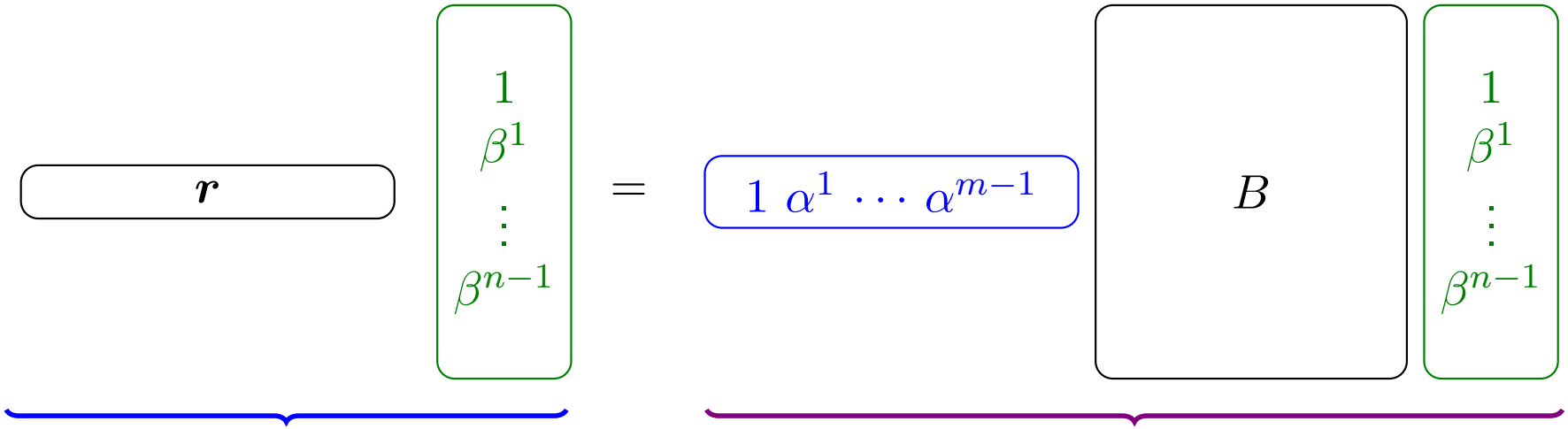
Sparse MVP


$\boldsymbol{r}$
1
β^1
$\vdots$
β^{n-1}
$=$
$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
B
1
β^1
$\vdots$
β^{n-1}
$r(\beta)$
???

Sparse MVP


$\boldsymbol{r}$
1
β^1
$\vdots$
β^{n-1}
$=$
$1\ \alpha^1\ \cdots\ \alpha^{m-1}$
B
1
β^1
$\vdots$
β^{n-1}
$r(\beta)$
$F_B(\alpha,\beta) := \sum_{k=1}^{K} c_k \cdot \alpha^{a_k} \cdot \beta^{b_k}$

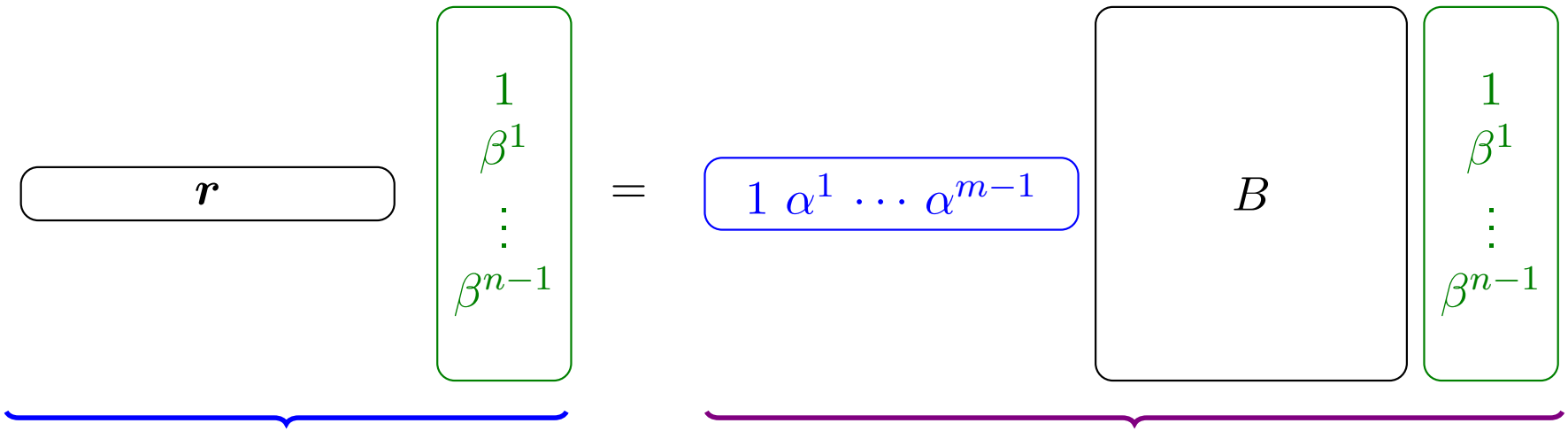
Sparse MVP



$r(\beta)$

$$F_B(\alpha, \beta) := \sum_{k=1}^{K} c_k \cdot \alpha^{a_k} \cdot \beta^{b_k}$$
$$= \boldsymbol{v}^{\mathsf{T}}(\boldsymbol{x} \circ \boldsymbol{y})$$

Sparse MVP

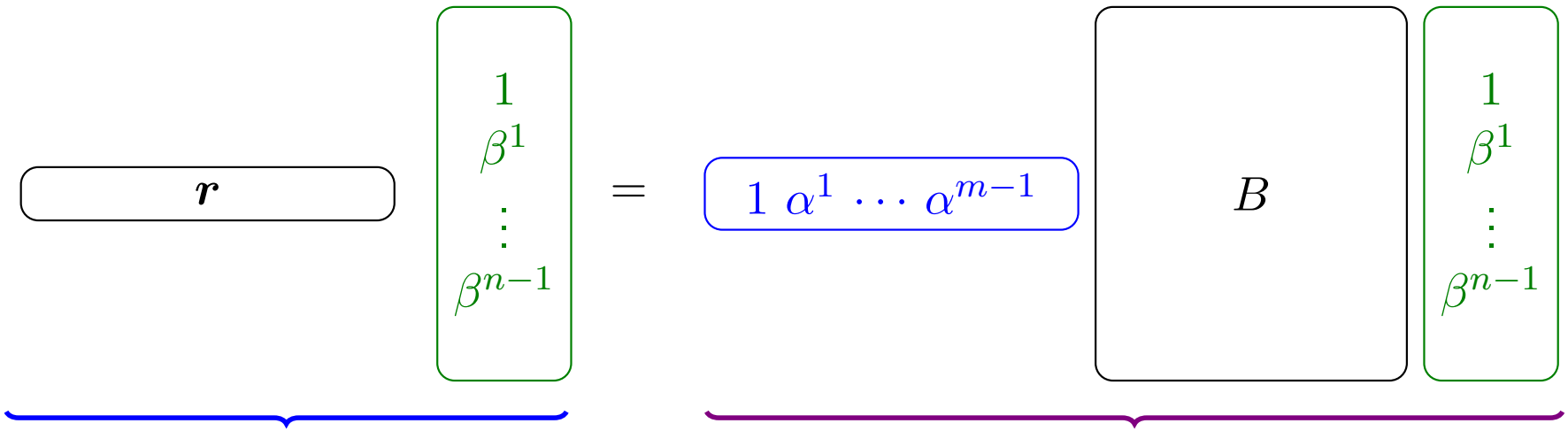


$r(\beta)$

$$F_B(\alpha, \beta) := \sum_{k=1}^{K} c_k \cdot \alpha^{a_k} \cdot \beta^{b_k}$$
$$= \boldsymbol{v}^{\mathsf{T}}(\boldsymbol{x} \circ \boldsymbol{y})$$

$$\mathsf{InnerProduct}([f_{\boldsymbol{v}}(X)],\ \mathsf{Hadamard}([f_{\boldsymbol{x}}(X)], [f_{\boldsymbol{y}}(X)]),\ r(\beta))$$

Sparse MVP



$r(\beta)$

$F_B(\alpha, \beta) := \sum_{k=1}^{K} c_k \cdot \alpha^{a_k} \cdot \beta^{b_k}$

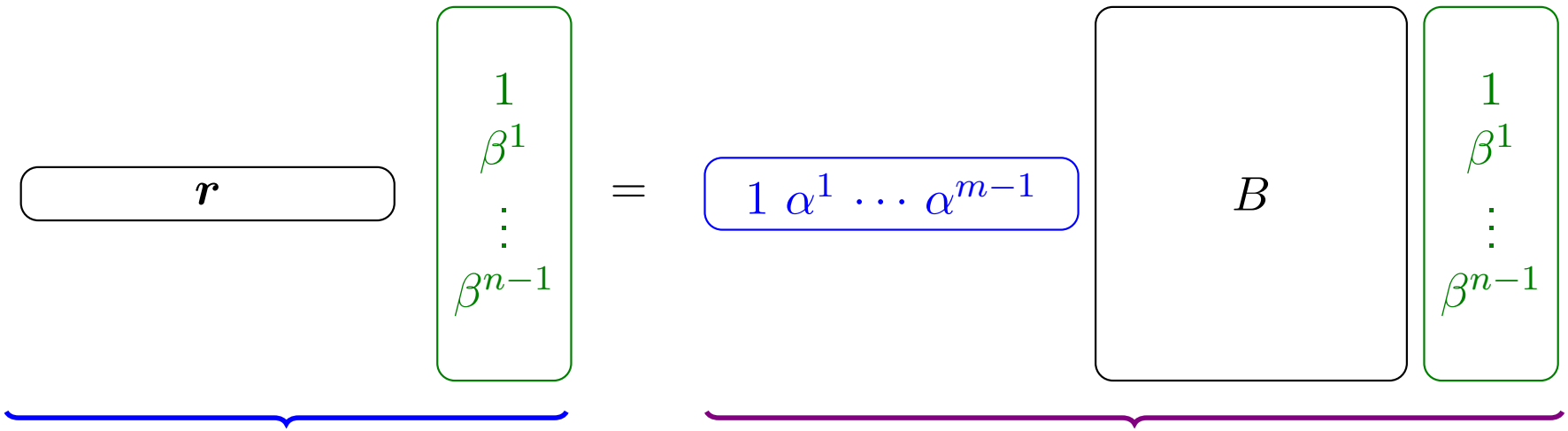
$= \boldsymbol{v}^{\mathsf{T}}(\boldsymbol{x} \circ \boldsymbol{y})$

InnerProduct($f_{\boldsymbol{v}}(X)$, Hadamard($[f_{\boldsymbol{x}}(X)], [f_{\boldsymbol{y}}(X)]$), $r(\beta)$)

Preprocessed

???

Sparse MVP


r
1
β^1
⋮
β^{n−1}
=
1 α^1 ⋯ α^{m−1}
B
1
β^1
⋮
β^{n−1}
r(β)
F_B(α, β) := Σ_{k=1}^{K} c_k · α^{a_k} · β^{b_k}
= v^T (x ∘ y)
InnerProduct([f_v(X)], Hadamard([f_x(X)], [f_y(X)]), r(β))
Preprocessed
SparseMonomialVector

Sparse MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)]$, $[f_{\boldsymbol{v}}(X)]$, $[f_{\boldsymbol{a}}(X)]$

where $\boldsymbol{v}$ are the nonzero values of B

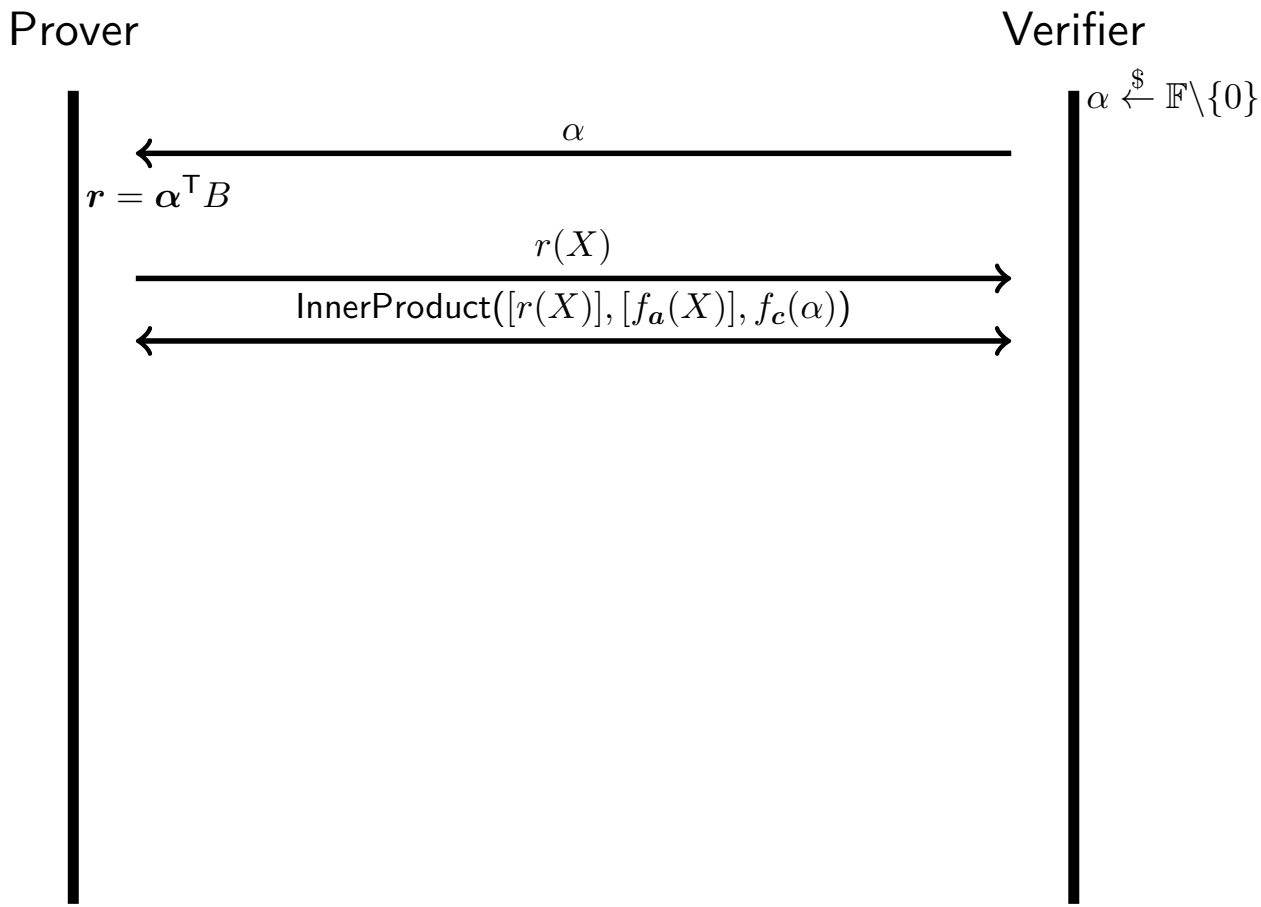
claim: $\boldsymbol{c} = B\boldsymbol{a}$

Sparse MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)]$, $[f_{\boldsymbol{v}}(X)]$, $[f_{\boldsymbol{a}}(X)]$

where $\boldsymbol{v}$ are the nonzero values of B

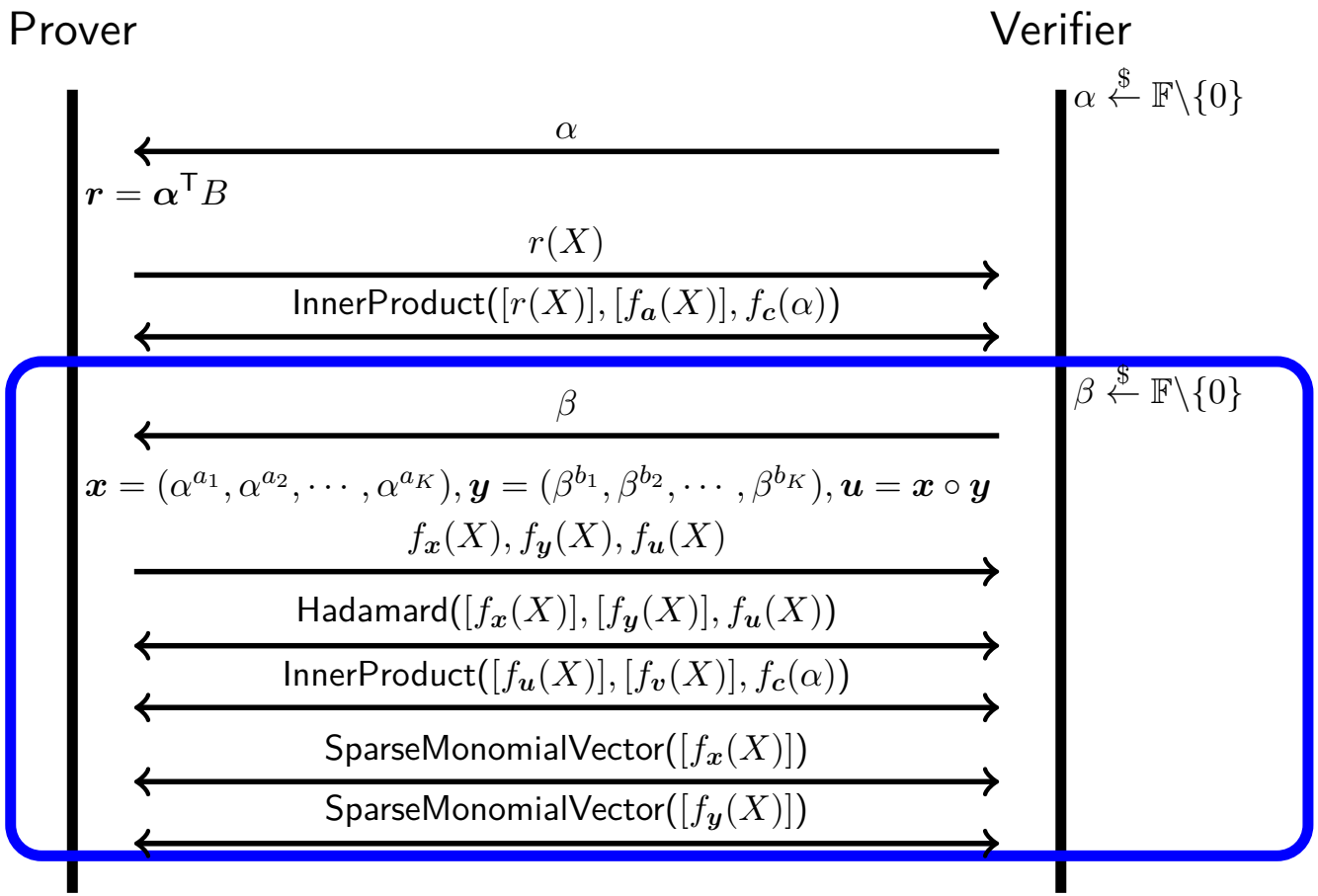
claim: $\boldsymbol{c} = B\boldsymbol{a}$



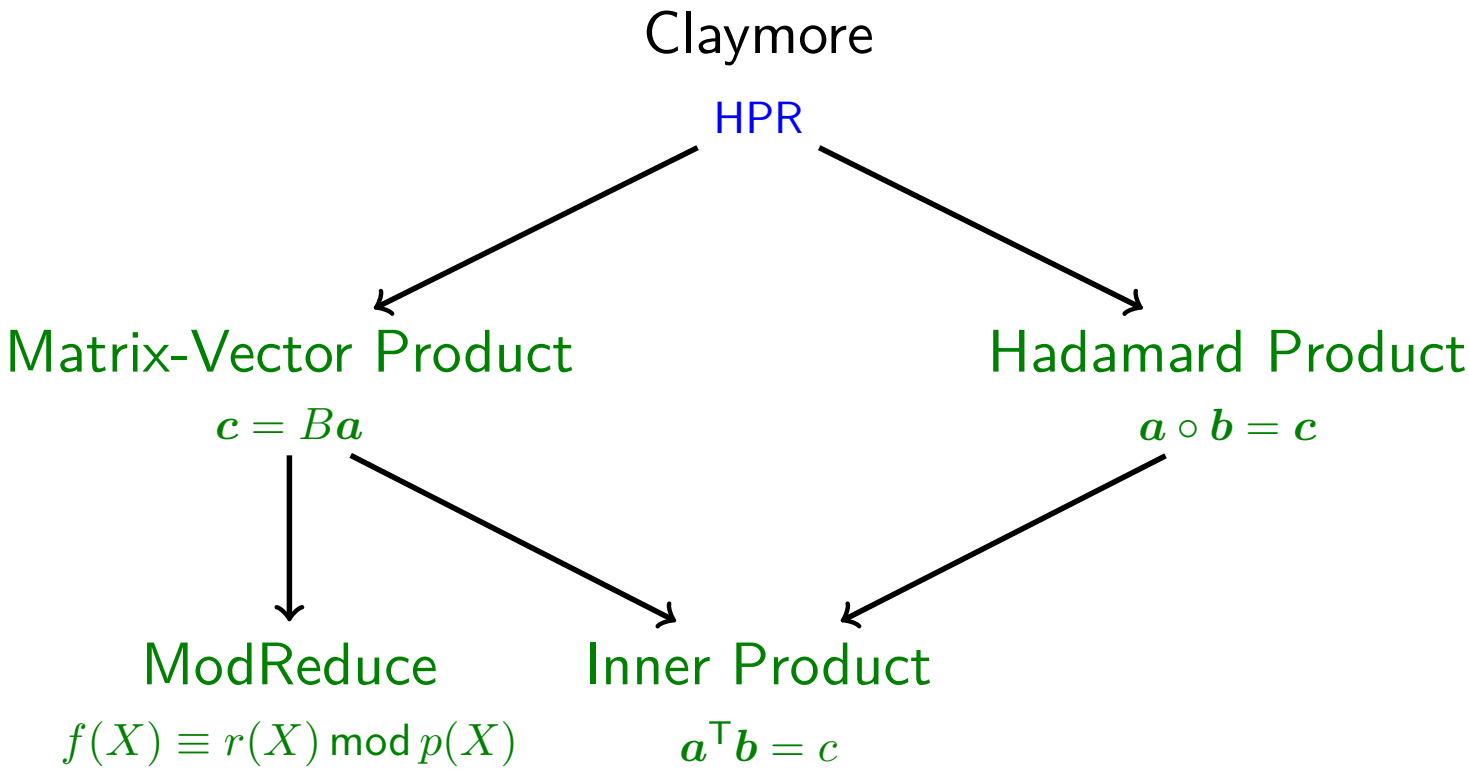
Sparse MVP

Verifier has oracles $[f_{\boldsymbol{c}}(X)]$, $[f_{\boldsymbol{v}}(X)]$, $[f_{\boldsymbol{a}}(X)]$
where $\boldsymbol{v}$ are the nonzero values of B

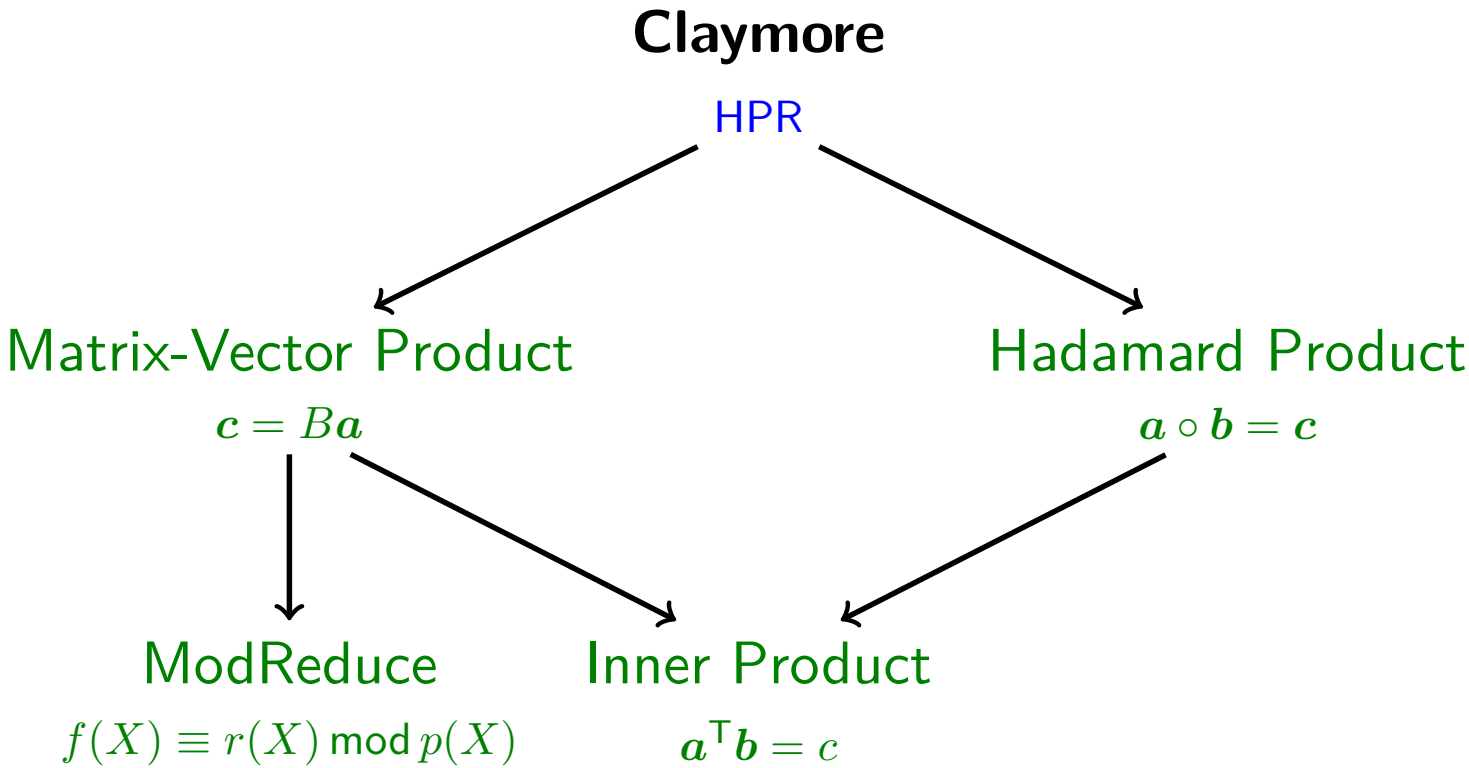
claim: $\boldsymbol{c} = B\boldsymbol{a}$



Subprotocols



Subprotocols



Claymore

Verifier has oracle $[f_M(X)]$

Verifier knows $\boldsymbol{x}$

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

$$\boldsymbol{x} = M \cdot (1|\boldsymbol{w}_l^{\mathsf{T}}|\boldsymbol{w}_r^{\mathsf{T}}|\boldsymbol{w}_o^{\mathsf{T}})^{\mathsf{T}} \wedge \boldsymbol{w}_l \circ \boldsymbol{w}_r = \boldsymbol{w}_o$$

Claymore

Verifier has oracle $[f_M(X)]$

Verifier knows $\boldsymbol{x}$

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

$$\boldsymbol{x} = M \cdot (1|\boldsymbol{w}_l^{\mathsf{T}}|\boldsymbol{w}_r^{\mathsf{T}}|\boldsymbol{w}_o^{\mathsf{T}})^{\mathsf{T}} \wedge \boldsymbol{w}_l \circ \boldsymbol{w}_r = \boldsymbol{w}_o$$

Prover

Verifier

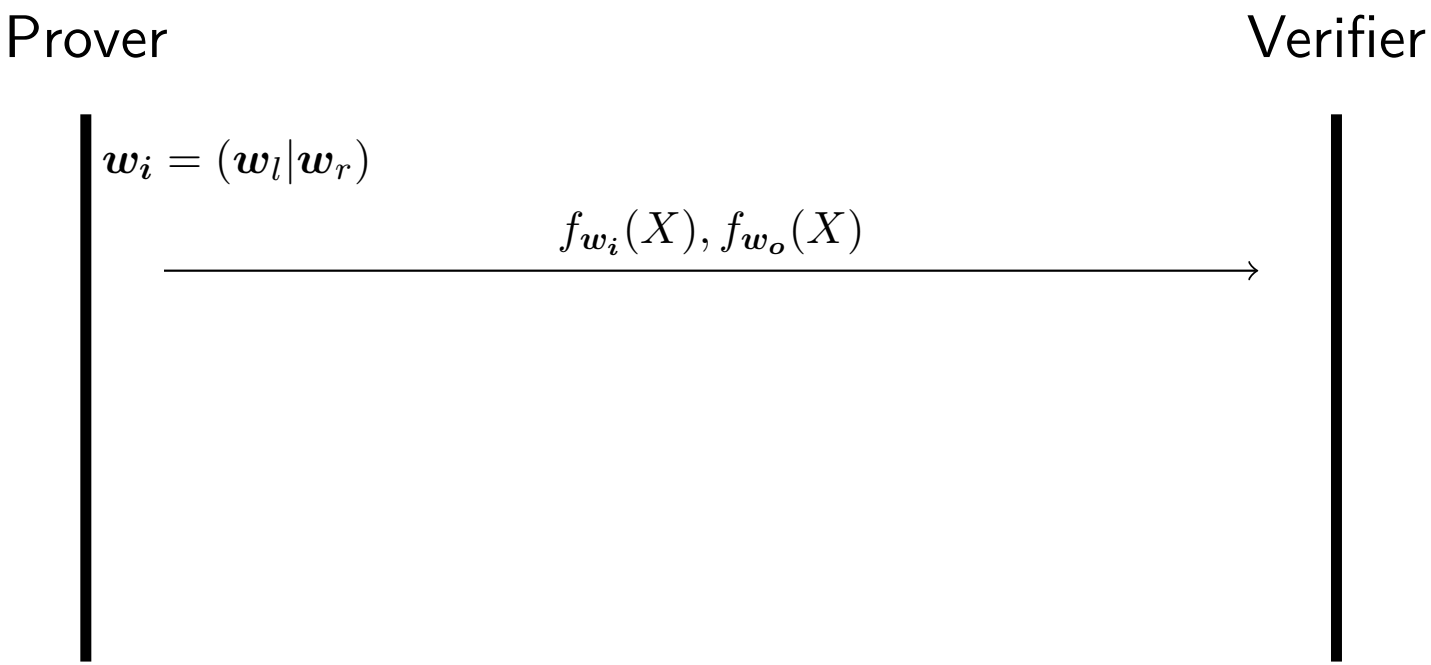
Claymore

Verifier has oracle $[f_M(X)]$

Verifier knows $\boldsymbol{x}$

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

$$\boldsymbol{x} = M \cdot (1|\boldsymbol{w}_l^{\mathsf{T}}|\boldsymbol{w}_r^{\mathsf{T}}|\boldsymbol{w}_o^{\mathsf{T}})^{\mathsf{T}} \wedge \boldsymbol{w}_l \circ \boldsymbol{w}_r = \boldsymbol{w}_o$$



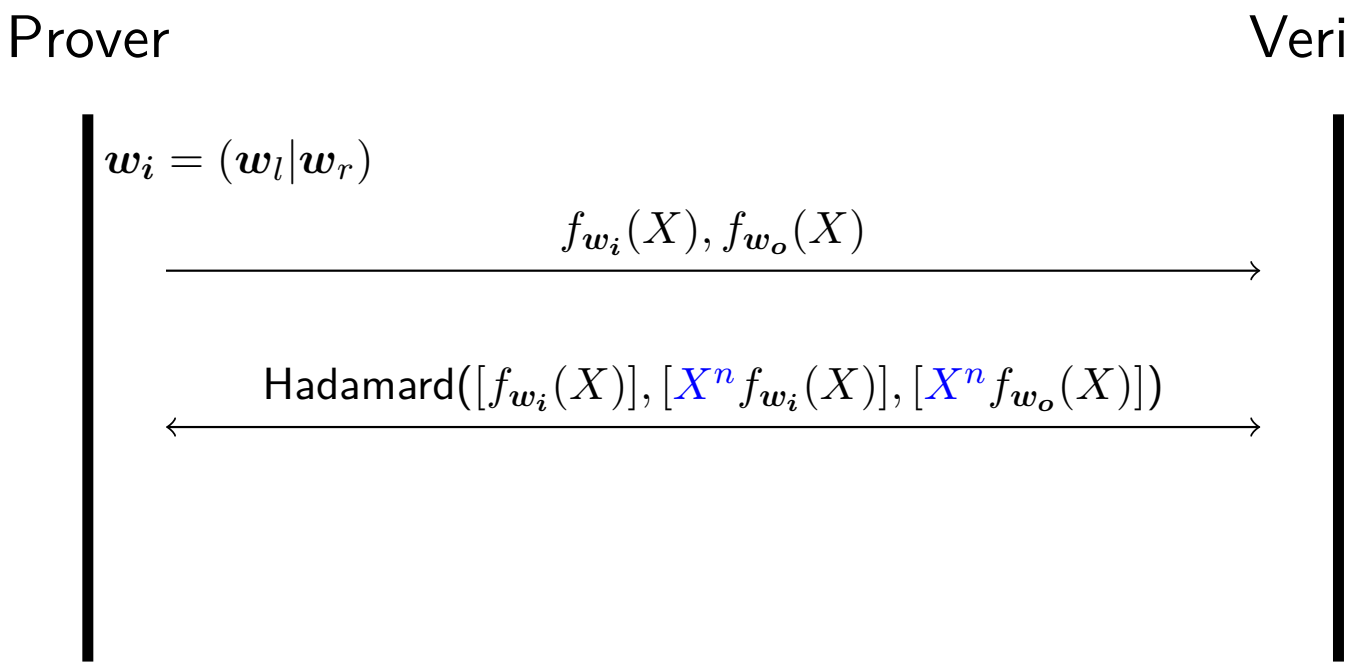
Claymore

Verifier has oracle $[f_M(X)]$

Verifier knows $\boldsymbol{x}$

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

$$\boldsymbol{x} = M \cdot (1|\boldsymbol{w}_l^{\mathsf{T}}|\boldsymbol{w}_r^{\mathsf{T}}|\boldsymbol{w}_o^{\mathsf{T}})^{\mathsf{T}} \wedge \boldsymbol{w}_l \circ \boldsymbol{w}_r = \boldsymbol{w}_o$$



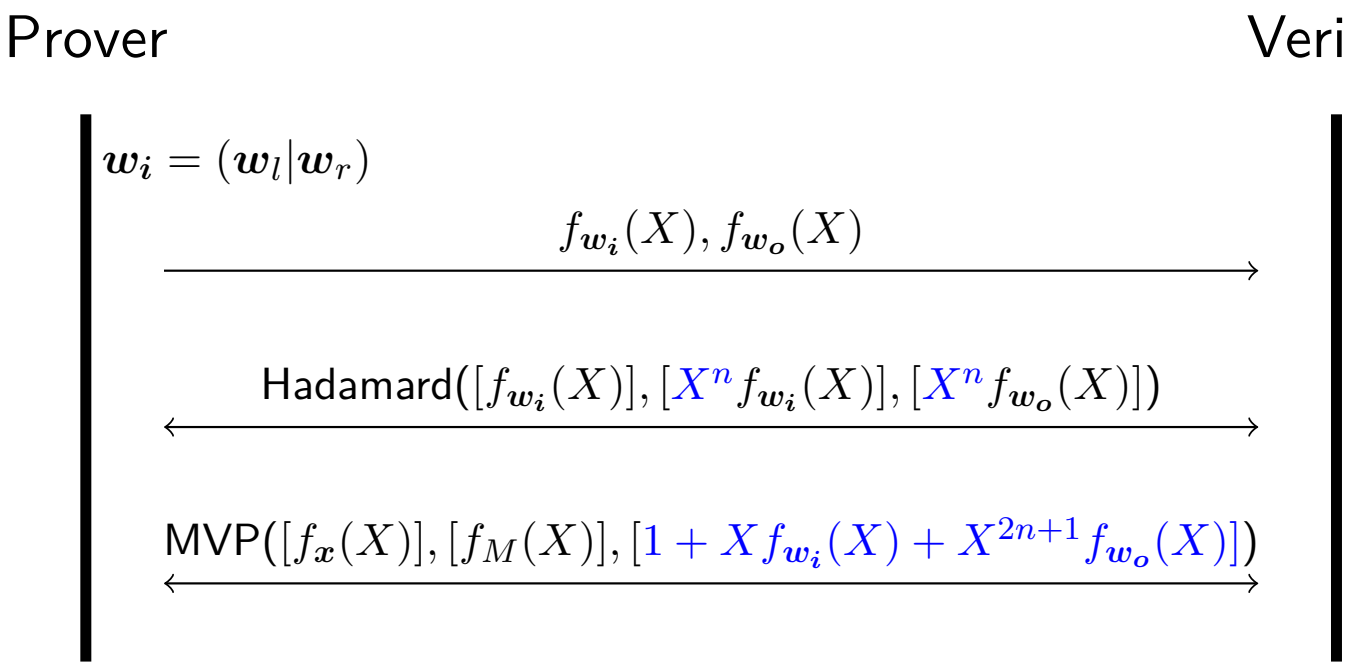
Claymore

Verifier has oracle $[f_M(X)]$

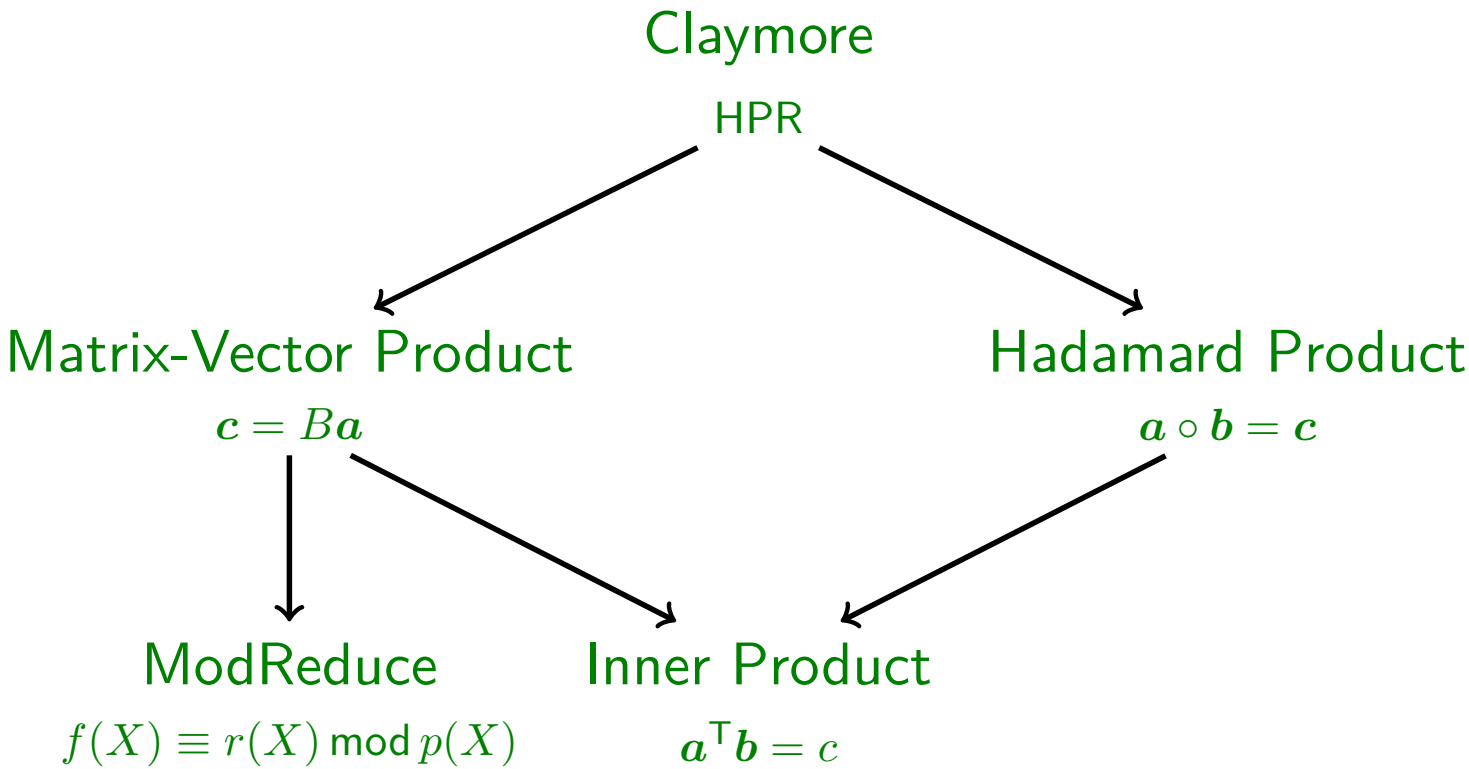
Verifier knows $\boldsymbol{x}$

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

$$\boldsymbol{x} = M \cdot (1|\boldsymbol{w}_l^{\mathsf{T}}|\boldsymbol{w}_r^{\mathsf{T}}|\boldsymbol{w}_o^{\mathsf{T}})^{\mathsf{T}} \wedge \boldsymbol{w}_l \circ \boldsymbol{w}_r = \boldsymbol{w}_o$$



Subprotocols



Zero-Knowledge

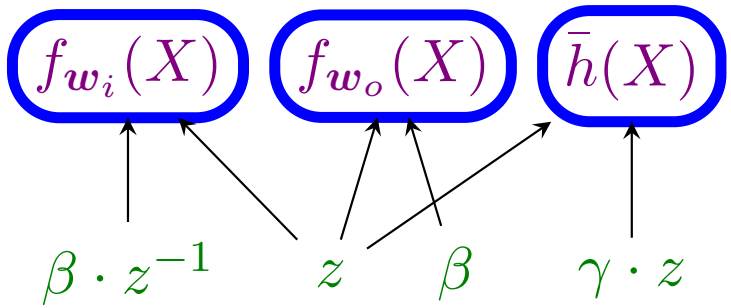
Witness-related polynomials

$$f_{\boldsymbol{w}_i}(X) \quad f_{\boldsymbol{w}_o}(X) \quad \bar{h}(X)$$

Zero-Knowledge

Witness-related polynomials

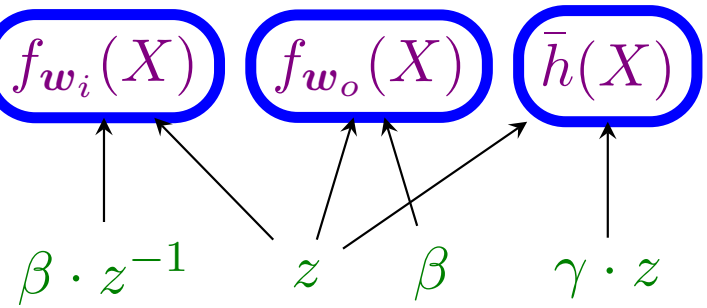
queries



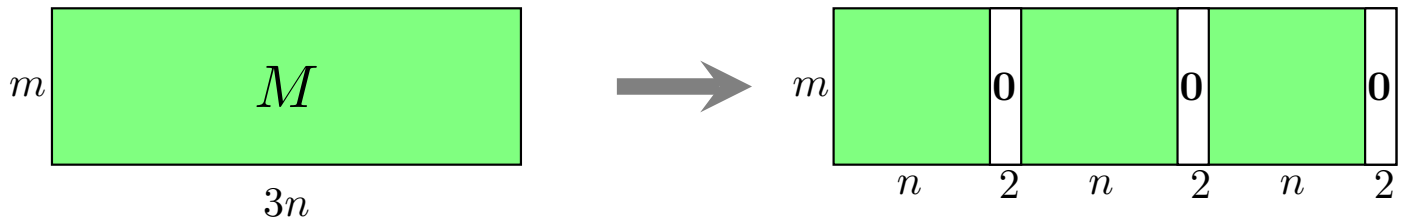
Zero-Knowledge

Witness-related polynomials

queries



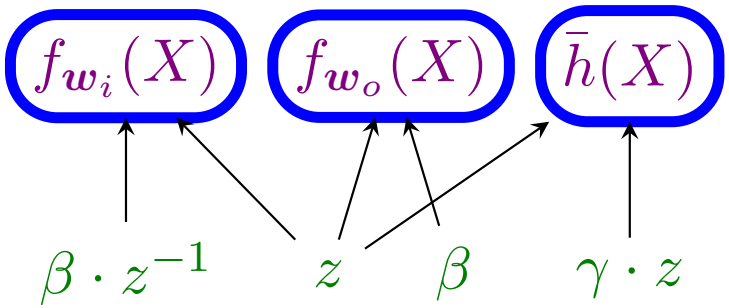
padding



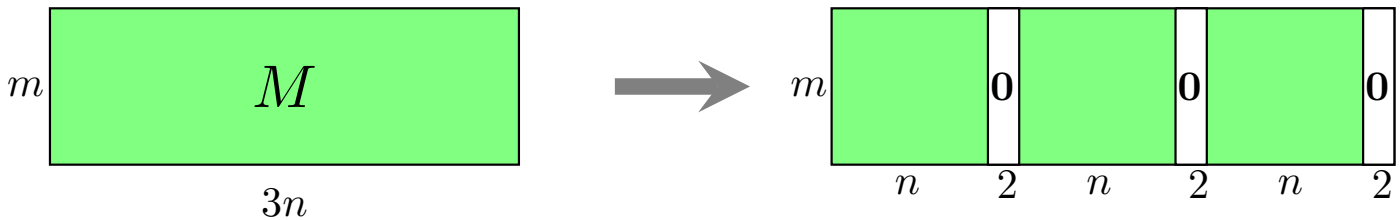
Zero-Knowledge

Witness-related polynomials

queries



padding

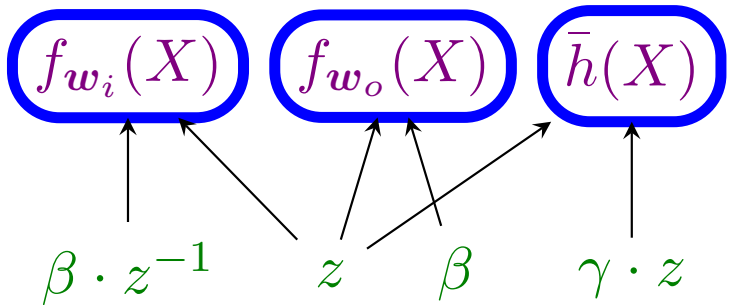


randomizing

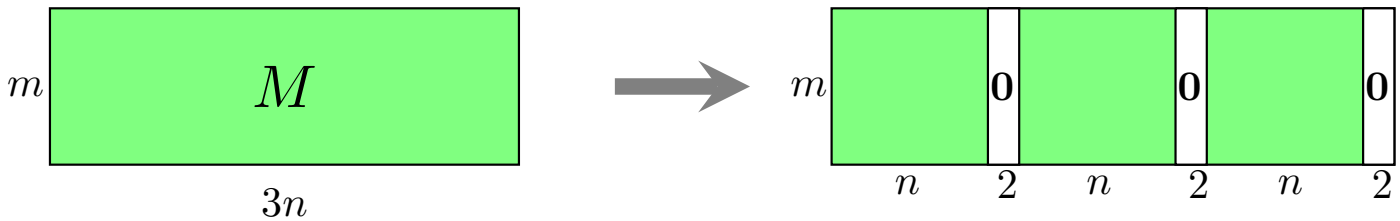
Zero-Knowledge

Witness-related polynomials

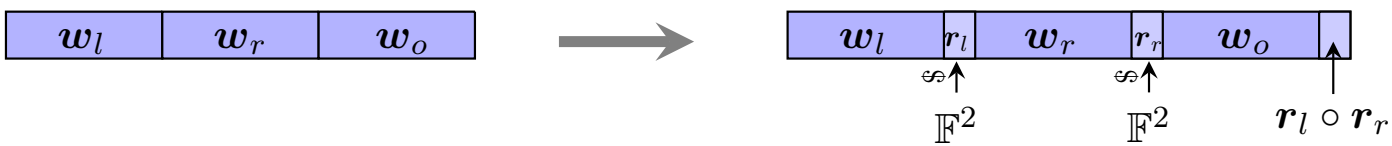
queries



padding



randomizing



Performance

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

Performance

	# polys offline / online	# evals	# points	max. degree
Sonic	$12f/$ $3f+7$	$11f+3$	$9f+2$	$O(n)$
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