polynomial commitments

building block for universal SNARKs

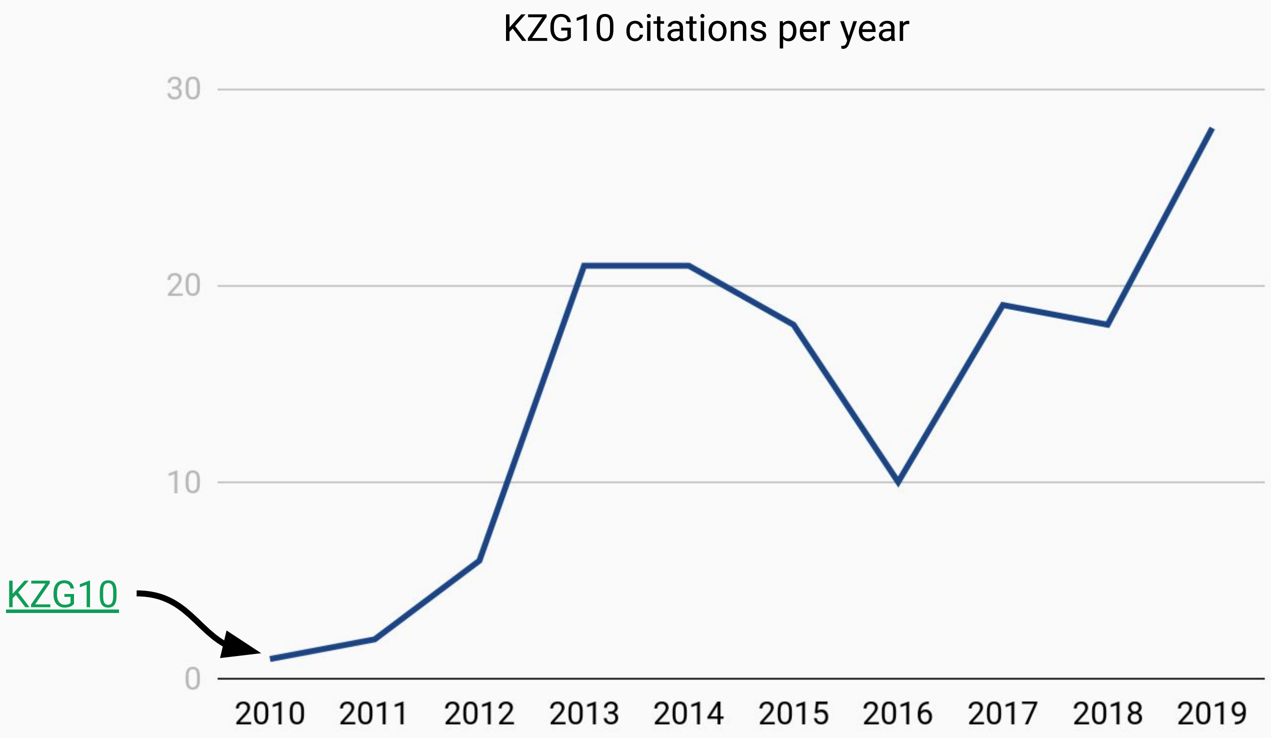
part 1—context

part 2—landscape

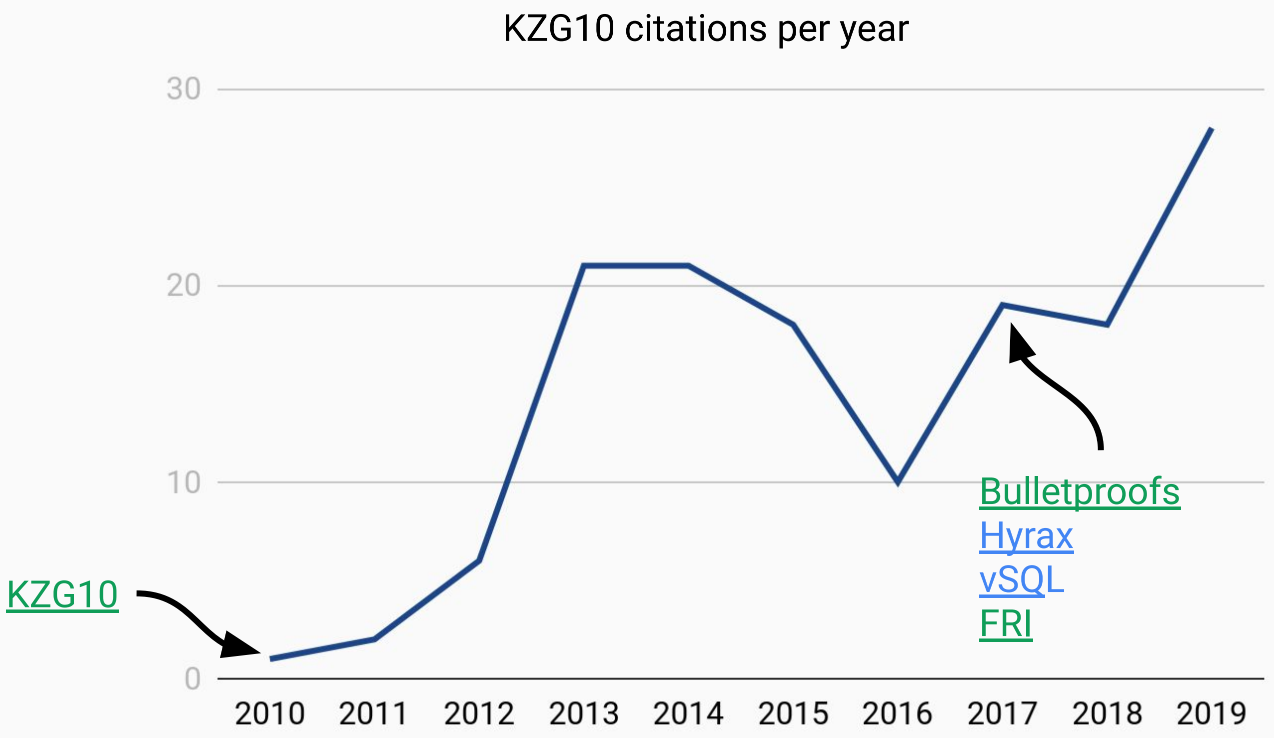
part 3—mechanics

part 4—gadgets

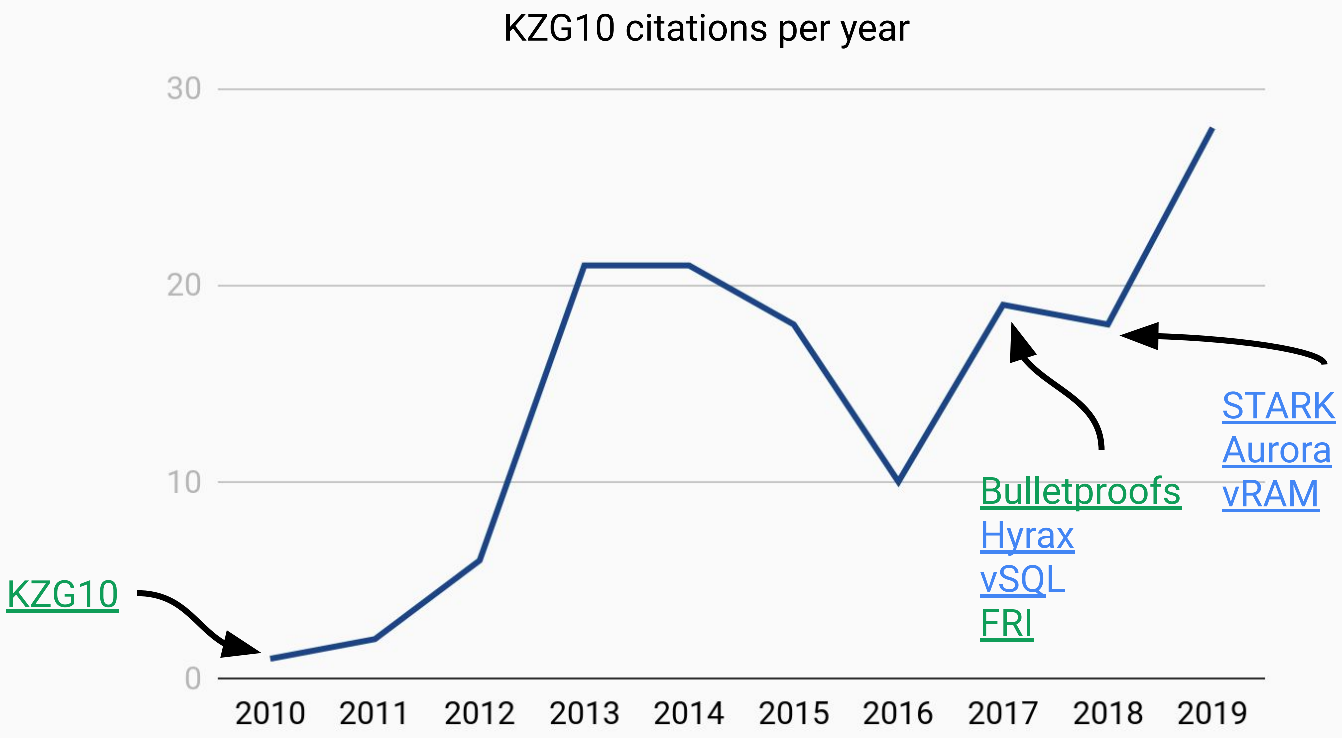
historical perspective



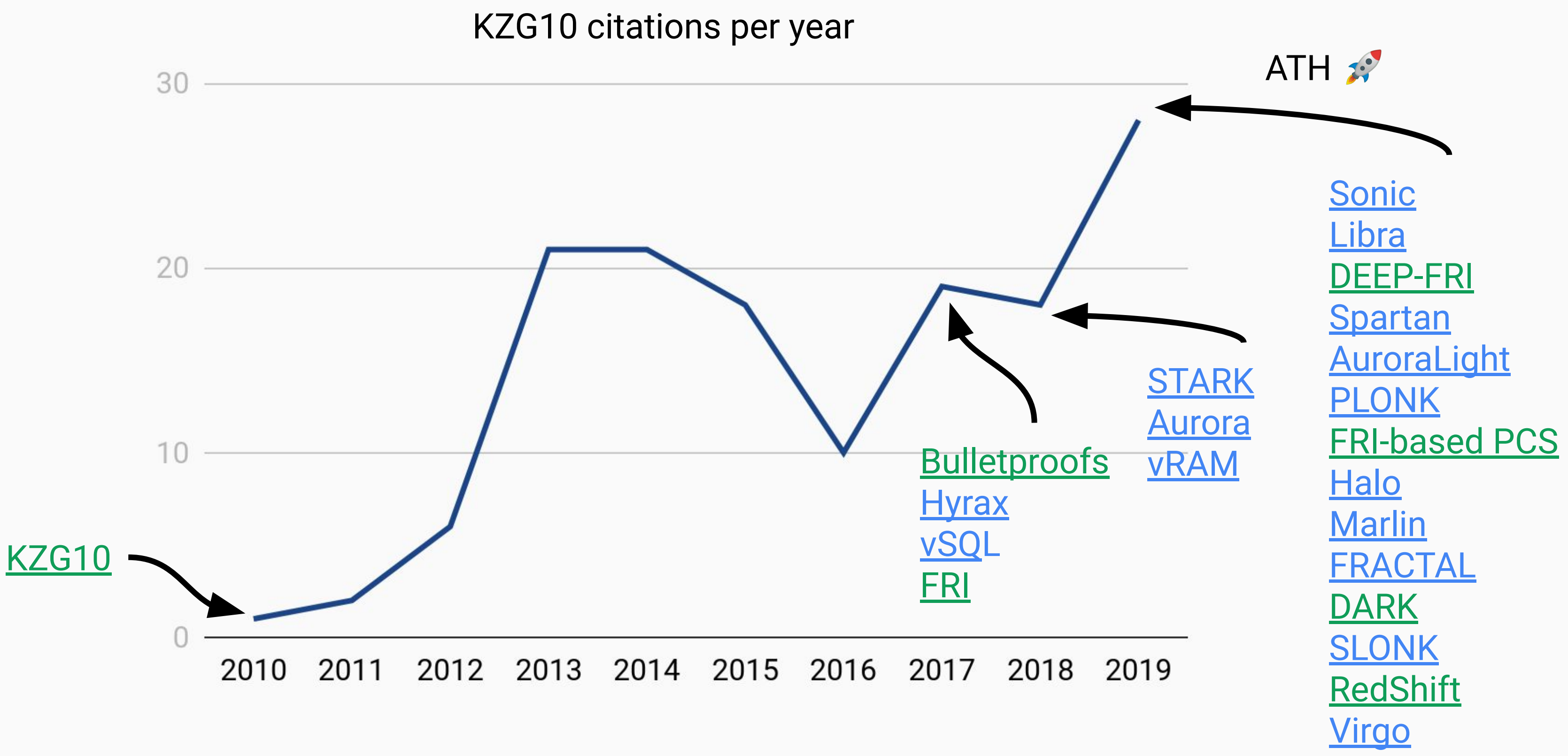
historical perspective



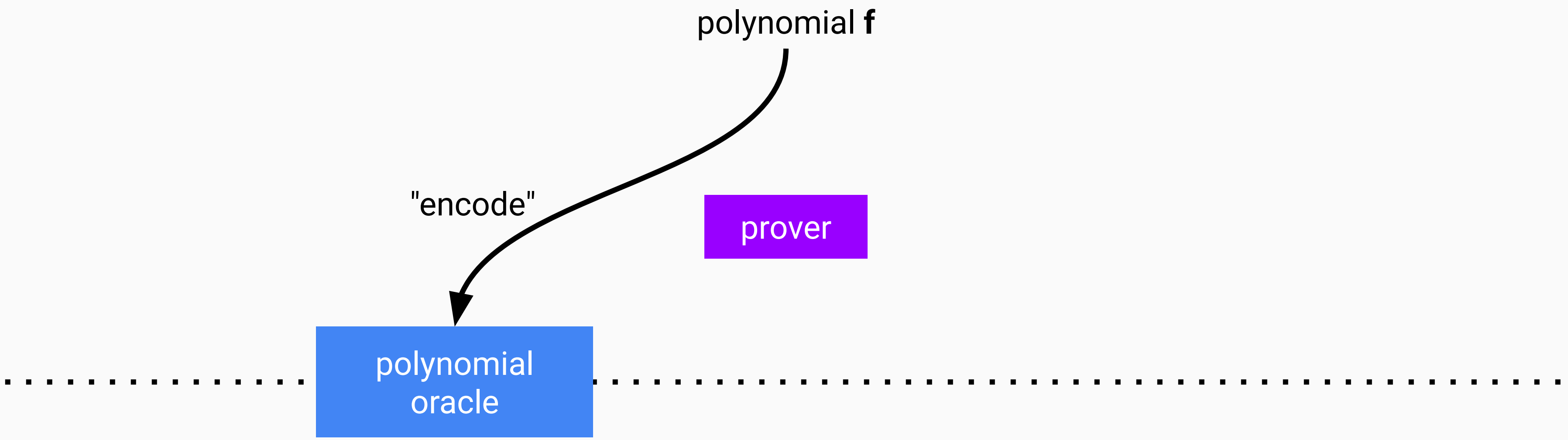
historical perspective



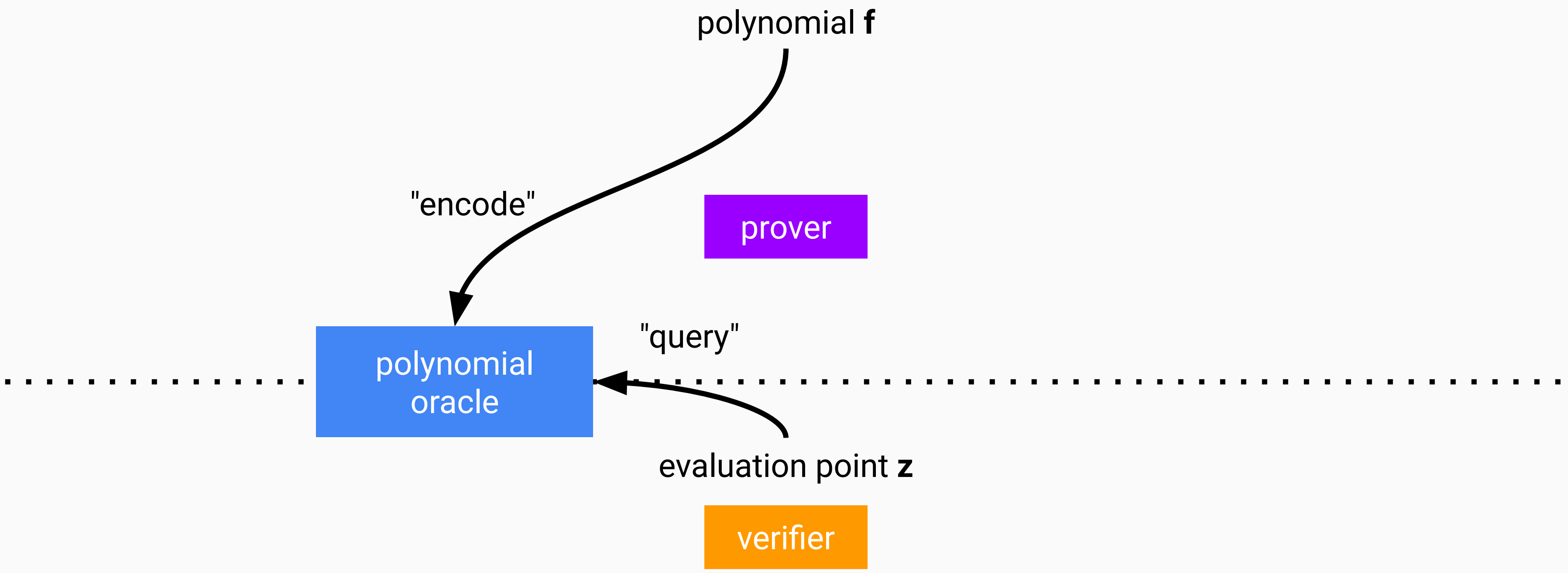
historical perspective



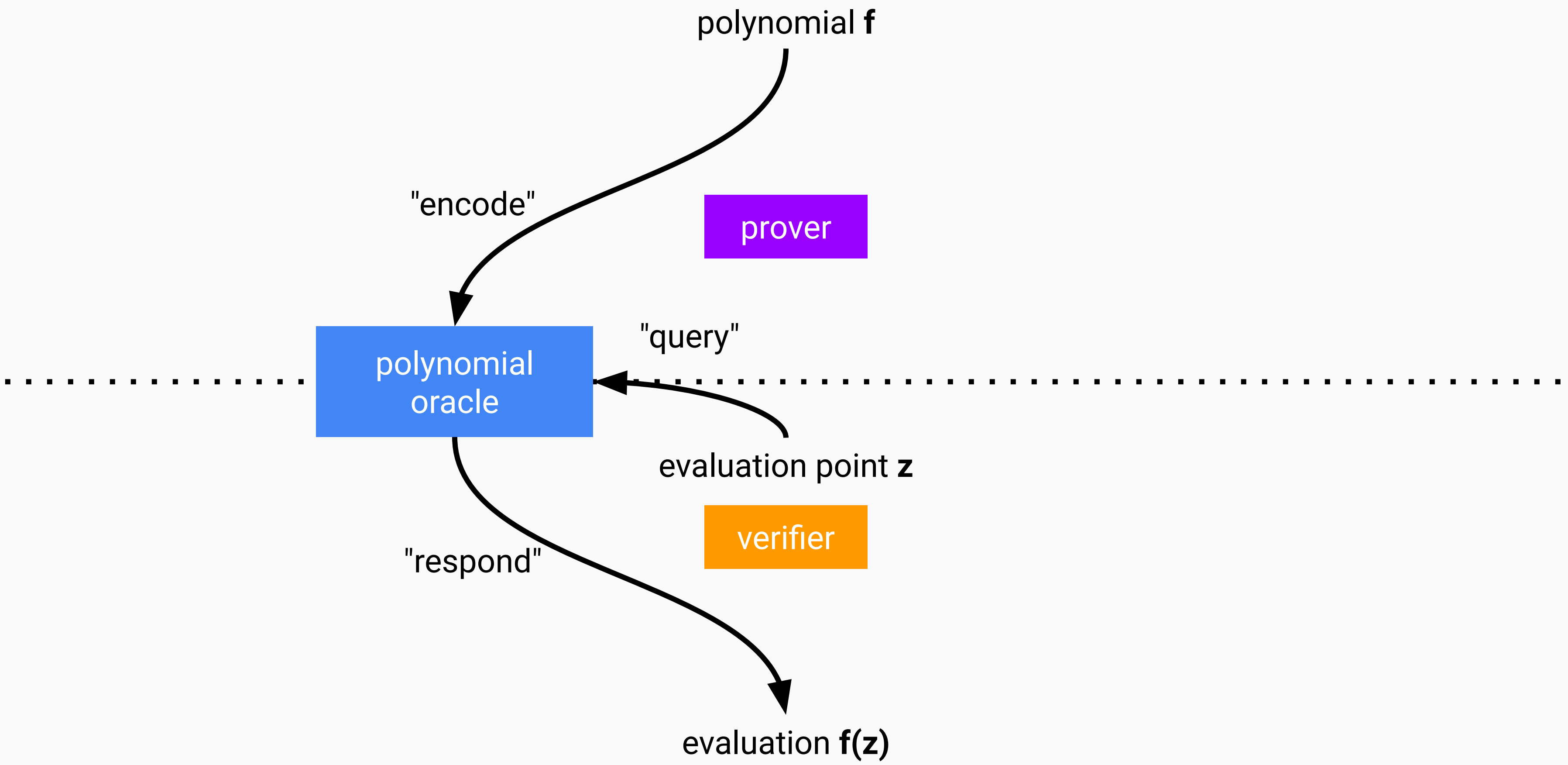
definitions



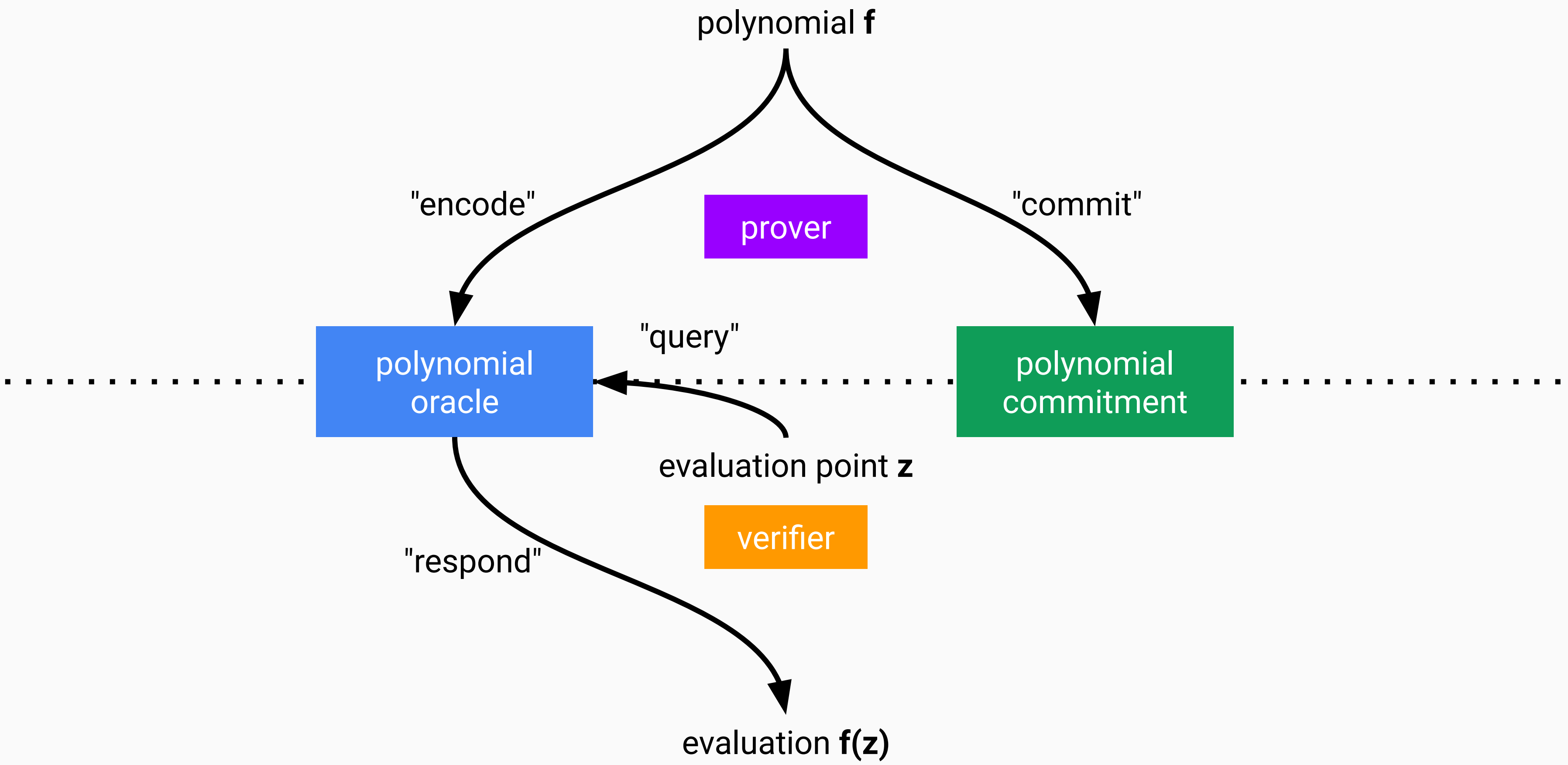
definitions



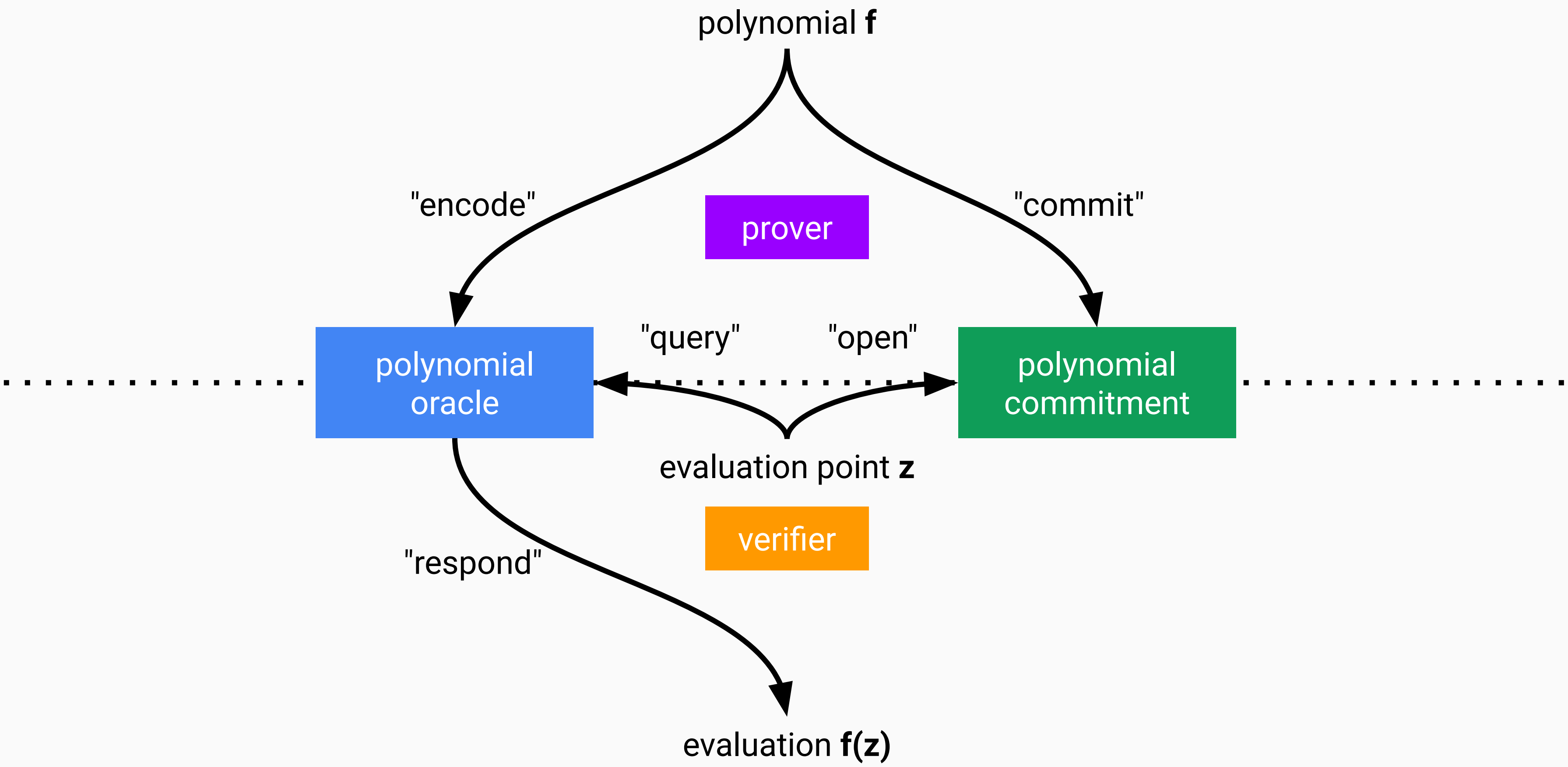
definitions



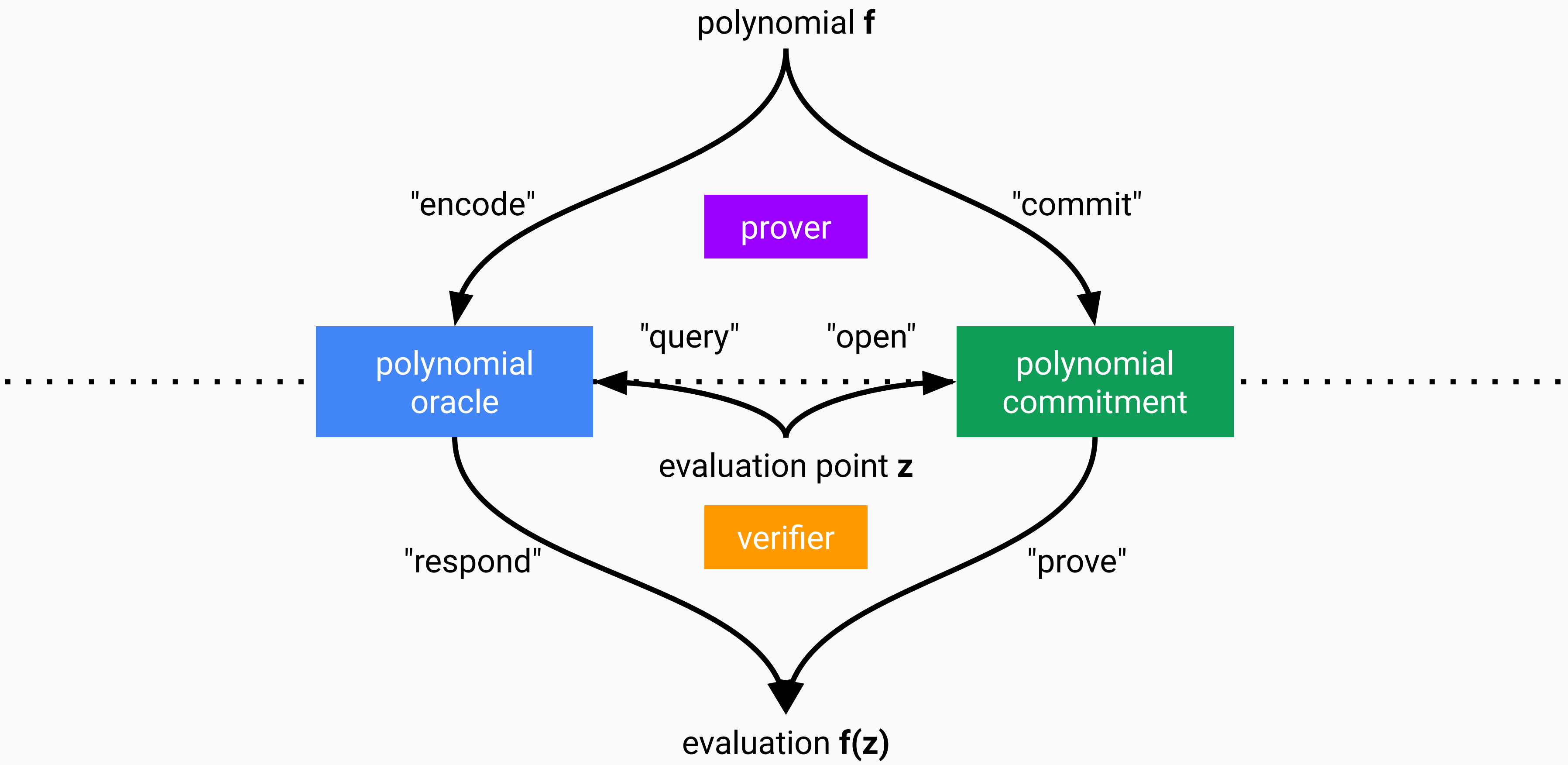
definitions



definitions



definitions



the fine print

B.1 Definition

A polynomial commitment scheme over a field family $\mathcal{F}$ for **a single degree bound and a single evaluation point** is a tuple of algorithms $\mathsf{PC_s} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Check})$ with the following syntax.

- $\mathsf{PC_s}.\mathsf{Setup}(1^\lambda, D) \to (\mathsf{ck}, \mathsf{rk})$. On input a security parameter λ (in unary), and a maximum degree bound $D \in \mathbb{N}$, $\mathsf{PC_s}.\mathsf{Setup}$ samples a key pair $(\mathsf{ck}, \mathsf{rk})$. The keys contain the description of a finite field $\mathbb{F} \in \mathcal{F}$.
- $\mathsf{PC_s}.\mathsf{Commit}(\mathsf{ck}, \boldsymbol{p}; \boldsymbol{\omega}) \to \boldsymbol{c}$. On input ck and univariate polynomials $\boldsymbol{p} = [p_i]_{i=1}^n$ over the field $\mathbb{F}$ with $\deg(p_i) \leq D$, $\mathsf{PC_s}.\mathsf{Commit}$ outputs commitments $\boldsymbol{c} = [c_i]_{i=1}^n$ to the polynomials $\boldsymbol{p}$. The randomness $\boldsymbol{\omega} = [\omega_i]_{i=1}^n$ is used if the commitments $\boldsymbol{c}$ are meant to be hiding.
- $\mathsf{PC_s}.\mathsf{Open}(\mathsf{ck}, \boldsymbol{p}, z, \xi; \boldsymbol{\omega}) \to \pi$. On input ck, univariate polynomials $\boldsymbol{p} = [p_i]_{i=1}^n$, evaluation point $z \in \mathbb{F}$, and opening challenge ξ, $\mathsf{PC_s}.\mathsf{Open}$ outputs an evaluation proof π. The randomness $\boldsymbol{\omega}$ must equal the one previously used in $\mathsf{PC_s}.\mathsf{Commit}$.
- $\mathsf{PC_s}.\mathsf{Check}(\mathsf{rk}, \boldsymbol{c}, z, \boldsymbol{v}, \pi, \xi) \in \{0, 1\}$. On input rk, commitments $\boldsymbol{c} = [c_i]_{i=1}^n$, evaluation point $z \in \mathbb{F}$, alleged evaluations $\boldsymbol{v} = [v_i]_{i=1}^n$, evaluation proof π, and opening challenge ξ, $\mathsf{PC_s}.\mathsf{Check}$ outputs 1 if π attests that, for each $i \in [n]$, the polynomial committed in c_i has degree at most D and evaluates to v_i at z.

The polynomial commitment scheme satisfies the completeness and extractability properties defined below. The polynomial commitment scheme is (perfectly) hiding if it also satisfies the hiding property defined below.

Definition B.1 (Completeness). *For every maximum degree bound $D \in \mathbb{N}$ and efficient adversary $\mathcal{A}$ it holds that*

$$\Pr\left[\begin{array}{c|r} \deg(\boldsymbol{p}) \leq D & (\mathsf{ck}, \mathsf{rk}) \leftarrow \mathsf{PC_s}.\mathsf{Setup}(1^\lambda, D) \\ \Downarrow & (\boldsymbol{p}, z, \xi) \leftarrow \mathcal{A}(\mathsf{ck}, \mathsf{rk}) \\ \mathsf{PC_s}.\mathsf{Check}(\mathsf{rk}, \boldsymbol{c}, z, \boldsymbol{v}, \pi, \xi) = 1 & \boldsymbol{c} \leftarrow \mathsf{PC_s}.\mathsf{Commit}(\mathsf{ck}, \boldsymbol{p}) \\ & \boldsymbol{v} \leftarrow \boldsymbol{p}(z) \\ & \pi \leftarrow \mathsf{PC_s}.\mathsf{Open}(\mathsf{ck}, \boldsymbol{p}, z, \xi) \end{array}\right] = 1 \; .$$

Definition B.2 (Extractability). *For every maximum degree bound $D \in \mathbb{N}$ and efficient adversary $\mathcal{A}$, there exists an efficient extractor $\mathcal{E}$ such that for every round bound $\mathsf{r} \in \mathbb{N}$, efficient public-coin challenger $\mathcal{C}$, efficient query sampler $\mathcal{Q}$, and efficient adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ the following probability is negligibly close*

$$\Pr\left[\begin{array}{c|r} & (\mathsf{ck}, \mathsf{rk}) \leftarrow \mathsf{PC_s}.\mathsf{Setup}(1^\lambda, D) \\ \hline & \textit{For } i = 1, \dots, \mathsf{r}\textit{:} \\ & \rho_i \leftarrow \mathcal{C}(\mathsf{ck}, \mathsf{rk}, i) \\ \mathsf{PC_s}.\mathsf{Check}(\mathsf{rk}, \boldsymbol{c}, z, \boldsymbol{v}, \pi, \xi) = 1 & \boldsymbol{c}_i \leftarrow \mathcal{A}(\mathsf{ck}, \mathsf{rk}, [\rho_j]_{j=1}^i) \\ & \boldsymbol{p}_i \leftarrow \mathcal{E}(\mathsf{ck}, \mathsf{rk}, [\rho_j]_{j=1}^i) \\ \hline \Downarrow & Q \leftarrow \mathcal{Q}(\mathsf{ck}, \mathsf{rk}, [\rho_j]_{j=1}^{\mathsf{r}}) \\ & (\boldsymbol{v}, \mathsf{st}) \leftarrow \mathcal{B}_1(\mathsf{ck}, \mathsf{rk}, [\rho_j]_{j=1}^{\mathsf{r}}, Q) \\ \deg(\boldsymbol{p}) \leq D \textit{ and } \boldsymbol{v} = \boldsymbol{p}(z) & \textit{Sample opening challenge } \xi \\ & \pi \leftarrow \mathcal{B}_2(\mathsf{st}, \xi) \\ & \textit{Set } [c_i]_{i=1}^n := [\boldsymbol{c}_i]_{i=1}^{\mathsf{r}},\ [p_i]_{i=1}^n := [\boldsymbol{p}_i]_{i=1}^{\mathsf{r}},\ [d_i]_{i=1}^n := [\boldsymbol{d}_i]_{i=1}^{\mathsf{r}} \\ & \textit{Parse } Q \textit{ as } T \times \{z\} \textit{ for some } T \subseteq [n] \textit{ and } z \in \mathbb{F} \\ & \textit{Set } \boldsymbol{c} := [c_i]_{i \in T},\ \boldsymbol{p} := [p_i]_{i \in T},\ \boldsymbol{d} := [d_i]_{i \in T} \end{array}\right] \; .$$

Definition B.3 (Succinctness). *A polynomial commitment scheme is* **succinct** *if the size of commitments, the size of evaluation proofs, and the time to check an opening are all independent of the degree of the committed polynomials. That is, $|\boldsymbol{c}| = n \cdot \mathrm{poly}(\lambda)$, $|\pi| = \mathrm{poly}(\lambda)$, and $\mathsf{time}(\mathsf{Check}) = n \cdot \mathrm{poly}(\lambda)$.*

Definition B.4 (Hiding). *There exists a polynomial-time simulator $\mathcal{S} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ such that, for every maximum degree bound $D \in \mathbb{N}$, round bound $\mathsf{r} \in \mathbb{N}$, and (even unbounded) non-uniform adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, the probability that $b = 1$ in the following two experiments is identical.*

$\mathsf{Real}(1^\lambda, D, \mathcal{A})$:

1. $(\mathsf{ck}, \mathsf{rk}) \leftarrow \mathsf{PC_s}.\mathsf{Setup}(1^\lambda, D)$.
2. *Letting* $\boldsymbol{c}_0 := \bot$, *for* $i = 1, \dots, \mathsf{r}$:
   (a) $(\boldsymbol{p}_i, \mathsf{h}_i) \leftarrow \mathcal{A}_1(\mathsf{ck}, \mathsf{rk}, \boldsymbol{c}_0, \boldsymbol{c}_1, \dots, \boldsymbol{c}_{i-1})$.
   (b) *If* $\mathsf{h}_i = 0$: *sample commitment randomness* $\boldsymbol{\omega}_i$.
   (c) *If* $\mathsf{h}_i = 1$: *set randomness* $\boldsymbol{\omega}_i$ *to* $\bot$.
   (d) $\boldsymbol{c}_i \leftarrow \mathsf{PC_s}.\mathsf{Commit}(\mathsf{ck}, \boldsymbol{p}_i; \boldsymbol{\omega}_i)$.
3. $\boldsymbol{c} := [\boldsymbol{c}_i]_{i=1}^{\mathsf{r}}$, $\boldsymbol{p} := [\boldsymbol{p}_i]_{i=1}^{\mathsf{r}}$, $\boldsymbol{\omega} := [\boldsymbol{\omega}_i]_{i=1}^{\mathsf{r}}$.
4. $([Q_j]_{j=1}^{\tau}, [\xi_j]_{j=1}^{\tau}, \mathsf{st}) \leftarrow \mathcal{A}_2(\mathsf{ck}, \mathsf{rk}, \boldsymbol{c})$.
5. *For* $j \in [\tau]$:
   $\pi_j \leftarrow \mathsf{PC_s}.\mathsf{Open}(\mathsf{ck}, \boldsymbol{p}, Q_j, \xi_j; \boldsymbol{\omega})$.
6. $b \leftarrow \mathcal{A}_3(\mathsf{st}, [\pi_j]_{j=1}^{\tau})$.

$\mathsf{Ideal}(1^\lambda, D, \mathcal{A})$:

1. $(\mathsf{ck}, \mathsf{rk}, \mathsf{trap}) \leftarrow \mathcal{S}.\mathsf{Setup}(1^\lambda, D)$.
2. *Letting* $\boldsymbol{c}_0 := \bot$, *for* $i = 1, \dots, \mathsf{r}$:
   (a) $(\boldsymbol{p}_i, \mathsf{h}_i) \leftarrow \mathcal{A}_1(\mathsf{ck}, \mathsf{rk}, \boldsymbol{c}_0, \boldsymbol{c}_1, \dots, \boldsymbol{c}_{i-1})$.
   (b) *If* $\mathsf{h}_i = 0$: *sample randomness* $\boldsymbol{\omega}_i$ *and compute simulated commitments* $\boldsymbol{c}_i \leftarrow \mathcal{S}.\mathsf{Commit}(\mathsf{trap}, |\boldsymbol{p}_i|; \boldsymbol{\omega}_i)$.
   (c) *If* $\mathsf{h}_i = 1$: *set* $\boldsymbol{\omega}_i := \bot$ *and compute (real) commitments* $\boldsymbol{c}_i \leftarrow \mathsf{PC_s}.\mathsf{Commit}(\mathsf{ck}, \boldsymbol{p}_i; \boldsymbol{\omega}_i)$.
3. $\boldsymbol{c} := [\boldsymbol{c}_i]_{i=1}^{\mathsf{r}}$, $\boldsymbol{p} := [\boldsymbol{p}_i]_{i=1}^{\mathsf{r}}$, $\boldsymbol{\omega} := [\boldsymbol{\omega}_i]_{i=1}^{\mathsf{r}}$.
4. *Zero out hidden polynomials:* $\boldsymbol{p}' := [\mathsf{h}_i \boldsymbol{p}_i]_{i=1}^{\mathsf{r}}$.
5. $([Q_j]_{j=1}^{\tau}, [\xi_j]_{j=1}^{\tau}, \mathsf{st}) \leftarrow \mathcal{A}_2(\mathsf{ck}, \mathsf{rk}, \boldsymbol{c})$.
6. *For* $j \in [\tau]$:
   $\pi_j \leftarrow \mathcal{S}.\mathsf{Open}(\mathsf{trap}, \boldsymbol{p}', \boldsymbol{p}(Q_j), Q_j, \xi_j; \boldsymbol{\omega})$.
7. $b \leftarrow \mathcal{A}_3(\mathsf{st}, [\pi_j]_{j=1}^{\tau})$.

Above we implicitly assume that $\mathcal{A}_1$ outputs $\mathrm{poly}(\lambda)$ polynomials in each round, and that $\mathcal{A}_2$ outputs $\tau = \mathrm{poly}(\lambda)$ query sets Q_j, so that $\mathsf{PC_s}.\mathsf{Commit}$, $\mathsf{PC_s}.\mathsf{Open}$, $\mathcal{S}.\mathsf{Commit}$, and $\mathcal{S}.\mathsf{Open}$ are all efficient.

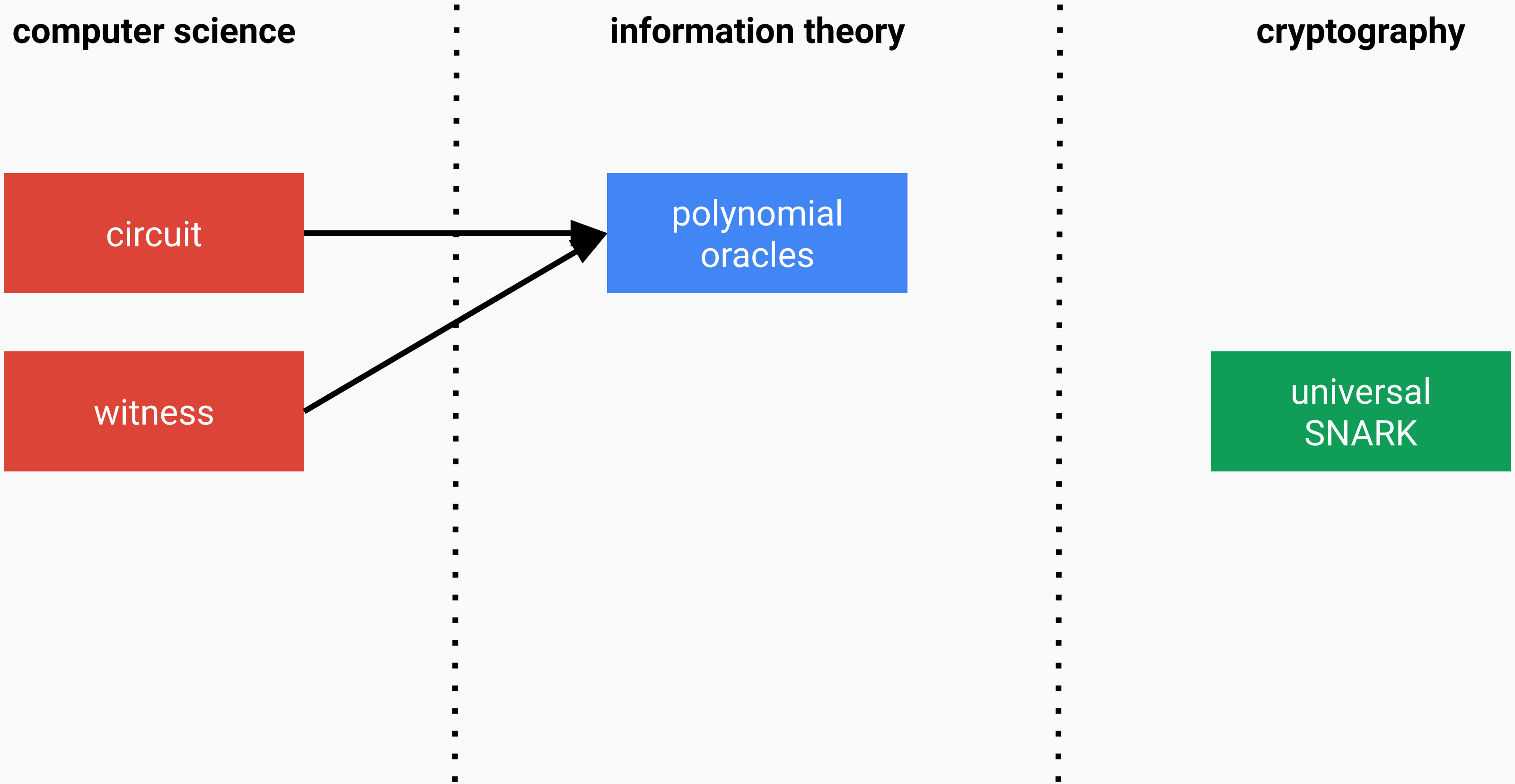
Marlin (Appendix B.1)

pages 54–55 of 75

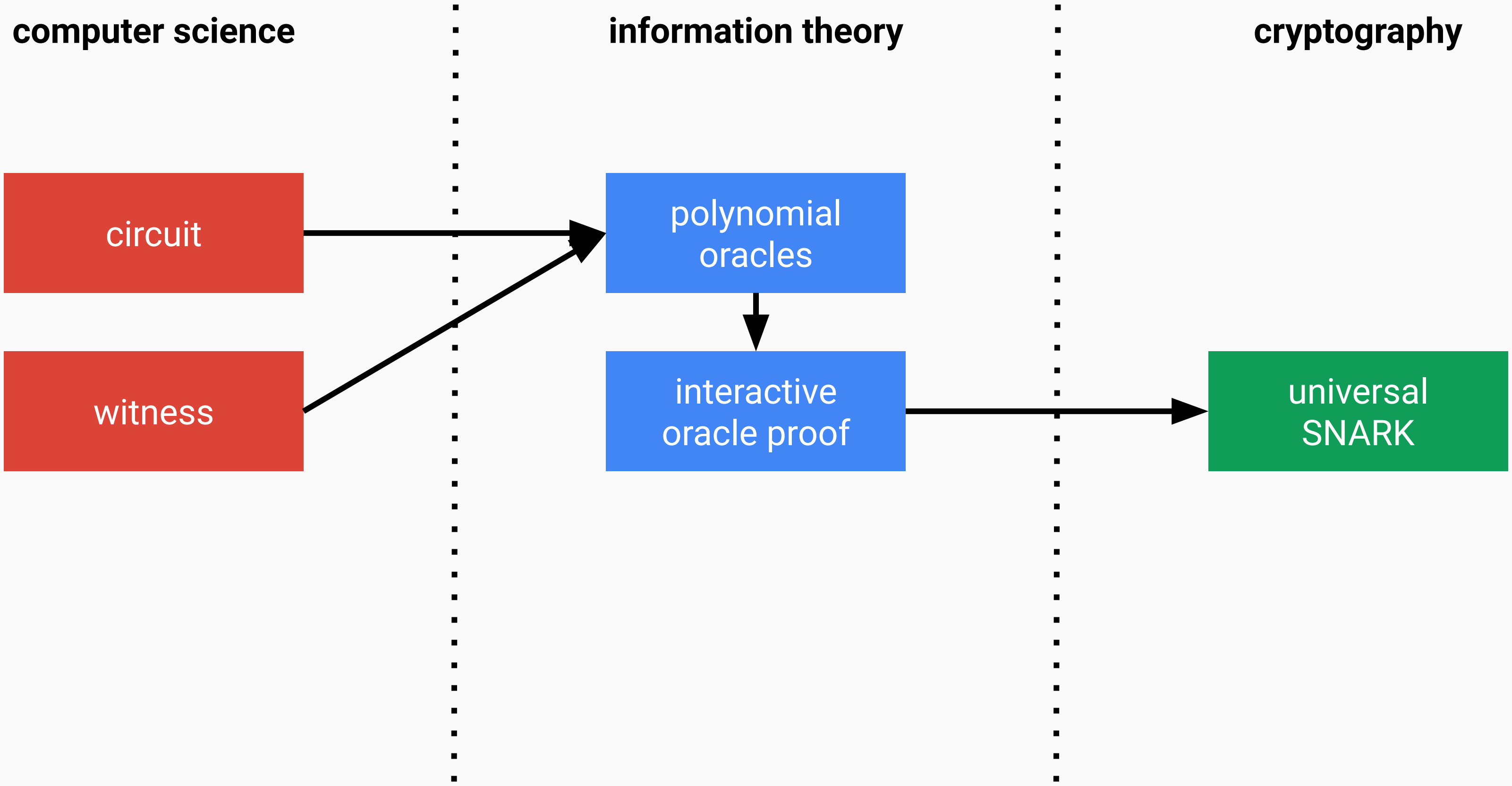
big picture

computer science	information theory	cryptography
circuit		
witness		universal SNARK

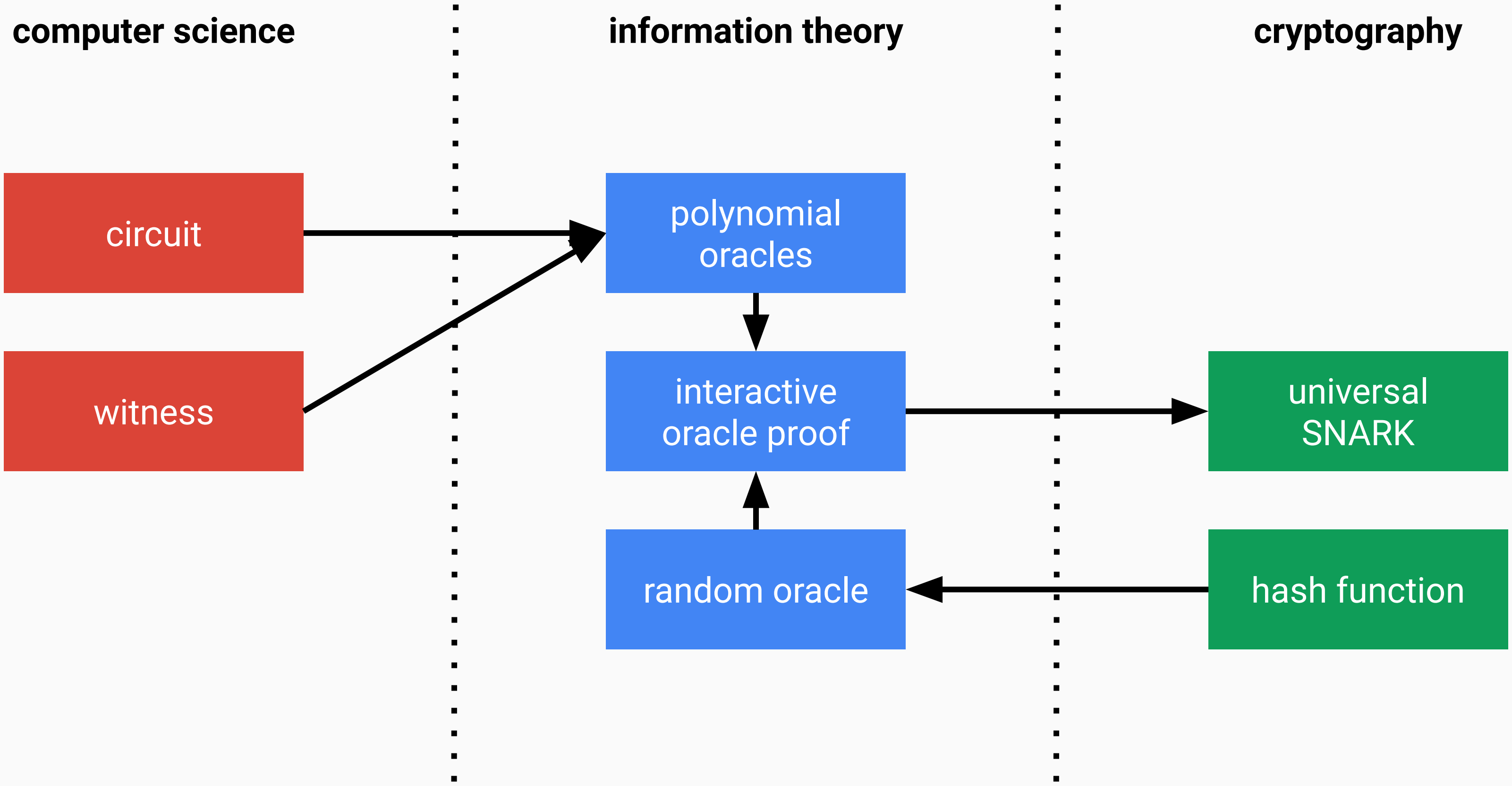
big picture


computer science
information theory
cryptography
circuit
witness
polynomial oracles
universal SNARK

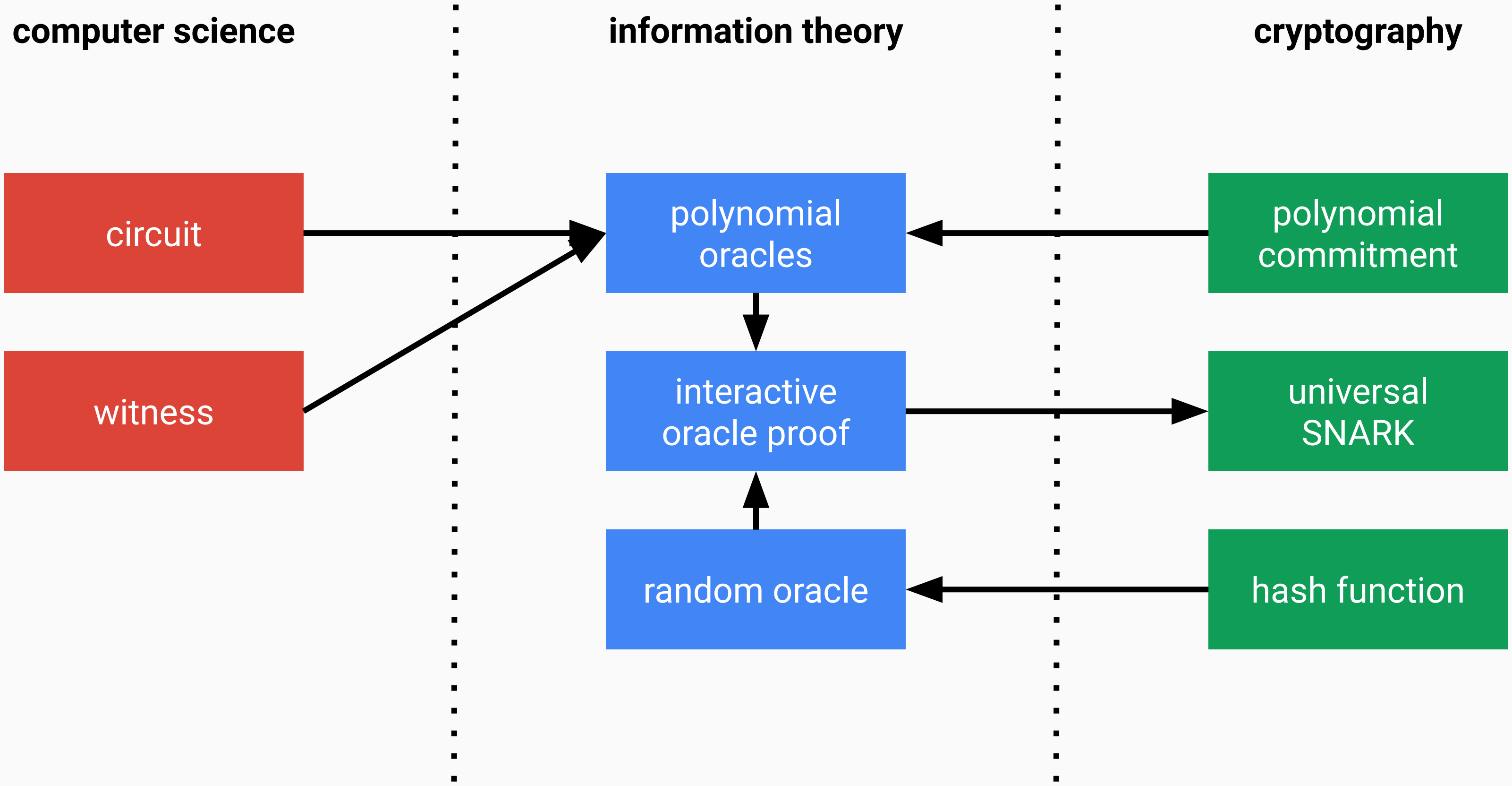
big picture


computer science
information theory
cryptography
circuit
witness
polynomial oracles
interactive oracle proof
universal SNARK

big picture

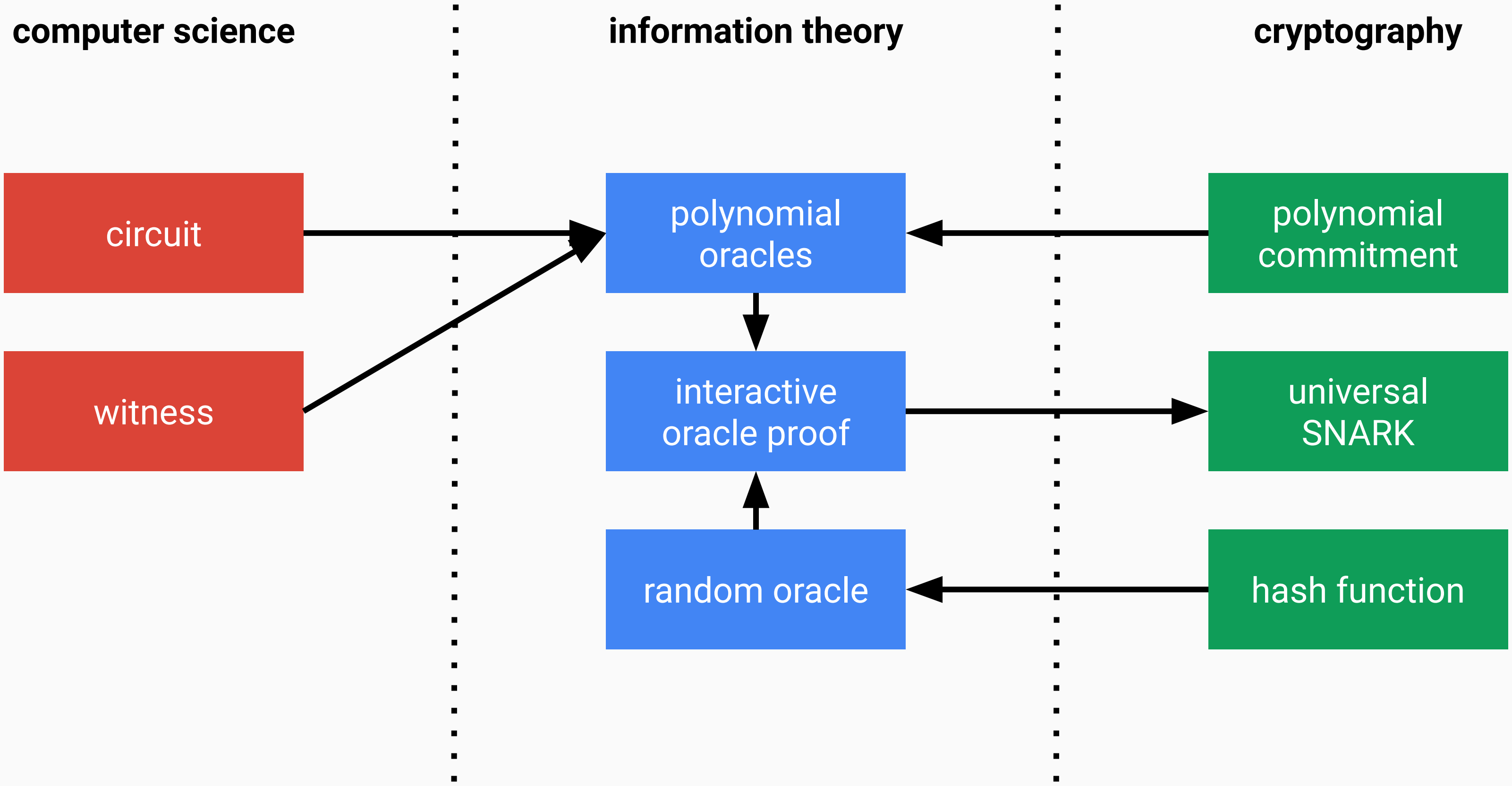

computer science
information theory
cryptography
circuit
witness
polynomial oracles
interactive oracle proof
random oracle
universal SNARK
hash function

big picture

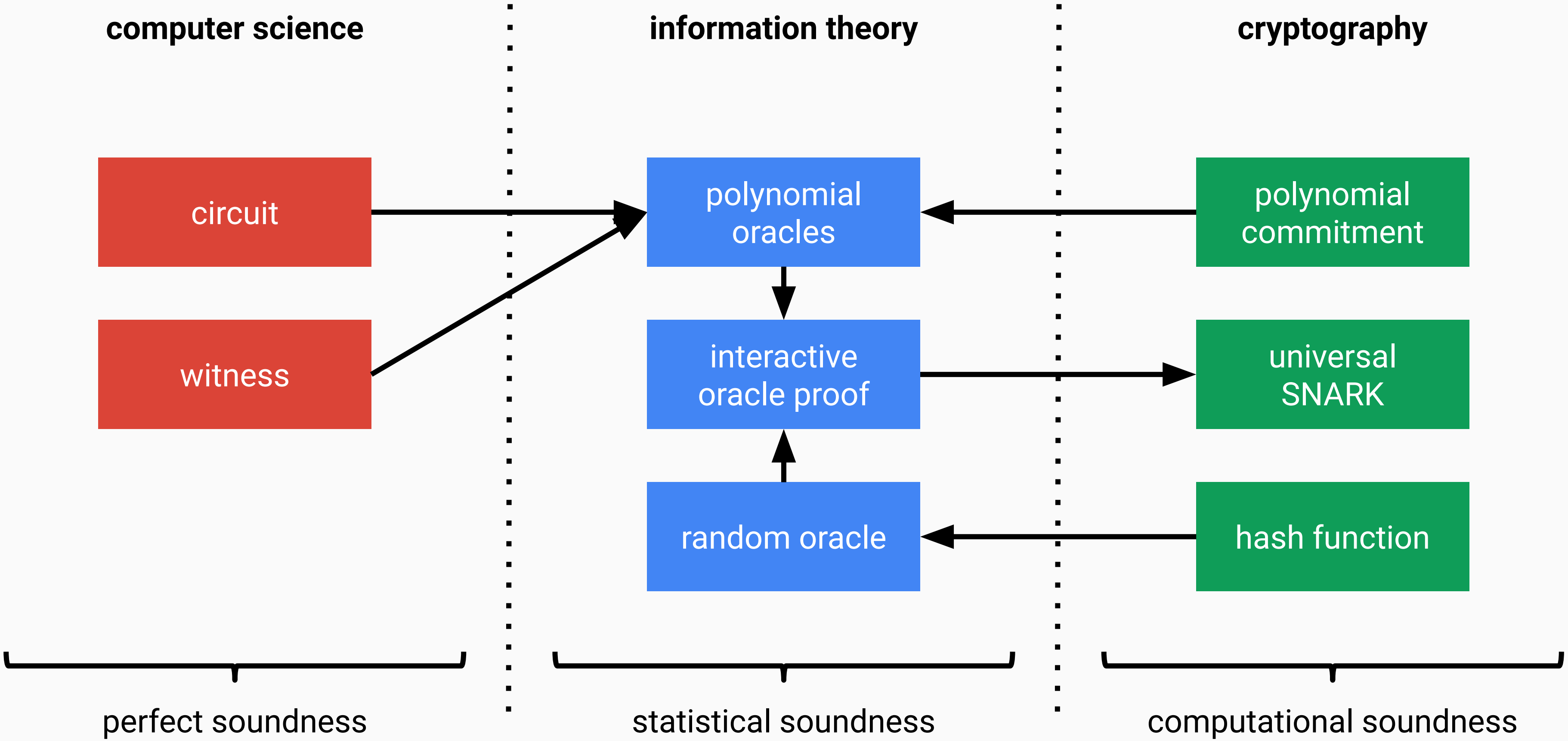


computer science
information theory
cryptography
circuit
witness
polynomial oracles
interactive oracle proof
random oracle
polynomial commitment
universal SNARK
hash function

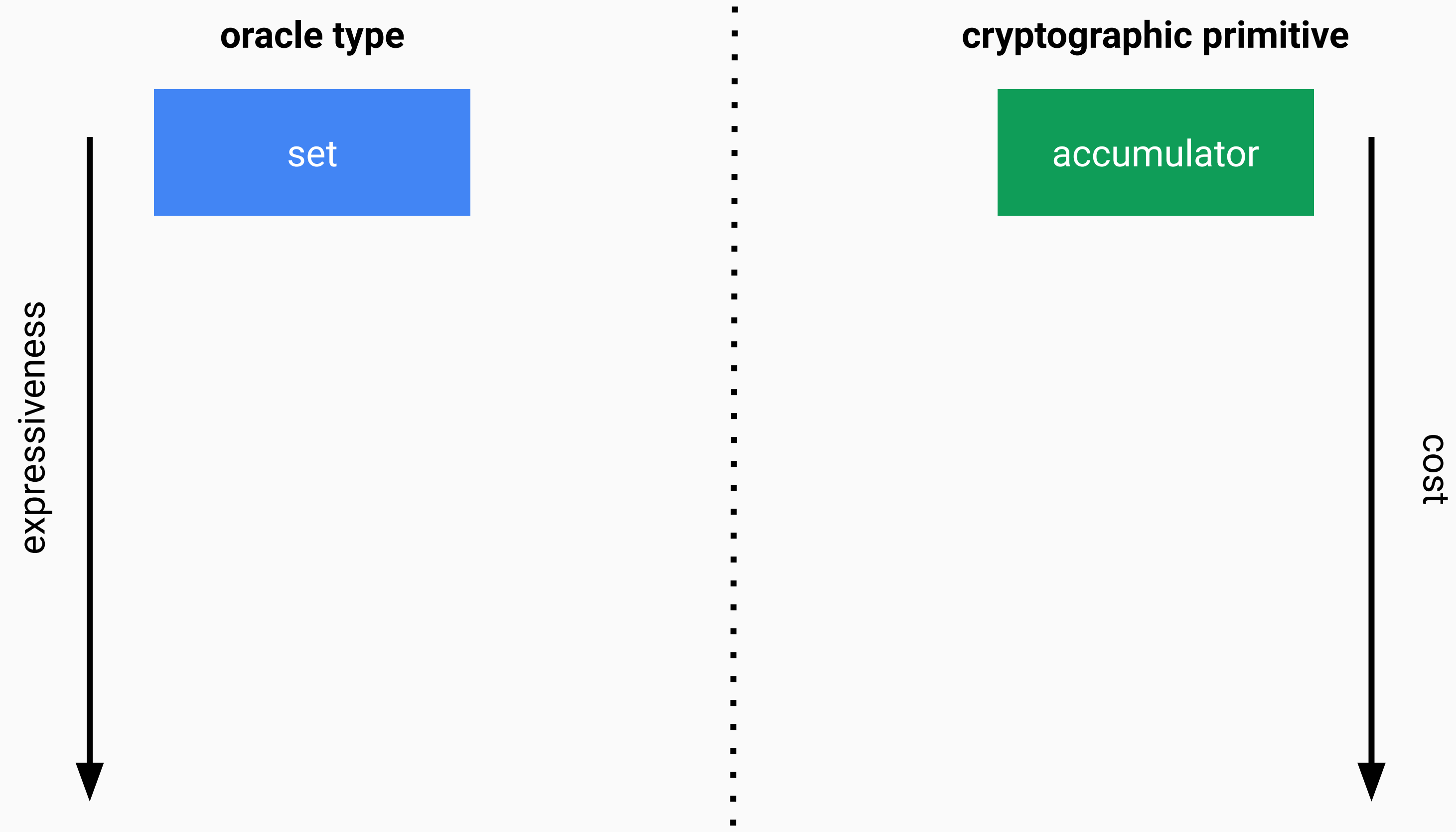
big picture


computer science
information theory
cryptography
circuit
witness
polynomial oracles
interactive oracle proof
random oracle
polynomial commitment
universal SNARK
hash function

big picture


computer science
information theory
cryptography
circuit
witness
polynomial oracles
interactive oracle proof
random oracle
polynomial commitment
universal SNARK
hash function
perfect soundness
statistical soundness
computational soundness

tug of war


oracle type
set
expressiveness
cryptographic primitive
accumulator
cost

tug of war

oracle type

set

vector

expressiveness

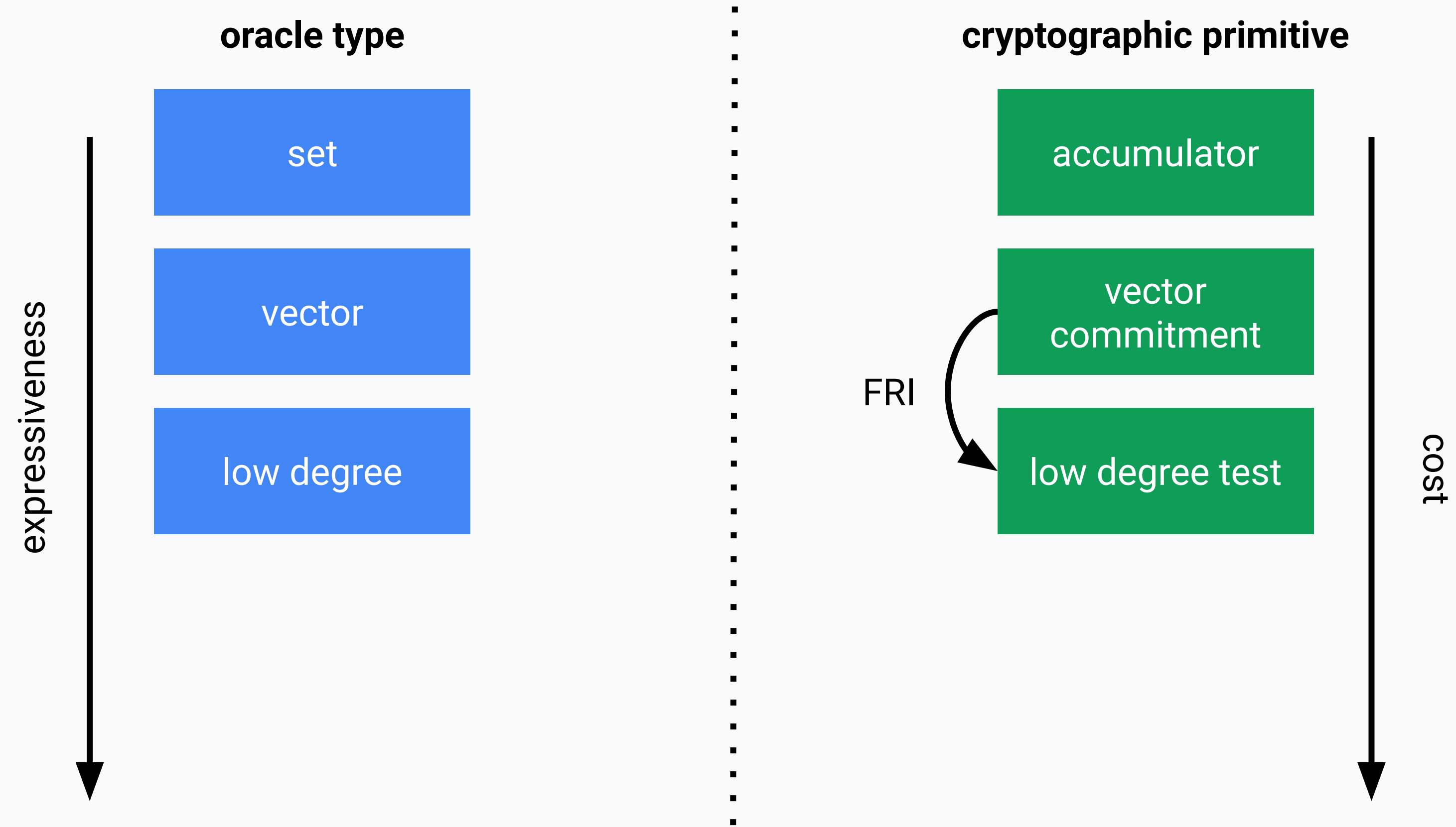
cryptographic primitive

accumulator

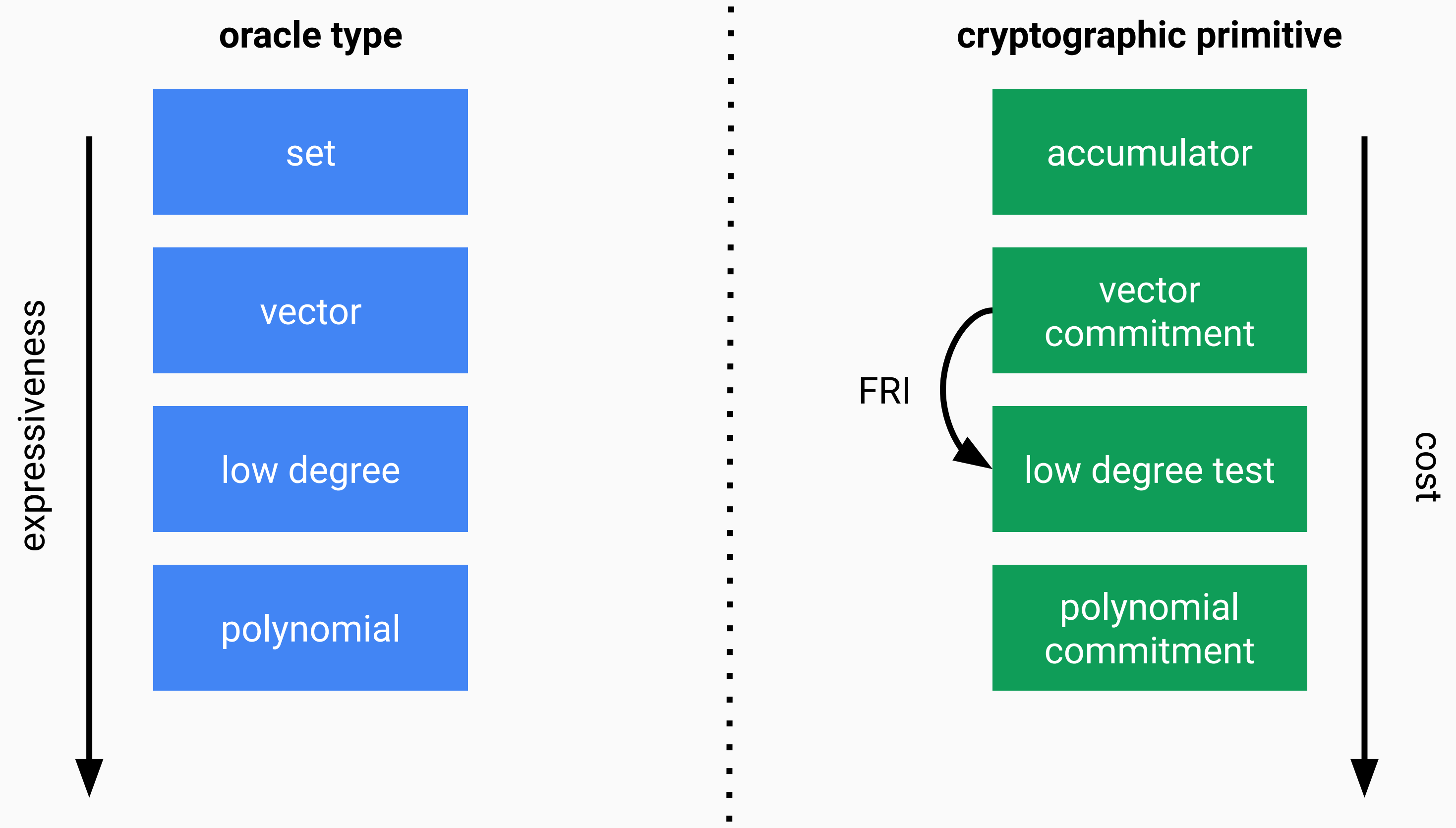
vector commitment

cost

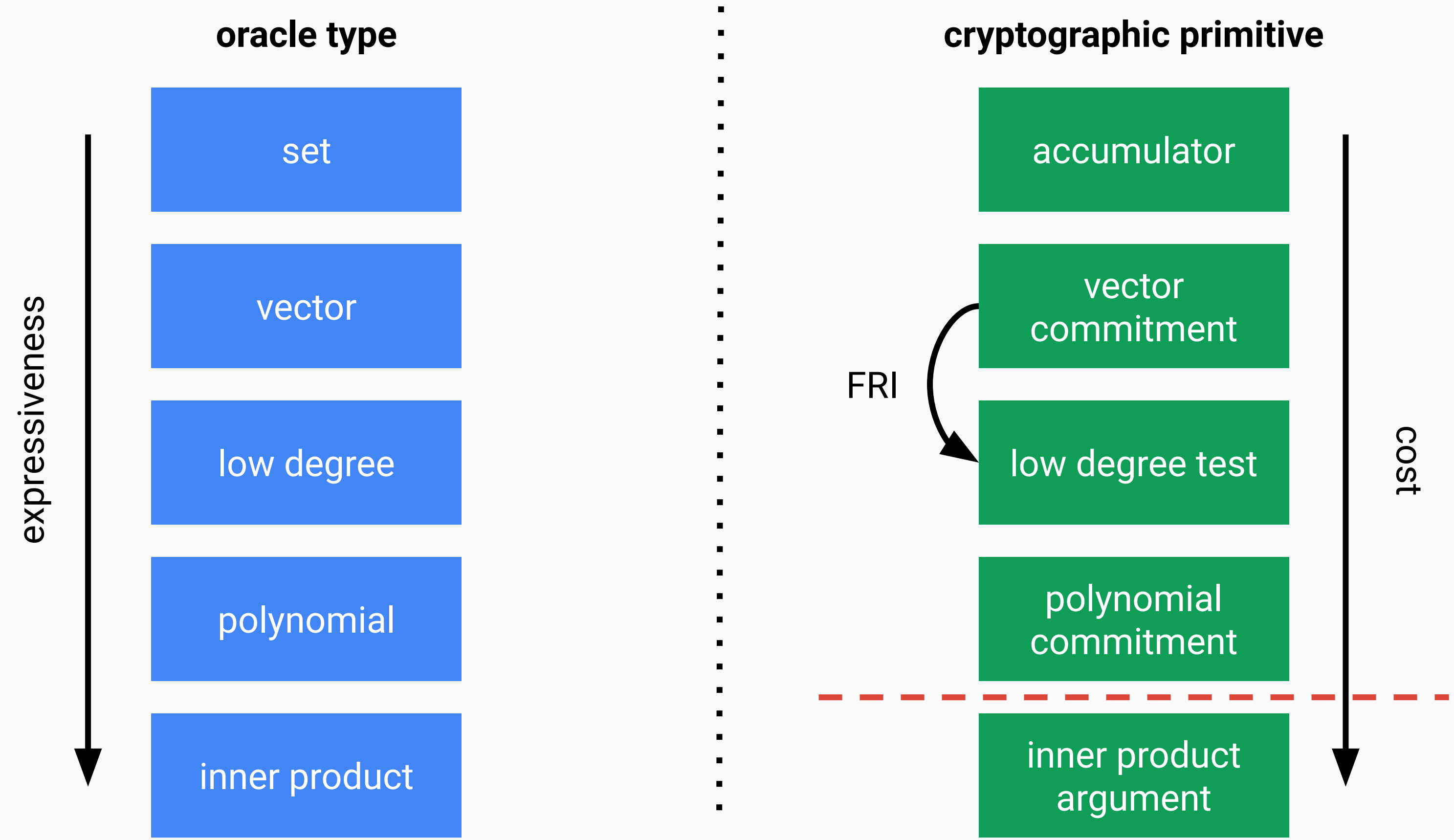
tug of war


oracle type
set
vector
low degree
expressiveness
cryptographic primitive
accumulator
vector commitment
low degree test
FRI
cost

tug of war


oracle type
set
vector
low degree
polynomial
expressiveness
cryptographic primitive
accumulator
vector commitment
FRI
low degree test
polynomial commitment
cost

tug of war



oracle type
set
vector
low degree
polynomial
inner product
expressiveness
cryptographic primitive
accumulator
vector commitment
FRI
low degree test
polynomial commitment
inner product argument
cost

part 1—context

part 2—landscape

part 3—mechanics

part 4—gadgets

setup and commitment

	FRI
	hash function
setup	**H** hash function **w** in **F** root of unity
commitment	**root(f(w^0), ..., f(w^{kd}))**

setup and commitment

	FRI	KZG
	hash function	**pairing group**
setup	**H** hash function **w** in **F** root of unity	G_1, G_2 groups g_1, g_2 generators **e** pairing **s** in **F** secret
commitment	$\mathrm{root}(f(w^0), \ldots, f(w^{kd}))$	$a_0 s^0 g_1 + \ldots + a_n s^d g_1$

setup and commitment

	FRI	KZG	DARK
	hash function	**pairing group**	**unknown order group**
setup	**H** hash function **w** in **F** root of unity	**G_1, G_2** groups **g_1, g_2** generators **e** pairing **s** in **F** secret	**G** unknown order **g** in **G** random **q** large integer
commitment	**root(f(w^0), ..., f(w^{kd}))**	**$a_0 s^0 g_1$ + ... + $a_n s^d g_1$**	**$a_0 q^0 g$ + ... + $a_d q^d g$**

setup and commitment

	FRI	KZG	DARK	Bulletproof
	hash function	**pairing group**	**unknown order group**	**discrete log group**
setup	**H** hash function **w** in **F** root of unity	$\mathbf{G_1}$, $\mathbf{G_2}$ groups $\mathbf{g_1}$, $\mathbf{g_2}$ generators **e** pairing **s** in **F** secret	**G** unknown order **g** in **G** random **q** large integer	**G** elliptic curve $\mathbf{g_i}$ in **G** independent
commitment	$\mathbf{root(f(w^0), ..., f(w^{kd}))}$	$\mathbf{a_0 s^0 g_1 + ... + a_n s^d g_1}$	$\mathbf{a_0 q^0 g + ... + a_d q^d g}$	$\mathbf{a_0 g_0 + ... + a_d g_d}$

$\underbrace{\text{KZG} \quad \text{DARK} \quad \text{Bulletproof}}_{\text{algebraic (with linear homomorphism)}}$

algebraic
(with linear homomorphism)

comparing setups

	FRI
	hash function
transparent	
succinct	
unbounded	
updatable	
post-quantum	

comparing setups

	FRI	KZG
	hash function	pairing group
transparent		
succinct		
unbounded		
updatable		
post-quantum		

comparing setups

	FRI	KZG	DARK		
	hash function	pairing group	RSA group	class group	Jacobian group
transparent					
succinct					
unbounded					
updatable					
post-quantum					

comparing setups

	FRI	KZG	DARK			Bulletproof
	hash function	pairing group	RSA group	class group	Jacobian group	discrete log group
transparent						
succinct						
unbounded						
updatable						
post-quantum						

asymptotic performance

max(commitment size, opening proof size)

	FRI
	hash function
size	$O(\log^2(d))$
verifier time	$O(\log^2(d))$
prover time	$O(d*\log(d))$

max(commitment time, opening time)

asymptotic performance

max(commitment size**,** opening proof size)

	FRI	KZG
	hash function	pairing group
size	$O(\log^2(d))$	$O(1)$
verifier time	$O(\log^2(d))$	$O(1)$
prover time	$O(d*\log(d))$	$O(d)$

max(commitment time**,** opening time)

asymptotic performance

max(commitment size, opening proof size)

	FRI	KZG	DARK
	hash function	pairing group	unknown order group
size	$O(\log^2(d))$	$O(1)$	$O(\log(d))$
verifier time	$O(\log^2(d))$	$O(1)$	$O(\log(d))$
prover time	$O(d*\log(d))$	$O(d)$	$O(d)$

max(commitment time, opening time)

asymptotic performance

max(commitment size, opening proof size)

	FRI	KZG	DARK	Bulletproof
	hash function	pairing group	unknown order group	discrete log group
size	$O(\log^2(d))$	$O(1)$	$O(\log(d))$	$O(\log(d))$
verifier time	$O(\log^2(d))$	$O(1)$	$O(\log(d))$	$O(d)$
prover time	$O(d*\log(d))$	$O(d)$	$O(d)$	$O(d)$

max(commitment time, opening time)

commitment size (with security parameter **λ** and **d** ≪ **λ**)

subexponential attacks
(sieve based)

polynomial attack

FRI	KZG	DARK			Bulletproof
hash function	**pairing group**	**RSA group**	**class group**	**Jacobian group**	**discrete log group**
$O(1) * O(\lambda)$	**$O(1) * O(\lambda^3)$**	**$O(1) * O(\lambda^3)$**	**$O(1) * O(\lambda^2)$**	**???**	**$O(1) * O(\lambda)$**

commitment size (with security parameter **λ** and **d** ≪ **λ**)

subexponential attacks
(sieve based)

polynomial attack

FRI	KZG	DARK			Bulletproof
hash function	**pairing group**	**RSA group**	**class group**	**Jacobian group**	**discrete log group**
$O(1) * O(\lambda)$	$O(1) * O(\lambda^3)$	$O(1) * O(\lambda^3)$	$O(1) * O(\lambda^2)$	???	$O(1) * O(\lambda)$

target
128 bits
security

SHA256	**BLS12-381**	**RSA**	**class group**	**Jacobian group**	**secp256k1**
256 bits	381 bits	4,096 bits	1,827 bits	303 bits	256 bits
(32 bytes)	(48 bytes)	(512 bytes)	(229 bytes)	(38 bytes)	(32 bytes)

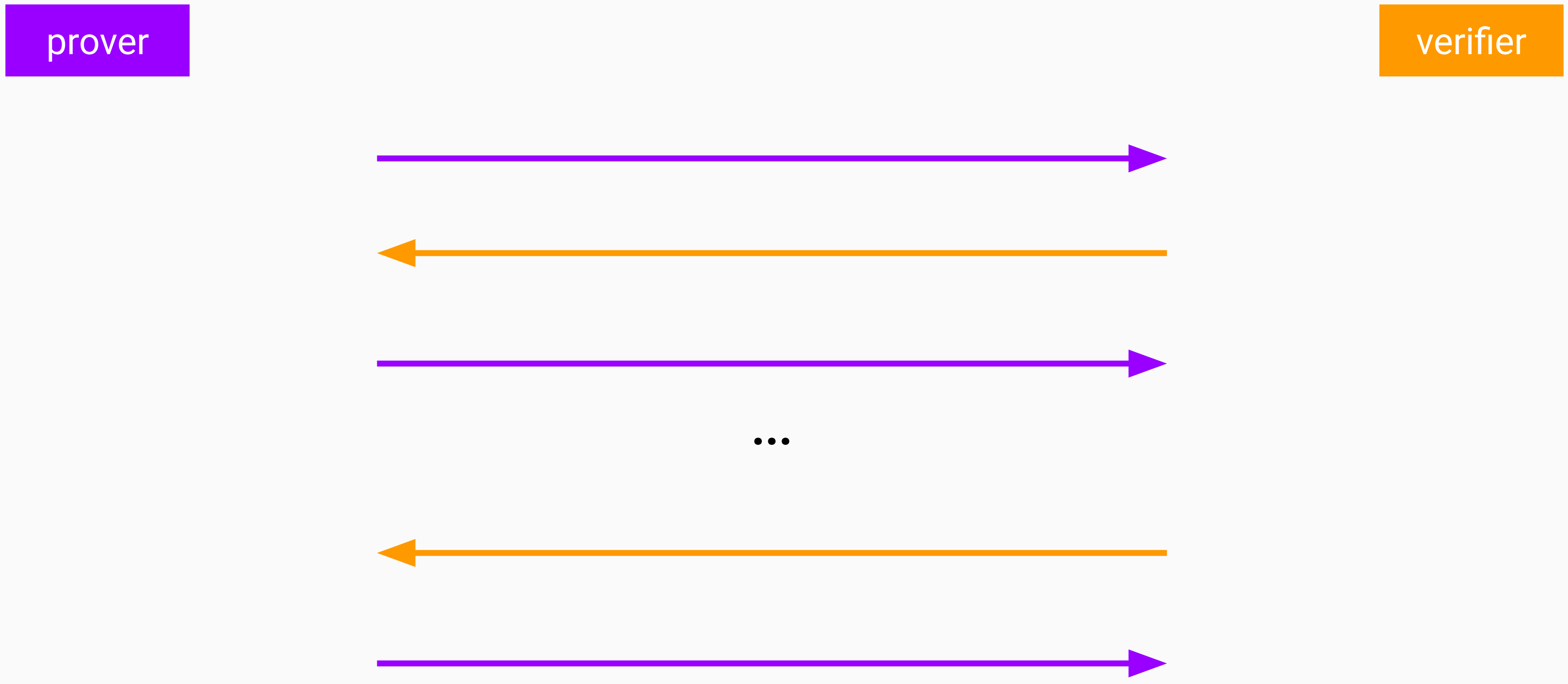
part 1—context

part 2—landscape

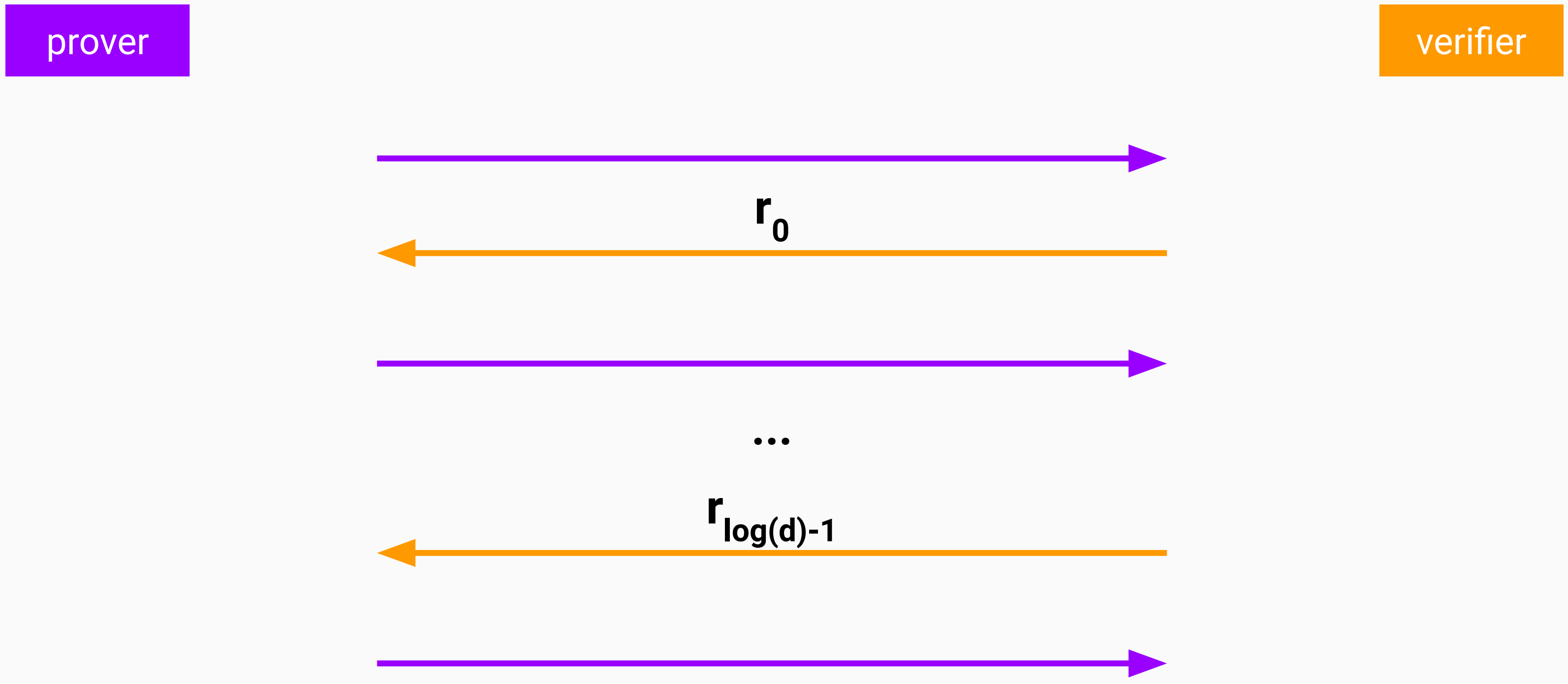
part 3—mechanics

part 4—gadgets

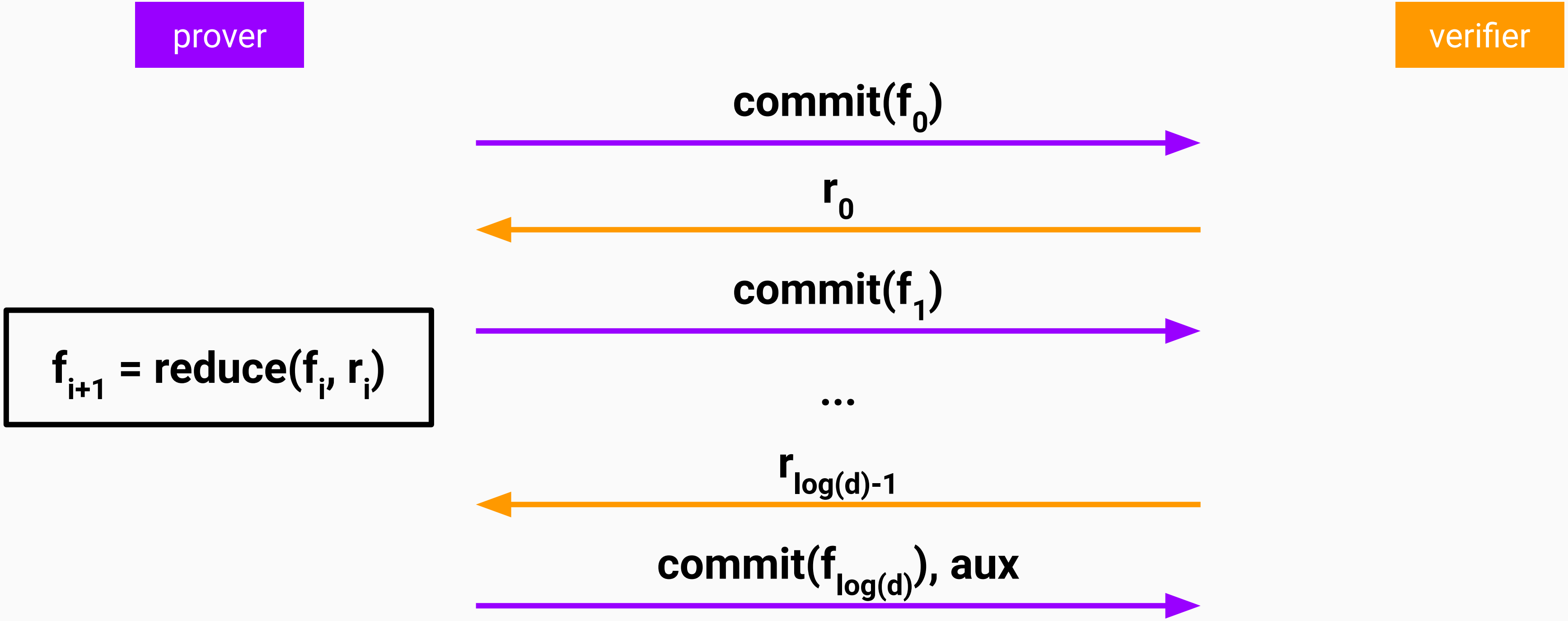
commit-reduce low degree test



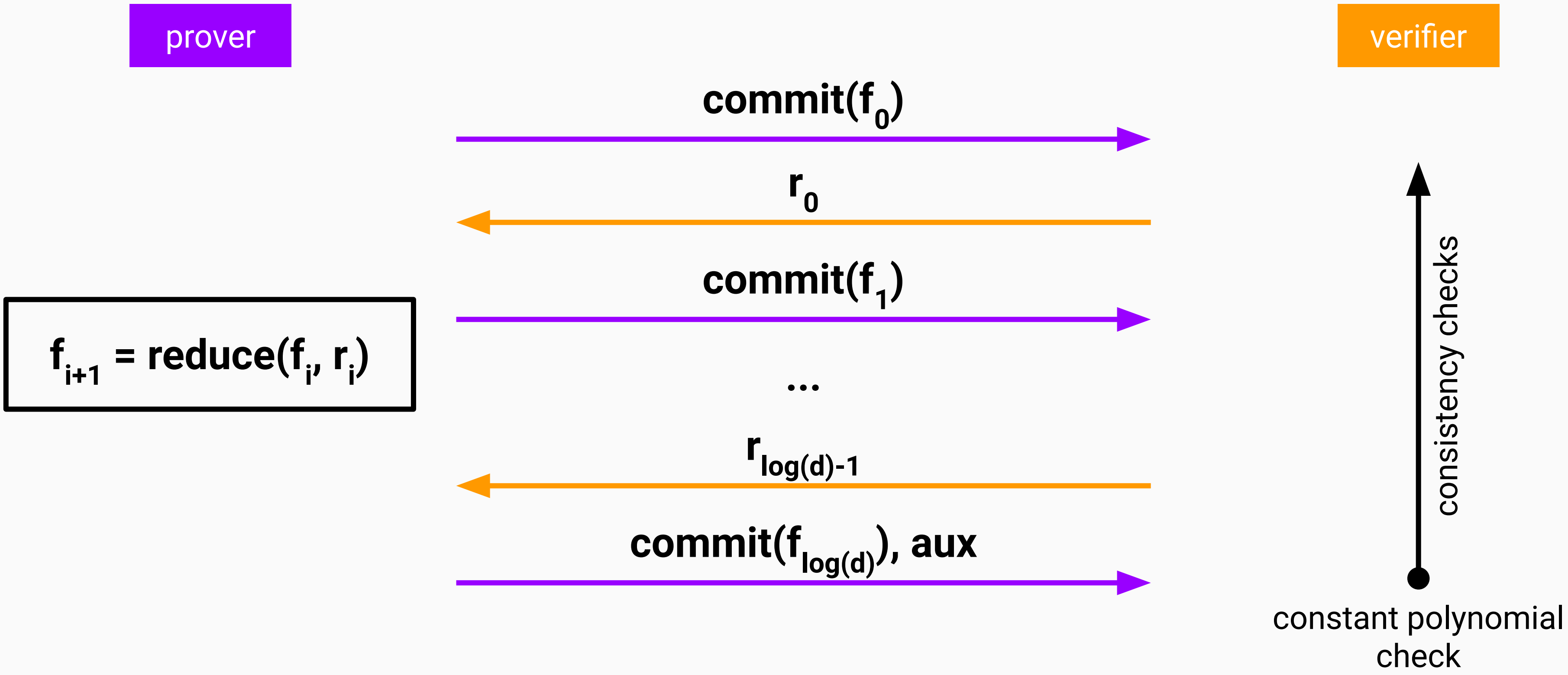
commit-reduce low degree test



commit-reduce low degree test



commit-reduce low degree test



even-odd and left-right decomposition

$$f(X) = even(f)(X^2) + X*odd(f)(X^2)$$

even-odd decomposition

$$f(X) = left(f)(X) + X^{d/2}*right(f)(X)$$

left-right decomposition

decompose-reduce

$$f(X) = even(f)(X^2) + X*odd(f)(X^2)$$

even-odd decomposition

$$f(X) = left(f)(X) + X^{d/2}*right(f)(X)$$

left-right decomposition

	hash function (FRI)	UO group (DARK)	discrete log group (Bulletproof)
coefficients	even(f) + r*odd(f)	even(f) + r*odd(f)	r*left(f) + r^{-1}*right(f)
basis	N/A	g	r^{-1}*left(g) + r*right(g)

consistency checks

FRI (hash function)

$$2z f_{i+1}(z^2) \stackrel{?}{=} z(f_i(z) + f_i(-z)) + r_i(f_i(z) - f_i(-z))$$

DARK (UO group)

Bulletproof (discrete log)

consistency checks

FRI (hash function)

$$2zf_{i+1}(z^2) \;?\!= z(f_i(z) + f_i(-z)) + r_i(f_i(z) - f_i(-z))$$

DARK (UO group)

$$\text{commit}(f_{i+1}) \;?\!= \text{commit}(\text{even}(f_i)) + r_i * q * \text{commit}(\text{odd}(f_i))$$

Bulletproof (discrete log)

consistency checks

FRI (hash function)

$$2zf_{i+1}(z^2) \mathrel{?=} z(f_i(z) + f_i(-z)) + r_i(f_i(z) - f_i(-z))$$

DARK (UO group)

$$\mathrm{commit}(f_{i+1}) \mathrel{?=} \mathrm{commit}(\mathrm{even}(f_i)) + r_i * q * \mathrm{commit}(\mathrm{odd}(f_i))$$

Bulletproof (discrete log)

$$\mathrm{commit}(f_{i+1}) \mathrel{?=} \mathrm{commit}(f_i) + (r_i)^2 L + (r_i)^{-2} R$$

quotient argument openings

$$\boxed{f(X) - f(z) = q(X)(X - z)}$$

FRI (hash function)

$(f(X) - f(z))/(X - z)$ low degree proof

(within unique decoding radius)

KZG10 (pairing group)

quotient argument openings

$$\boxed{f(X) - f(z) = q(X)(X - z)}$$

FRI (hash function)

$(f(X) - f(z))/(X - z)$ low degree proof

(within unique decoding radius)

KZG10 (pairing group)

$$e(\text{commit}(f) - f(z), g_2) \stackrel{?}{=} e(\text{commit}(q), (s - z)g_2)$$

other openings

DARK (UO group)

$\text{even}(f_i)(z), \text{odd}(f_i)(z)$

Bulletproof (discrete log)

$\langle \text{coeff}(f), \text{powers}(x) \rangle$

recent developments

novel constructions

- lattice-based polynomial commitment
- Jacobian groups with unknown order
- sparse polynomial commitments

part 1—context
part 2—landscape
part 3—mechanics
part 4—gadgets

modularity

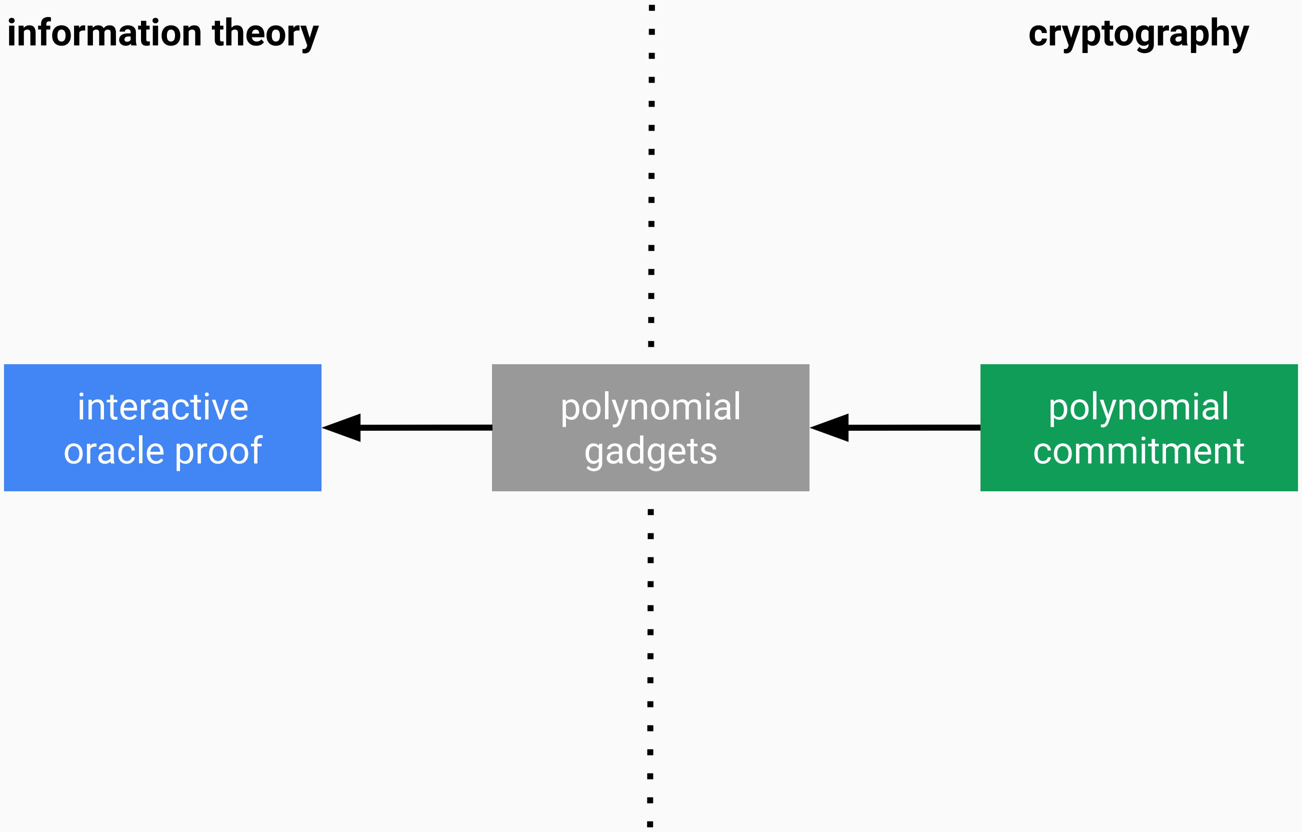
information theory

cryptography

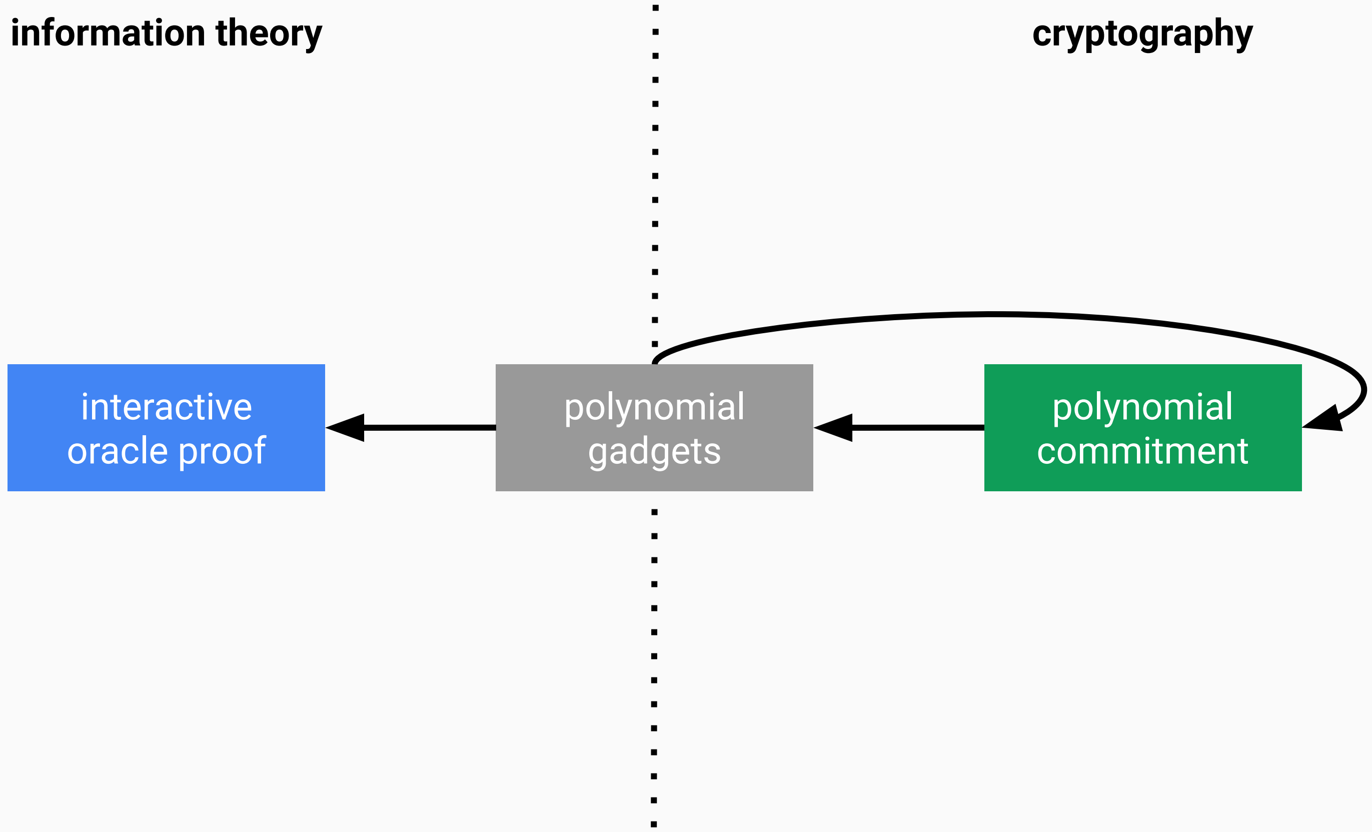
interactive oracle proof

polynomial commitment

modularity


information theory
cryptography
interactive oracle proof
polynomial gadgets
polynomial commitment

modularity


information theory
cryptography
interactive oracle proof
polynomial gadgets
polynomial commitment

testing polynomial identities

fundamental theorem of algebra

f_1, f_2 low-degree polynomials

$f_1 = f_2$ with high probability

$\Leftrightarrow$

$f_1(z) = f_2(z)$ at random point z

Schwartz–Zippel lemma

$f_1(X), \ldots, f_k(X)$ low-degree polynomials

$G(X_1, \ldots, X_K, Y)$ low-degree

$G(f_1, \ldots, f_n, Y) = 0$ with high probability

$\Leftrightarrow$

$G(f_1, \ldots, f_n, Y)|_{X=z}$ at random point z

basic tricks

	trick
range	**$(f \mathbin{//} Z_s)*Z_s$**

range check

commitments to **f, f // $\mathbf{Z_s}$**

range check

commitments to **f**, **f** // $\mathbf{Z_s}$

$\xrightarrow{\hspace{6em}}$

z

$\xleftarrow{\hspace{6em}}$

range check

commitments to $\mathbf{f}$, $\mathbf{f} \;//\; \mathbf{Z}_s$

$\mathbf{z}$

openings of $\mathbf{f}$, $\mathbf{f} \;//\; \mathbf{Z}_s$ at $\mathbf{z}$

range check

commitments to $\mathbf{f}, \mathbf{f} \mathbin{/\!/} \mathbf{Z_S}$

$\mathbf{z}$

openings of $\mathbf{f}, \mathbf{f} \mathbin{/\!/} \mathbf{Z_S}$ at $\mathbf{z}$

check that $\mathbf{f(z)} = \mathbf{(f \mathbin{/\!/} Z_S)(z)*Z_S(z)}$

basic tricks

	trick
range	$(f \,//\, Z_S)*Z_S$
multi-point opening	$(f \,//\, Z_S)*Z_S + f \,\%\, Z_S$

multi-point opening

in the Lagrange basis
(evaluations of **f** on **S**)

commitments to **f**, **f // Z_S** plus **f % Z_S**

multi-point opening

in the Lagrange basis
(evaluations of **f** on **S**)

commitments to **f**, **f** // $\mathbf{Z_S}$ plus **f** % $\mathbf{Z_S}$

z

multi-point opening

in the Lagrange basis
(evaluations of **f** on **S**)

commitments to **f**, **f // $\mathbf{Z_s}$** plus **f % $\mathbf{Z_s}$**

z

openings of **f**, **f // $\mathbf{Z_s}$** at **z**

multi-point opening

in the Lagrange basis
(evaluations of **f** on **S**)

commitments to **f, f // Z_S** plus **f % Z_S**

z

openings of **f, f // Z_S** at **z**

check that **f(z) = (f // Z_S)(z)*Z_S(z) + (f % Z_S)(z)**

basic tricks

	trick
range	$(f \;//\; Z_S)^*Z_S$
multi-point opening	$(f \;//\; Z_S)^*Z_S + f \;\%\; Z_S$
multi-polynomial opening	$Y^0f_0 + \ldots + Y^kf_k$

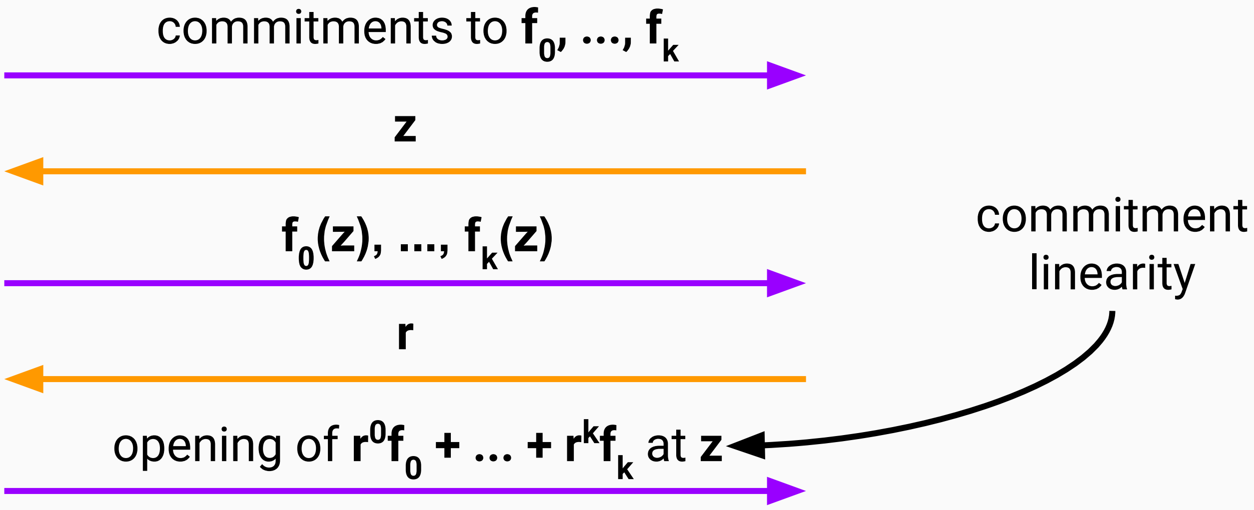
multi-point opening

$$\xrightarrow{\text{commitments to } \mathbf{f_0, \ldots, f_k}}$$

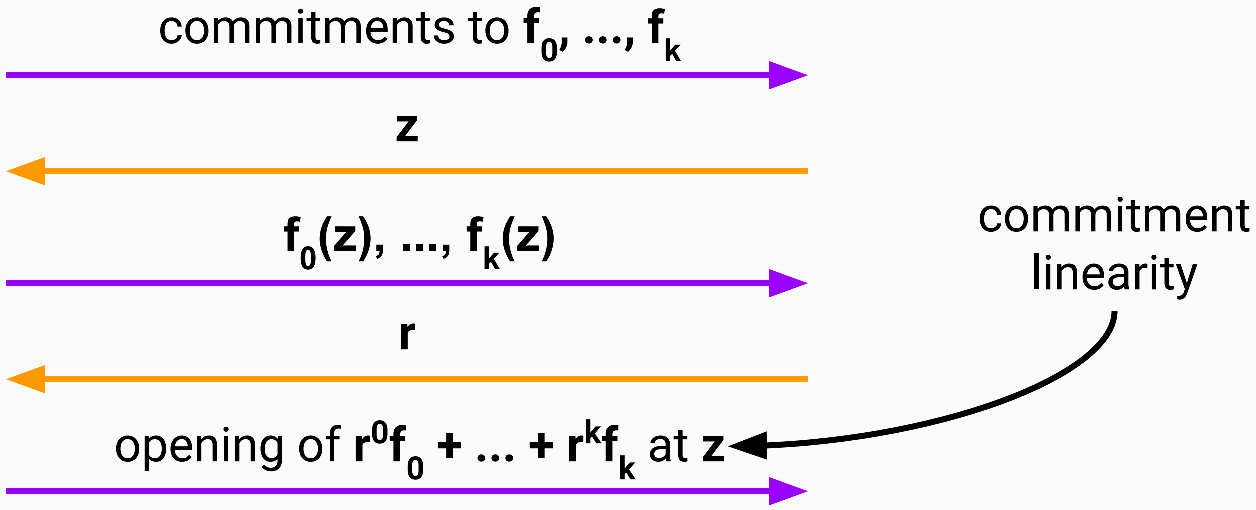
$$\xleftarrow{\mathbf{z}}$$

$$\xrightarrow{\mathbf{f_0(z), \ldots, f_k(z)}}$$

multi-point opening



multi-point opening



$$\boxed{\text{check that } (r^0 f_0 + \ldots + r^k f_k)(z) = r^0 f_0(z) + \ldots + r^k f_k(z)}$$

basic tricks

	trick
range	$(f \mathbin{//} Z_S)*Z_S$
multi-point opening	$(f \mathbin{//} Z_S)*Z_S + f \% Z_S$
multi-polynomial opening	$Y^0f_0 + \ldots + Y^kf_k$
multi-{point, polynomial}	see here
degree bound	$X^{N-d}f(X)$

side note—Lagrange basis

monomial and Lagrange bases

$(\mathbf{a}_0, \dots, \mathbf{a}_d)$

vector

monomial and Lagrange bases

$$(a_0, \ldots, a_d)$$

vector

$$a_0X^0 + \ldots + a_dX^d$$

monomial basis

monomial and Lagrange bases

$$(a_0, \ldots, a_d)$$

vector

$$a_0X^0 + \ldots + a_dX^d$$

monomial basis

$$a_0L_0(X) + \ldots + a_dL_d(X)$$

Lagrange basis

monomial and Lagrange bases

$(a_0, \ldots, a_d)$

vector

$a_0X^0 + \ldots + a_dX^d$

monomial basis

$a_0L_0(X) + \ldots + a_dL_d(X)$

Lagrange basis

FFT

inverse FFT

barycentric formula

evaluation on **i**'th root of unity

$$\boxed{f(z) = \frac{z^n - 1}{n} \sum_{i=1}^{n} \frac{f(\omega^i)}{(z - \omega^i)\omega^i}}$$

Montgomery batch inversion

a_0
a_1
a_2
a_3
a_4
a_5

Montgomery batch inversion

a_0		a_0
a_1		a_0a_1
a_2		$a_0a_1a_2$
a_3		$a_0a_1a_2a_3$
a_4		$a_0a_1a_2a_3a_4$
a_5	$\downarrow$	$\boxed{a_0a_1a_2a_3a_4a_5}$

Montgomery batch inversion

$$
\begin{array}{lll}
a_0 & a_0 & \\
a_1 & a_0a_1 & \\
a_2 & a_0a_1a_2 & \\
a_3 & a_0a_1a_2a_3 & \\
a_4 & a_0a_1a_2a_3a_4 & \\
a_5 & \boxed{a_0a_1a_2a_3a_4a_5} & \boxed{(a_0a_1a_2a_3a_4a_5)^{-1}}
\end{array}
$$

Montgomery batch inversion

a_0	a_0	$(a_0)^{-1}$
a_1	a_0a_1	$(a_0a_1)^{-1}$
a_2	$a_0a_1a_2$	$(a_0a_1a_2)^{-1}$
a_3	$a_0a_1a_2a_3$	$(a_0a_1a_2a_3)^{-1}$
a_4	$a_0a_1a_2a_3a_4$	$(a_0a_1a_2a_3a_4)^{-1}$
a_5	$\boxed{a_0a_1a_2a_3a_4a_5}$	$\boxed{(a_0a_1a_2a_3a_4a_5)^{-1}}$

Montgomery batch inversion

$$
\begin{array}{llll}
a_0 & a_0 & (a_0)^{-1} & (a_0)^{-1} \\
a_1 & a_0 a_1 & (a_0 a_1)^{-1} & (a_1)^{-1} \\
a_2 & a_0 a_1 a_2 & (a_0 a_1 a_2)^{-1} & (a_2)^{-1} \\
a_3 & a_0 a_1 a_2 a_3 & (a_0 a_1 a_2 a_3)^{-1} & (a_3)^{-1} \\
a_4 & a_0 a_1 a_2 a_3 a_4 & (a_0 a_1 a_2 a_3 a_4)^{-1} & (a_4)^{-1} \\
a_5 & \boxed{a_0 a_1 a_2 a_3 a_4 a_5} & \boxed{(a_0 a_1 a_2 a_3 a_4 a_5)^{-1}} & (a_5)^{-1}
\end{array}
$$

/side note—Lagrange basis

more tricks

	Lagrange basis	monomial basis
encode	$a_0L_0(X) + \ldots + a_dL_d(X)$	$a_0X^0 + \ldots + a_dX^d$

more tricks

	Lagrange basis	monomial basis
encode	$a_0 L_0(X) + \ldots + a_d L_d(X)$	$a_0 X^0 + \ldots + a_d X^d$
query	$f(w^i)$	$f_L(X) + a_i X^i + X^{i+1} f_R(X)$

more tricks

	Lagrange basis	monomial basis
encode	$a_0L_0(X) + \ldots + a_dL_d(X)$	$a_0X^0 + \ldots + a_dX^d$
query	$f(w^i)$	$f_L(X) + a_iX^i + X^{i+1}f_R(X)$
shift	$f(w^iX)$	$X^if(X)$

more tricks

	Lagrange basis	monomial basis
encode	$a_0L_0(X) + \dots + a_dL_d(X)$	$a_0X^0 + \dots + a_dX^d$
query	$f(w^i)$	$f_L(X) + a_iX^i + X^{i+1}f_R(X)$
shift	$f(w^iX)$	$X^if(X)$
sum	$g(wX) = f(X) + g(X)$	$f(1)$

sum argument

$\mathbf{q = (shift(g) - g - f) \,//\, Z_S}$

commitments to **f, g, q**

sum argument

$$\mathbf{q = (shift(g) - g - f) \,/\!/\, Z_S}$$

commitments to **f, g, q**

z

sum argument

q = (shift(g) - g - f) // Z_S

commitments to **f, g, q** $\longrightarrow$

z $\longleftarrow$

openings of **f, g, shift(g), q** at **z** $\longrightarrow$

sum argument

q = (shift(g) - g - f) // Z_S

commitments to **f, g, q**

z

openings of **f, g, shift(g), q** at **z**

shift(g)(z)

check **$g(wz) - g(z) - f(z) = q(z)Z_S(z)$**

more tricks

	Lagrange basis	monomial basis
encode	$a_0L_0(X) + \ldots + a_dL_d(X)$	$a_0X^0 + \ldots + a_dX^d$
query	$f(w^i)$	$f_L(X) + a_iX^i + X^{i+1}f_R(X)$
shift	$f(w^iX)$	$X^if(X)$
sum	$g(wX) = f(X) + g(X)$	$f(1)$

sum check alternative

$$|S|^{-1}(f(X) \% Z_S(X))|_{X=0}$$

more tricks

	Lagrange basis	monomial basis
encode	$a_0L_0(X) + \dots + a_dL_d(X)$	$a_0X^0 + \dots + a_dX^d$
query	$f(w^i)$	$f_L(X) + a_iX^i + X^{i+1}f_R(X)$
shift	$f(w^iX)$	$X^if(X)$
sum	$g(wX) = f(X) + g(X)$	$f(1)$
grand product	$g(wX) = f(X)g(X)$	see Sonic appendix B

more tricks

	Lagrange basis	monomial basis
encode	$a_0L_0(X) + \dots + a_dL_d(X)$	$a_0X^0 + \dots + a_dX^d$
query	$f(w^i)$	$f_L(X) + a_iX^i + X^{i+1}f_R(X)$
shift	$f(w^iX)$	$X^if(X)$
sum	$g(wX) = f(X) + g(X)$	$f(1)$
grand product	$g(wX) = f(X)g(X)$	see Sonic appendix B
permutation	$f(X) + Y\sigma(X) + Z$ and $f(X) + YX + Z$ grand products	see Sonic appendix A

permutation argument

$\boldsymbol{\sigma}$**:** $(\mathbf{a}_i) \rightarrow (\mathbf{a}_j)$ is a permutation

permutation argument

$$\boldsymbol{\sigma}\textbf{: (}\mathbf{a_i}\textbf{)} \rightarrow \textbf{(}\mathbf{a_j}\textbf{)} \text{ is a permutation}$$

$$\Leftrightarrow$$

$$\mathbf{a_i = a_j} \text{ whenever } \mathbf{i = \boldsymbol{\sigma}(j)}$$

permutation argument

$$\boldsymbol{\sigma}\textbf{: (}\mathbf{a_i}\textbf{)} \rightarrow \textbf{(}\mathbf{a_j}\textbf{)} \text{ is a permutation}$$

$$\Leftrightarrow$$

$$\mathbf{a_i = a_j} \text{ whenever } \mathbf{i = \sigma(j)}$$

$$\Leftrightarrow$$

$$\mathbf{a_i + i{*}X = a_j + \sigma(j){*}X} \text{ whenever } \mathbf{i = \sigma(j)}$$

permutation argument

$$\boldsymbol{\sigma}\textbf{: (}\mathbf{a_i}\textbf{)} \rightarrow \textbf{(}\mathbf{a_j}\textbf{)} \text{ is a permutation}$$

$$\Leftrightarrow$$

$$\mathbf{a_i = a_j} \text{ whenever } \mathbf{i = \boldsymbol{\sigma}(j)}$$

$$\Leftrightarrow$$

$$\mathbf{a_i + i{*}X = a_j + \boldsymbol{\sigma}(j){*}X} \text{ whenever } \mathbf{i = \boldsymbol{\sigma}(j)}$$

$$\Leftrightarrow$$

$$\mathbf{\{a_i + i{*}X\} = \{a_j + \boldsymbol{\sigma}(j){*}X\}} \text{ as multisets}$$

permutation argument

$\boldsymbol{\sigma}$**:** $\mathbf{(a_i)} \rightarrow \mathbf{(a_j)}$ is a permutation

$\Leftrightarrow$

$\mathbf{a_i = a_j}$ whenever $\mathbf{i = \sigma(j)}$

$\Leftrightarrow$

$\mathbf{a_i + i{*}X = a_j + \sigma(j){*}X}$ whenever $\mathbf{i = \sigma(j)}$

$\Leftrightarrow$

$\mathbf{\{a_i + i{*}X\} = \{a_j + \sigma(j){*}X\}}$ as multisets

$\Leftrightarrow$

$\mathbf{product_i(a_i + i{*}X + Y) = product_j(a_j + \sigma(j){*}X + Y)}$ as polynomials in $\mathbf{X, Y}$

permutation argument

$\boldsymbol{\sigma}$**:** $(\mathbf{a_i}) \rightarrow (\mathbf{a_j})$ is a permutation

$\Leftrightarrow$

$\mathbf{a_i = a_j}$ whenever $\mathbf{i = \sigma(j)}$

$\Leftrightarrow$

$\mathbf{a_i + i{*}X = a_j + \sigma(j){*}X}$ whenever $\mathbf{i = \sigma(j)}$

$\Leftrightarrow$

$\mathbf{\{a_i + i{*}X\} = \{a_j + \sigma(j){*}X\}}$ as multisets

$\Leftrightarrow$

$\mathbf{product_i(a_i + i{*}X + Y) = product_j(a_j + \sigma(j){*}X + Y)}$ as polynomials in $\mathbf{X, Y}$

$\Leftrightarrow$

$\mathbf{product_i(a_i + i{*}r_1 + r_2) = product_j(a_j + \sigma(j){*}r_1 + r_2)}$ for random challenges $\mathbf{r_1, r_2}$

even more tricks

Lagrange superpower

	Lagrange basis trick
Hadamard product	$f_1(X)f_2(X)$

even more tricks

Lagrange superpower

	Lagrange basis trick
Hadamard product	$\mathbf{f_1(X)f_2(X)}$
inner product	sum over $\mathbf{f_1(X)f_2(X)}$

even more tricks

Lagrange superpower

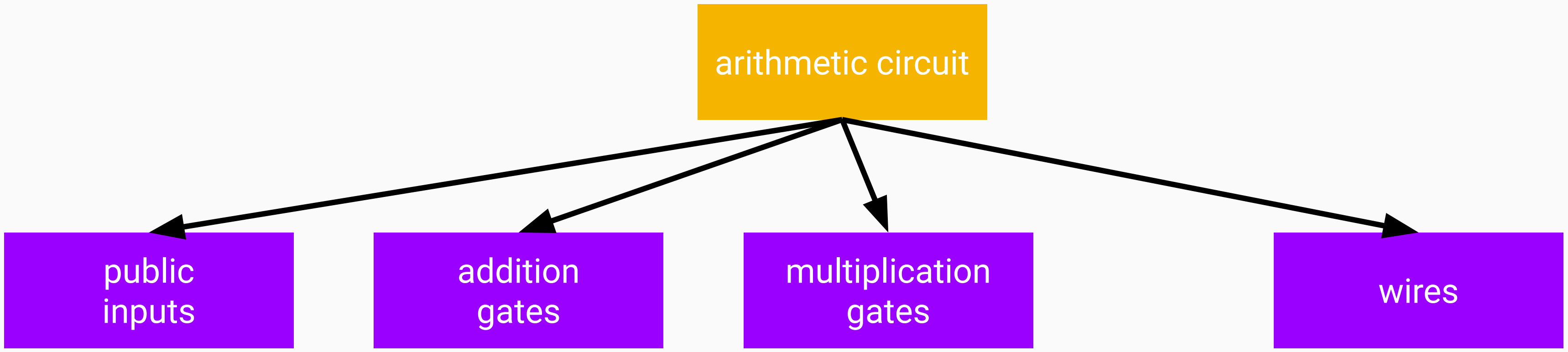
	Lagrange basis trick
Hadamard product	$\mathbf{f_1(X)f_2(X)}$
inner product	sum over $\mathbf{f_1(X)f_2(X)}$
sparse matrix multiplication	two sum cheks; see [here]()

even more tricks

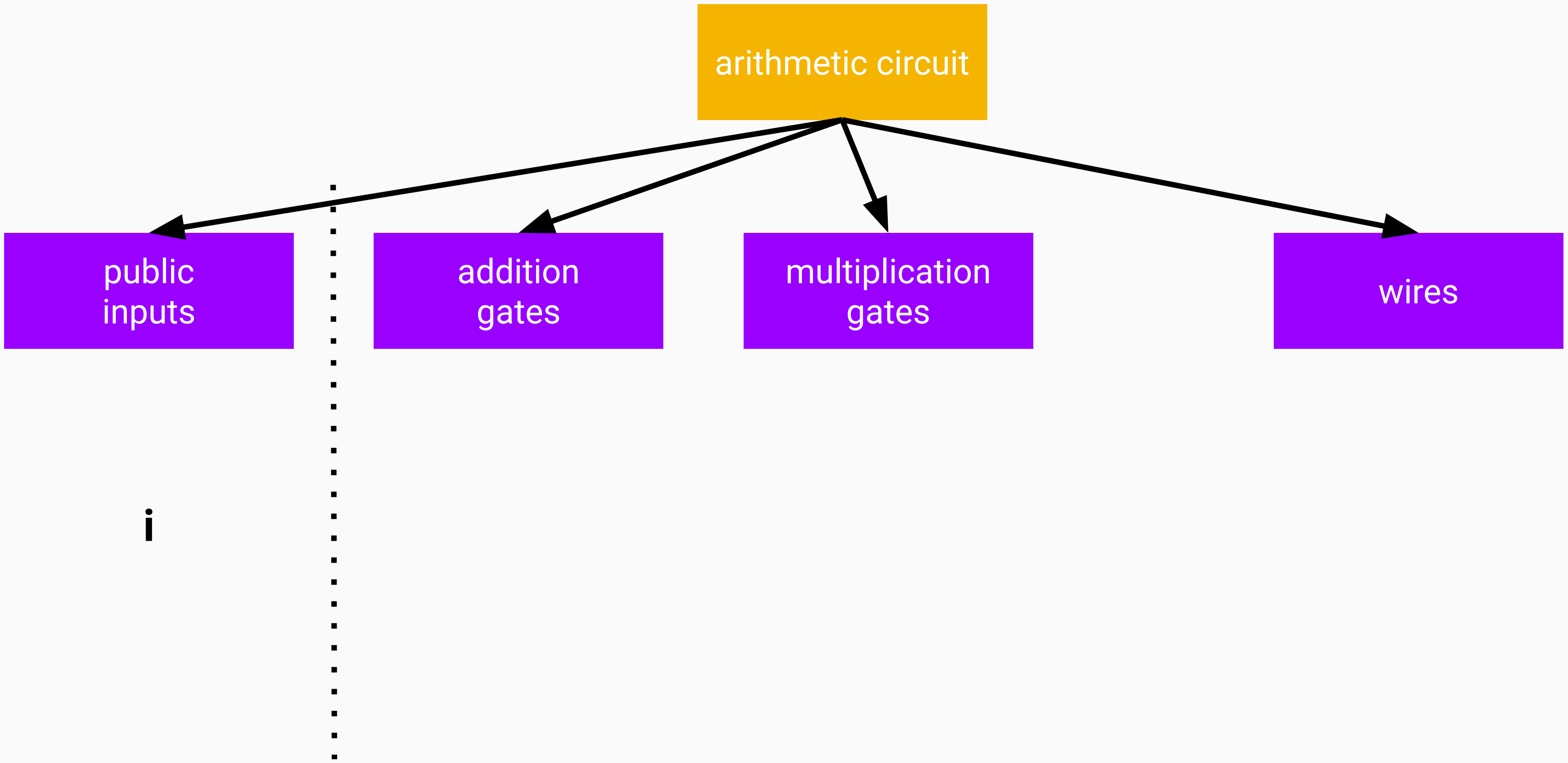
Lagrange superpower

	Lagrange basis trick
Hadamard product	$\mathbf{f_1(X)f_2(X)}$
inner product	sum over $\mathbf{f_1(X)f_2(X)}$
sparse matrix multiplication	two sum cheks; see here
range checks	see Aztec research
RAM read and write	see Aztec research

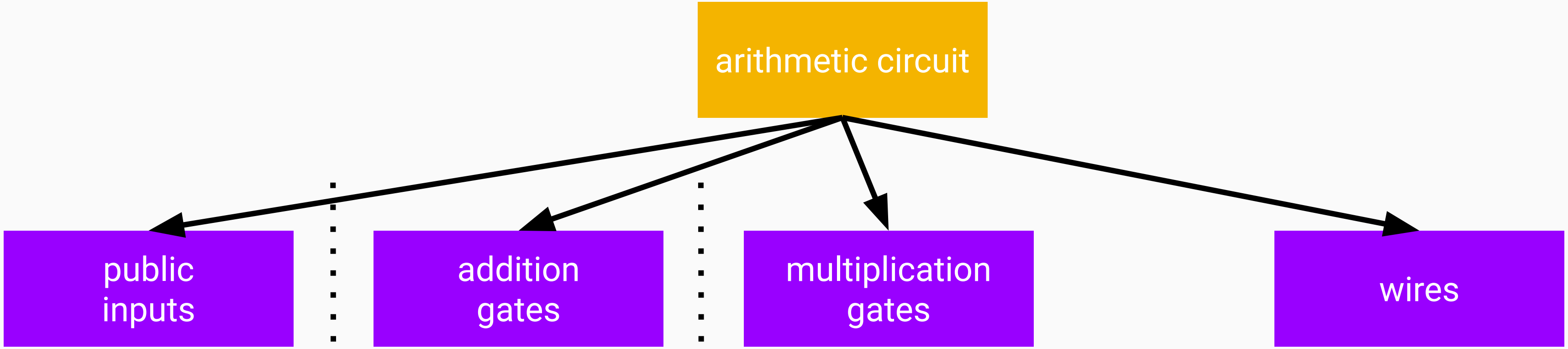
PLONK constraint system



PLONK constraint system



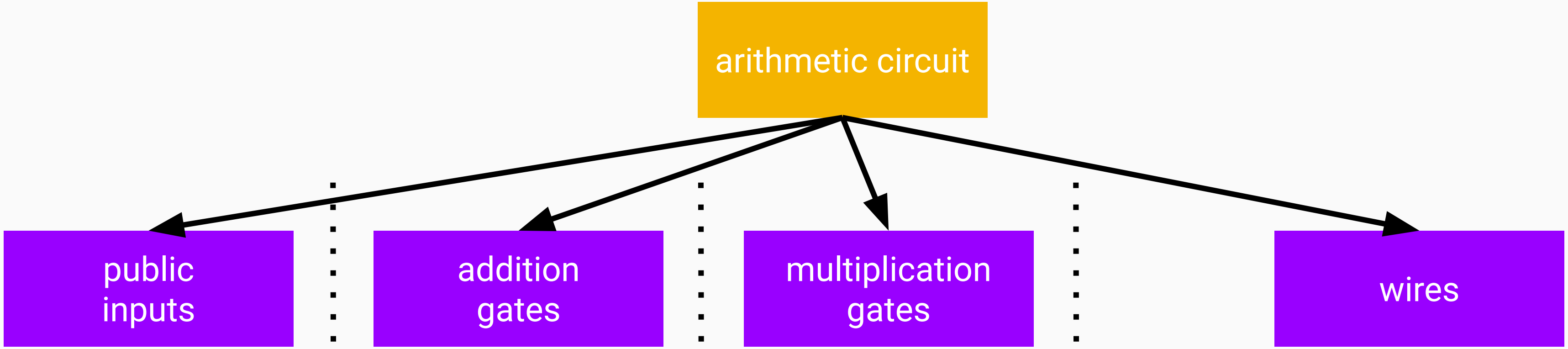
PLONK constraint system



i

$a_L + a_R = a_O$

PLONK constraint system

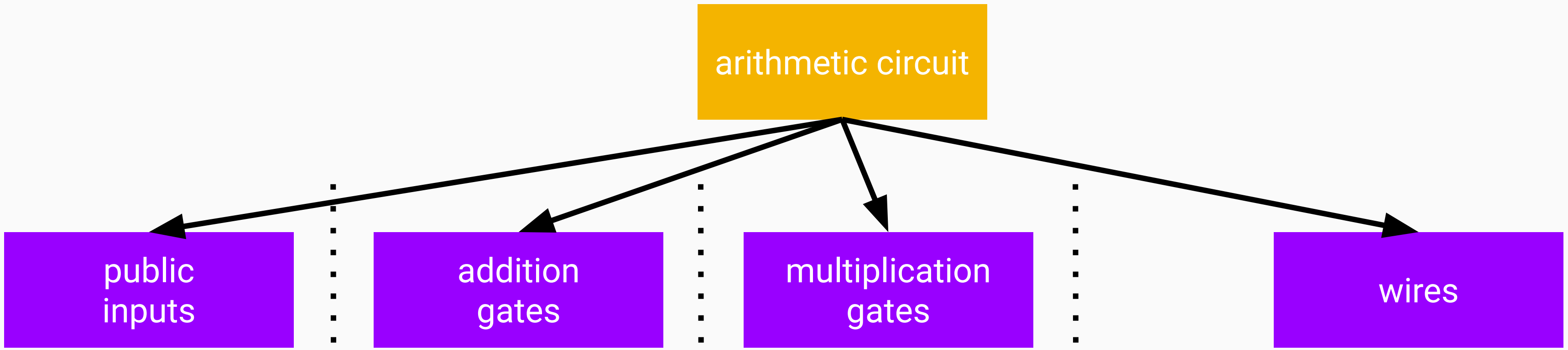


i

$a_L + a_R = a_O$

$m_L m_R = m_O$

PLONK constraint system



i

$a_L + a_R = a_O$

$m_L m_R = m_O$

w = concat(
i, # input wires
a_L, a_R, a_O, # ADD wires
m_L, m_R, m_O, # MUL wires
)

w = permute(w, σ)

thank you :)