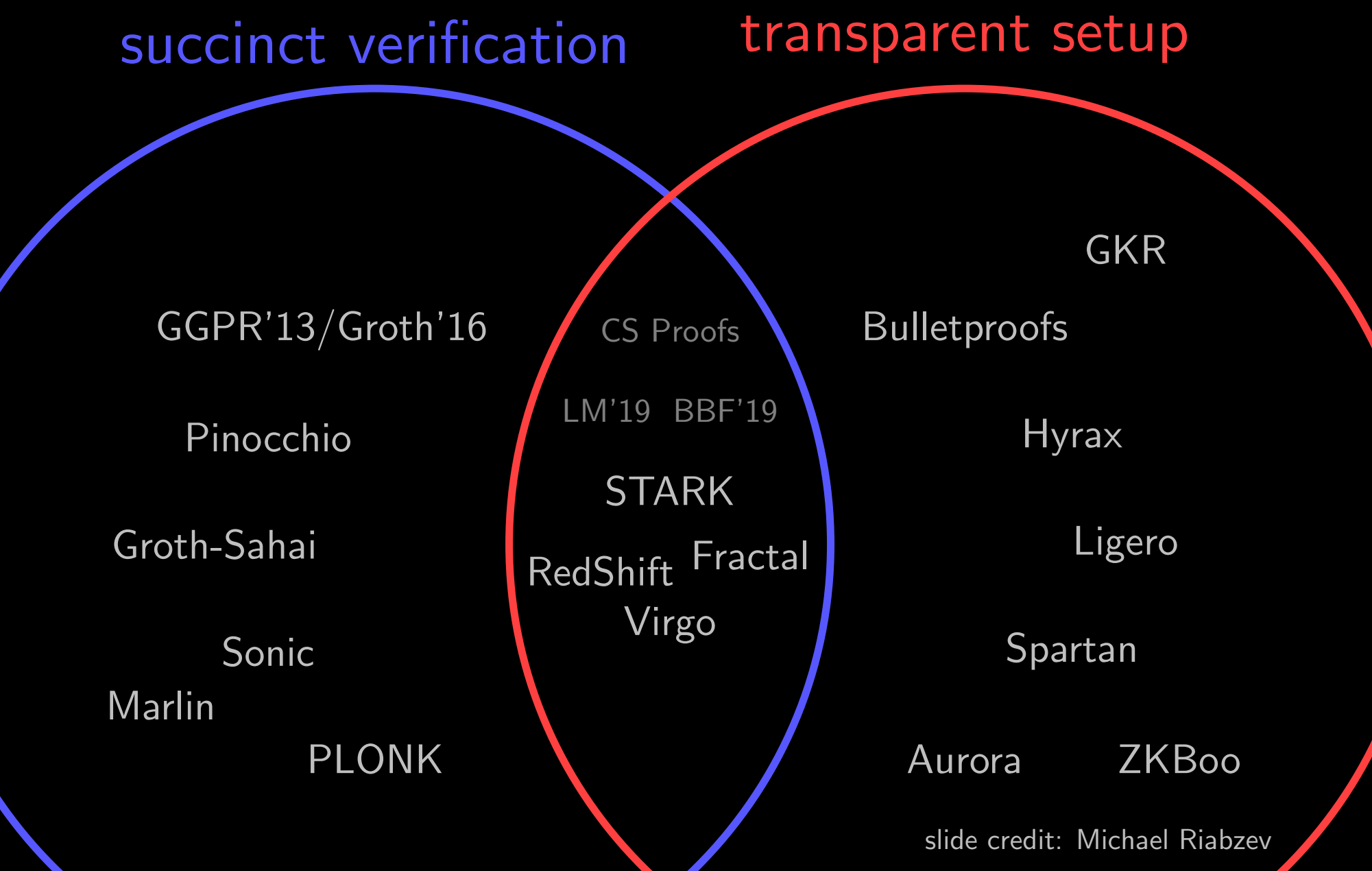
Transparent SNARKs from DARK Compilers

Benedikt Bünz[1] and Ben Fisch[1] and **Alan Szepieniec**[2]

[1]Stanford [2]Nervos Foundation

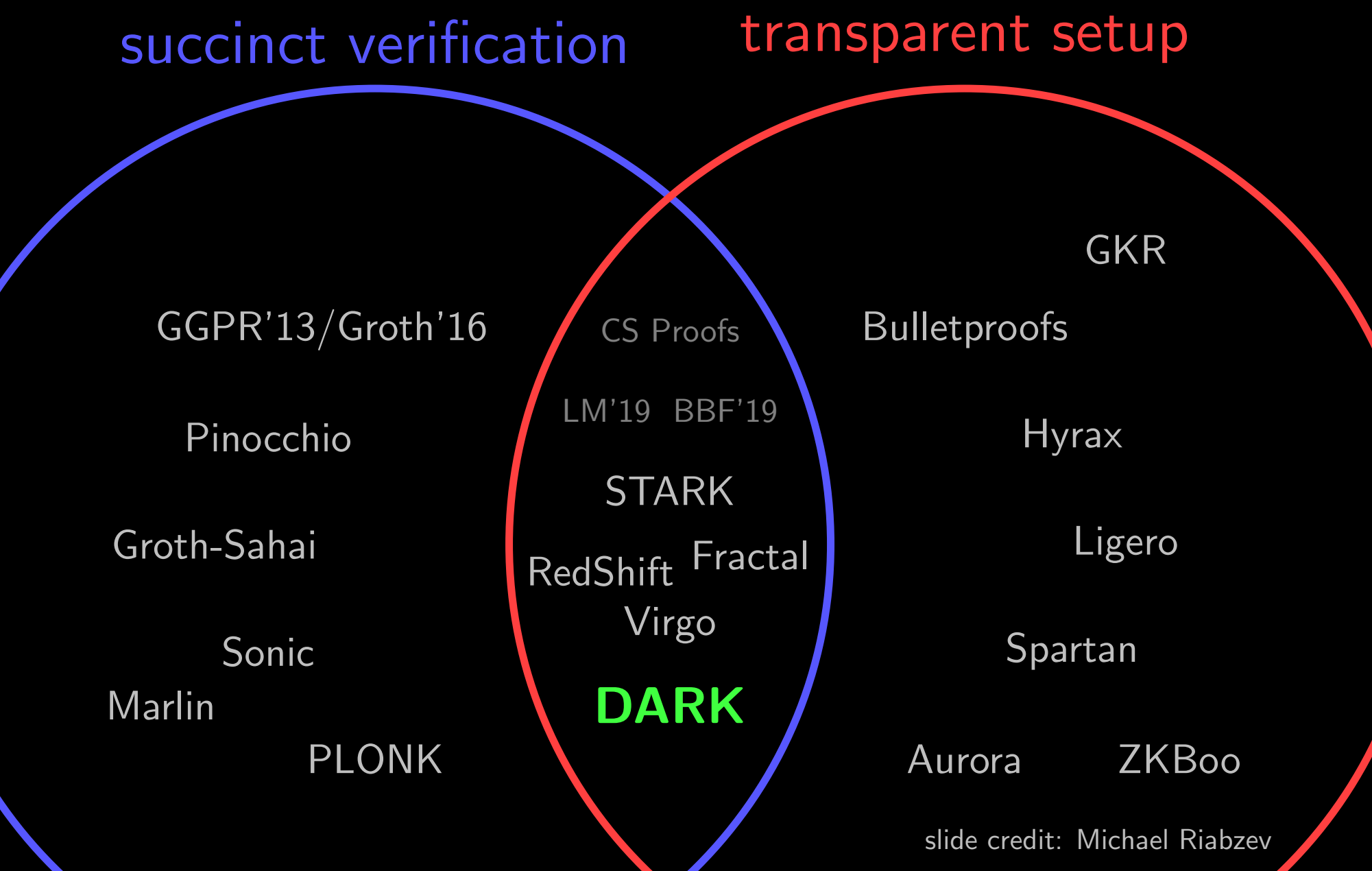
Eurocrypt 2020

General Purpose Zero-Knowledge Proofs



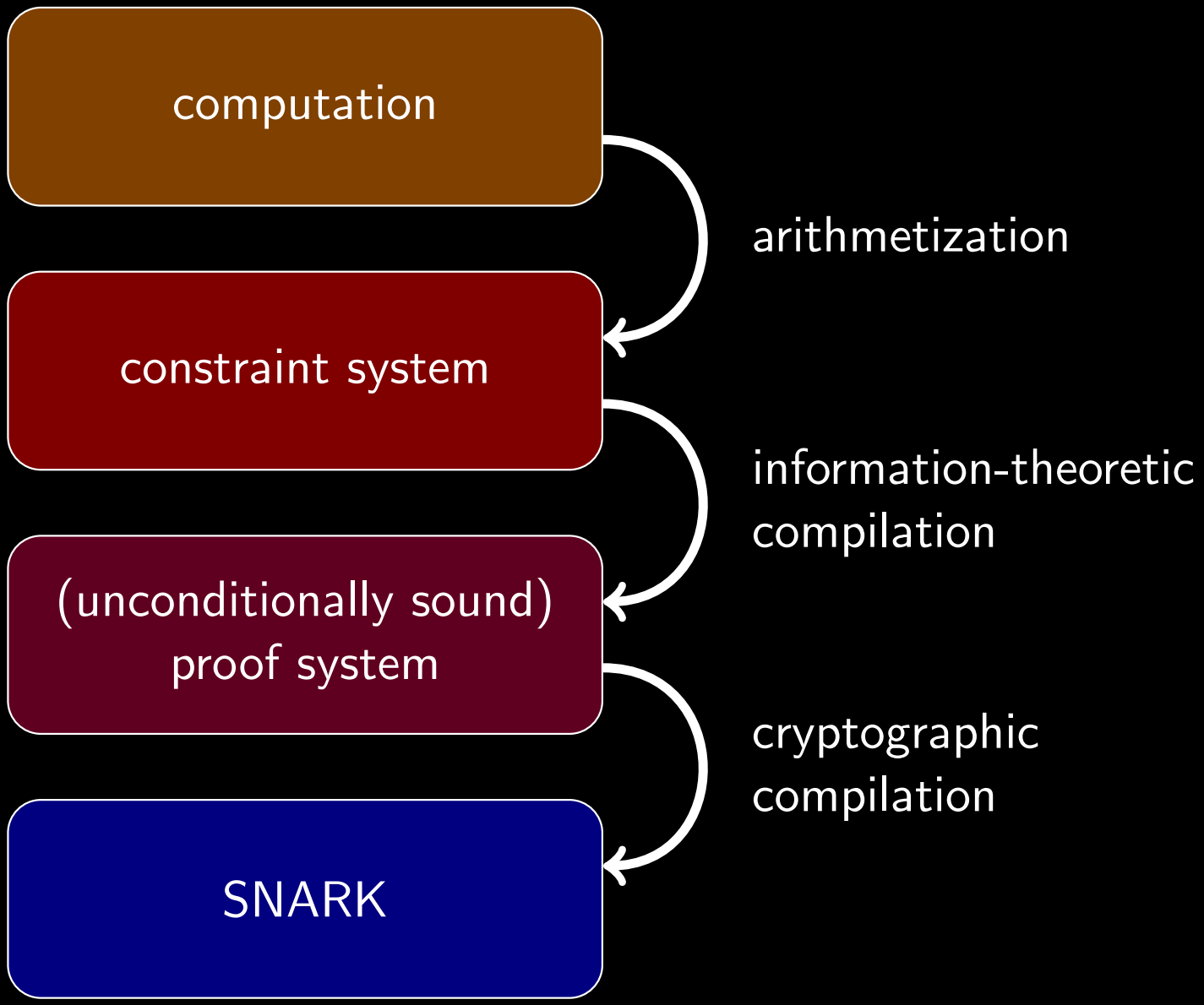
slide credit: Michael Riabzev

General Purpose Zero-Knowledge Proofs

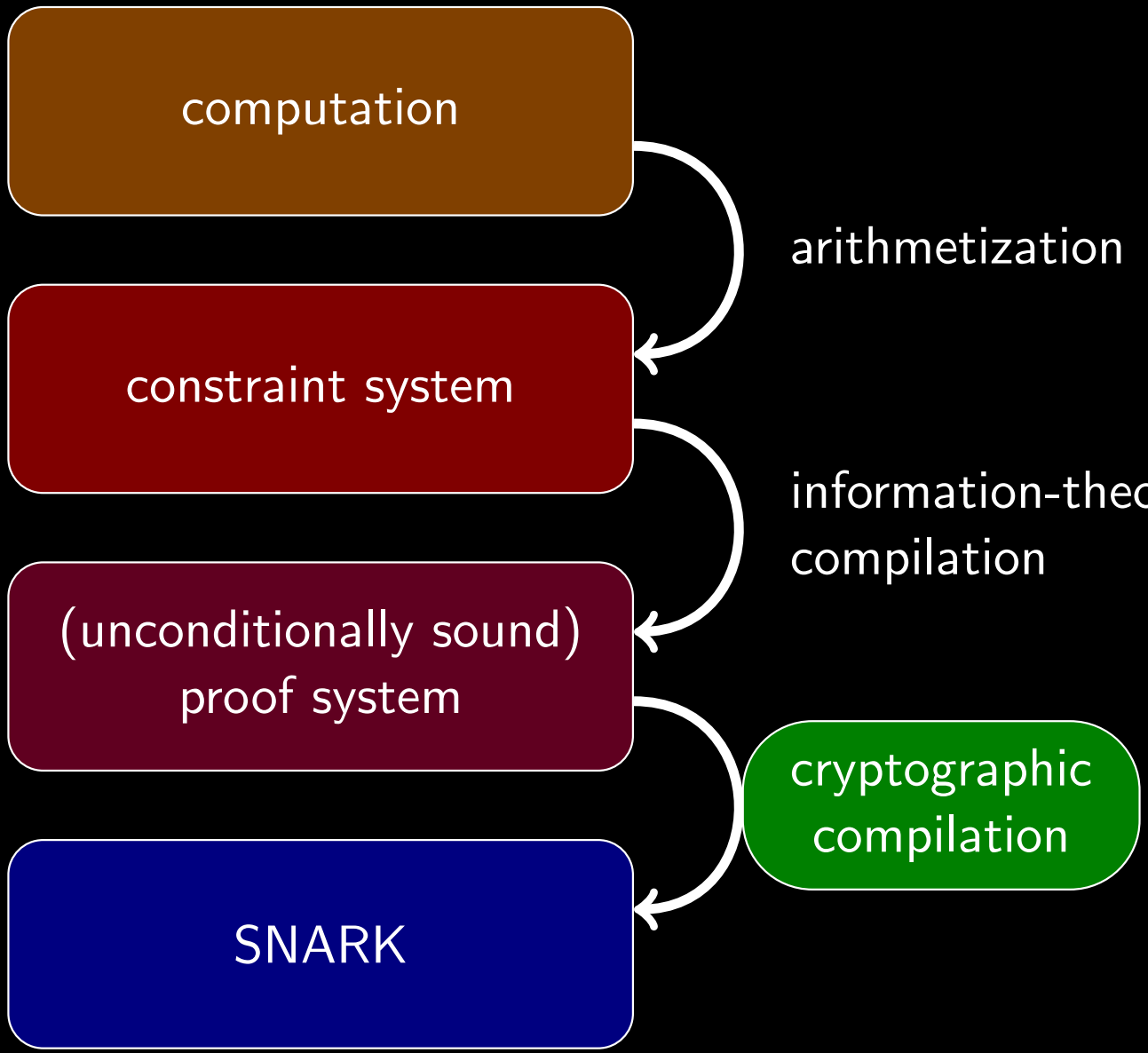


slide credit: Michael Riabzev

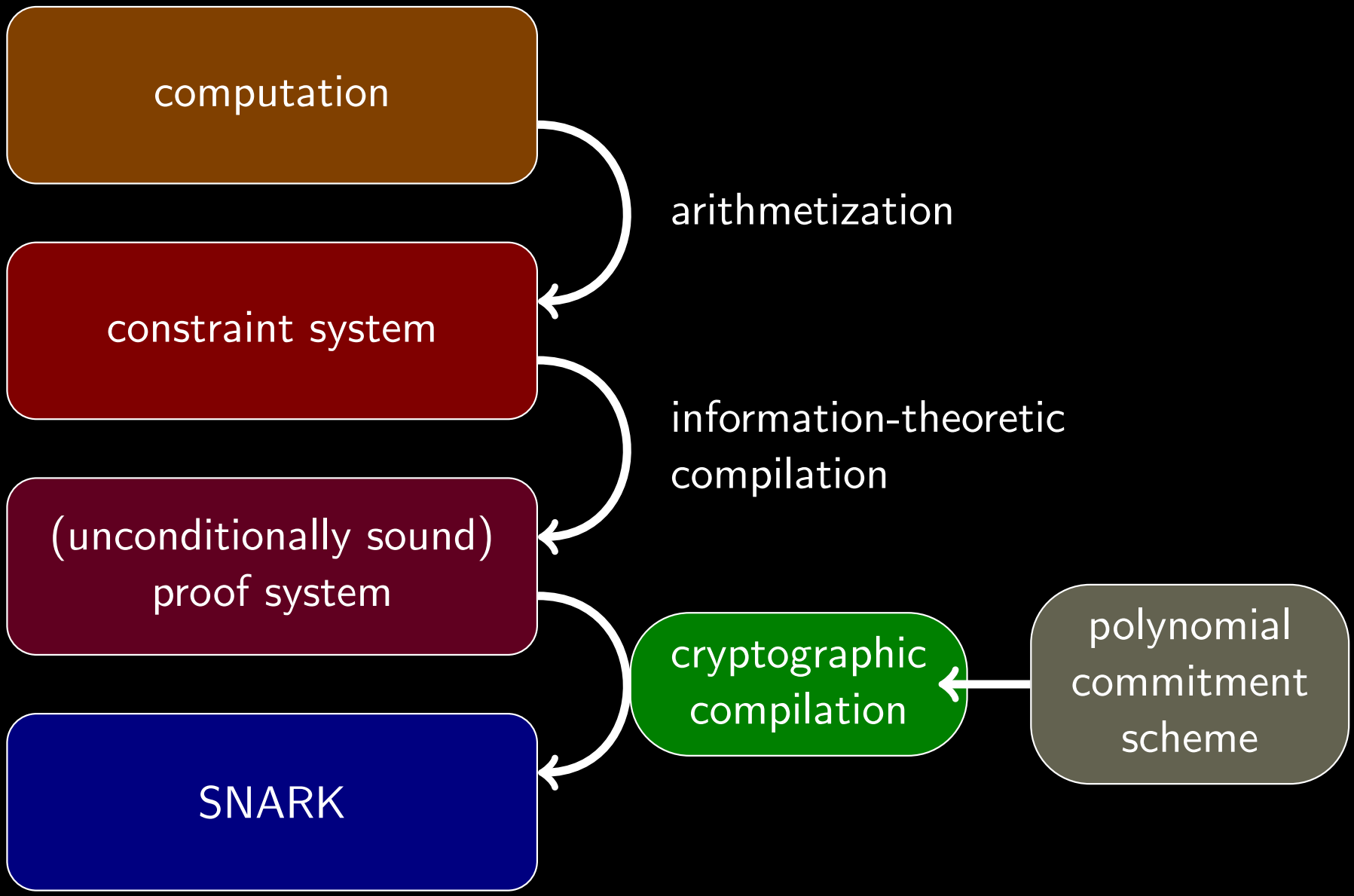
Compilation Pipeline



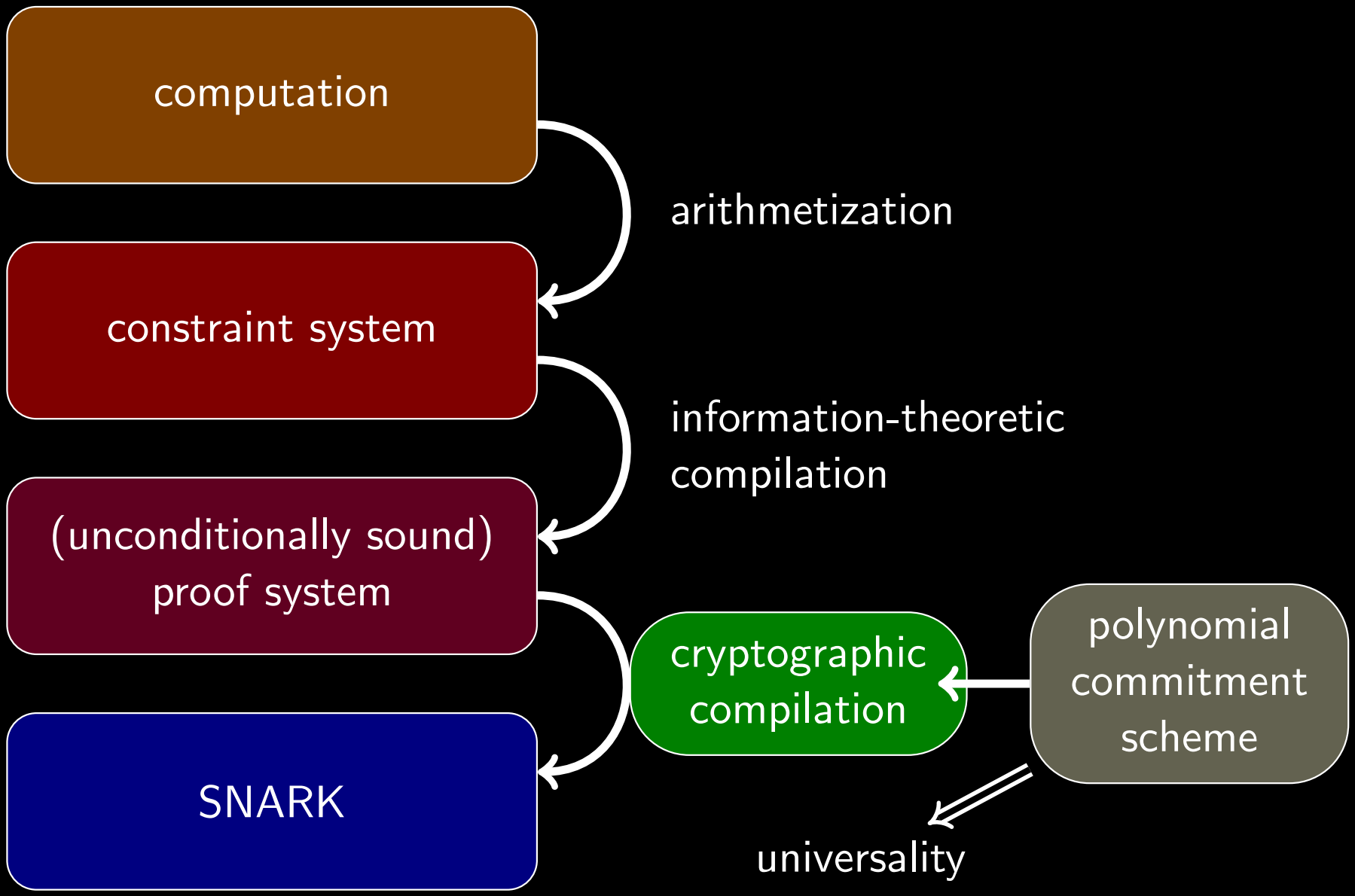
Compilation Pipeline



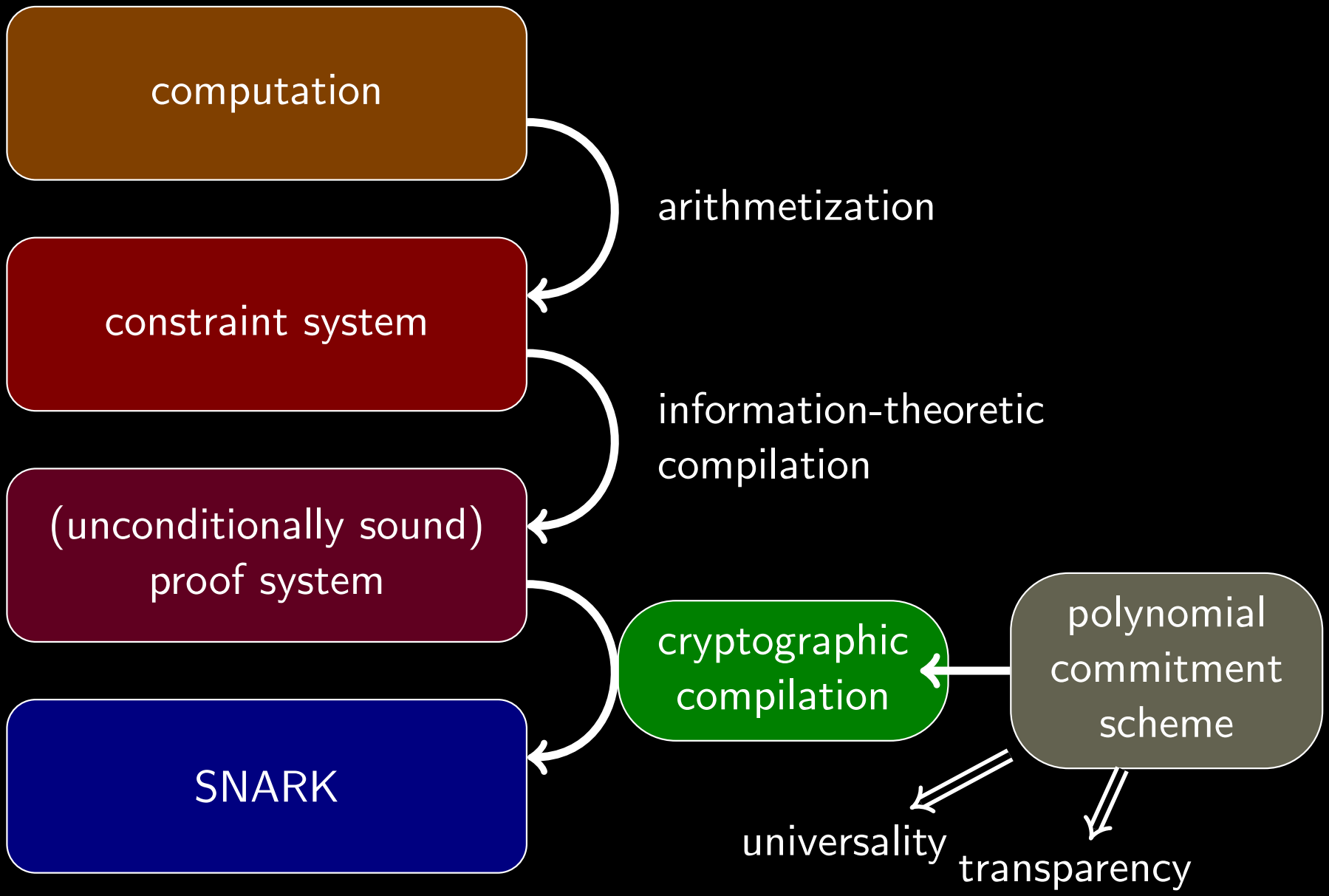
Compilation Pipeline



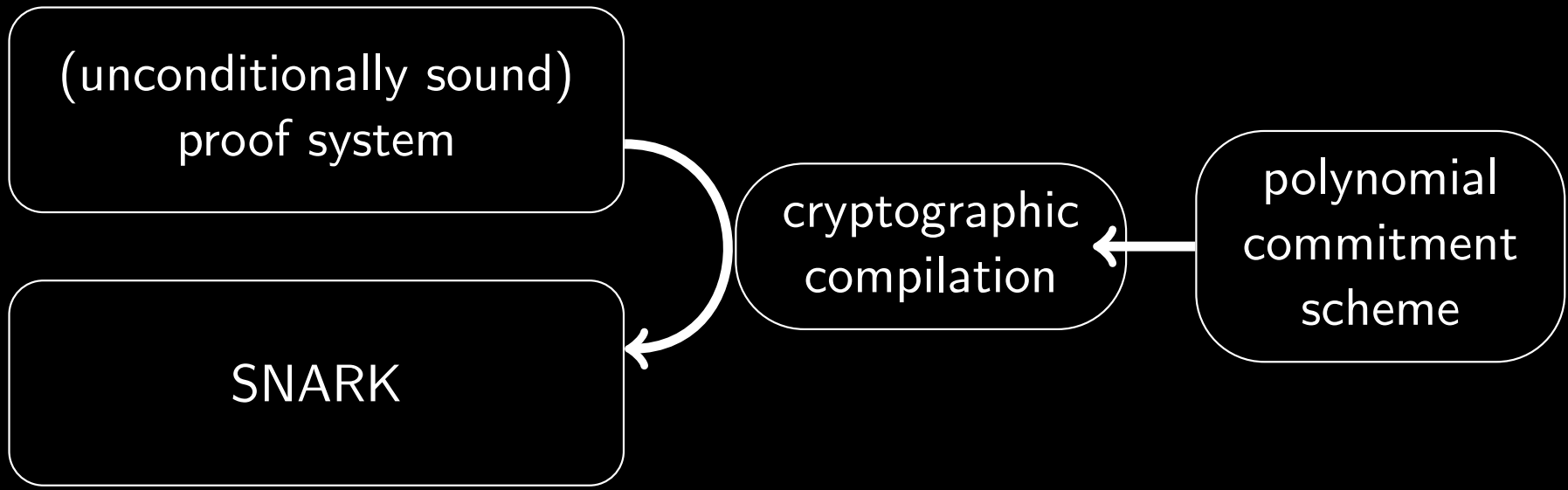
Compilation Pipeline



Compilation Pipeline

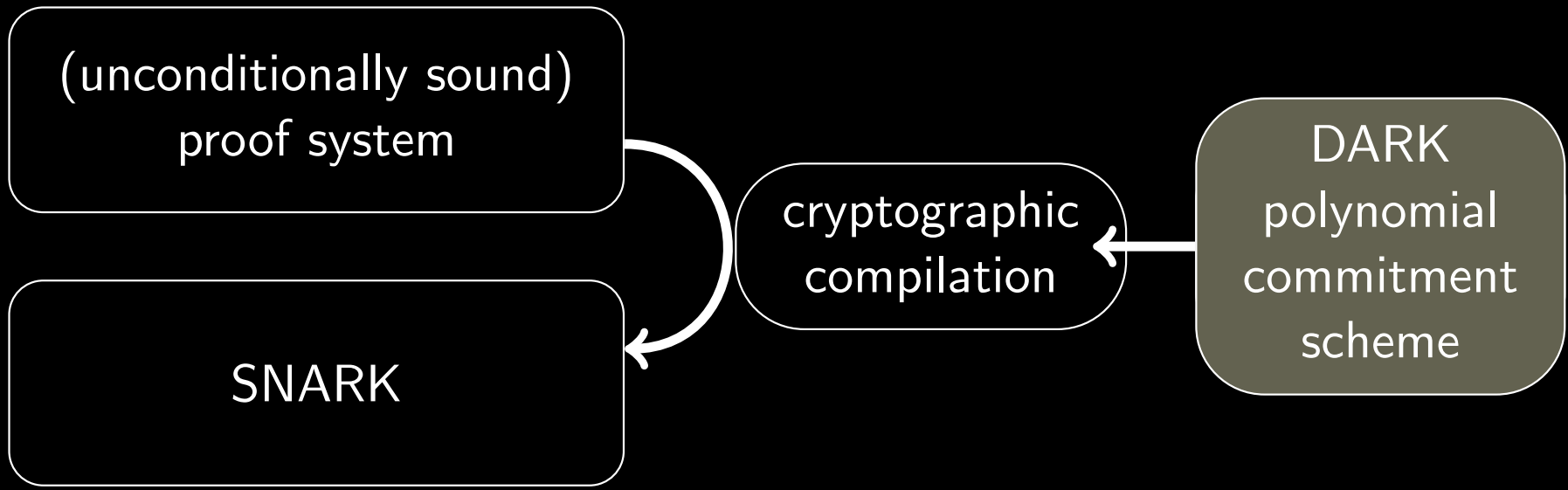


Contributions



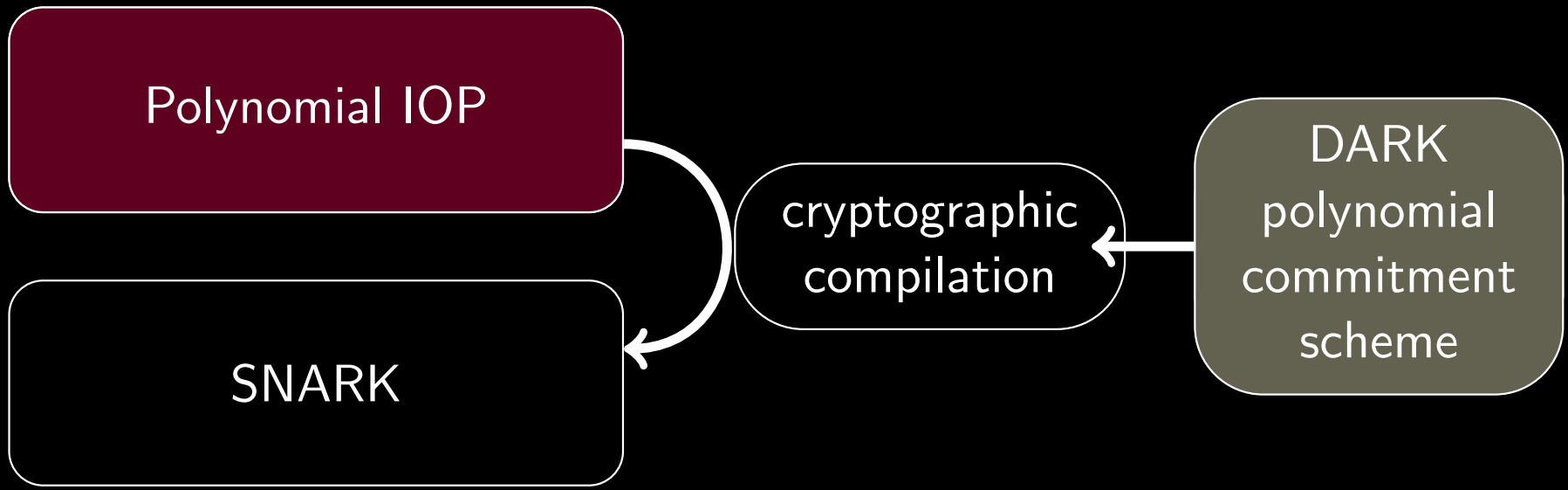
Contributions

1. **polynomial commitment scheme** based on **groups of unknown order**



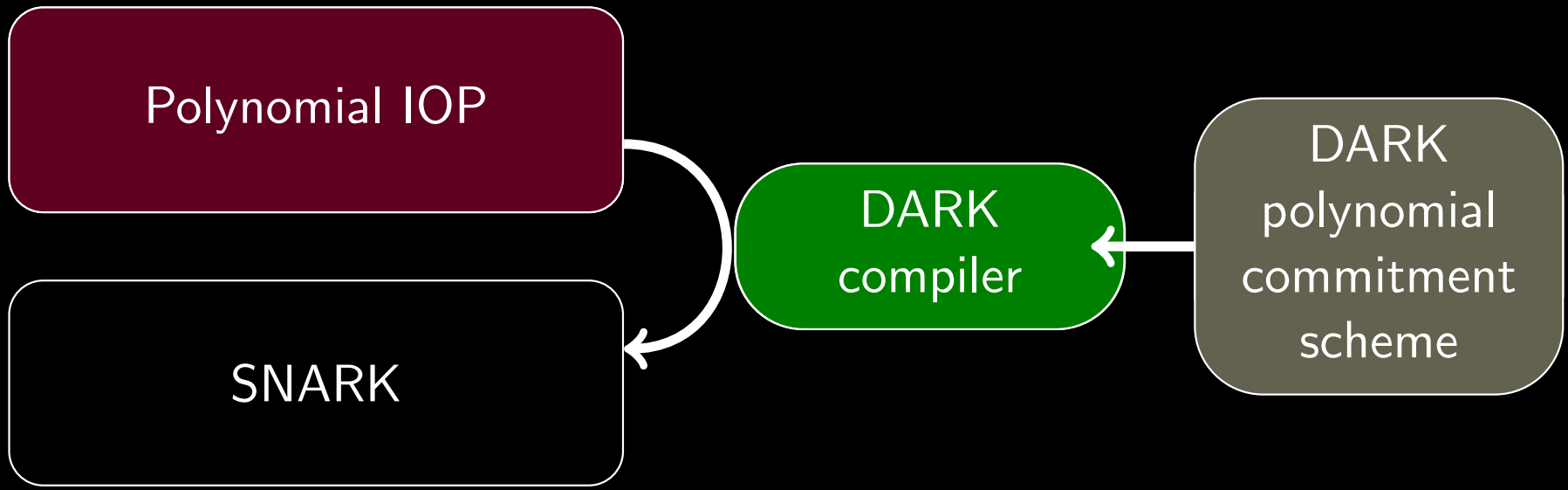
Contributions

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**



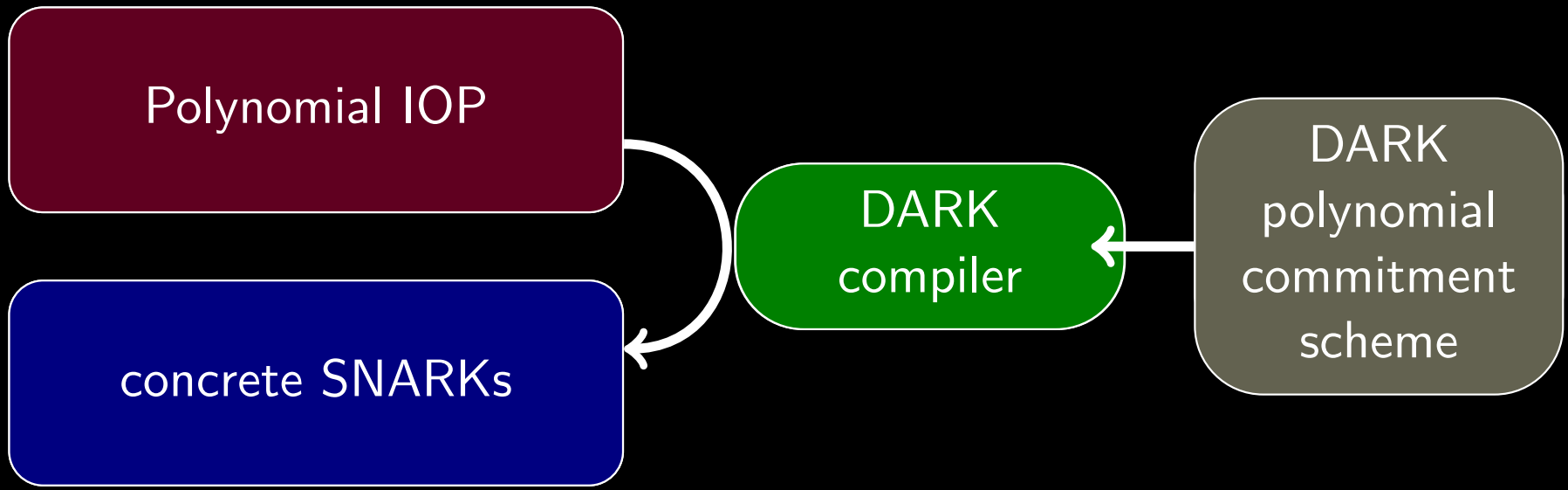
Contributions

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**

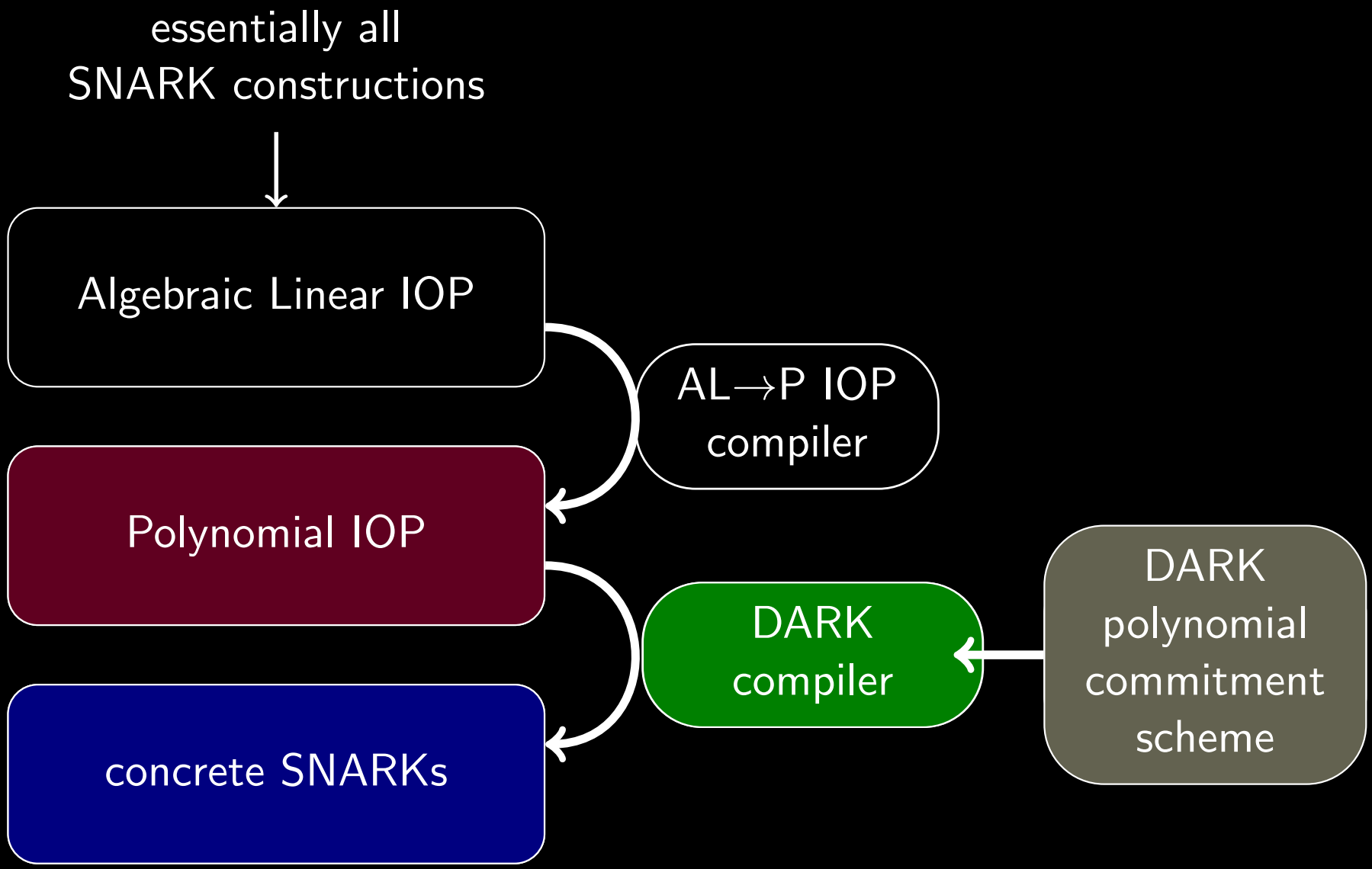


Contributions

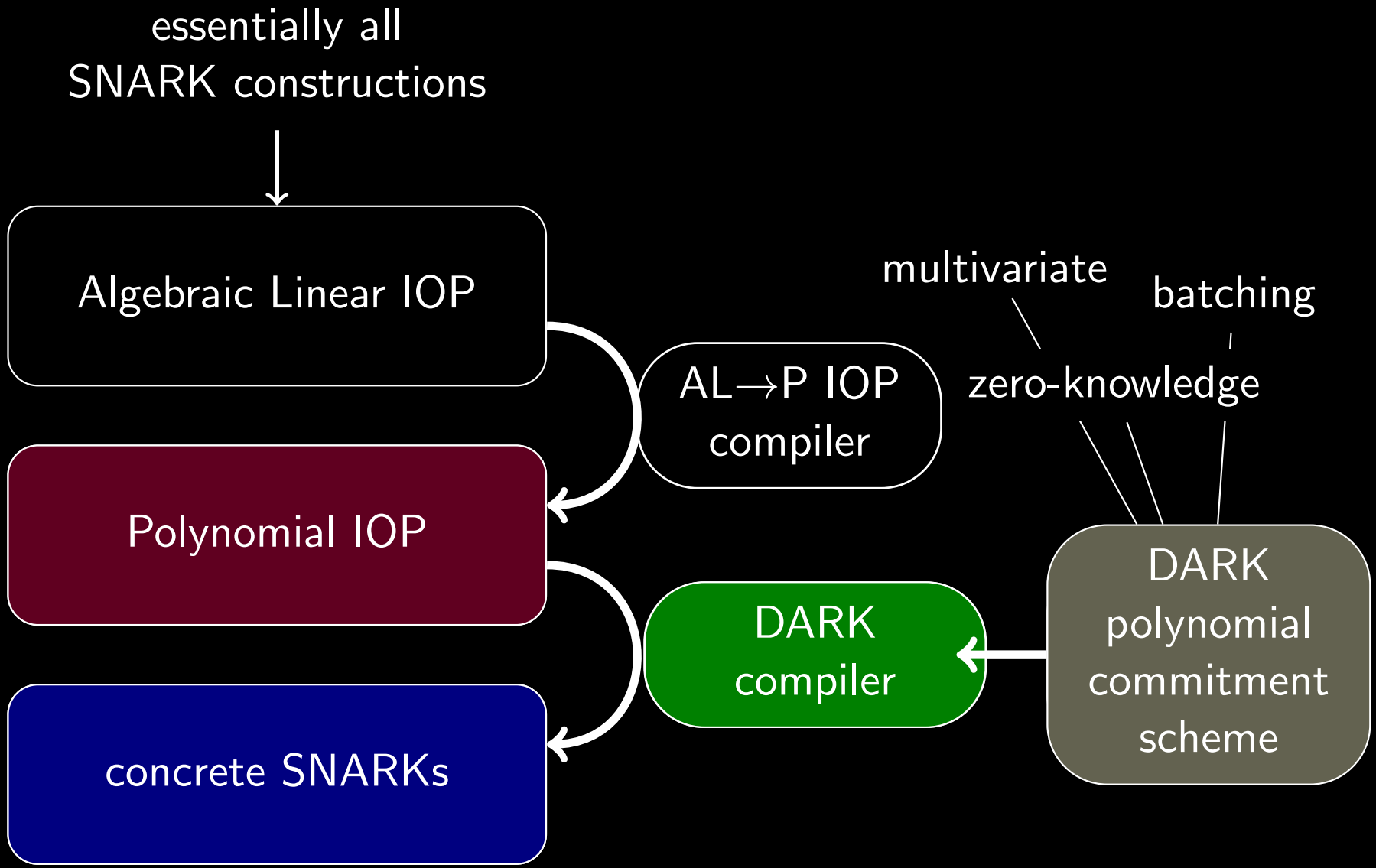
1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** $\leftarrow$ transparent SNARK (and other constructions)



Contributions

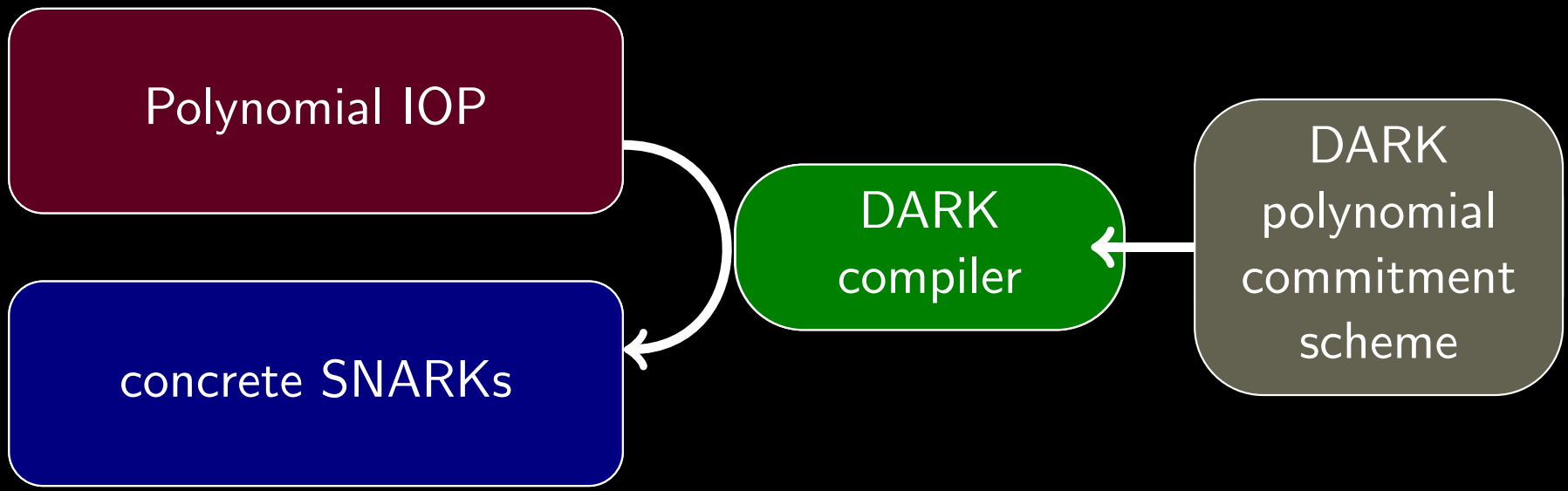


Contributions



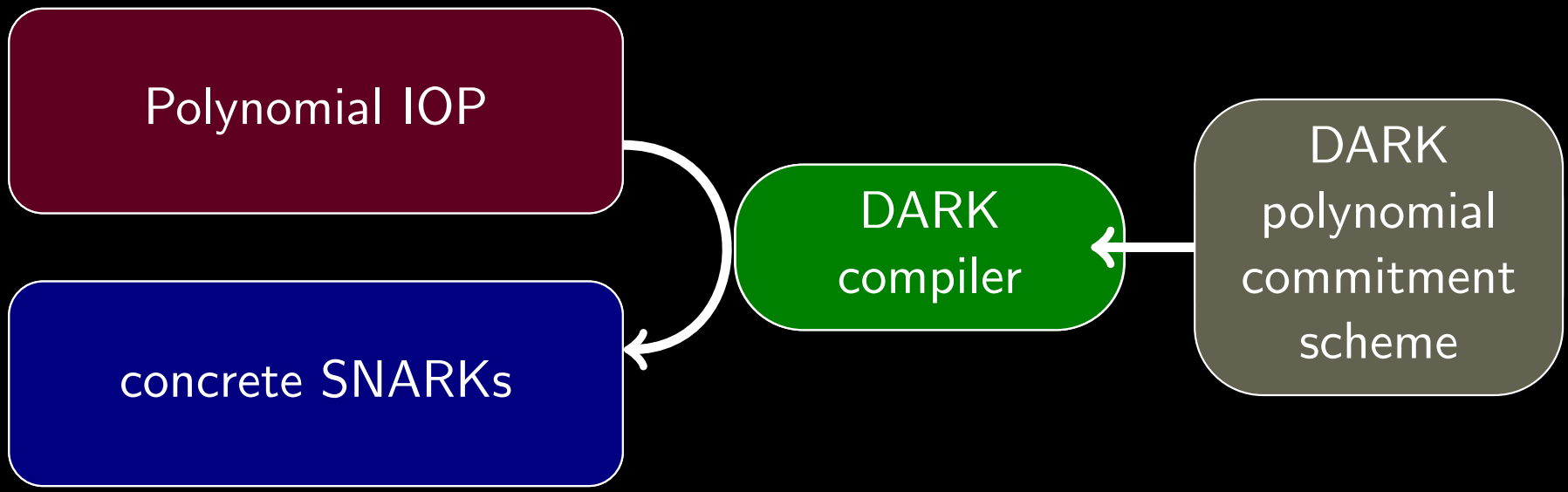
Talk Outline

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)

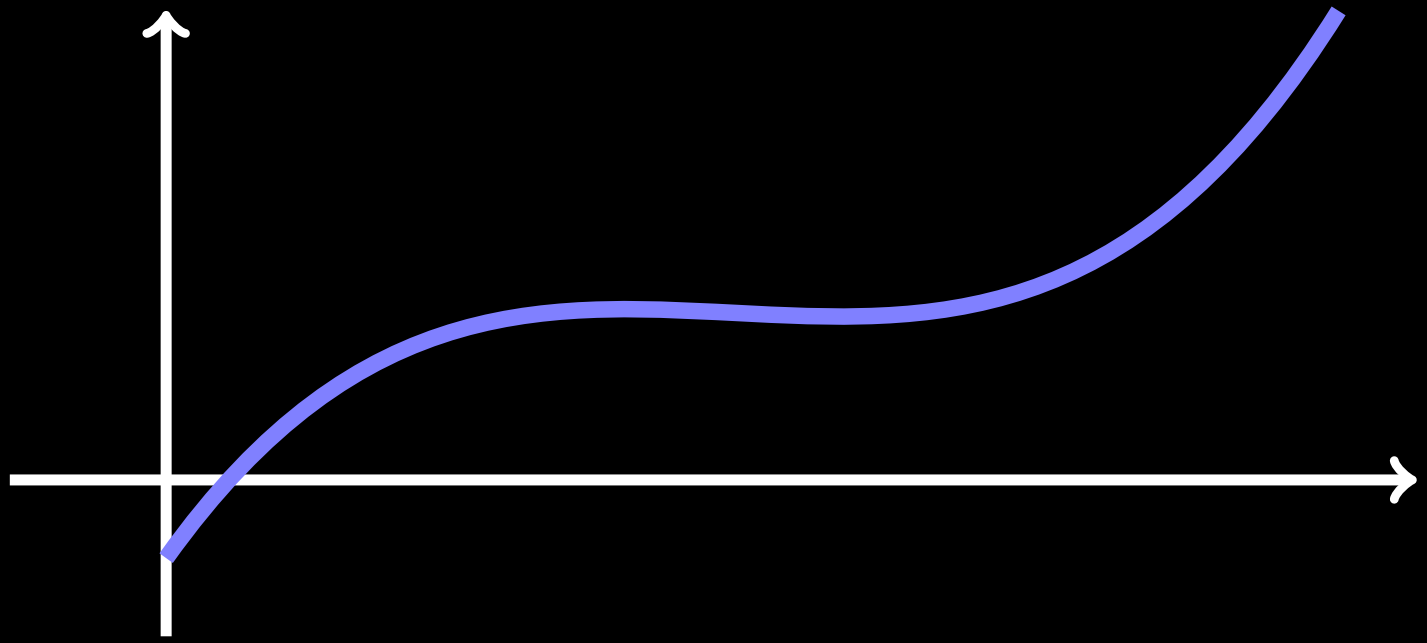


Talk Outline

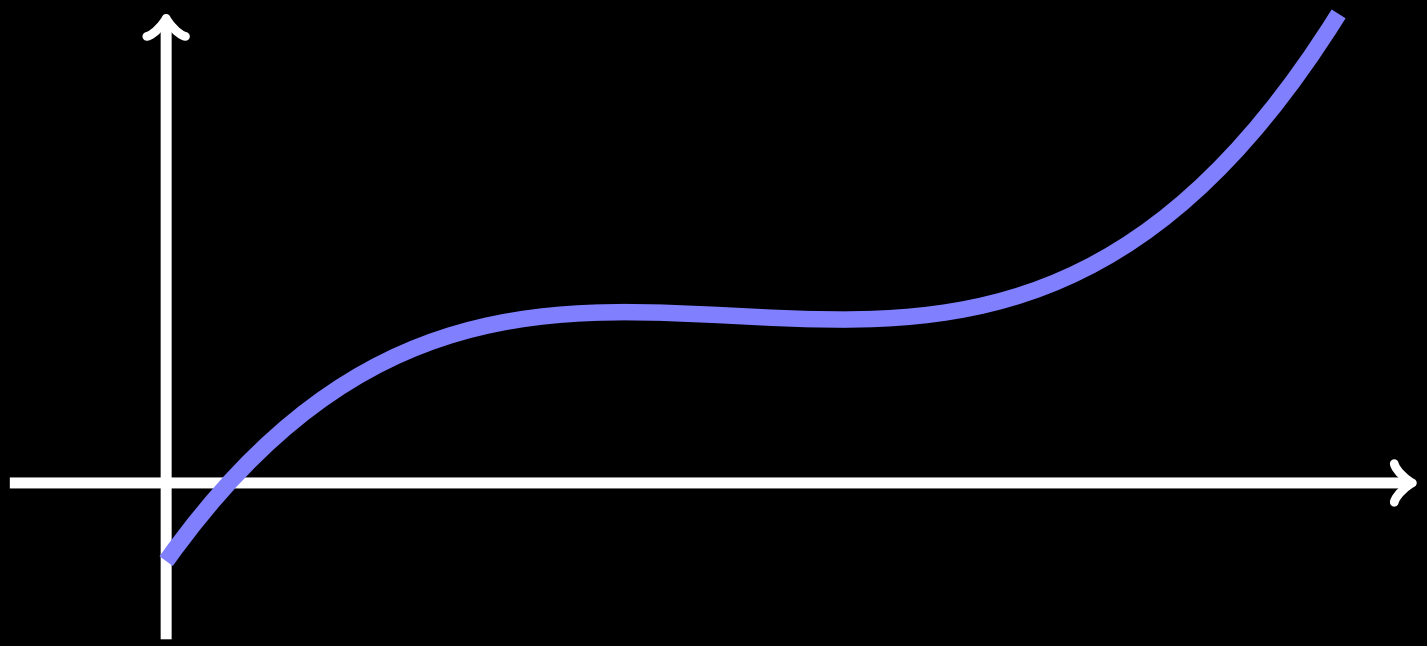
1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)



Polynomial Commitment Scheme

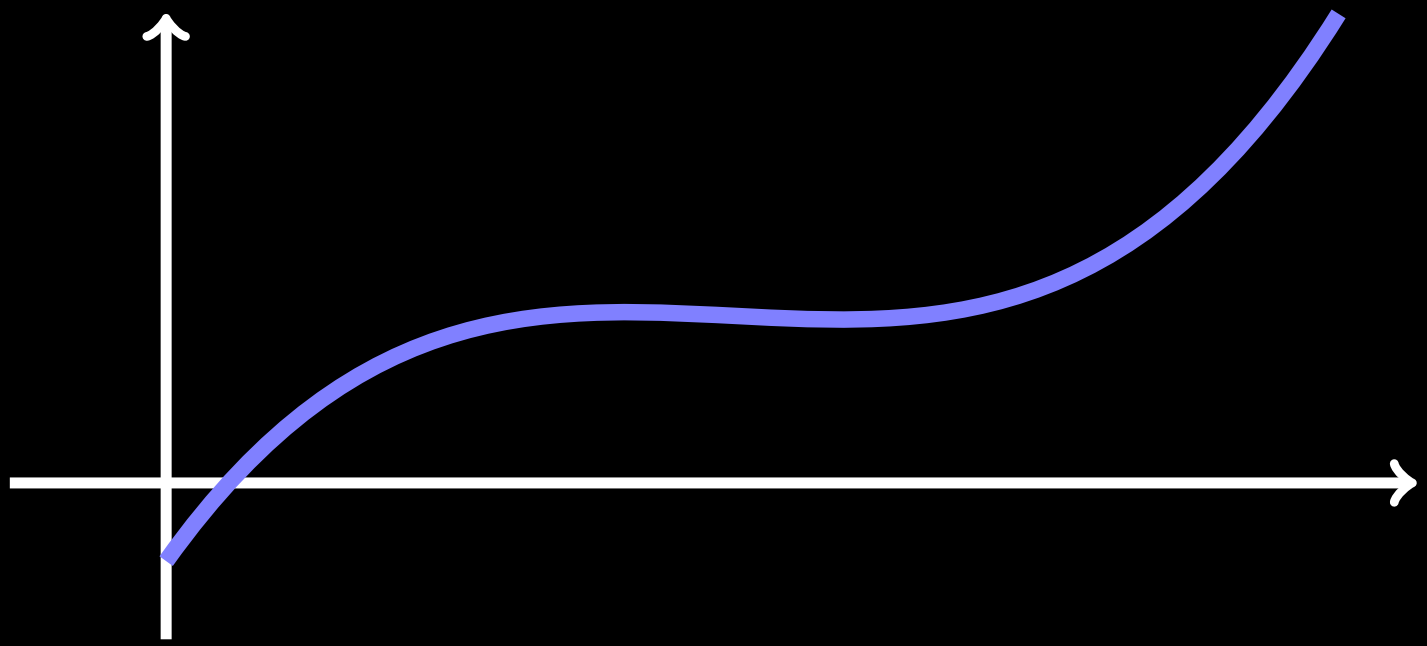


Polynomial Commitment Scheme

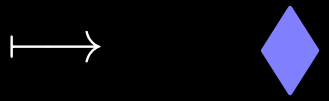


lots of coefficients

Polynomial Commitment Scheme

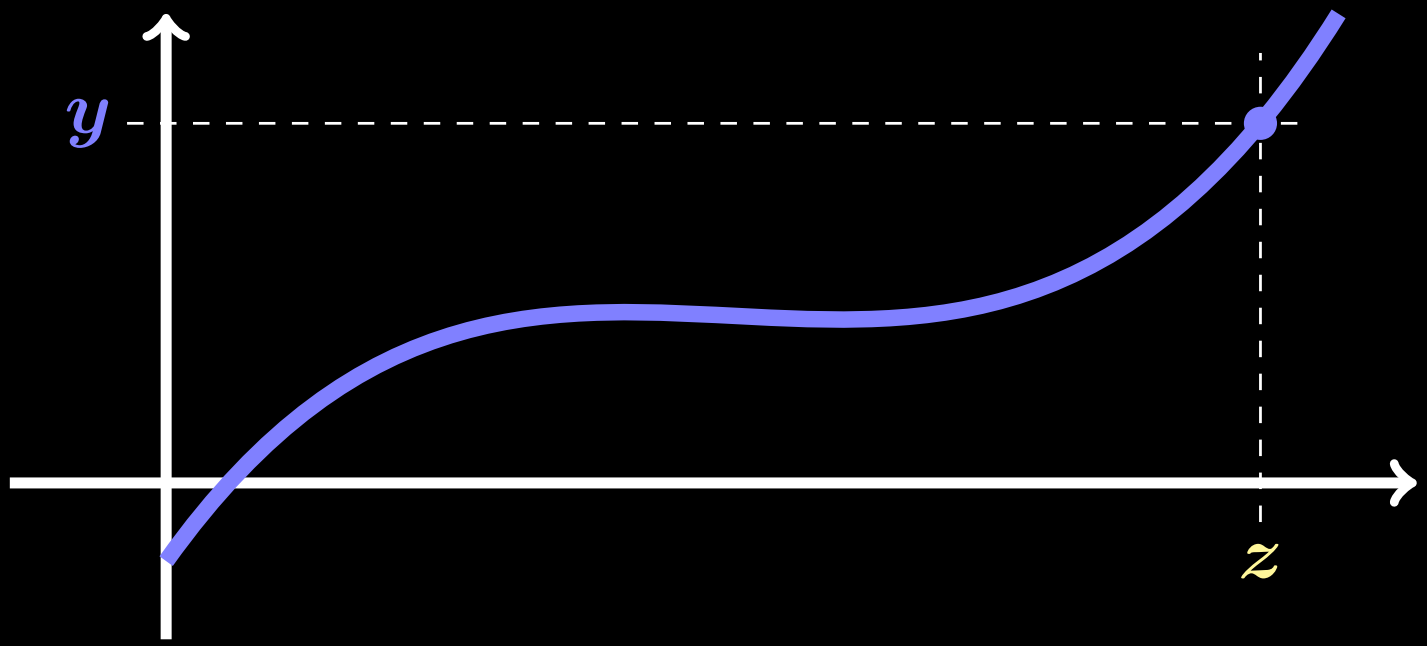


lots of coefficients



short commitment

Polynomial Commitment Scheme



lots of coefficients

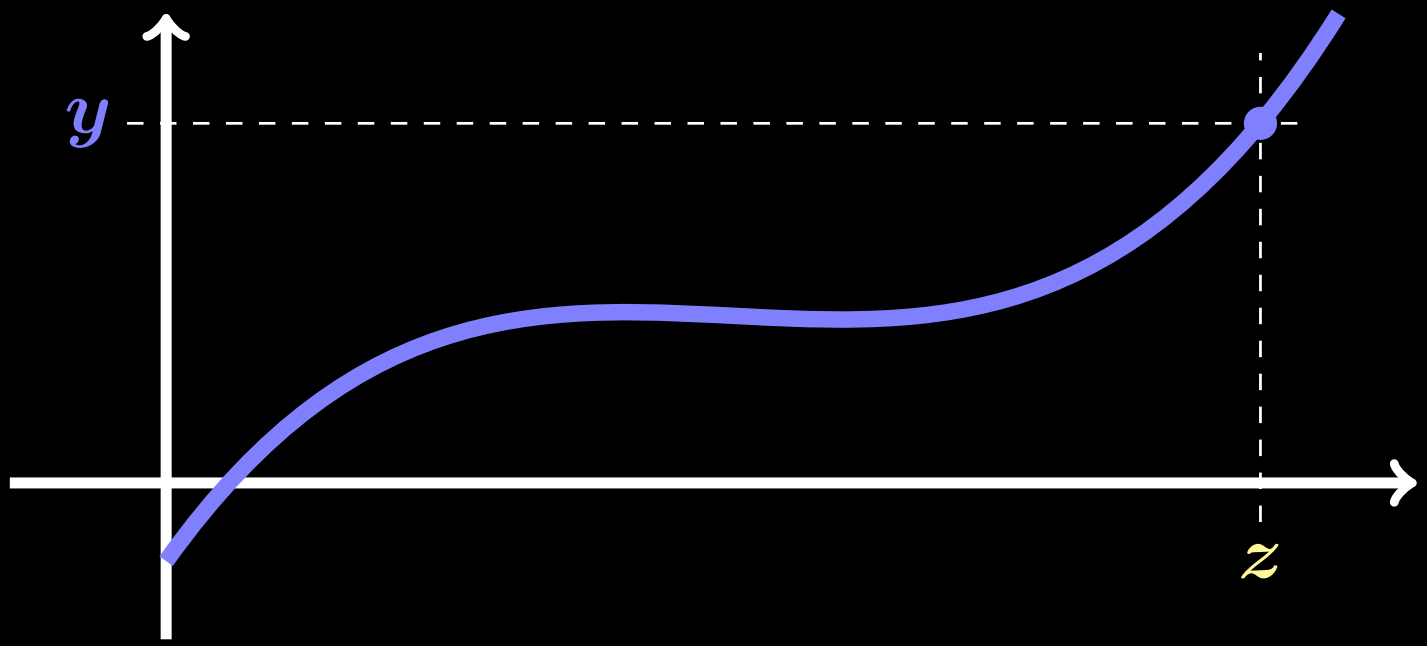
$\longmapsto$

short commitment

$f(z) = y$ $\longmapsto$ π

evaluation proof

Polynomial Commitment Scheme



lots of coefficients

$\longmapsto$

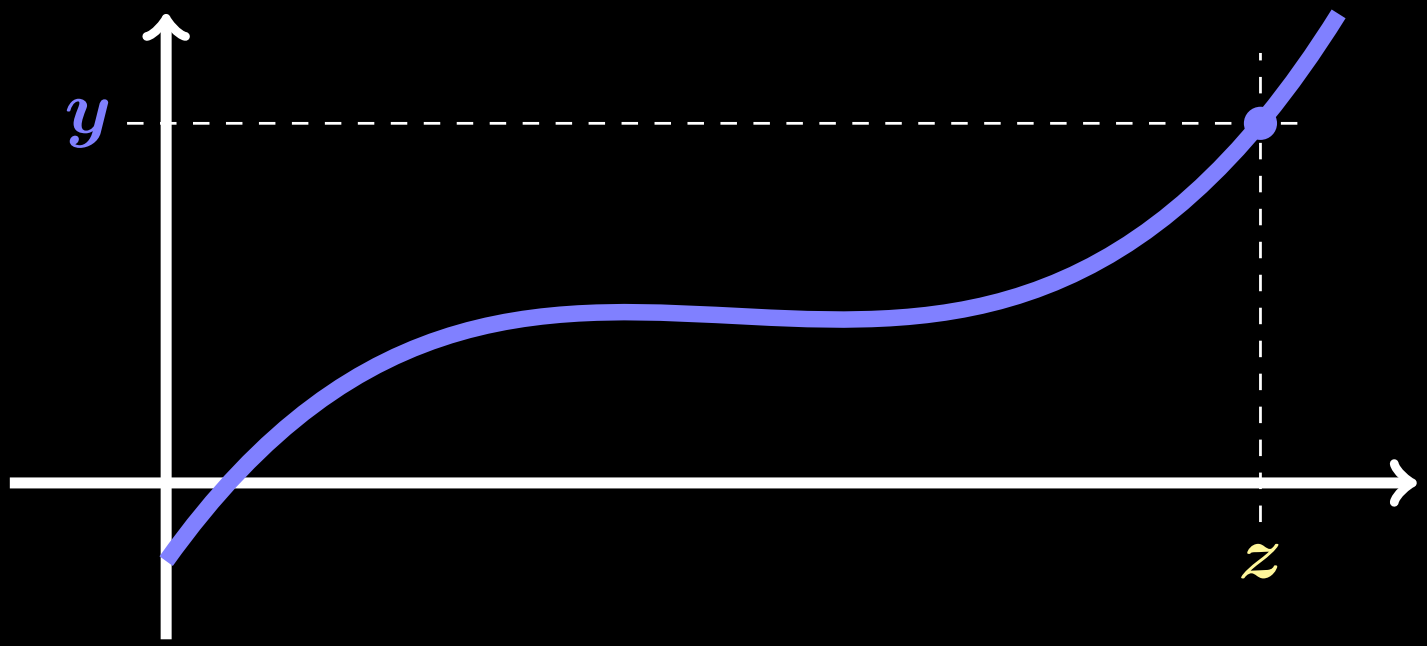
short commitment

$\leftarrow$ binding

$f(z) = y$ $\longmapsto$ π

evaluation proof

Polynomial Commitment Scheme



lots of coefficients

$\longmapsto$ ♦ $\longleftarrow$ binding

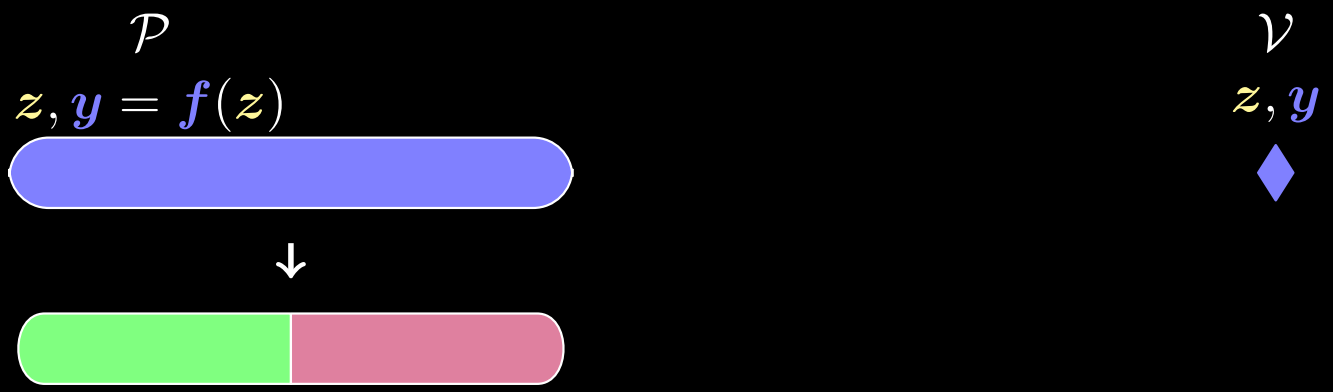
short commitment

$f(z) = y$ $\longmapsto$ π $\longleftarrow$ extractable

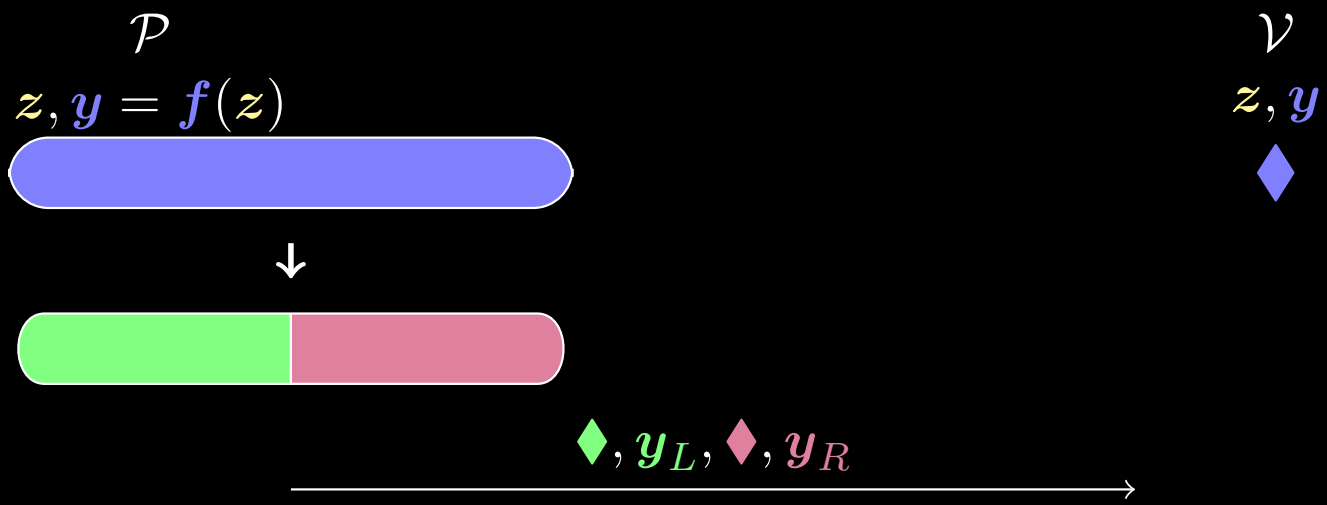
evaluation proof

Evaluation Protocol with Special Homomorphic Commitment

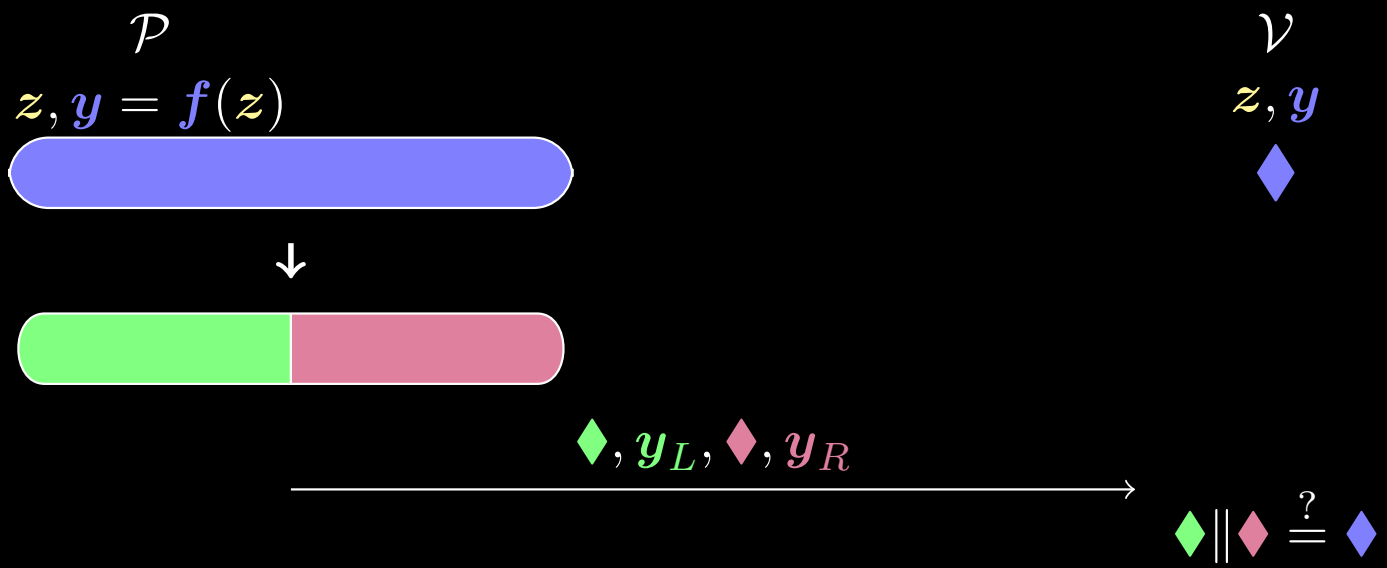
Evaluation Protocol with Special Homomorphic Commitment



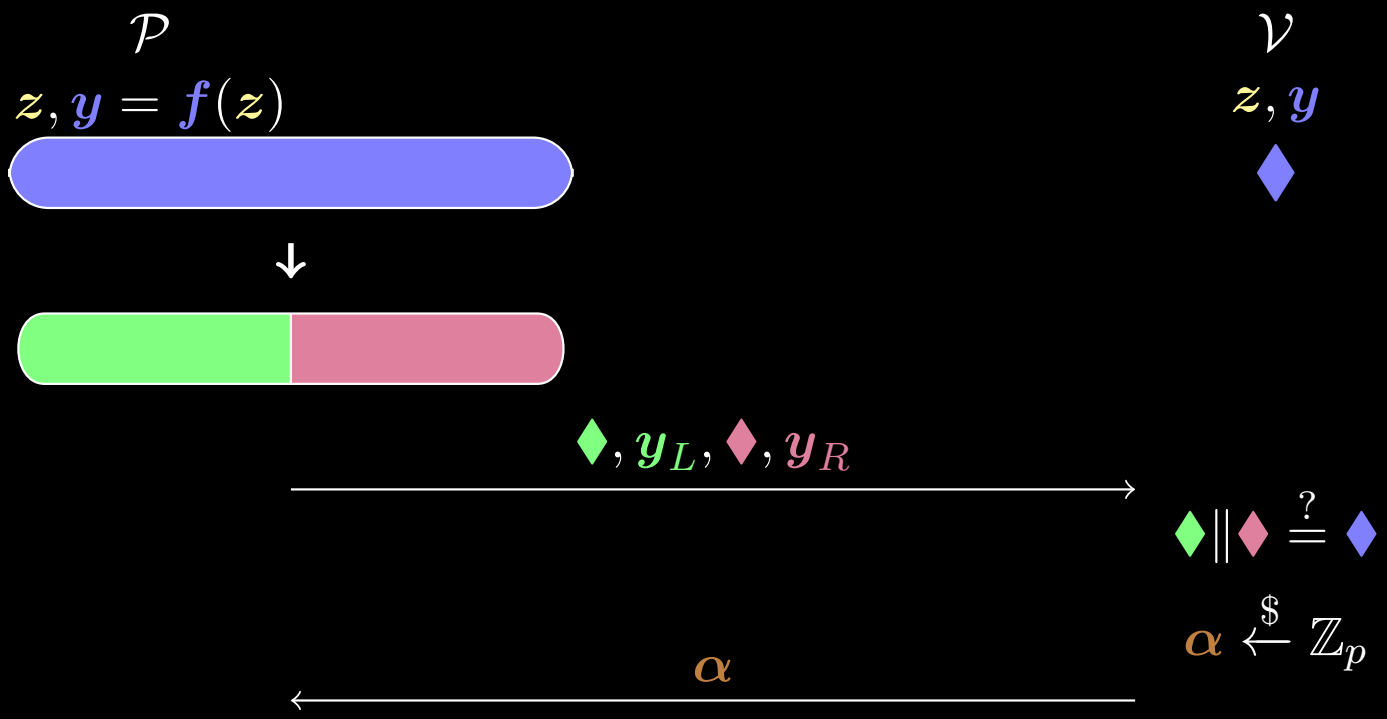
Evaluation Protocol with Special Homomorphic Commitment



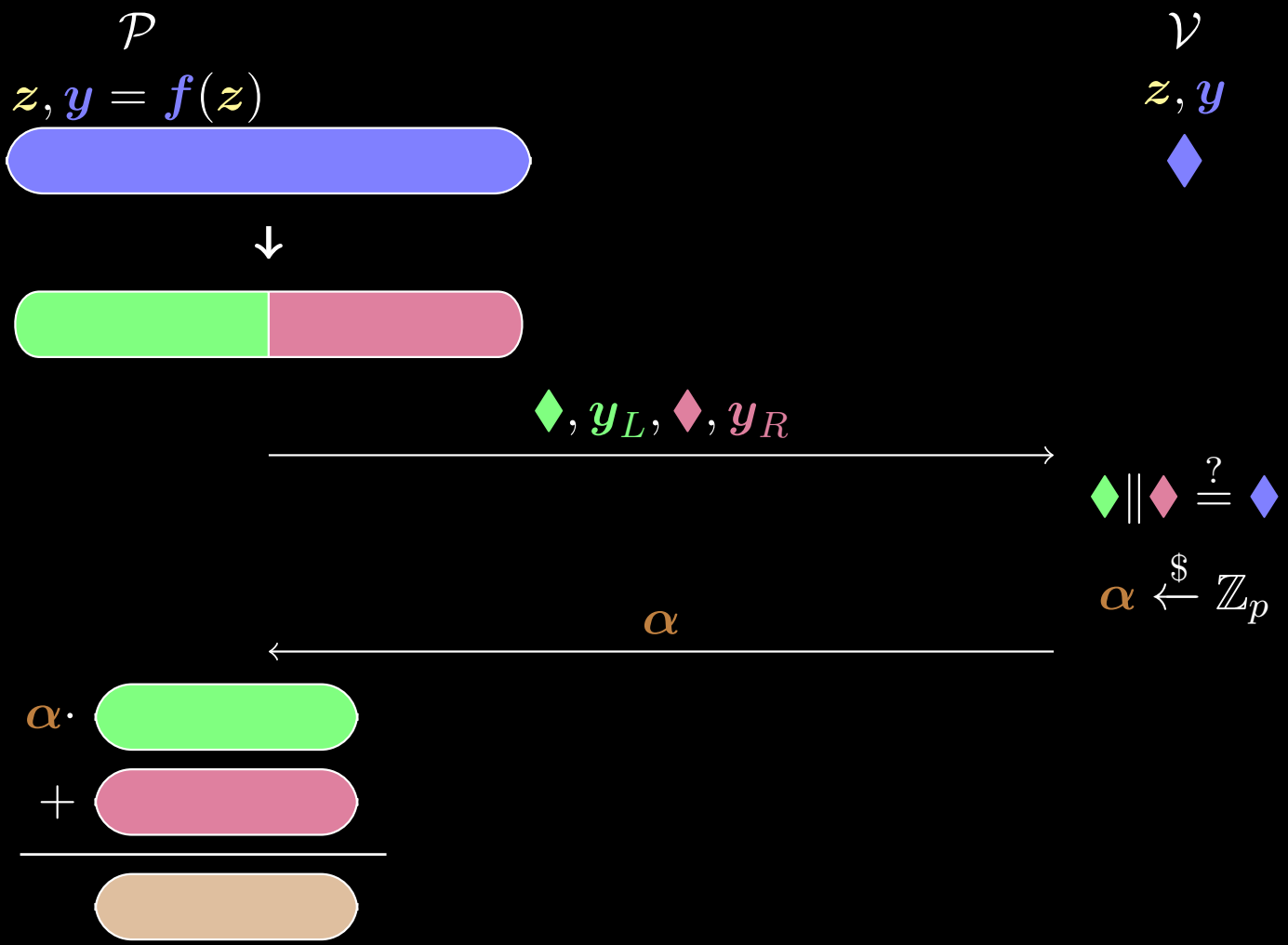
Evaluation Protocol with Special Homomorphic Commitment



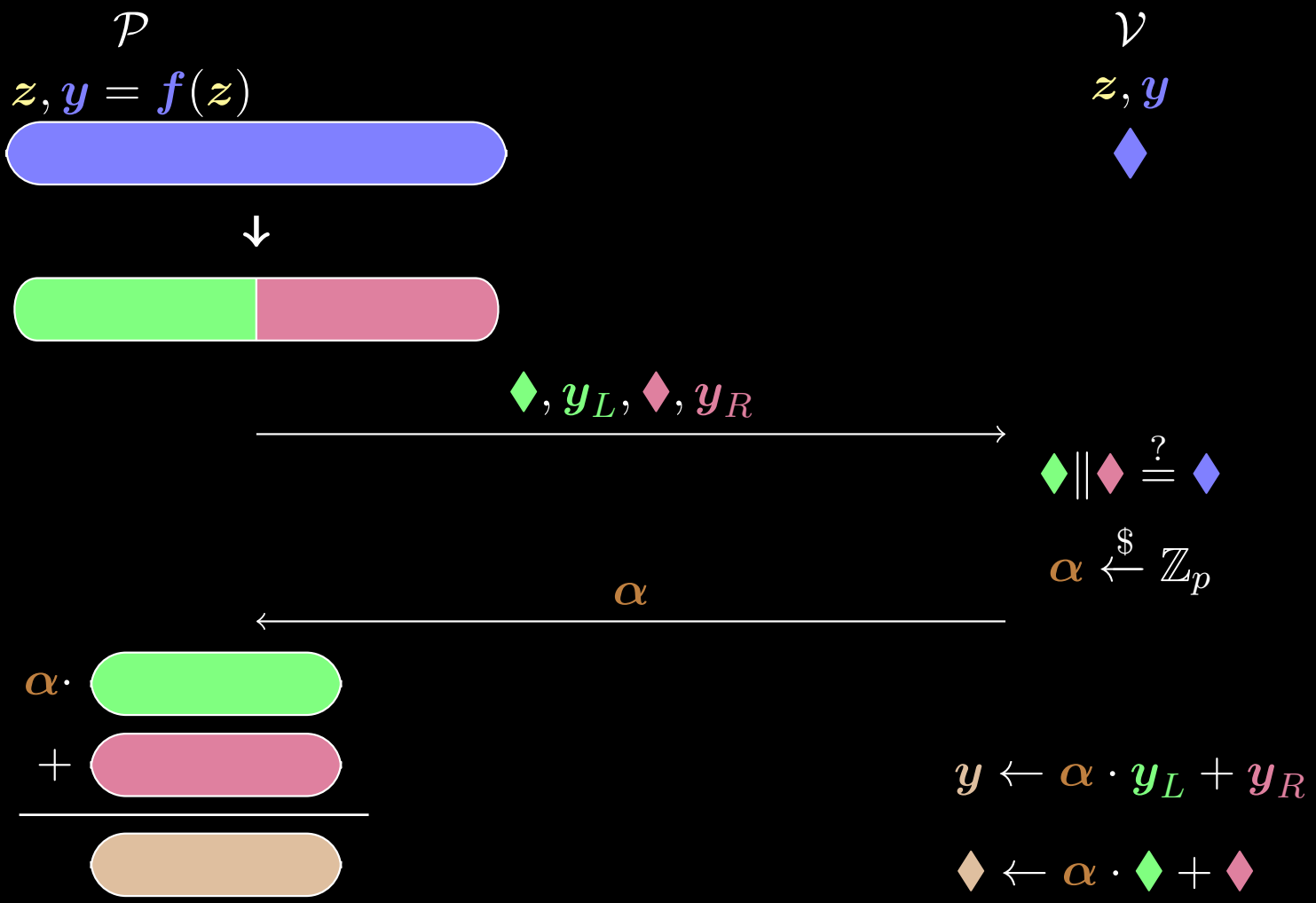
Evaluation Protocol with Special Homomorphic Commitment



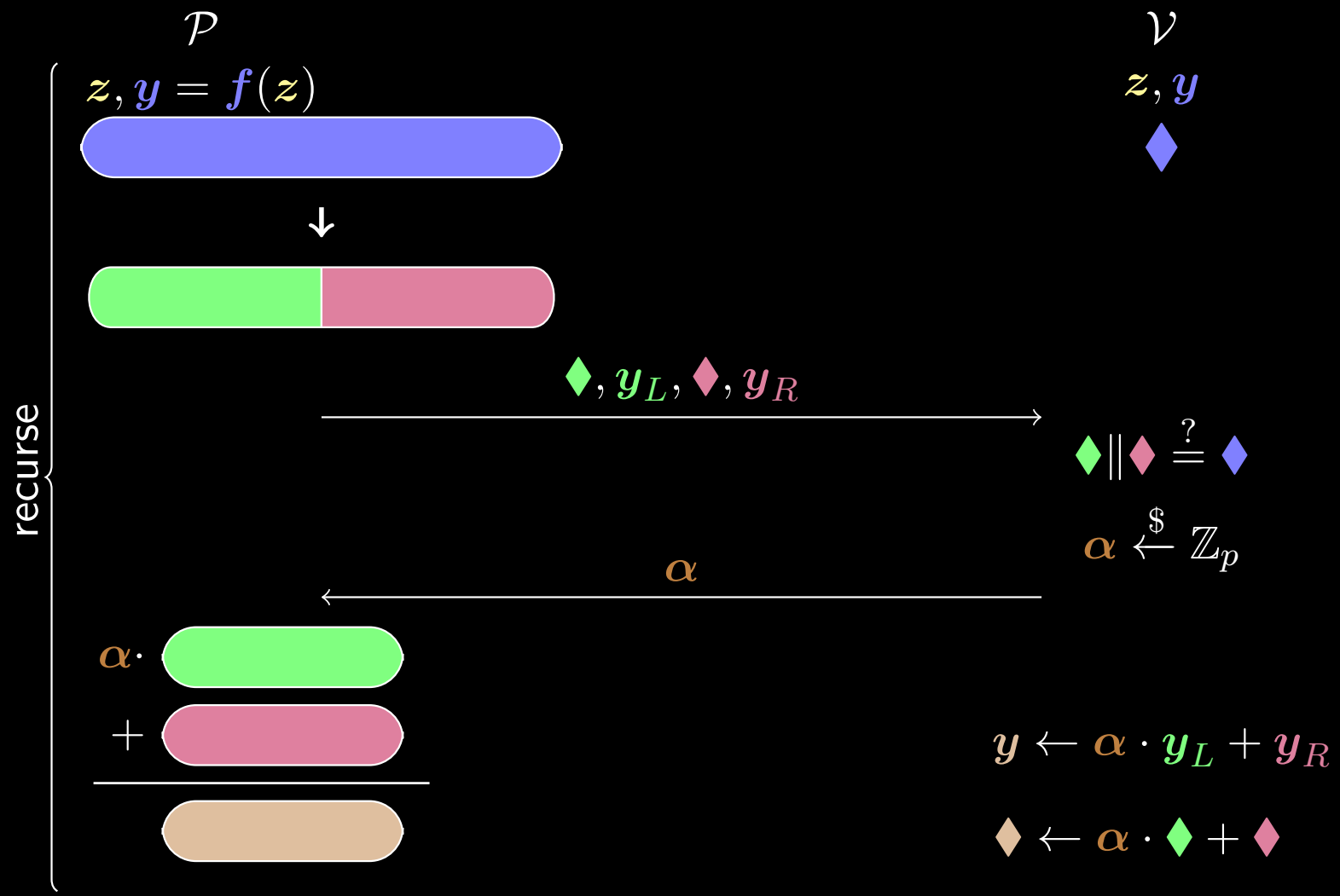
Evaluation Protocol with Special Homomorphic Commitment



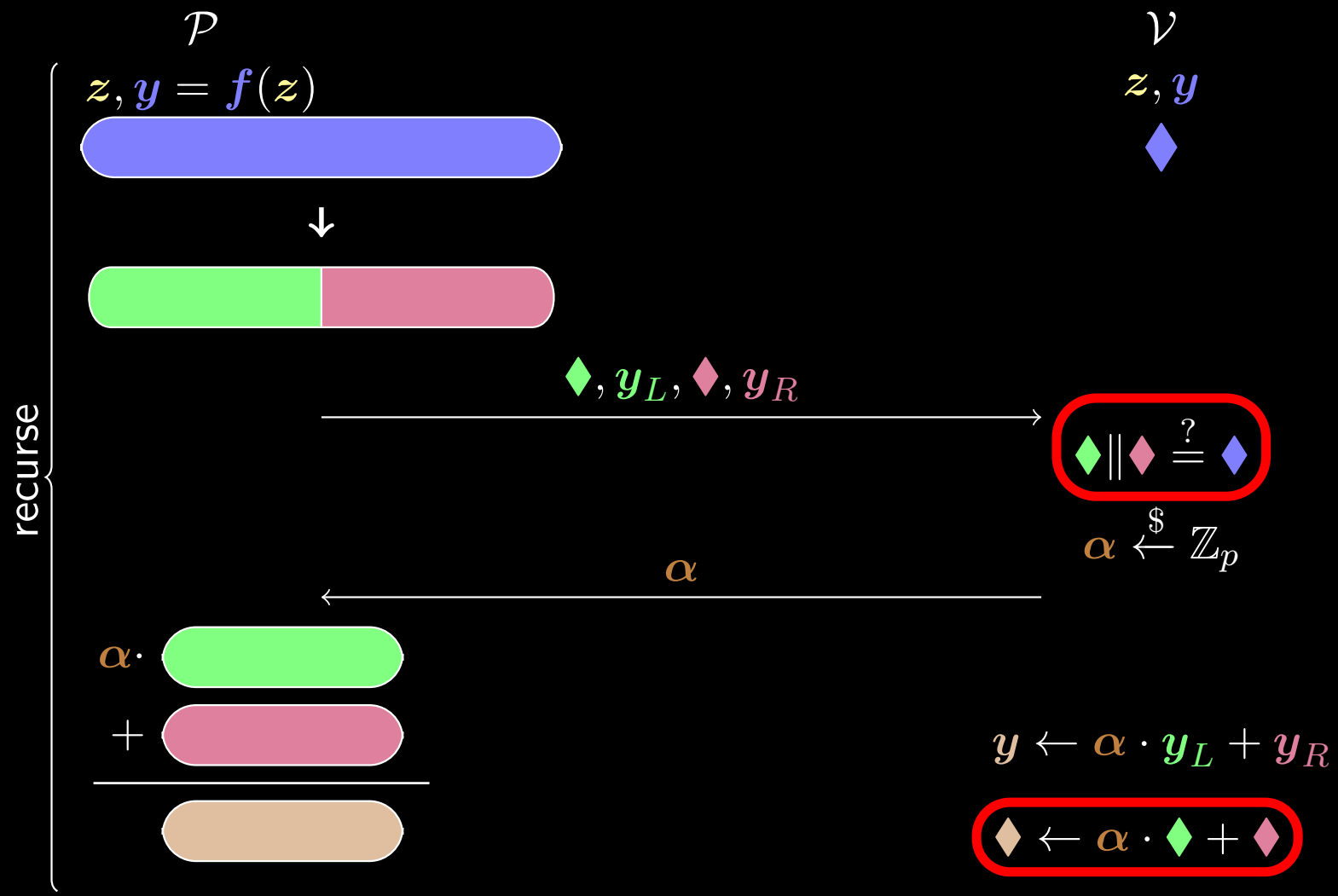
Evaluation Protocol with Special Homomorphic Commitment



Evaluation Protocol with Special Homomorphic Commitment

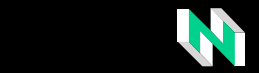


Evaluation Protocol with Special Homomorphic Commitment



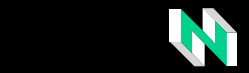
Special Homomorphic Commitment ♦

Special Homomorphic Commitment ♦



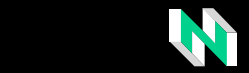
0. commitment to elements from infinite set

Special Homomorphic Commitment ♦



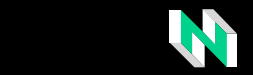
0. commitment to elements from infinite set
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$

Special Homomorphic Commitment ♦



0. commitment to elements from infinite set
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$

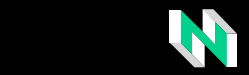
Special Homomorphic Commitment ♦



0. commitment to elements from infinite set
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$

?

Special Homomorphic Commitment ♦



0. commitment to elements from infinite set
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$

$\rightarrow$ Groups of Unknown Order

Groups of Unknown Order

RSA Group

- $\mathbb{Z}_n^*/\langle -1 \rangle, \times$ $n = pq$
- big
- trusted setup

Class Group

- $\mathcal{C}\ell(\Delta), \times$ of $\mathbb{Q}(\sqrt{\Delta})$
- big
- no trusted setup

Groups of Unknown Order

RSA Group

- $\mathbb{Z}_n^*/\langle -1 \rangle, \times$ $n = pq$
- big
- trusted setup

Class Group

- $\mathcal{C}\ell(\Delta), \times$ of $\mathbb{Q}(\sqrt{\Delta})$
- big
- no trusted setup

not post quantum

Hardness Assumptions

Strong RSA Assumption:

$$\frac{\begin{array}{l}\mathsf{g} \xleftarrow{\$} \mathbb{G} \\ (\mathsf{u}, \ell) \leftarrow \mathcal{A}(\mathsf{g})\end{array}}{\mathcal{A} \text{ wins} \Leftrightarrow \mathsf{u}^{\ell} = \mathsf{g}}$$

Adaptive Root Assumption:

$$\frac{\begin{array}{l}(\mathsf{g}, \mathsf{st}) \leftarrow \mathcal{A}_0(\mathbb{G}) \\ \ell \xleftarrow{\$} \mathsf{Primes}(\lambda 2^{\lambda}) \\ \mathsf{u} \leftarrow \mathsf{A}_1(\ell, \mathsf{st})\end{array}}{(\mathcal{A}_0, \mathcal{A}_1) \text{ wins} \Leftrightarrow \mathsf{u}^{\ell} = \mathsf{g}}$$

Hardness Assumptions

Strong RSA Assumption:

$$\frac{\begin{array}{l}\mathsf{g} \xleftarrow{\$} \mathbb{G} \\ (\mathsf{u}, \ell) \leftarrow \mathcal{A}(\mathsf{g})\end{array}}{\mathcal{A} \text{ wins} \Leftrightarrow \mathsf{u}^{\ell} = \mathsf{g}}$$

Adaptive Root Assumption:

$$\frac{\begin{array}{l}(\mathsf{g}, \mathsf{st}) \leftarrow \mathcal{A}_0(\mathbb{G}) \\ \ell \xleftarrow{\$} \mathsf{Primes}(\lambda 2^{\lambda}) \\ \mathsf{u} \leftarrow \mathsf{A}_1(\ell, \mathsf{st})\end{array}}{(\mathcal{A}_0, \mathcal{A}_1) \text{ wins} \Leftrightarrow \mathsf{u}^{\ell} = \mathsf{g}}$$

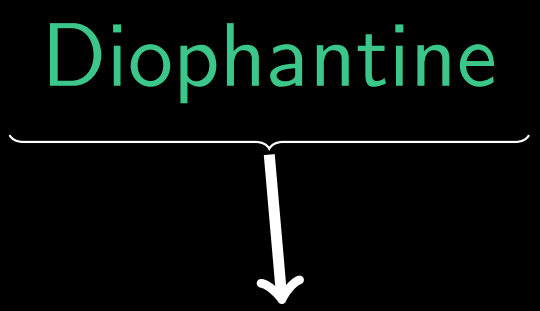
falsifiable ✓

What are GUOs good for?

What are GUOs good for?

Diophantine Arguments of Knowledge

What are GUOs good for?

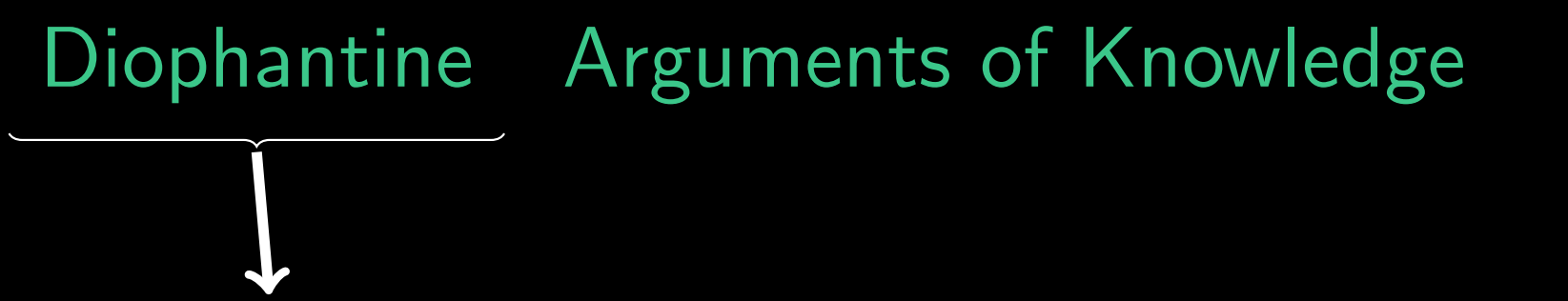


Arguments of Knowledge

polynomial equations

over the integers

What are GUOs good for?

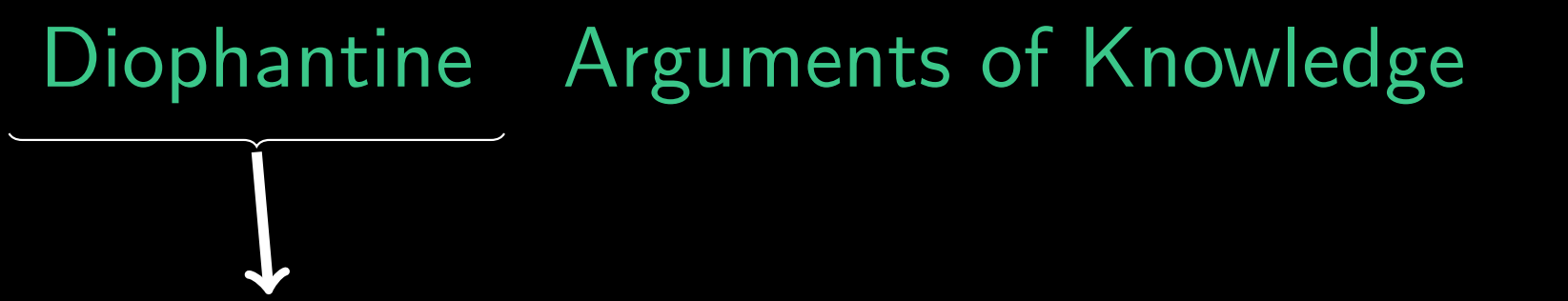


polynomial equations

over the integers

$a^n + b^n = c^n$ ✓

What are GUOs good for?



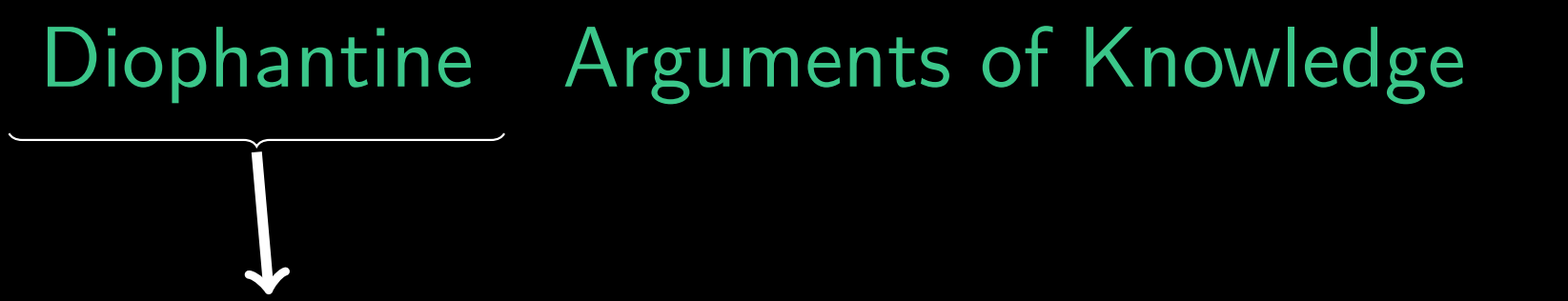
polynomial equations

over the integers

$a^n + b^n = c^n$ ✓

$x^2 - ny^2 = \pm 1$ ✓

What are GUOs good for?

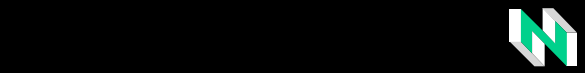


$a^n + b^n = c^n$ ✓

$x^2 - ny^2 = \pm 1$ ✓

$e^\pi = (-1)^i$ ×

What are GUOs good for?



Diophantine Arguments of Knowledge

polynomial equations

over the integers

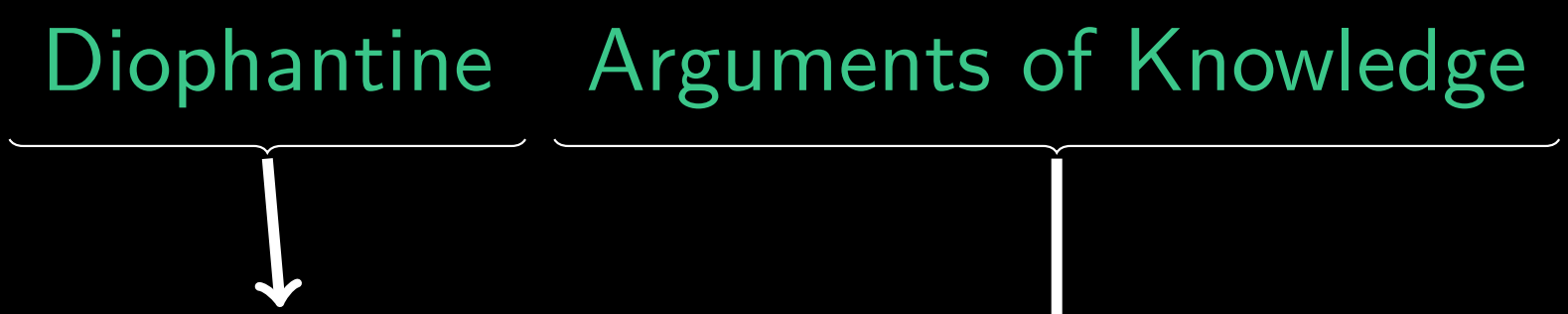
$a^n + b^n = c^n$ ✓

$x^2 - ny^2 = \pm 1$ ✓

$e^\pi = (-1)^i$ ×

$\lceil e \rceil - \lfloor \pi \rfloor = 0$ ✓

What are GUOs good for?



polynomial equations

over the integers

"I know $\mathbf{y} \in \mathbb{Z}^m$ such that $\mathcal{P}(\mathbf{x}, \mathbf{y}) = 0$"

$a^n + b^n = c^n$ ✓

$x^2 - ny^2 = \pm 1$ ✓

$e^\pi = (-1)^i$ ×

$\lceil e \rceil - \lfloor \pi \rfloor = 0$ ✓

Integer Encoding

Integer Encoding

solution:

example:

Integer Encoding

have:
arguments for $\mathbb{Z}$

want:
arguments for $\mathbb{F}_p[X]$

solution:

example:

$$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$$

Integer Encoding

have:
arguments for $\mathbb{Z}$

$\xrightarrow{?}$

want:
arguments for $\mathbb{F}_p[X]$

solution:

example:

$$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$$

1. choose integer $q \gg p$

Integer Encoding

solution:

1. choose integer $q \gg p$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

Integer Encoding

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$

example:

$$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$$

$$q = 10$$

Integer Encoding

have:
arguments for $\mathbb{Z}$

$\xrightarrow{?}$

want:
arguments for $\mathbb{F}_p[X]$

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{Z}[X]$

Integer Encoding

have:
arguments for $\mathbb{Z}$

$\xrightarrow{?}$

want:
arguments for $\mathbb{F}_p[X]$

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$
3. evaluate: $\hat{f}(q)$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{Z}[X]$

Integer Encoding

have:
arguments for $\mathbb{Z}$

want:
arguments for $\mathbb{F}_p[X]$

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$
3. evaluate: $\hat{f}(q)$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{Z}[X]$

2341

Integer Encoding

have: arguments for $\mathbb{Z}$ $\xrightarrow{?}$ want: arguments for $\mathbb{F}_p[X]$

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$
3. evaluate: $\hat{f}(q)$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{Z}[X]$

2341

homomorphic properties:

Integer Encoding

have:
arguments for $\mathbb{Z}$

want:
arguments for $\mathbb{F}_p[X]$

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$
3. evaluate: $\hat{f}(q)$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{Z}[X]$

2341

homomorphic properties:

$$a \cdot \hat{f}(q) + b \cdot \hat{g}(q) =^* \widehat{(a \cdot f + b \cdot g)}(q)$$

Integer Encoding

have: arguments for $\mathbb{Z}$ $\xrightarrow{?}$ want: arguments for $\mathbb{F}_p[X]$

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$
3. evaluate: $\hat{f}(q)$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{Z}[X]$

2341

homomorphic properties:

$$a \cdot \hat{f}(q) + b \cdot \hat{g}(q) =^* \widehat{(a \cdot f + b \cdot g)}(q)$$

$$\hat{f}(q) \cdot \hat{g}(q) =^* \widehat{(f \cdot g)}(q)$$

Integer Encoding

have: arguments for $\mathbb{Z}$ $\xrightarrow{?}$ want: arguments for $\mathbb{F}_p[X]$

solution:

1. choose integer $q \gg p$
2. lift $f(X) \mapsto \hat{f}(X) \in \mathbb{Z}[X]$
3. evaluate: $\hat{f}(q)$

example:

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{F}_5[X]$

$q = 10$

$2X^3 + 3X^2 + 4X + 1 \in \mathbb{Z}[X]$

2341

homomorphic properties:

$$a \cdot \hat{f}(q) + b \cdot \hat{g}(q) =^* \widehat{(a \cdot f + b \cdot g)}(q)$$

$$\hat{f}(q) \cdot \hat{g}(q) =^* \widehat{(f \cdot g)}(q)$$

*: if q is large enough

Commitment

0. commitment
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$

Commitment

0. commitment $\longrightarrow \mathsf{com}(f(X)) := \mathrm{g}^{\hat{f}(q)}$ for $\mathrm{g} \in \mathbb{G}$
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$

Commitment

0. commitment $\longrightarrow \mathsf{com}(f(X)) := \mathrm{g}^{\hat{f}(q)}$ for $\mathrm{g} \in \mathbb{G}$
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$ $\longrightarrow$ **free** (large q)
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$

Commitment

0. commitment $\longrightarrow \mathsf{com}(f(X)) := \mathrm{g}^{\hat{f}(q)}$ for $\mathrm{g} \in \mathbb{G}$
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$ $\longrightarrow$ **free** (large q)
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$ $\mathrm{g}^{\hat{f}(q)} \stackrel{?}{=} \mathrm{g}^{\hat{f}_L(q)} \cdot (\mathrm{g}^{\hat{f}_R(q)})^{q^{d'+1}}$

Commitment

0. commitment $\longrightarrow \mathsf{com}(f(X)) := \mathsf{g}^{\hat{f}(q)}$ for $\mathsf{g} \in \mathbb{G}$
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$ $\longrightarrow$ **free** (large q)
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$ $\mathsf{g}^{\hat{f}(q)} \stackrel{?}{=} \mathsf{g}^{\hat{f}_L(q)} \cdot (\mathsf{g}^{\hat{f}_R(q)})^{q^{d'+1}}$

$\longrightarrow$ prover sends $\mathsf{C}_R = \mathsf{g}^{\hat{f}_R(q)}$ and $\mathsf{C}_R^\star = \mathsf{C}_R^{q^{d'+1}}$

$\longrightarrow$ verifier checks $\mathsf{C}_R^\star = \mathsf{C}_R^{q^{d'+1}}$

Commitment

0. commitment $\longrightarrow \mathsf{com}(f(X)) := \mathsf{g}^{\hat{f}(q)}$ for $\mathsf{g} \in \mathbb{G}$
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$ $\longrightarrow$ **free** (large q)
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$ $\mathsf{g}^{\hat{f}(q)} \stackrel{?}{=} \mathsf{g}^{\hat{f}_L(q)} \cdot (\mathsf{g}^{\hat{f}_R(q)})^{q^{d'+1}}$

$\longrightarrow$ prover sends $\mathsf{C}_R = \mathsf{g}^{\hat{f}_R(q)}$ and $\mathsf{C}_R^\star = \mathsf{C}_R^{q^{d'+1}}$

$\longrightarrow$ verifier checks $\mathsf{C}_R^\star = \mathsf{C}_R^{q^{d'+1}}$ $\$\$\$$

Commitment

0. commitment $\longrightarrow \mathsf{com}(f(X)) := \mathsf{g}^{\hat{f}(q)}$ for $\mathsf{g} \in \mathbb{G}$
1. compute linear relations: $\blacklozenge \leftarrow \alpha \cdot \blacklozenge + \blacklozenge$ $\longrightarrow$ **free** (large q)
2. verify monomial relations: $\blacklozenge \| \blacklozenge \stackrel{?}{=} \blacklozenge$ $\mathsf{g}^{\hat{f}(q)} \stackrel{?}{=} \mathsf{g}^{\hat{f}_L(q)} \cdot (\mathsf{g}^{\hat{f}_R(q)})^{q^{d'+1}}$

$\longrightarrow$ prover sends $\mathsf{C}_R = \mathsf{g}^{\hat{f}_R(q)}$ and $\mathsf{C}_R^\star = \mathsf{C}_R^{q^{d'+1}}$

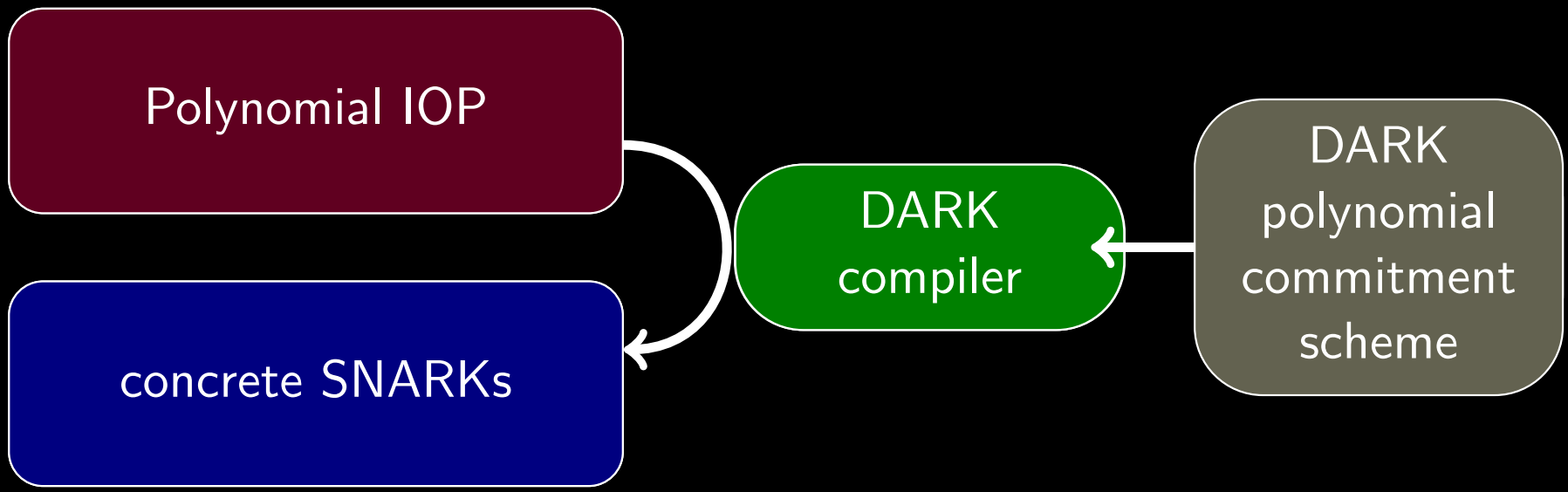
$\longrightarrow$ verifier checks $\mathsf{C}_R^\star = \mathsf{C}_R^{q^{d'+1}}$ $\$\$\$$

Wesolowski 2019: PoE (proof of exponentiation)

$\mathcal{P}$	claim: $\mathsf{u}^x = \mathsf{v}$	$\mathcal{V}$
		$\ell \stackrel{\$}{\leftarrow} \mathsf{Primes}(\lambda \cdot 2^{2\lambda})$
$q = \lfloor x/\ell \rfloor \quad r = x \bmod \ell$		
$\mathsf{Q} = \mathsf{u}^q$		
		$\mathsf{Q}^\ell \mathsf{u}^r \stackrel{?}{=} \mathsf{v}$

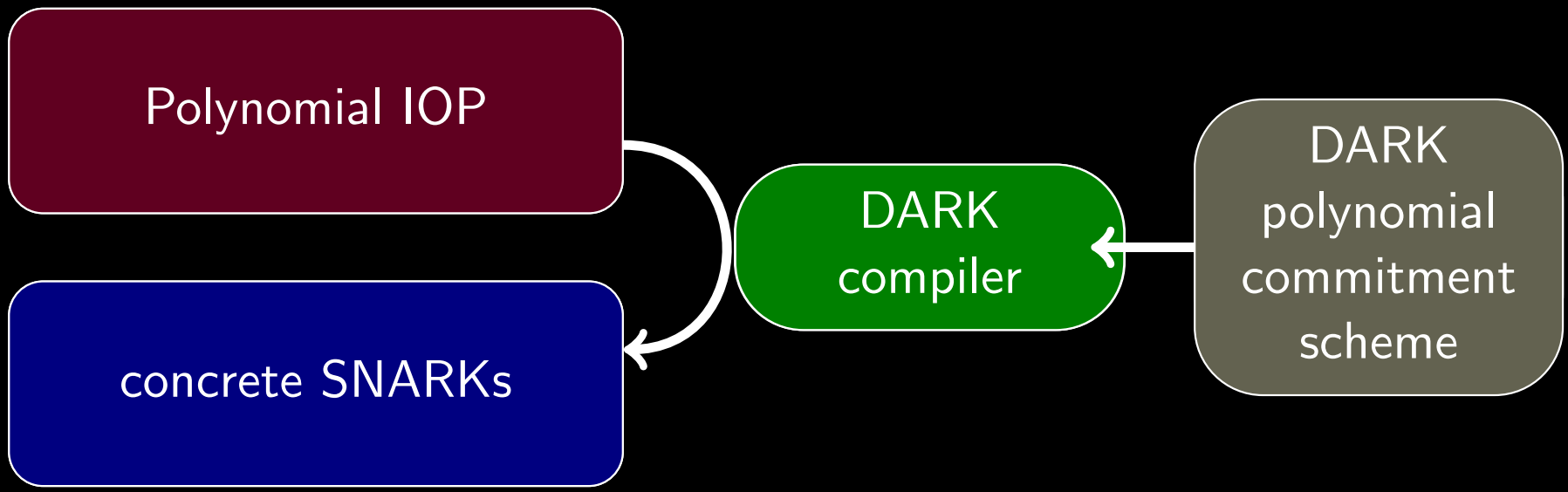
Talk Outline

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)



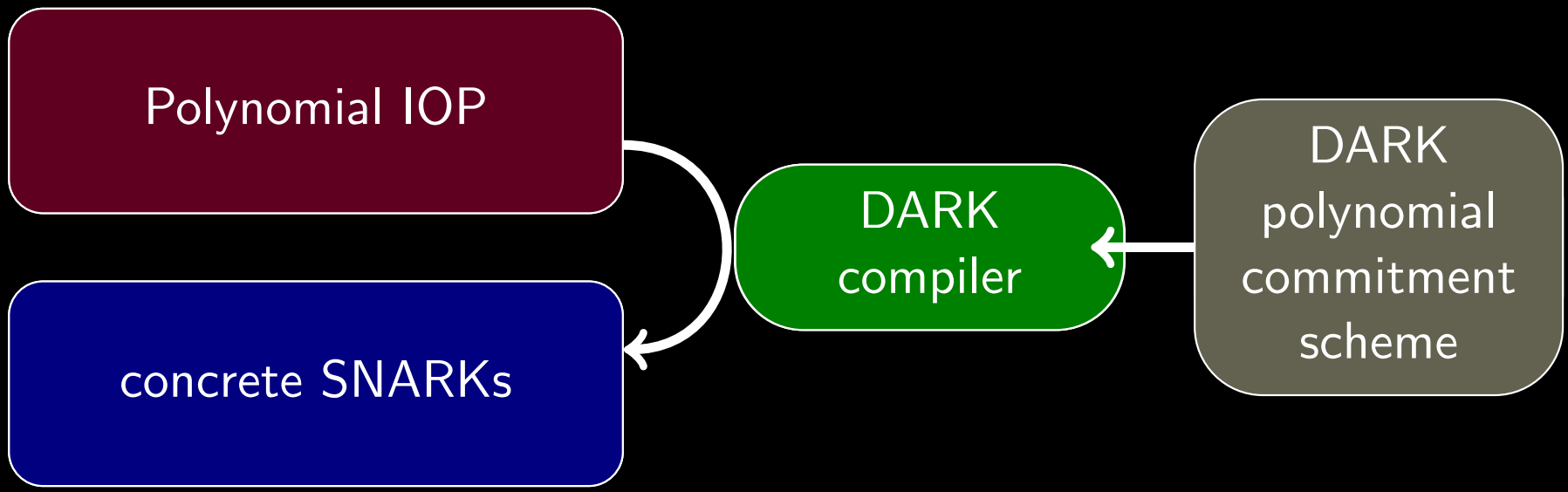
Talk Outline

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)

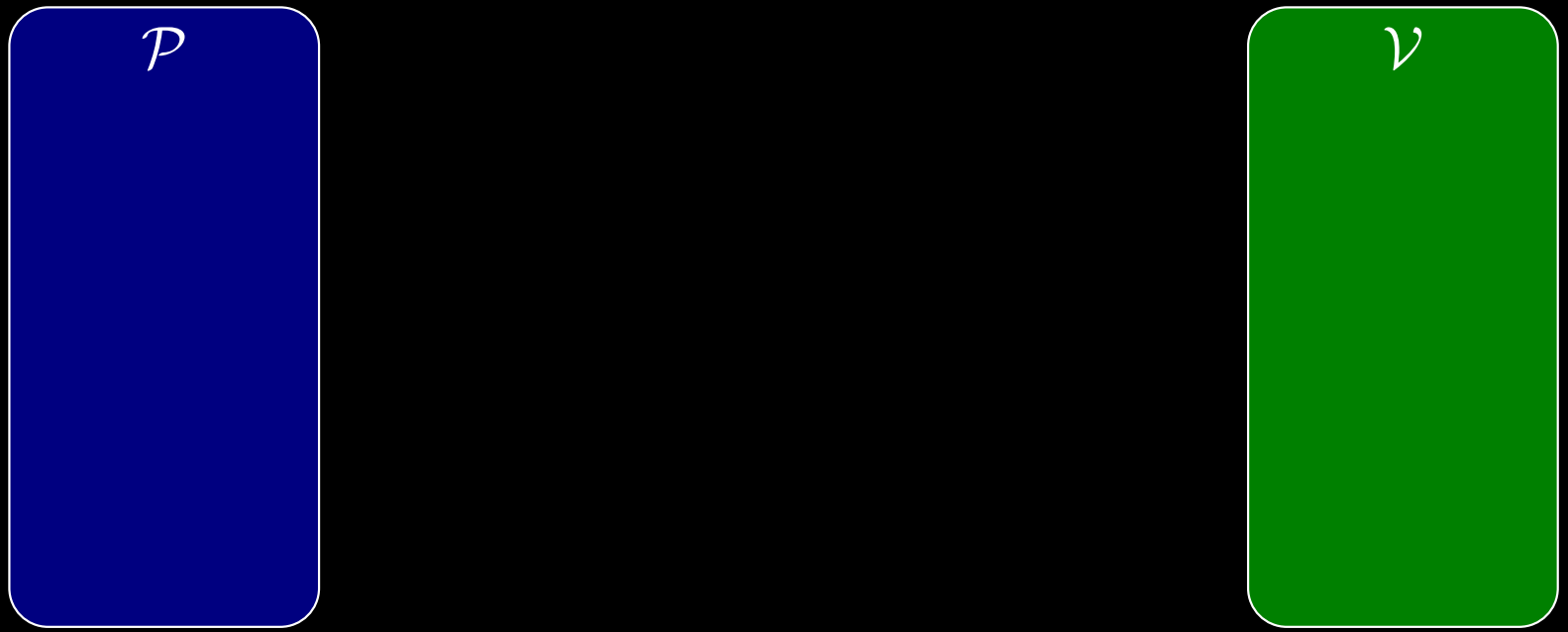


Talk Outline

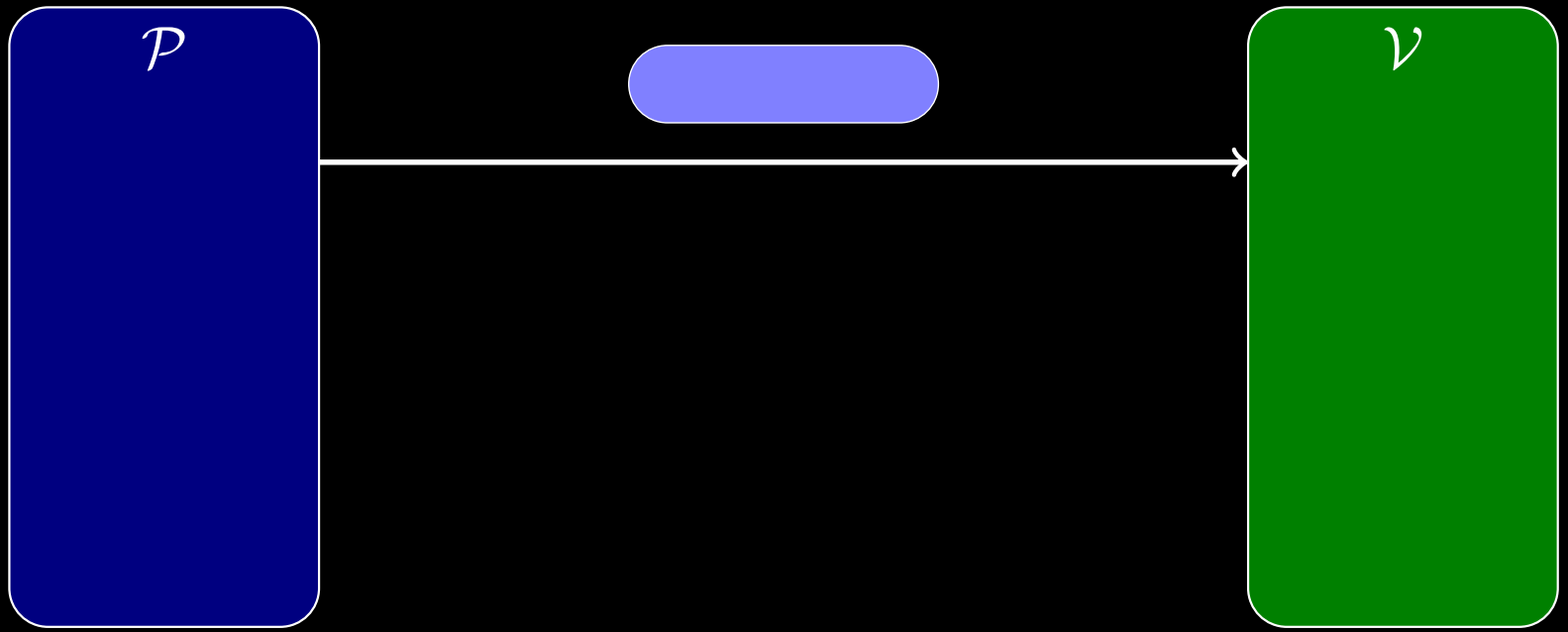
1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)



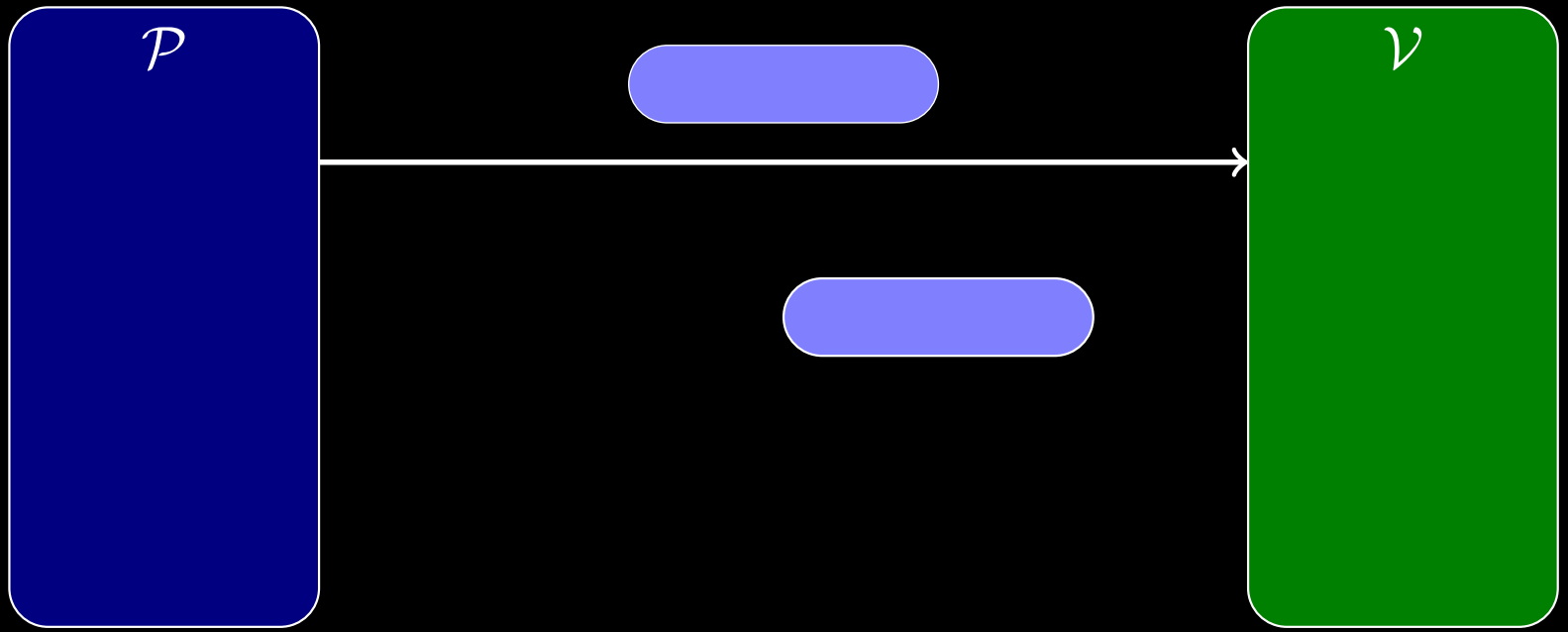
Polynomial IOP



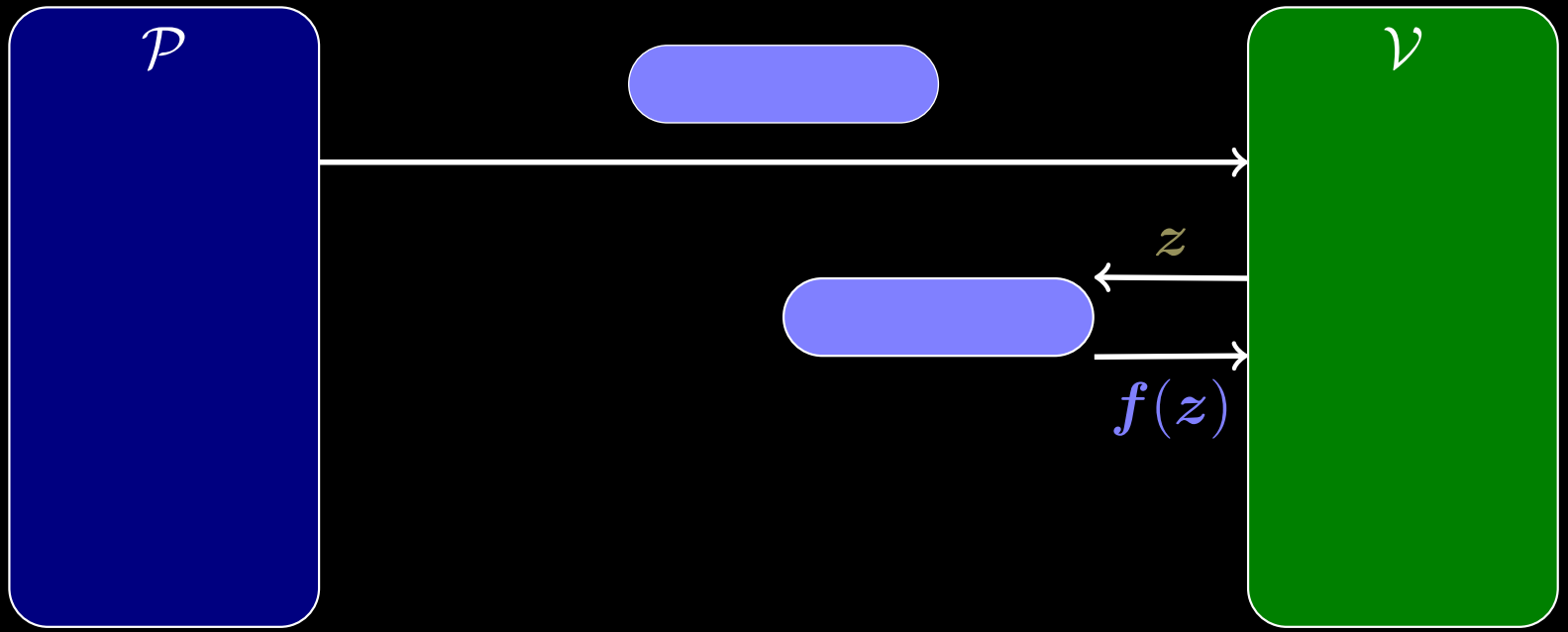
Polynomial IOP



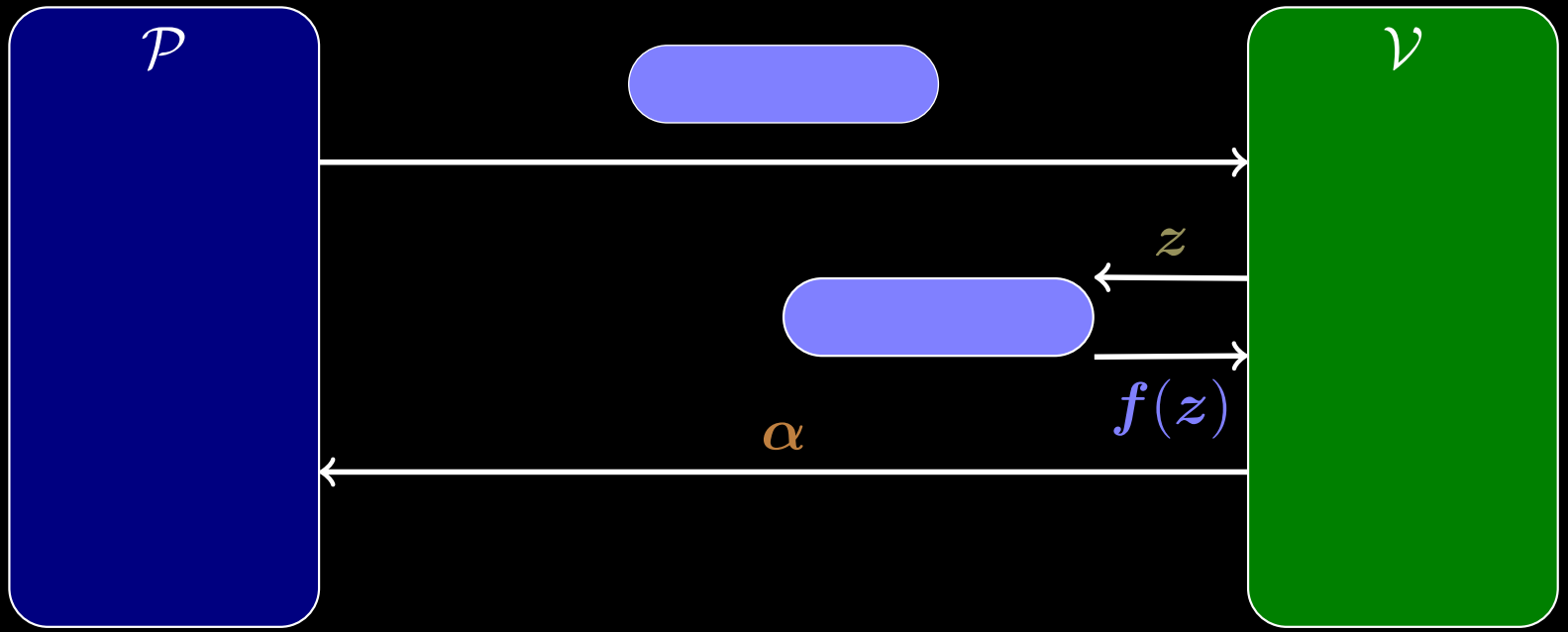
Polynomial IOP



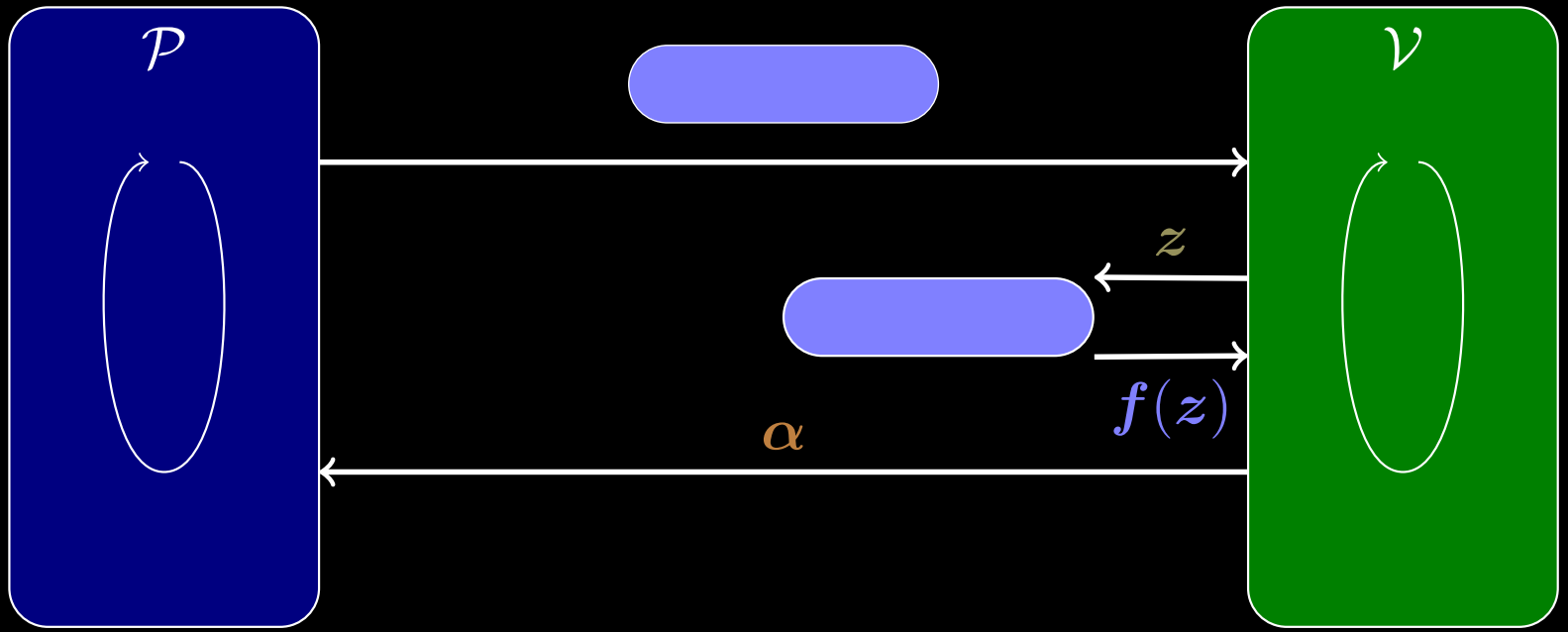
Polynomial IOP



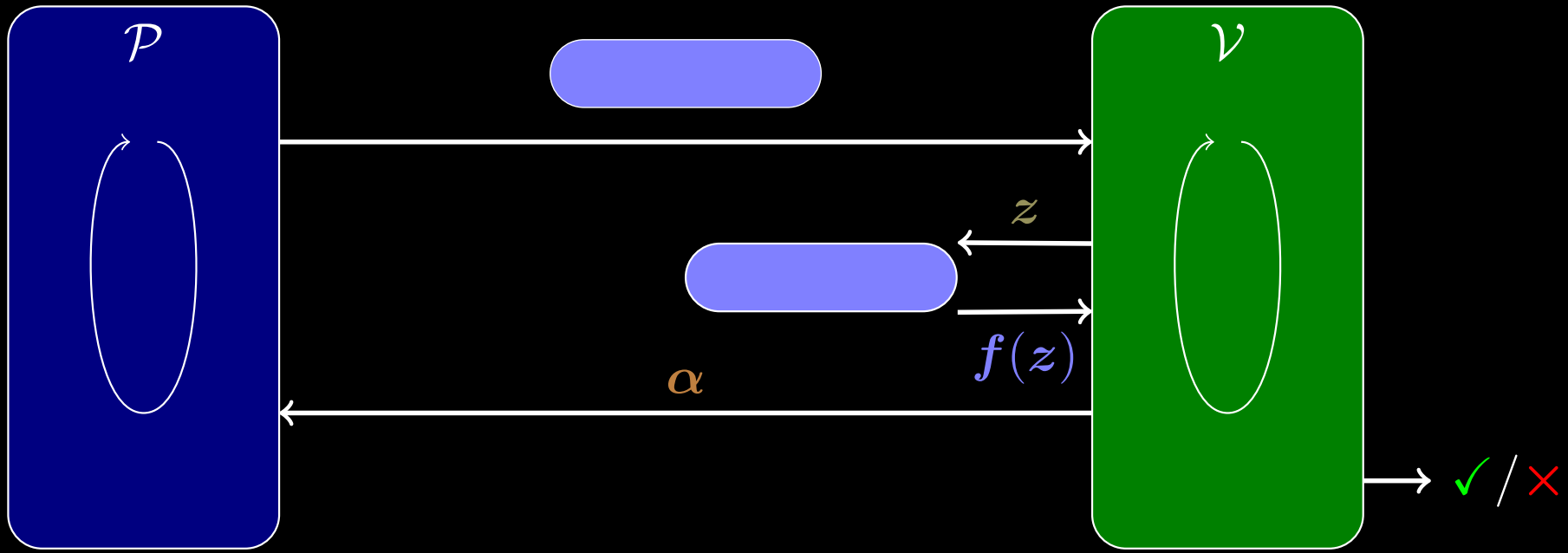
Polynomial IOP



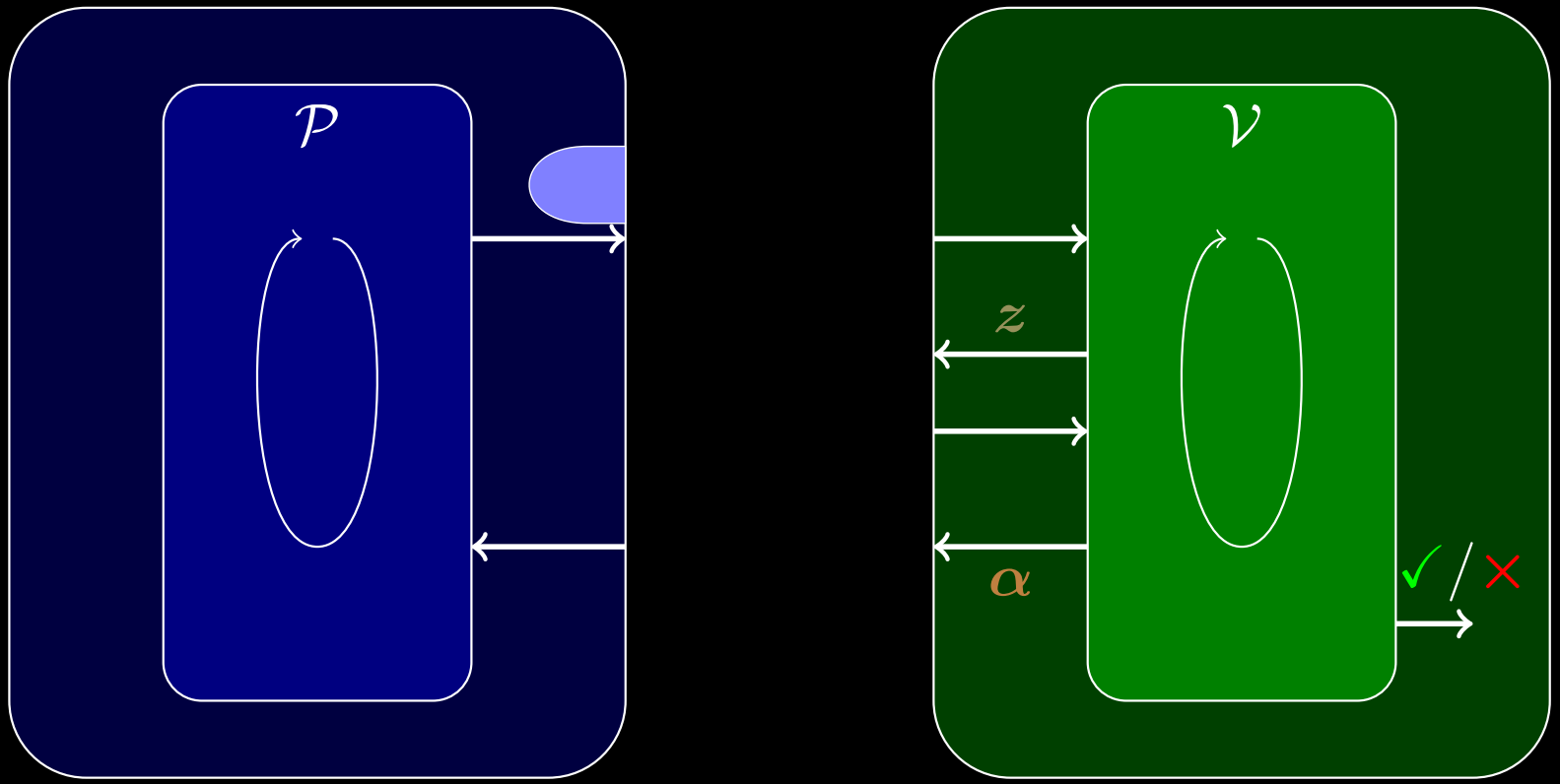
Polynomial IOP



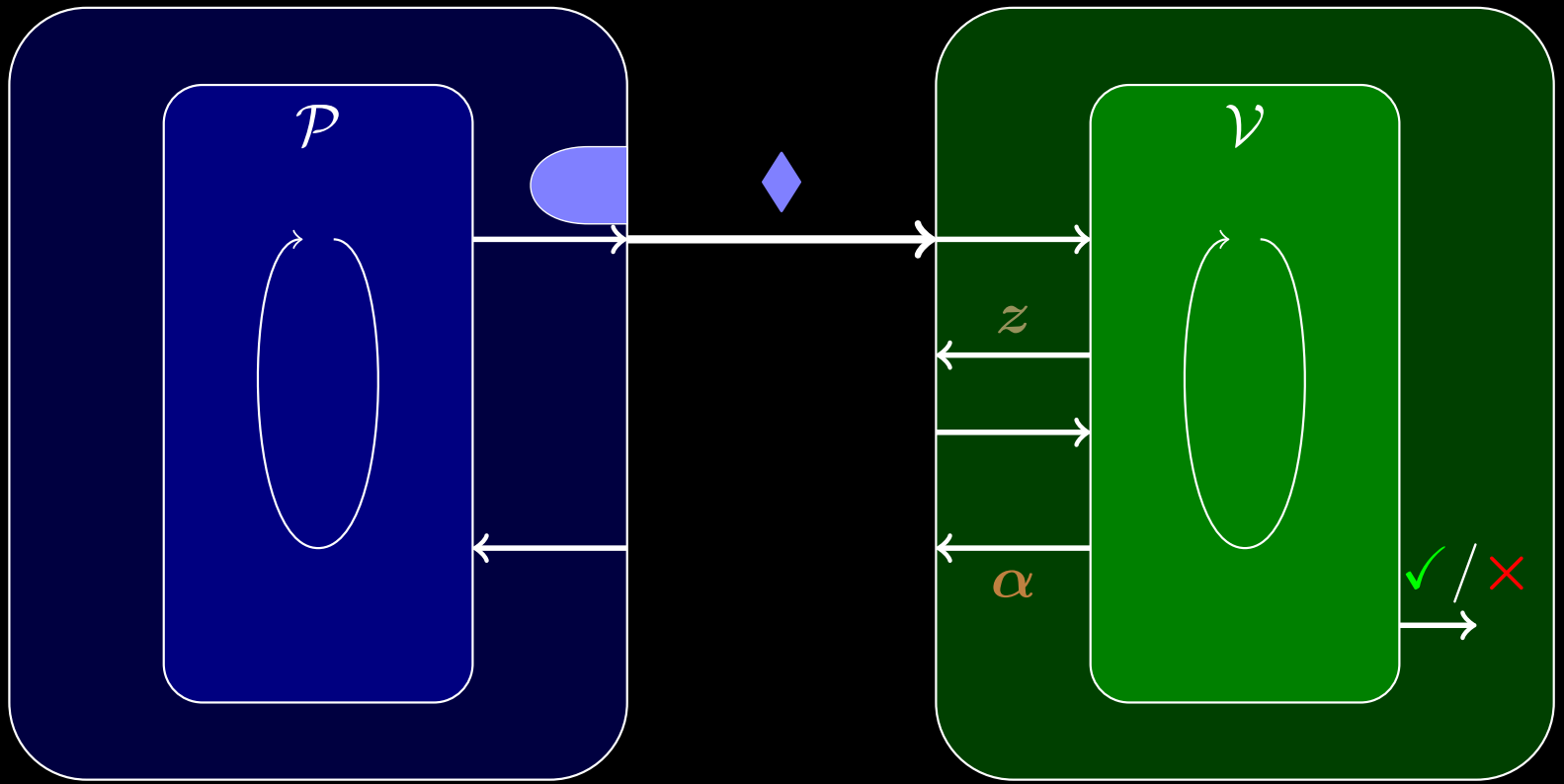
Polynomial IOP



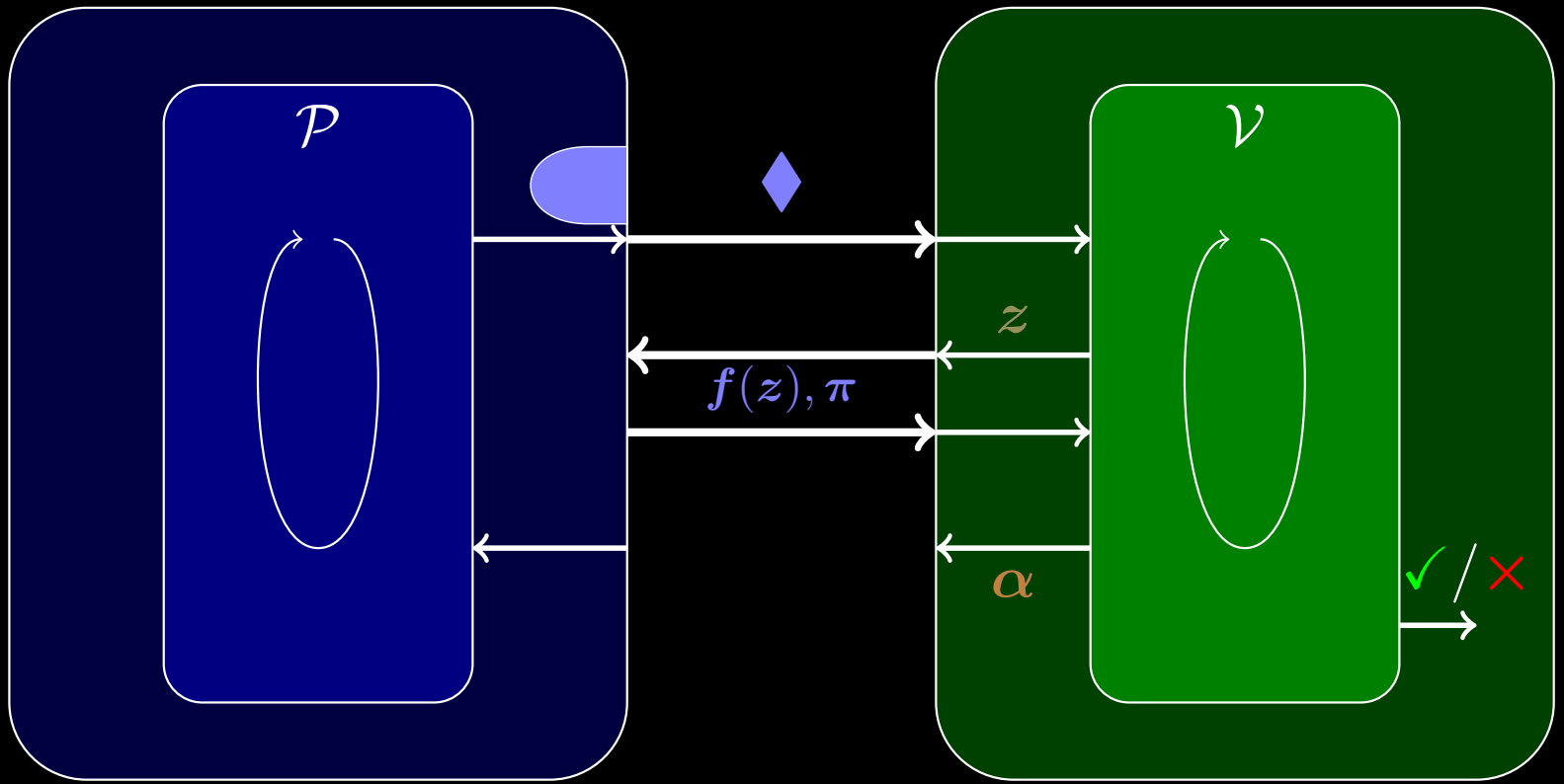
Polynomial IOP Compiler



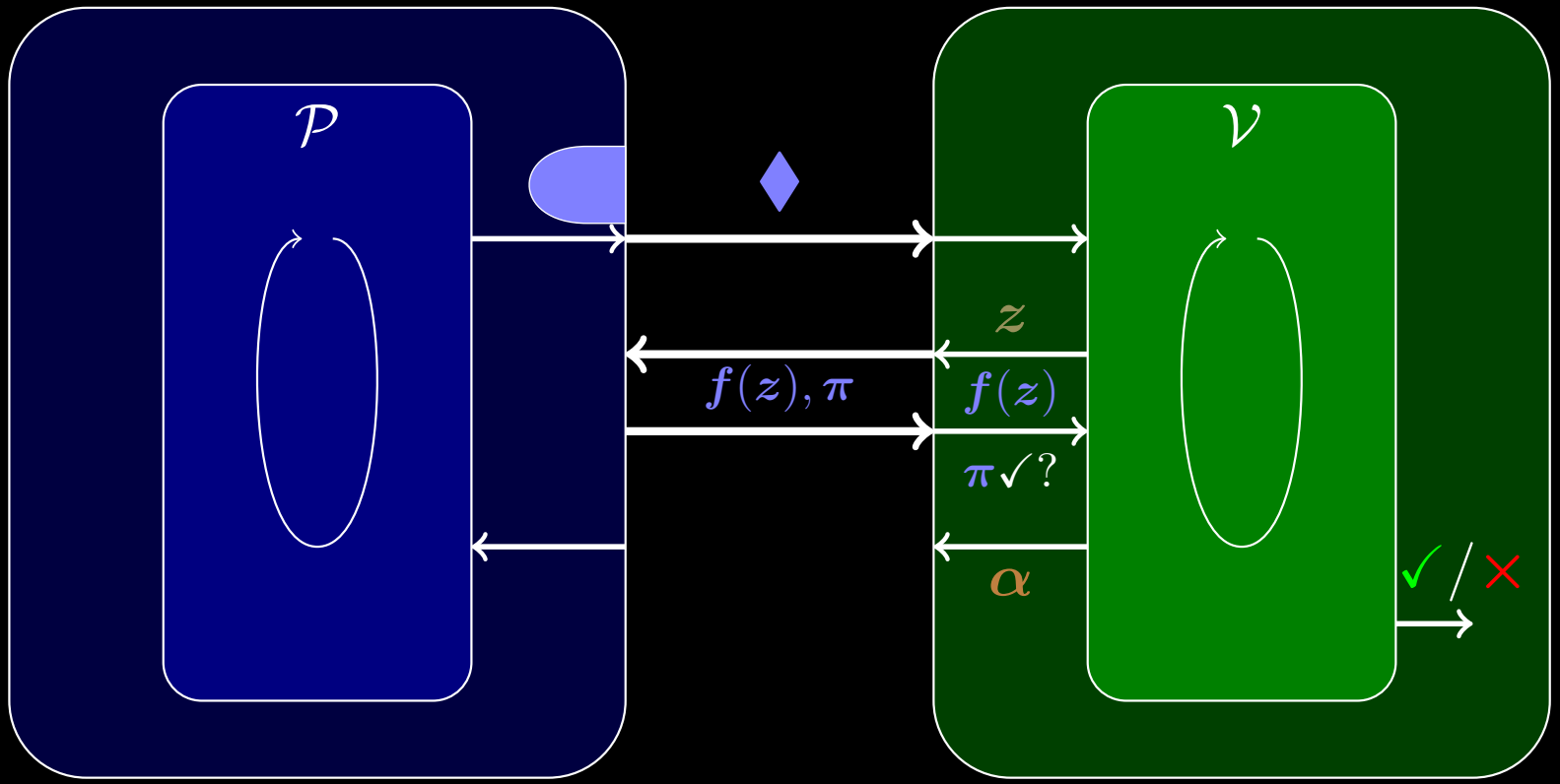
Polynomial IOP Compiler



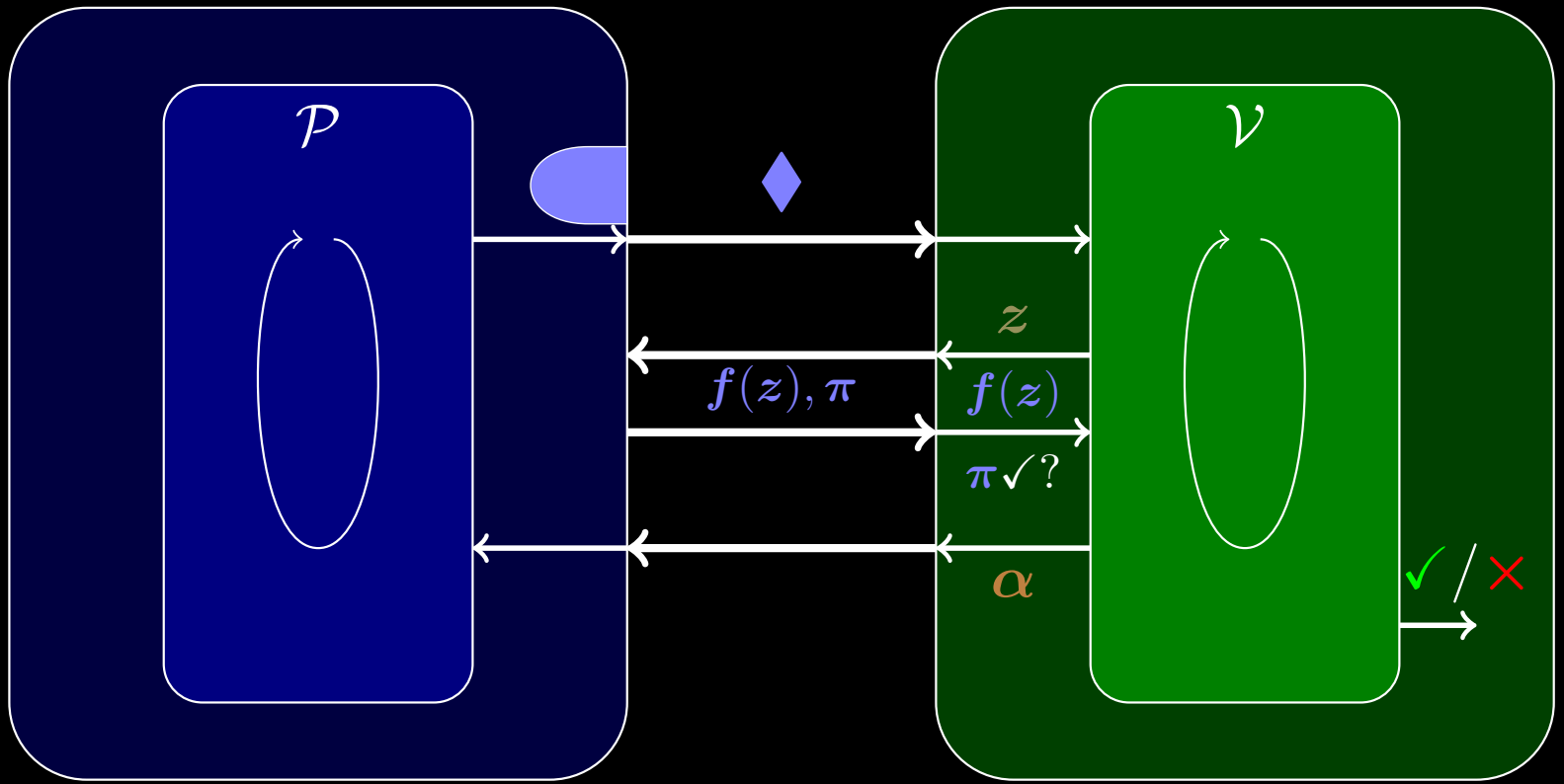
Polynomial IOP Compiler



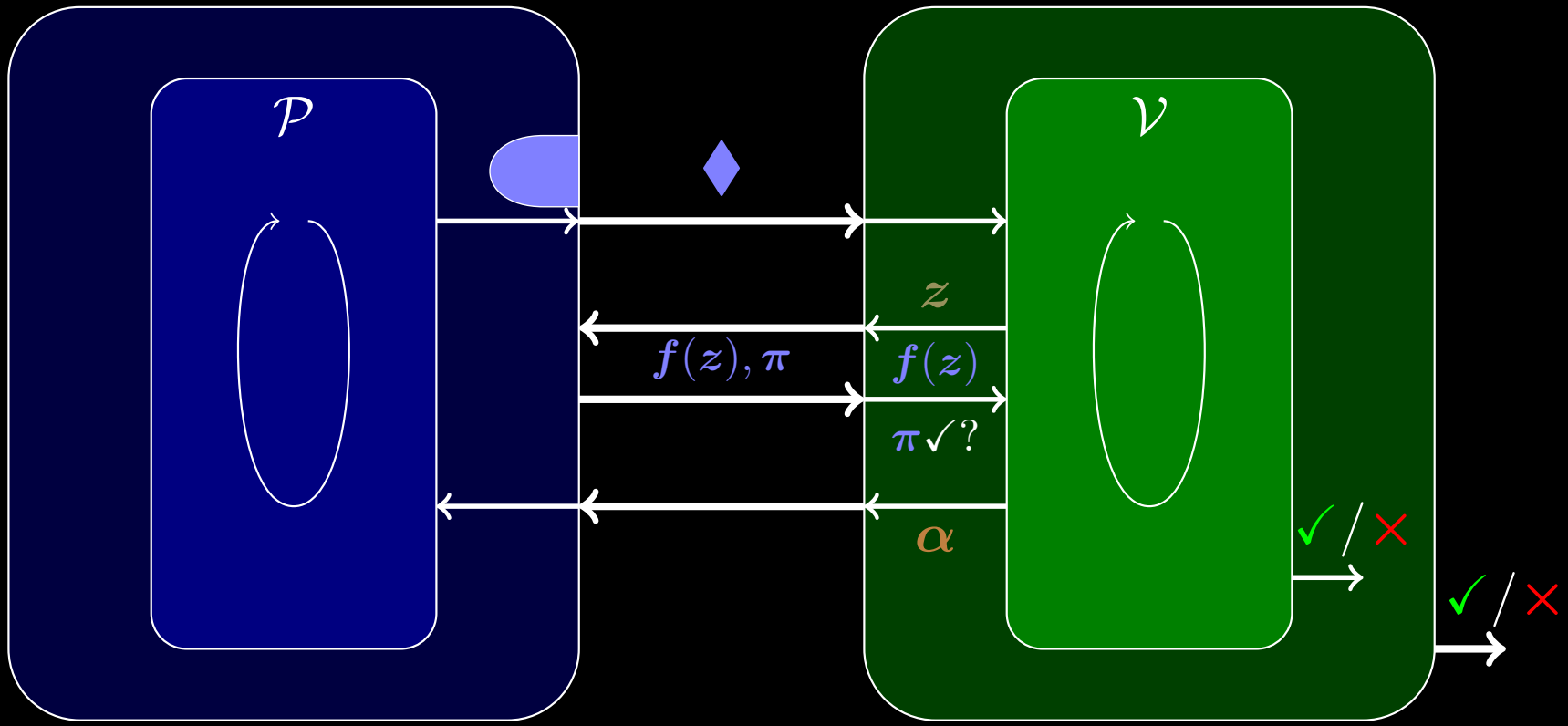
Polynomial IOP Compiler



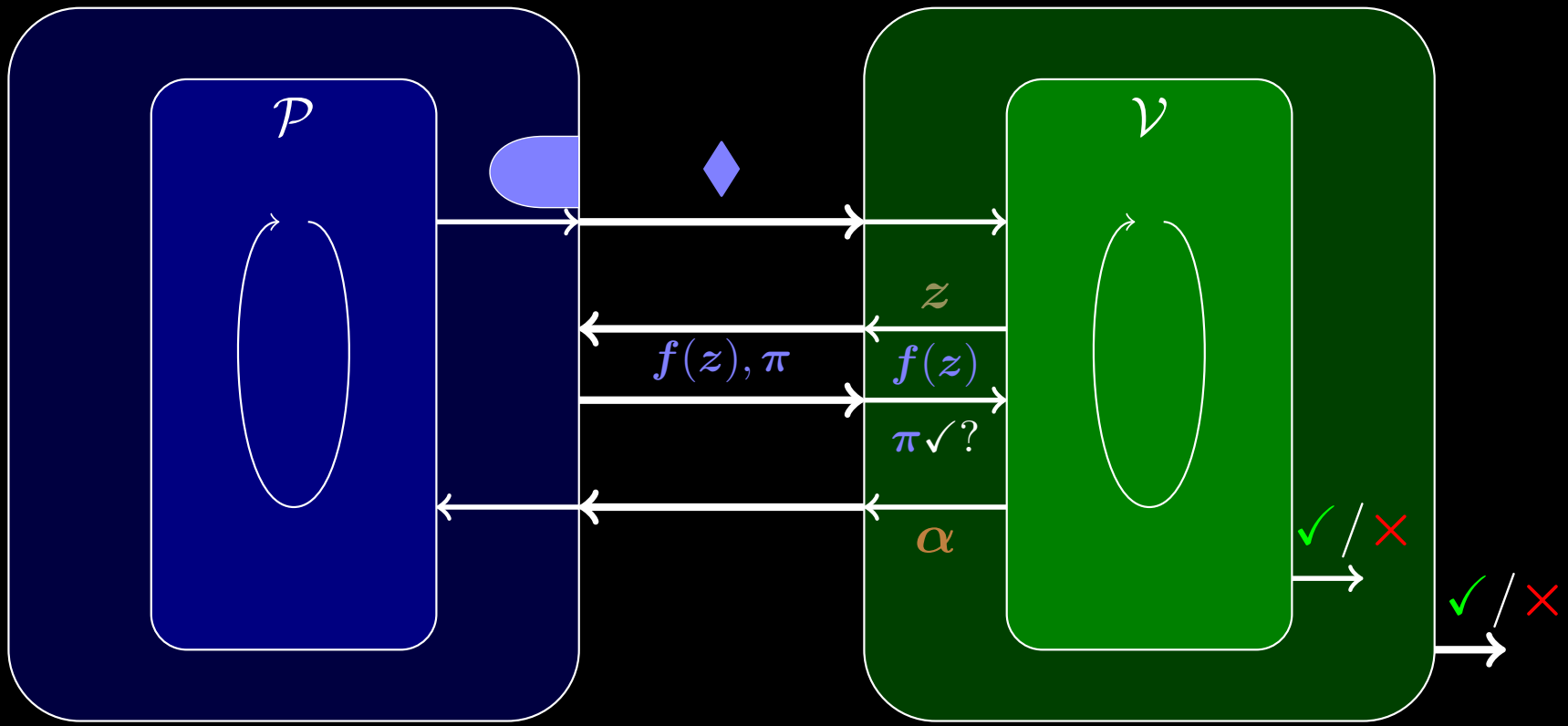
Polynomial IOP Compiler



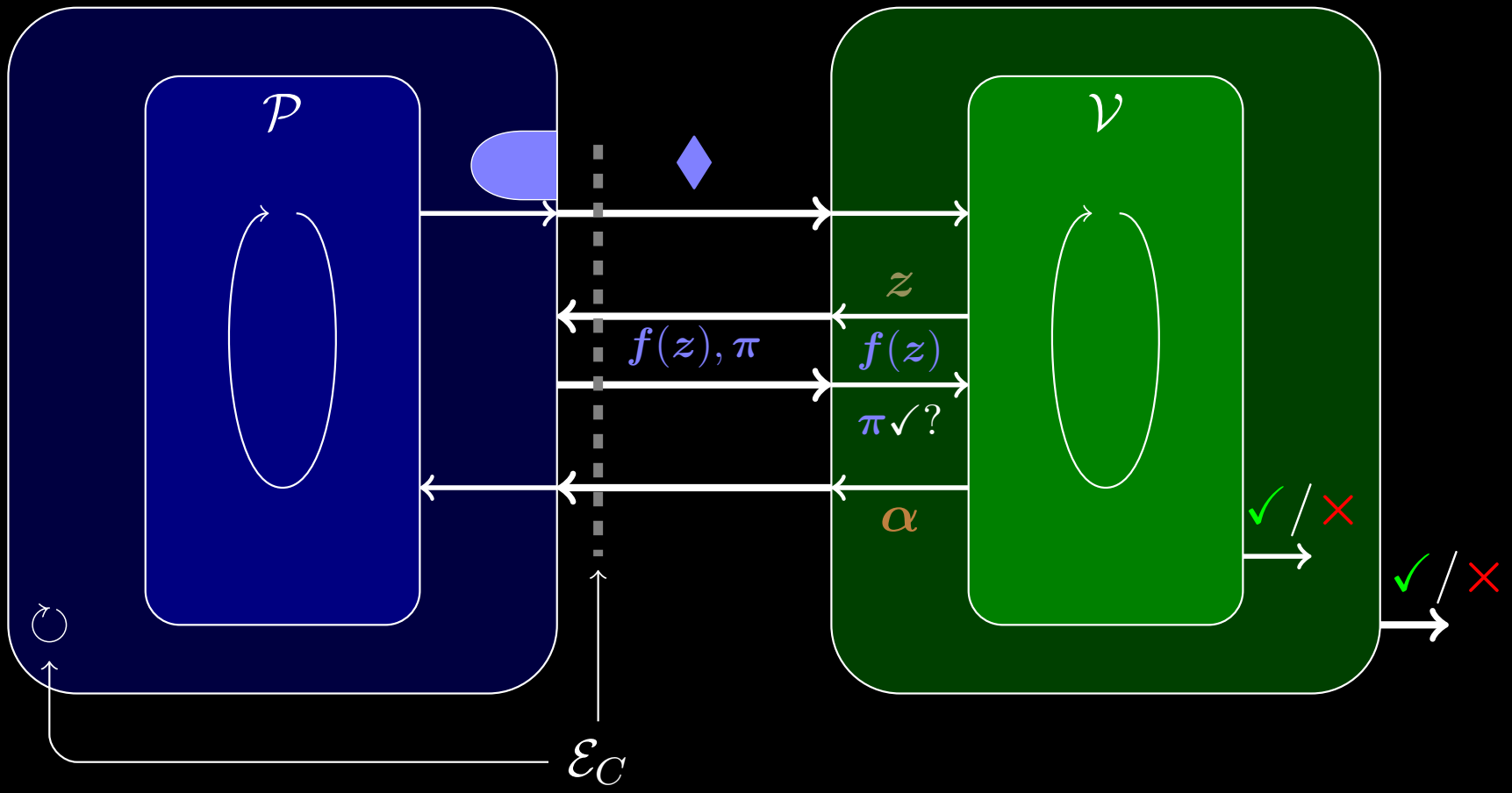
Polynomial IOP Compiler



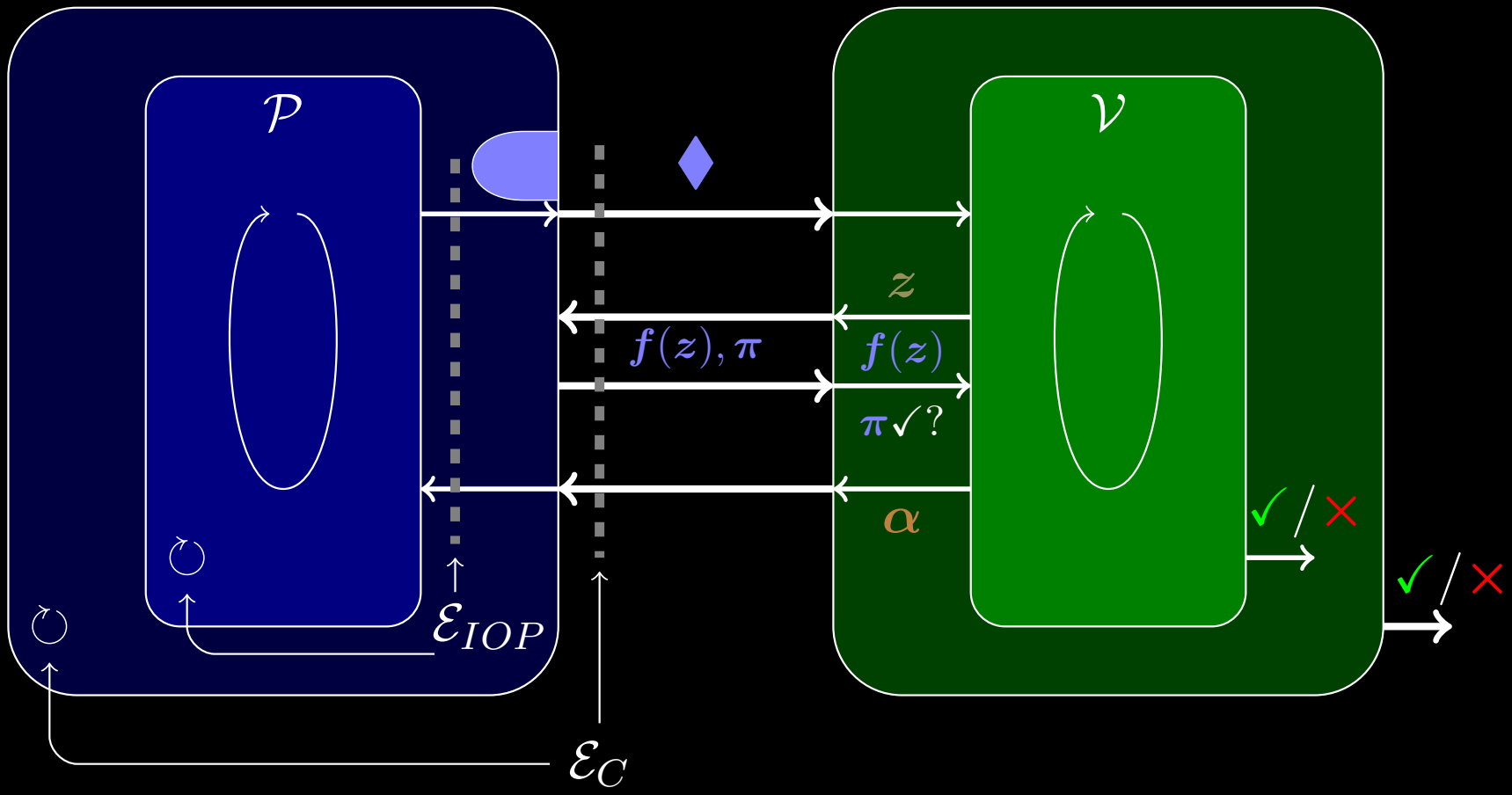
Compiled Polynomial IOP — Soundness



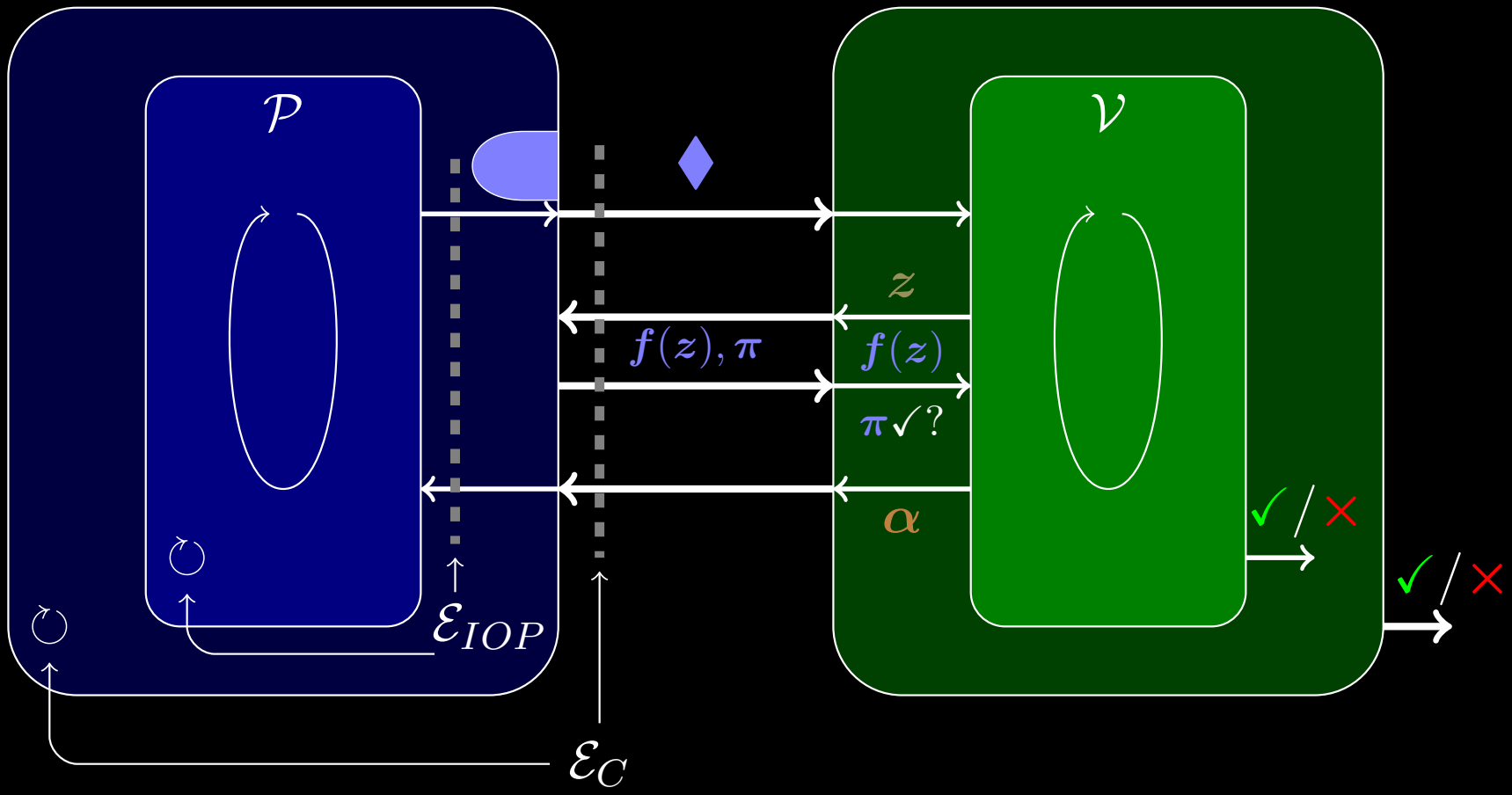
Compiled Polynomial IOP — Soundness



Compiled Polynomial IOP — Soundness

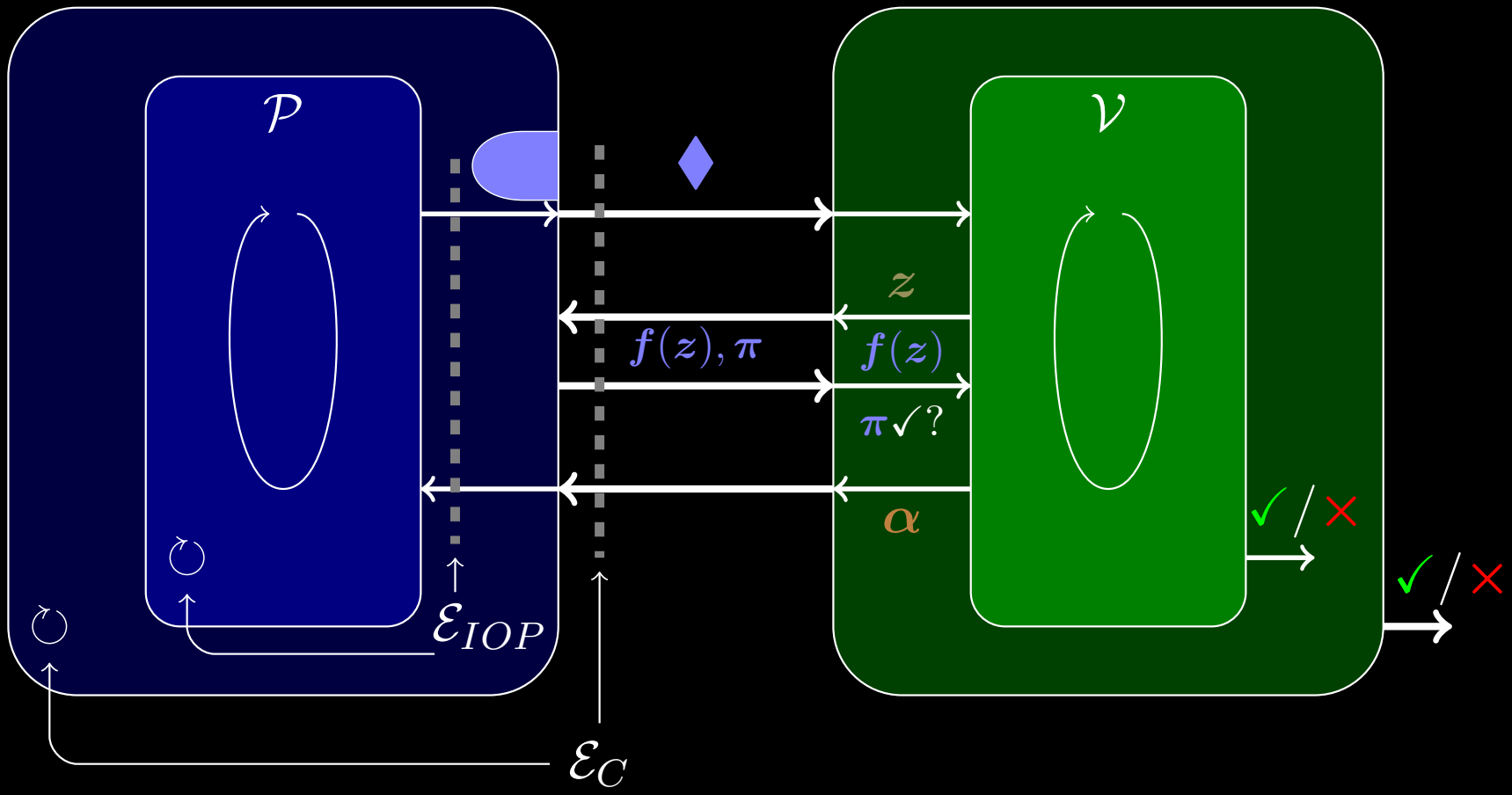


Compiled Polynomial IOP — Soundness



$$\mathcal{E}_{IOP} \xrightarrow{???} \mathcal{E}_C$$

Compiled Polynomial IOP — Soundness

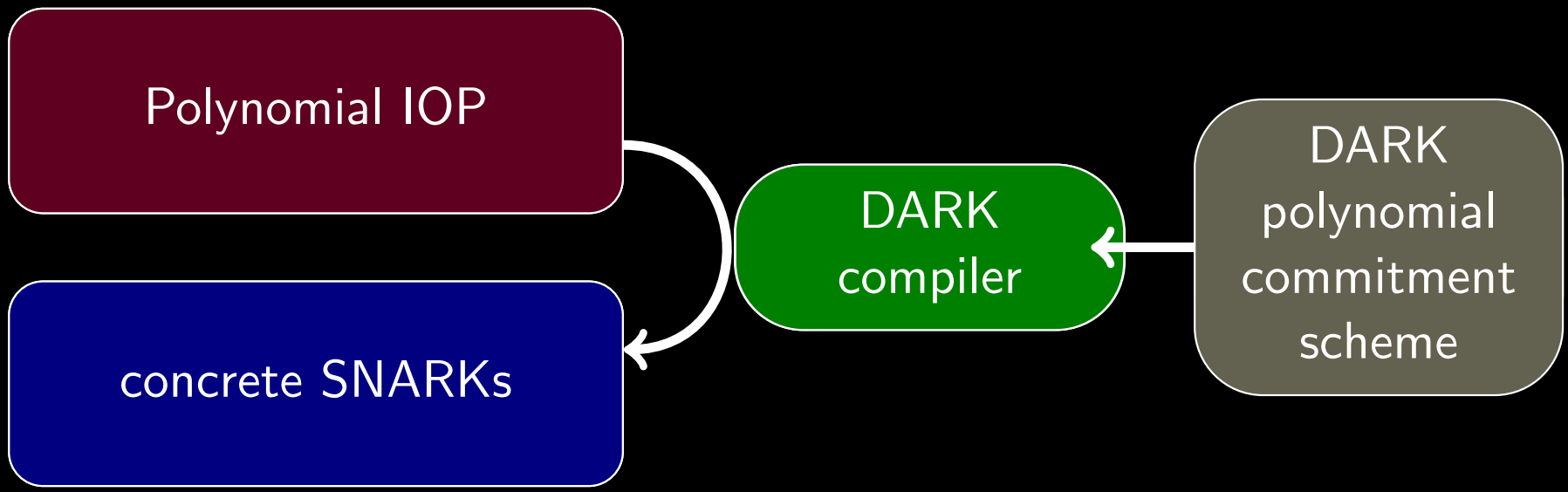


$$\mathcal{E}_{PCS}(\pi) \Rightarrow f(X)$$

$$\mathcal{E}_{IOP} \longrightarrow \mathcal{E}_C$$

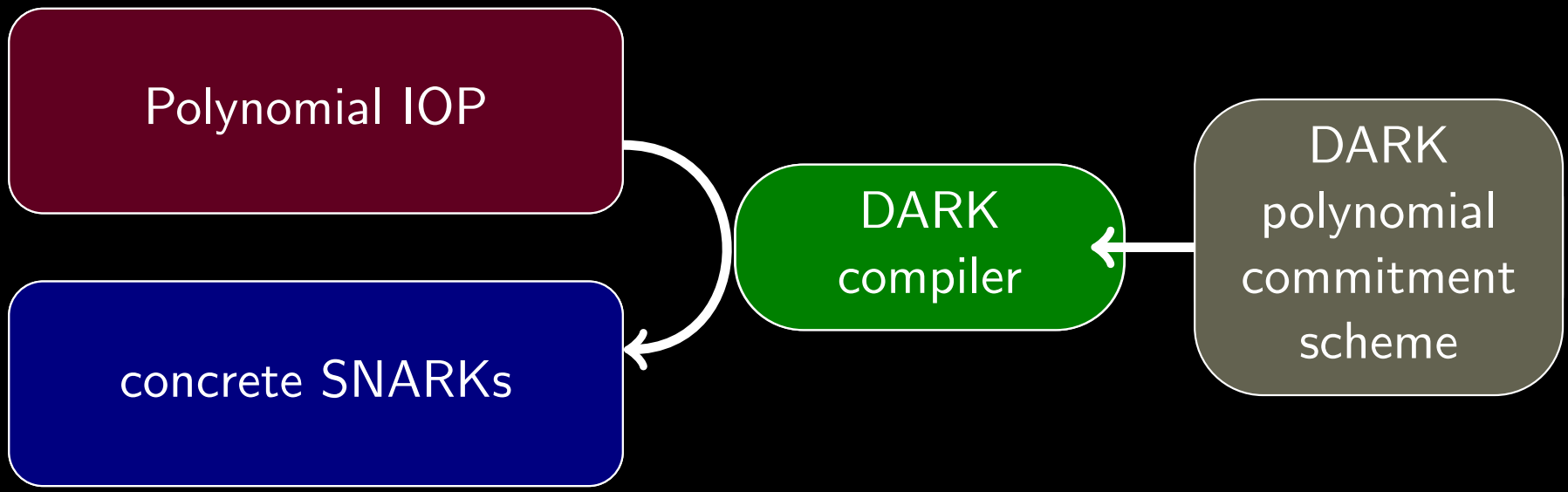
Talk Outline

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)



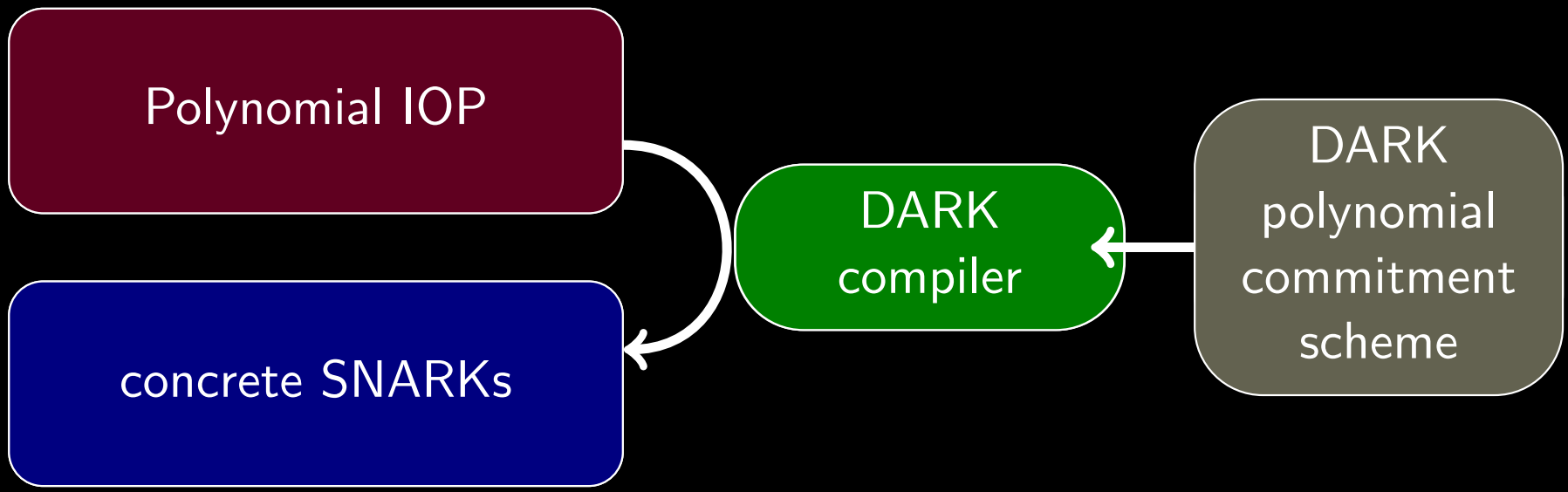
Talk Outline

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)



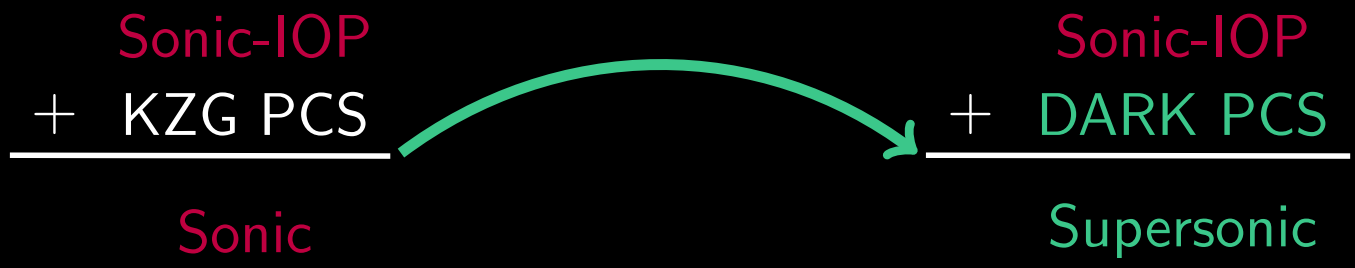
Talk Outline

1. **polynomial commitment scheme** based on **groups of unknown order**
2. **information-theoretic formalism**
3. **cryptographic compiler**
4. **DARK + Sonic = Supersonic** ← transparent SNARK (and other constructions)

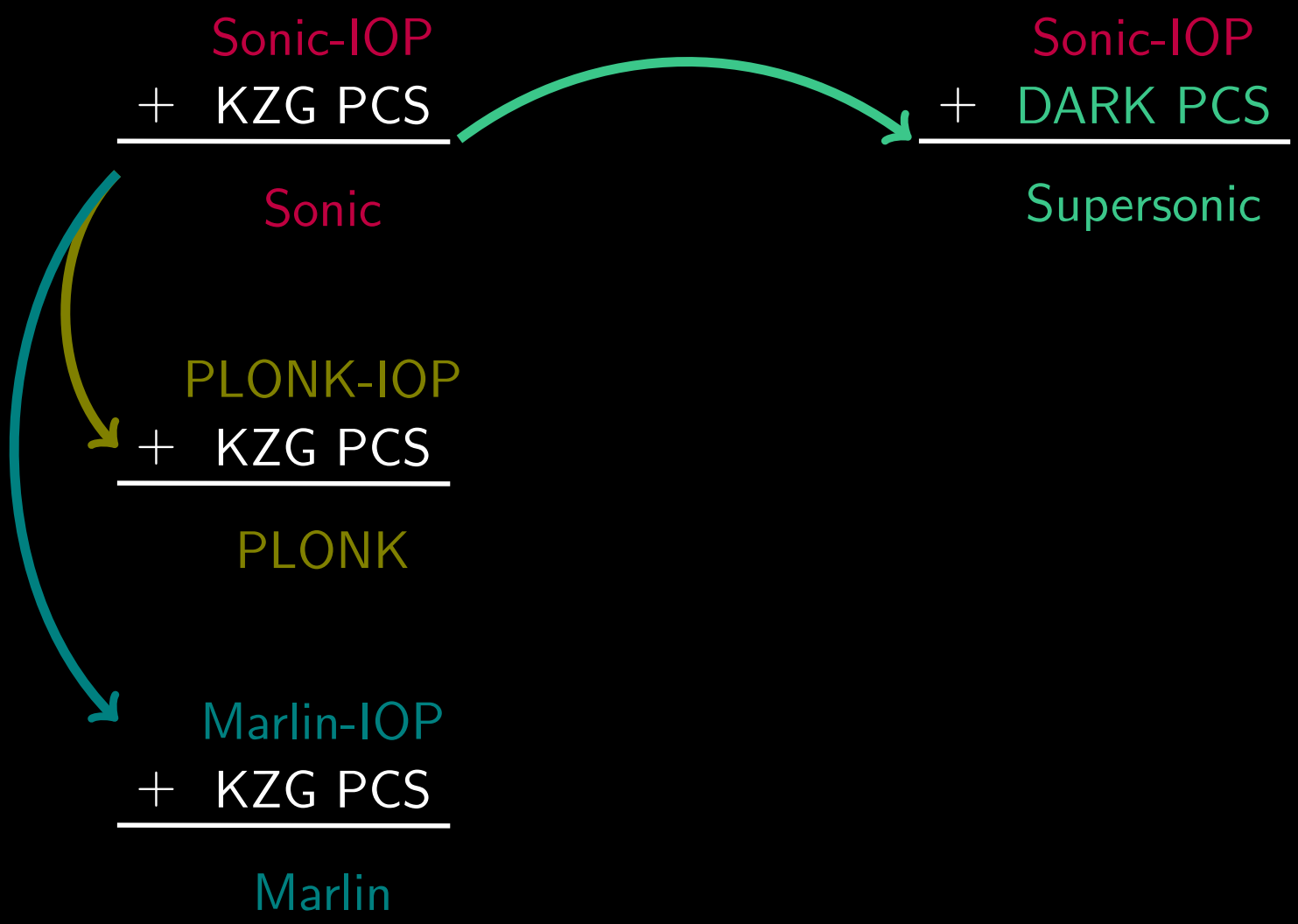


Supersonic / Other Constructions

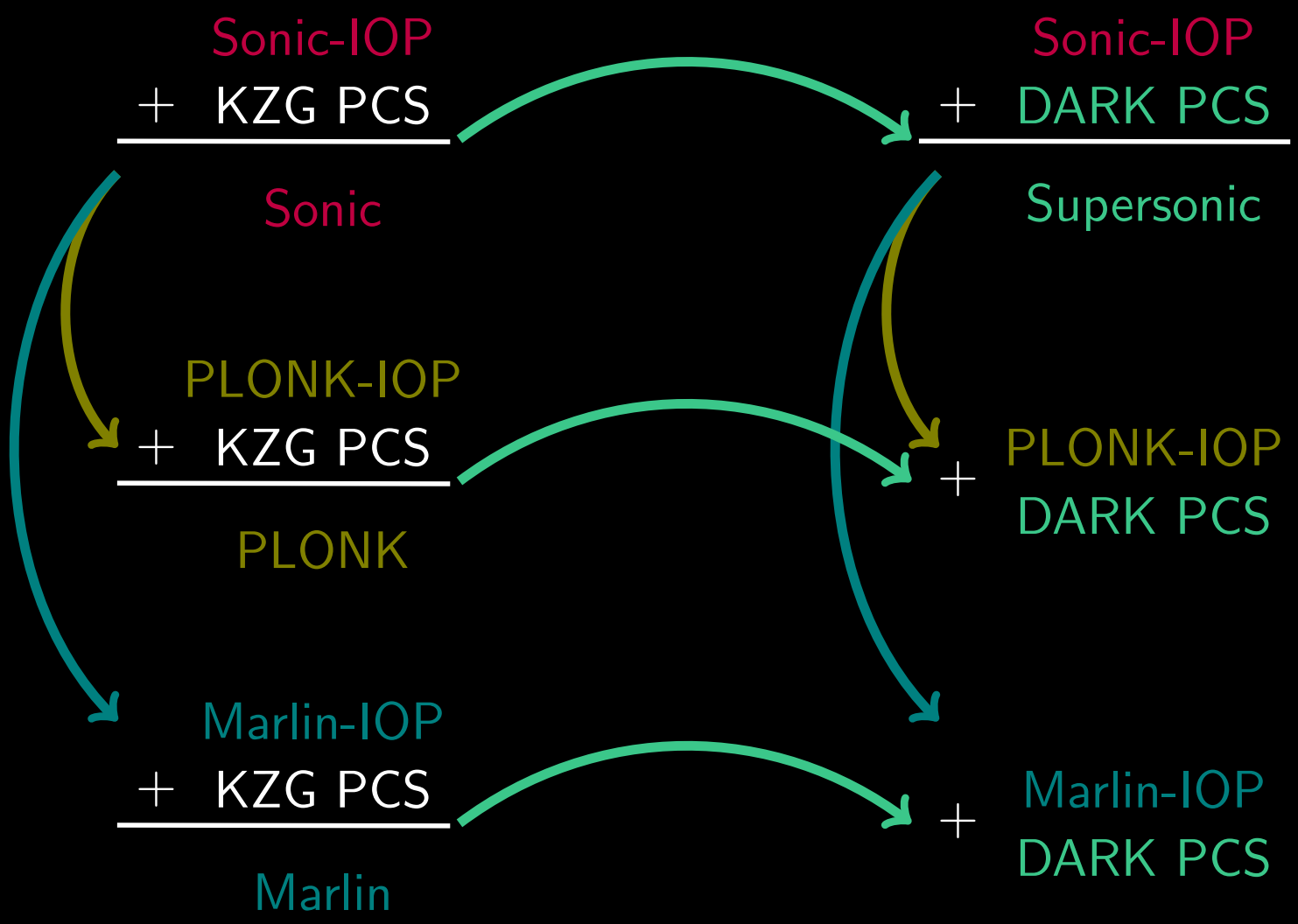
Supersonic / Other Constructions



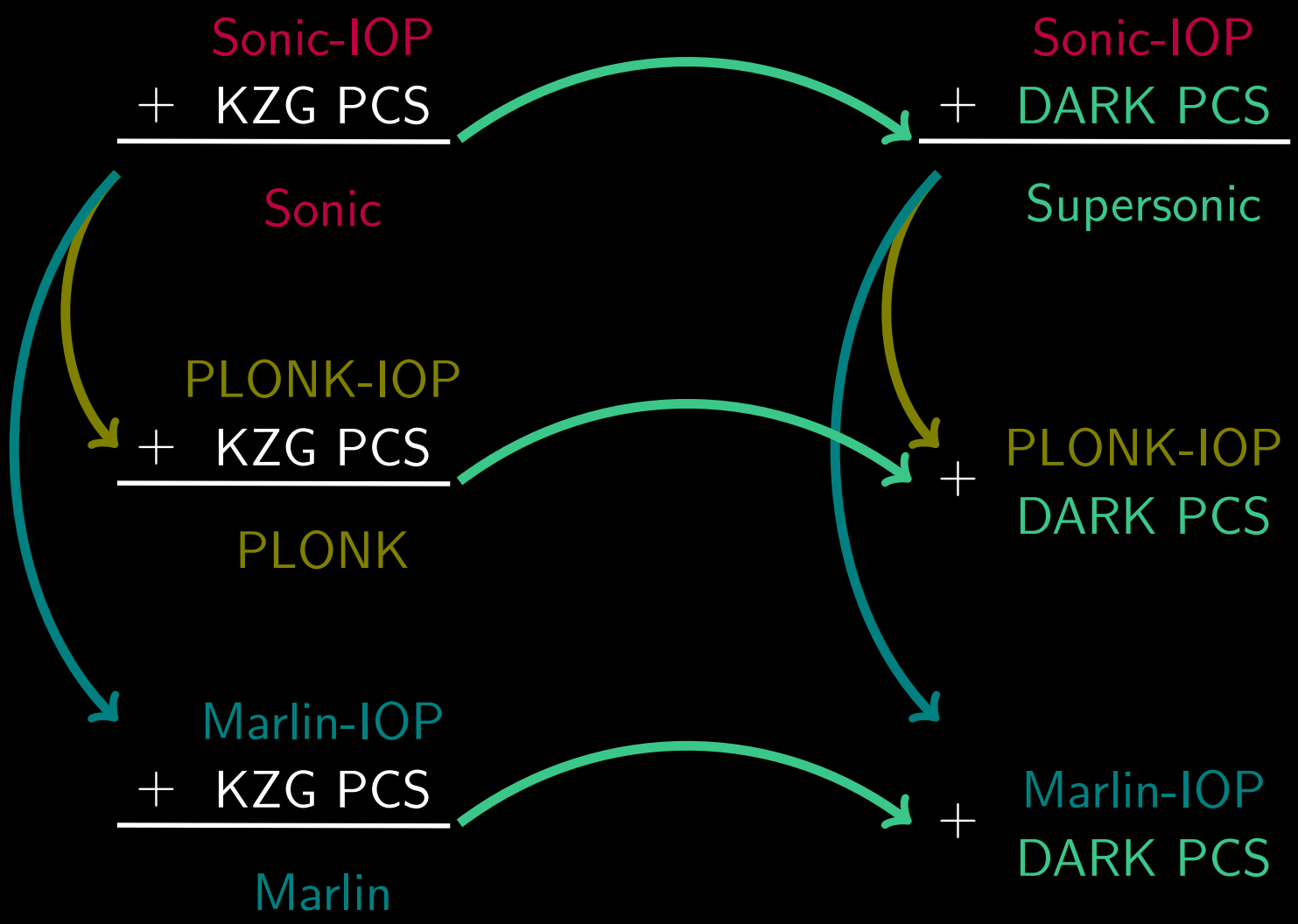
Supersonic / Other Constructions



Supersonic / Other Constructions



Supersonic / Other Constructions



transparent setup (class groups)

constant-size SRS (RSA groups)

Supersonic Performance

Polynomial IOP	# Polynomials	# Eval points	$\|\pi\|$
Sonic	$12 + 15$	12	15.3 KB
PLONK	$7 + 7$	2	10.1 KB
Marlin	$9 + 10$	3	12.3 KB

Scheme	Transp.	|CRS|	Prover	Verifier	$\|\pi\|$	$\|\pi\|$
Supersonic	✓	1	$n\log(n)$	$3\log(n)$	$2\log(n)$	10.1KB
PLONK	×	$2n$	n	1	1	720b
Groth16	×	$2n$	n	1	1	192b
Bulletproofs	✓	$2n$	n	n	$2\log(n)$	1.7KB
STARK	✓	1	λT	$\lambda\log^2(T)$	$\lambda\log^2(T)$	600 KB
Virgo	✓	1	λn	$\lambda\log^2(n)$	$\lambda\log^2(n)$	271 KB

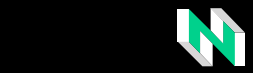
concrete quantities for $n = 2^{20}$ and $\lambda = 120$

Conclusions

Conclusions

- This work:
  - trustless SNARKs from class groups
  - theoretical baggage: Polynomial IOP
  - all SNARKs have a DARK analogue

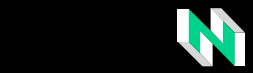
Conclusions



- This work:
  - trustless SNARKs from class groups
  - theoretical baggage: Polynomial IOP
  - all SNARKs have a DARK analogue

- Open questions:
  - class groups
  - other groups of unknown order
  - Polynomial IOPs
  - multi-round Fiat-Shamir

Conclusions



- This work:
  - trustless SNARKs from class groups
  - theoretical baggage: Polynomial IOP
  - all SNARKs have a DARK analogue
- Open questions:
  - class groups
  - other groups of unknown order
  - Polynomial IOPs
  - multi-round Fiat-Shamir
- Future work:
  - improvements
  - implementation

Thank you!

benedikt@cs.stanford.edu

bfisch@cs.stanford.edu

alan@nervos.org

https://eprint.iacr.org/2019/1229