ZKET Core Program 2025

STARK

Scalable Transparent ARgument of Knowledge

Presenter: Ninh Quốc Bảo

Orochi ZKP Labs

Đà Nẵng, 5/7/2025

Table of Content

0. Computational Integrity & STARK

Computational Integrity mean the output of a certain computation is **correct**.

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

- **f :** computation (public)
- **w :** witness (private input)
- **x :** instances (public input)

STARK aims to prove the computational integrity of a statement f(w,x)=0,

- **without revealing** the witness w (optionally),
- and allows the verifier to **verify** the proof **efficiently**.

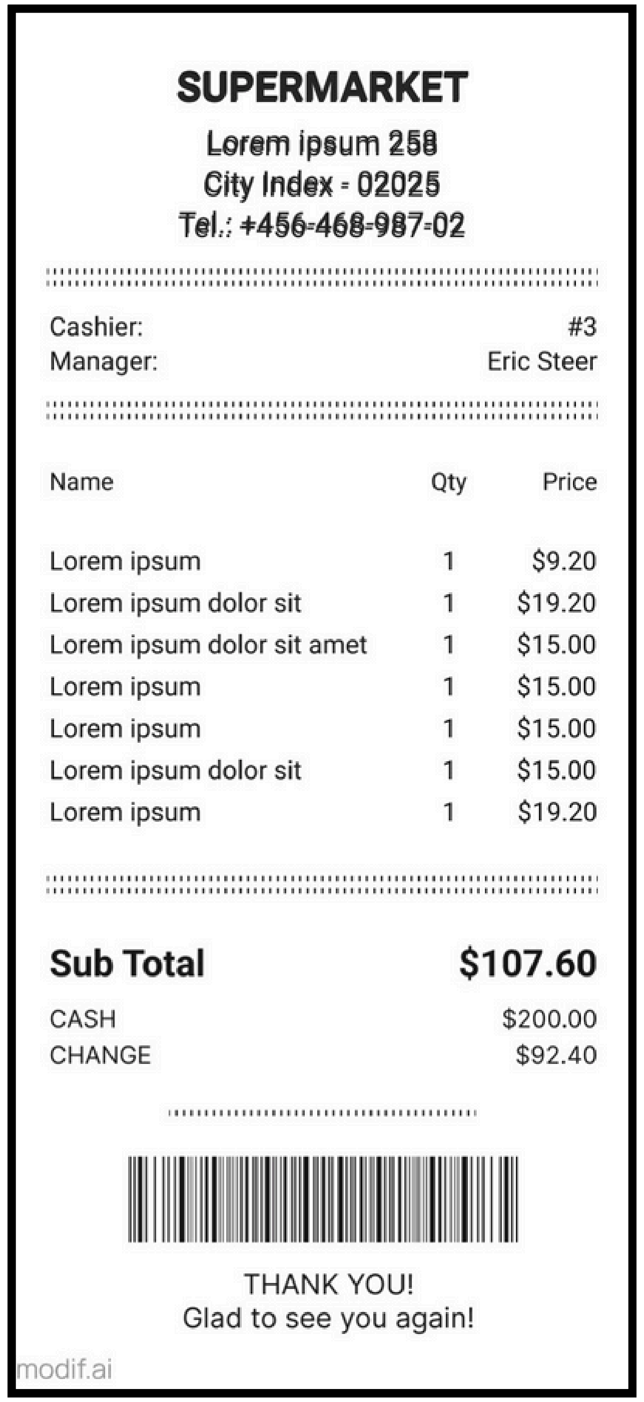
0. Computational Integrity & STARK

Example:

- Receipt, bill, ...
- Account Balance
- Fibonacci number 1M-th

```
a_0 = 1
a_1 = 1
  ...
a_1M = 19532821287077577316 ...
    ... 68996526838242546875
```

(208 988 digits)



SUPERMARKET
Lorem ipsum 258
City Index - 02025
Tel.: +456-468-987-02

Cashier: #3
Manager: Eric Steer

Name	Qty	Price
Lorem ipsum	1	$9.20
Lorem ipsum dolor sit	1	$19.20
Lorem ipsum dolor sit amet	1	$15.00
Lorem ipsum	1	$15.00
Lorem ipsum	1	$15.00
Lorem ipsum dolor sit	1	$15.00
Lorem ipsum	1	$19.20

Sub Total **$107.60**
CASH $200.00
CHANGE $92.40

THANK YOU!
Glad to see you again!

modif.ai

How to prove?

-> STARK

1. Introducing STARKs

- **Scalable :** Prover time is *quasi-linear* in problem size
  Verifier time is *poly-logarithm* in problem size
- **Transparent :** Do *not* have trusted-setup
- **ARgument :** A proof system where we consider it secure only for provers that have *bounded computational resources*
- **of Knowledge :** It is not possible for the prover to construct a valid proof *without knowing witness* for the statement

Is STARK a SNARK? Yes, STARK is SNARK without trusted-setup

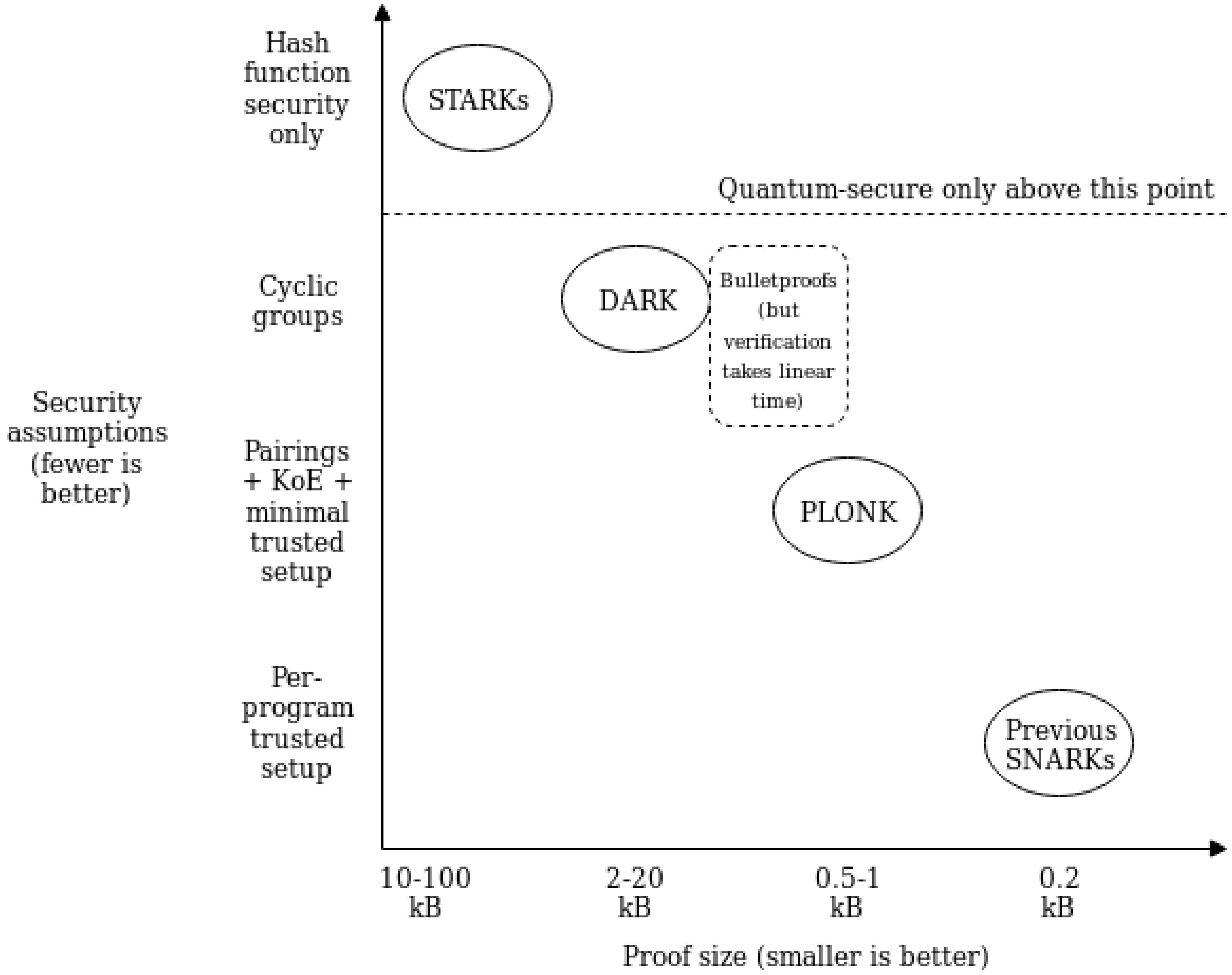
1.1 Key Features of STARKs

- **Scalable :** Offer low prover time (compare to traditional SNARKs), while maintaining affordable verifier time
- **Transparent :** Do *not* have trusted-setup → lower trust assumption
- **Post-Quantum Secure :** Secure against attacks from both classical and quantum computers
- **Security Assumption :** Based on Hash function security only

1.2 Drawback

- **Big proof size :** STARK proof size is much bigger compare to other SNARKs proof

1.3 Compare Security Assumption & Proof Size



Source: Understanding PLONK - Vitalik

2. Preliminaries

- Finite Field
- Multiplicative subgroup & Coset
- Polynomial Interpolation
- Merkle Tree
- Quotienting

2.0 Finite Field

Finite Field is **field** have **finite** elements

- Integer modulo prime p

$$\mathbb{F}_p = \{0, 1, 2, \ldots, p-1\}$$

$$E.g. \quad \mathbb{F}_{17} = \{0, 1, 2, \ldots, 16\}$$

2.1 Multiplicative subgroup & Coset

- **Multiplicative subgroup :** $G = \{1, 4, 16, 13\} \quad \text{with} \quad g = 4$

  Set of elements of the form:

$$G = \{g^0, g^1, \ldots, g^{n-1}\}$$

  and multiply $(\cdot)$ operation.

- **Coset :** $D = \{5, 3, 12, 14\} \quad \text{with} \quad g = 4, w = 5$

  Set of elements of the form:

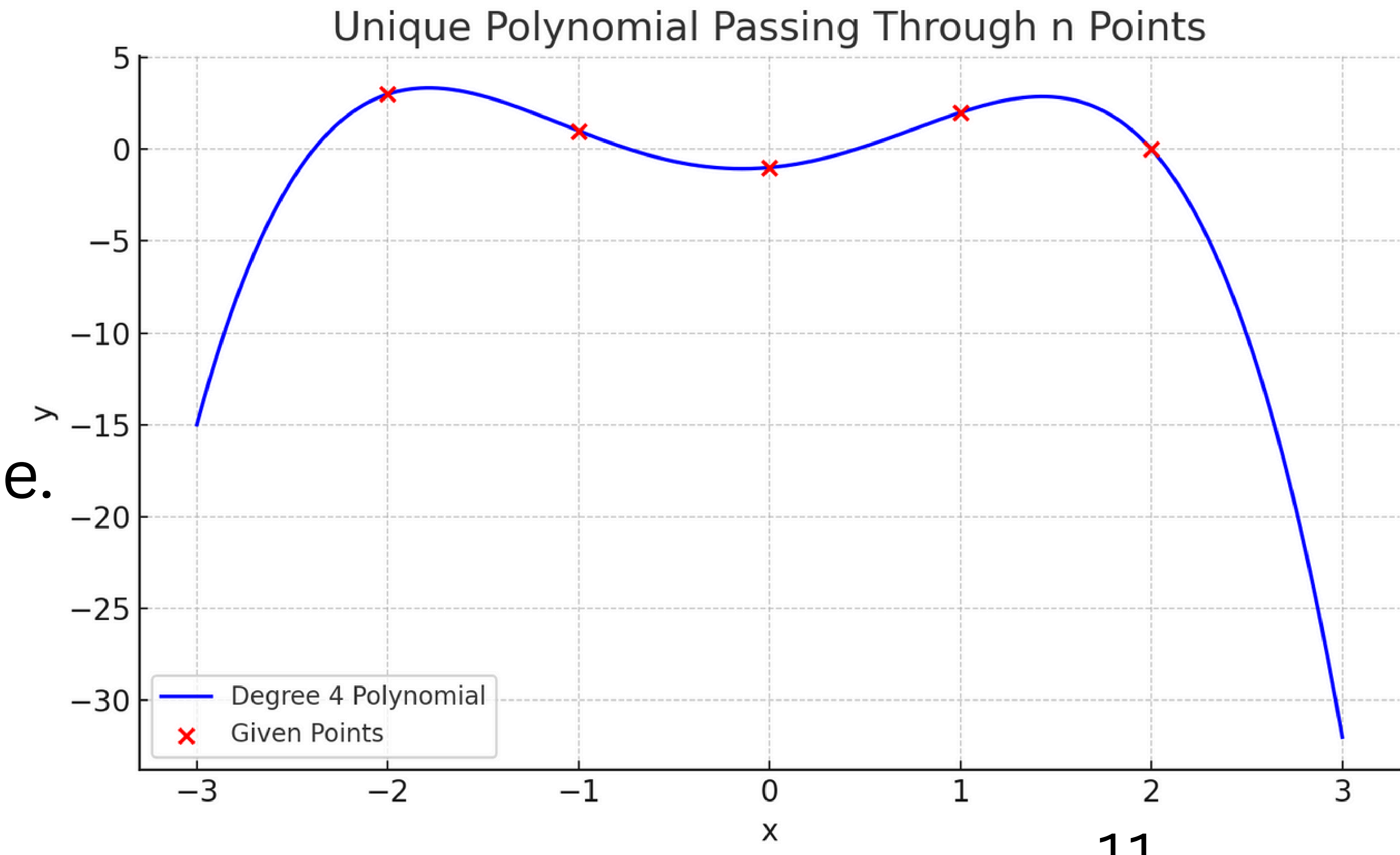
$$D = \{w \cdot g^0, w \cdot g^1, \ldots, w \cdot g^{n-1}\}$$

2.2 Polynomial Interpolation

Given a set of n values: $y_0, y_1, \ldots, y_{n-1}$

Choose set of n x-coordinates: $x_0, x_1, \ldots, x_{n-1}$

Question: Find polynomial degree n-1 pass through all points (xi,yi)



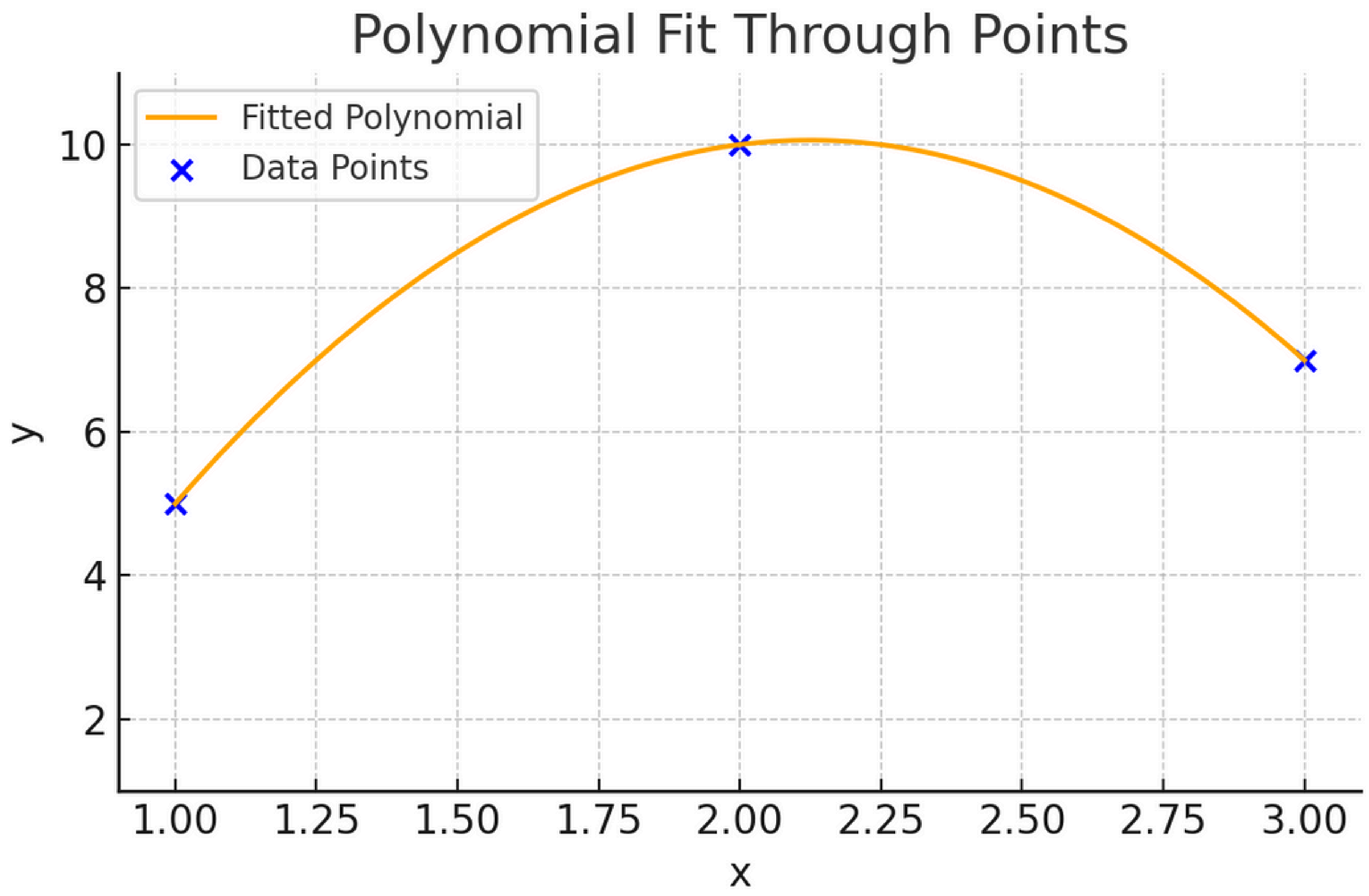
Theorem: Exists a unique polynomial degree n-1 pass through n points in plane.

Lagrange Interpolation

Fast Fourier Transform

2.2 Polynomial Interpolation

x	y
1	5
2	10
3	7



2.3 Merkle Tree

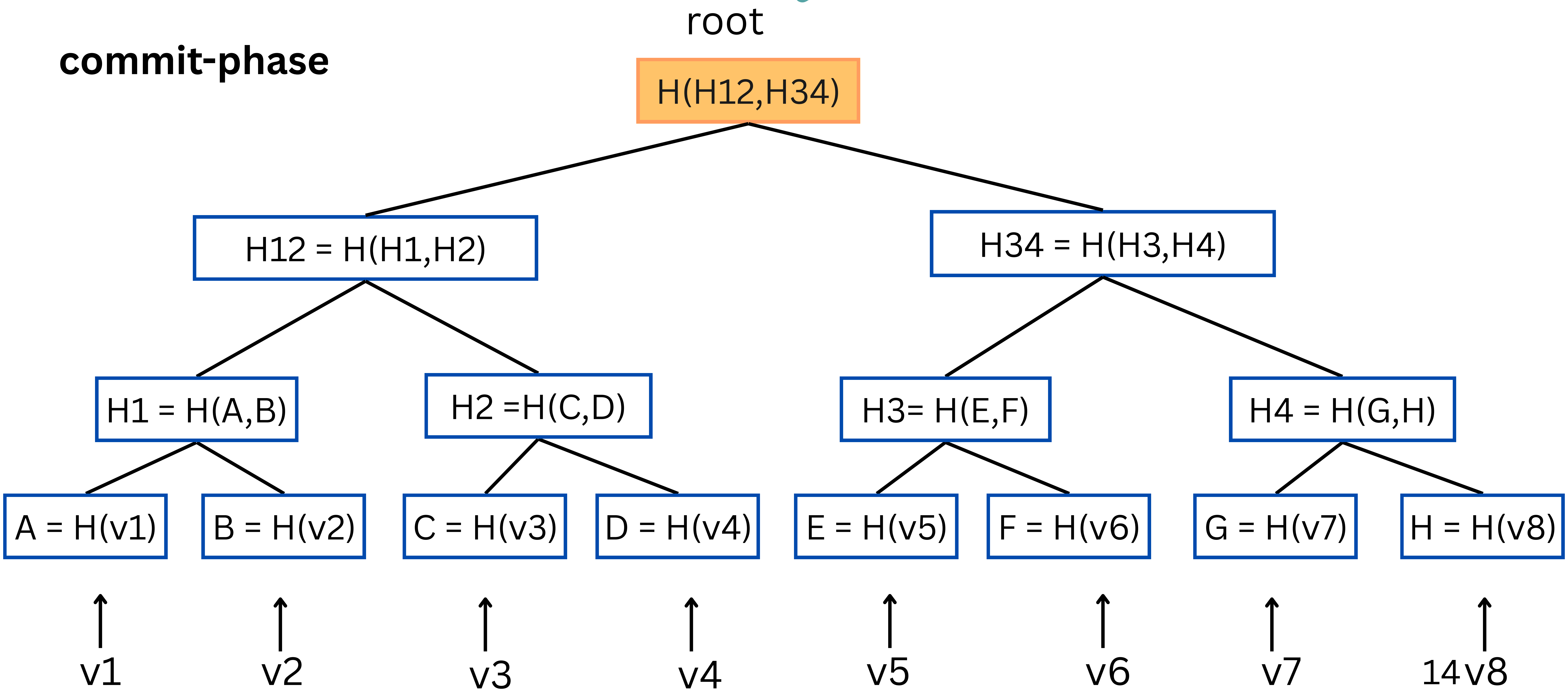
Merkle Tree is **binary tree** with :

- Each leaf node contains the hash of a data block.
- Each internal (parent) node contains the hash of the concatenation of its two child nodes' hashes.
- The root hash (called the Merkle root) summarizes the entire data set and can be used to verify integrity efficiently.

Merkle Root is commitment of all data v1->v8

2.3 Merkle Tree

commit-phase



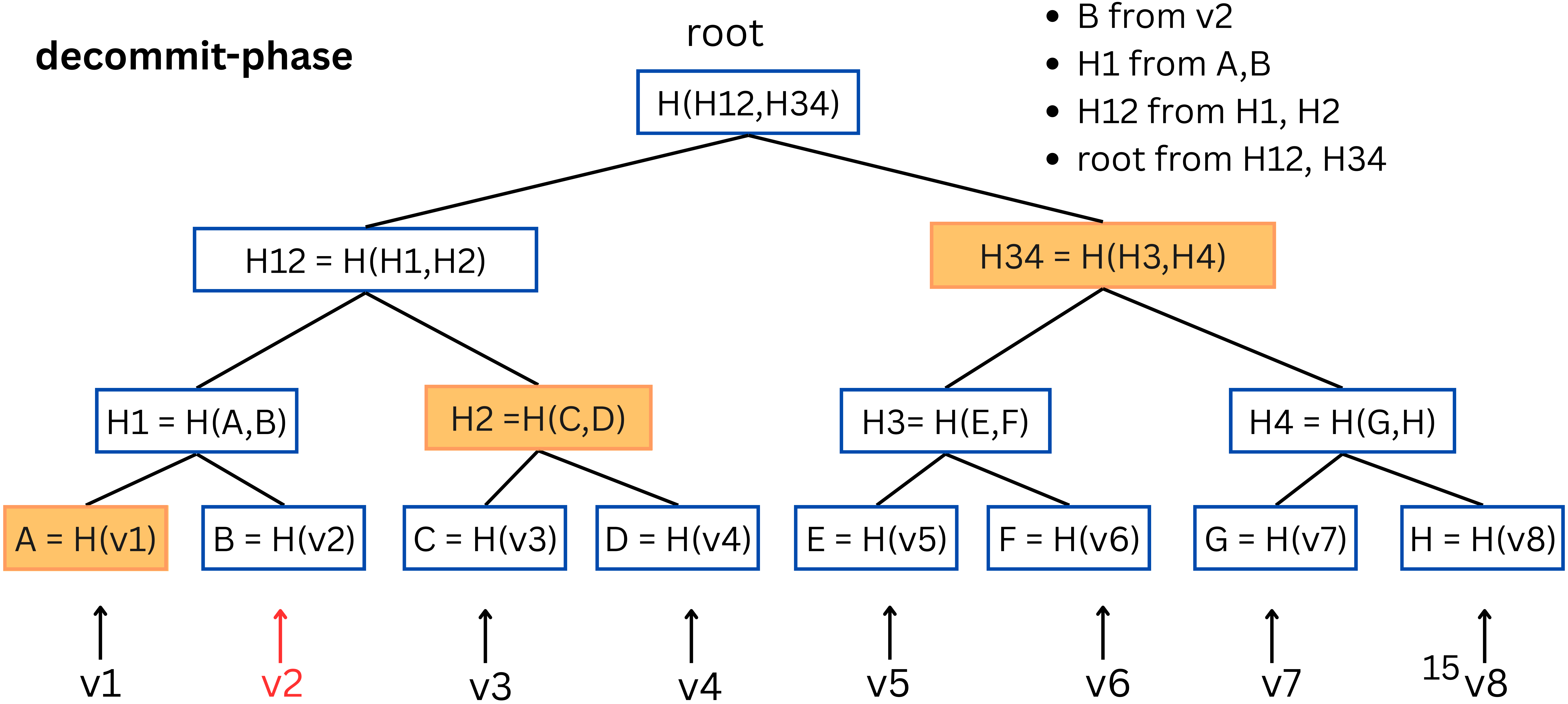
2.3 Merkle Tree

decommit-phase

A, H2, H34 is decommitment of v2

we can verify by recompute:

- B from v2
- H1 from A,B
- H12 from H1, H2
- root from H12, H34



2.4 Quotienting

Let p(x) is polynomial,

- To prove p(x) = y at x = z
- We prove : Exists polynomial q(x) that q(x) = (p(x) - y) / (x - z)

$$p(x) = y \quad \text{at } x = z$$

$$\Leftrightarrow \exists \text{ polynomial } q(x) : q(x) = \frac{p(x) - y}{x - z}$$

2.4 Quotienting

Let p(x) is polynomial,

- To prove p(x+1) = p(x) + 1 for all x from 1 to 100
- We prove : Exists polynomial q(x) that q(x) = (p(x+1) - p(x) -1) / V(x)

$$p(x+1) = p(x) + 1 \quad \forall x \in \{1, \ldots, 100\}$$

$$\Leftrightarrow \exists \text{ polynomial } q(x) : q(x) = \frac{p(x+1) - p(x) - 1}{V(x)}$$

$$\text{where :} V(x) = (x-1)(x-2)\ldots(x-100)$$

$$\text{is vanishing polynomial for } x \in \{1, \ldots, 100\}$$

2.4 Quotienting

Let p(x) is polynomial, with x in **Multiplicative Subgroup:**

$$G = \{g^0, g^1, \ldots, g^{n-1}\}$$

p(x+1) → p(g*x)

- To prove p(gx) = p(x) + 1 for all x from g^0 to g^100
- We prove : Exists polynomial q(x) that q(x) = (p(gx) - p(x) -1) / V(x)

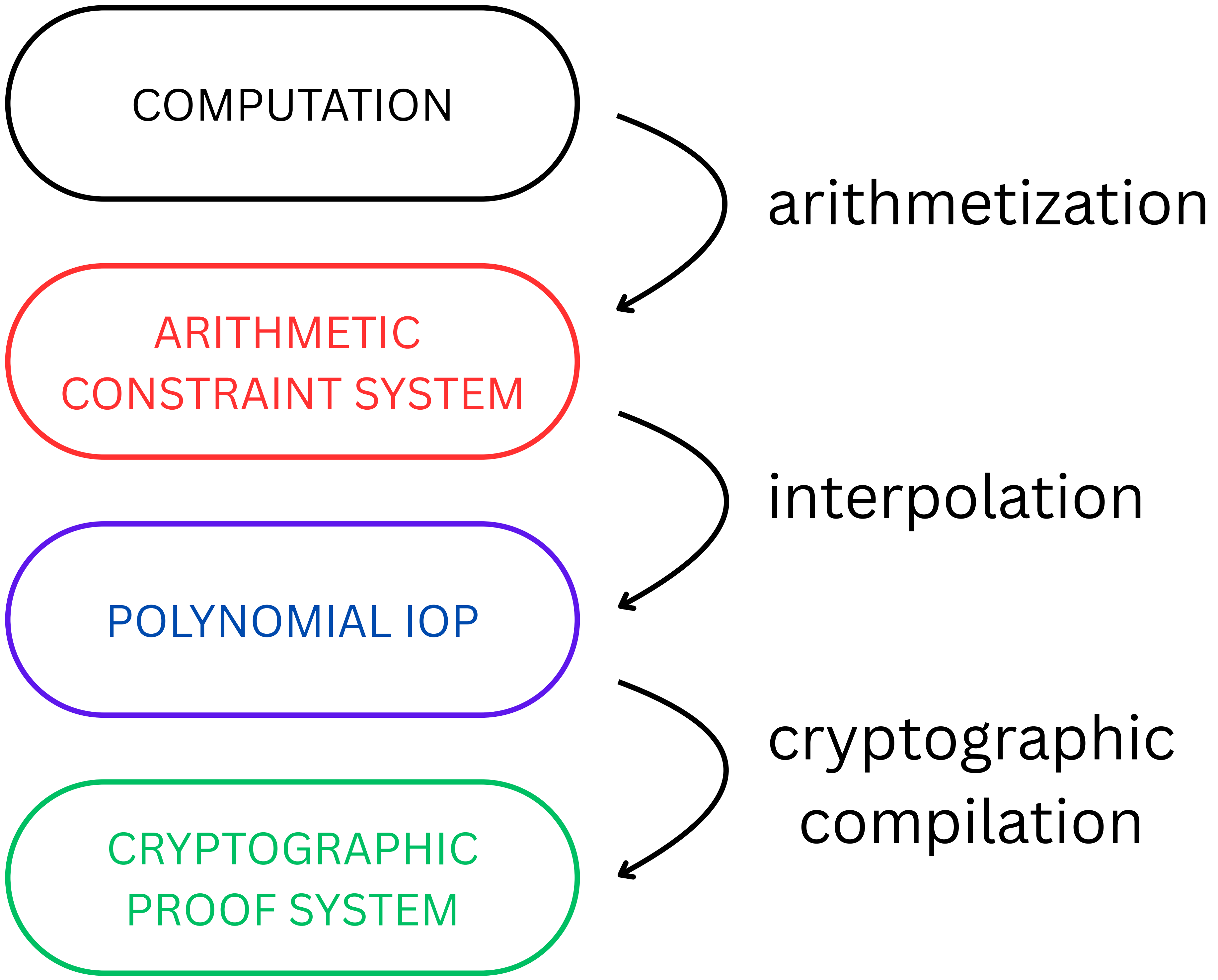
$$p(g \cdot x) = p(x) + 1 \quad \forall x \in \{g^0, \ldots, g^{100}\}$$

$$\Leftrightarrow \exists \text{ polynomial } q(x) : q(x) = \frac{p(g \cdot x) - p(x) - 1}{V(x)}$$

$$\text{where } : V(x) = (x - g^0)(x - g^1) \ldots (x - g^{100})$$

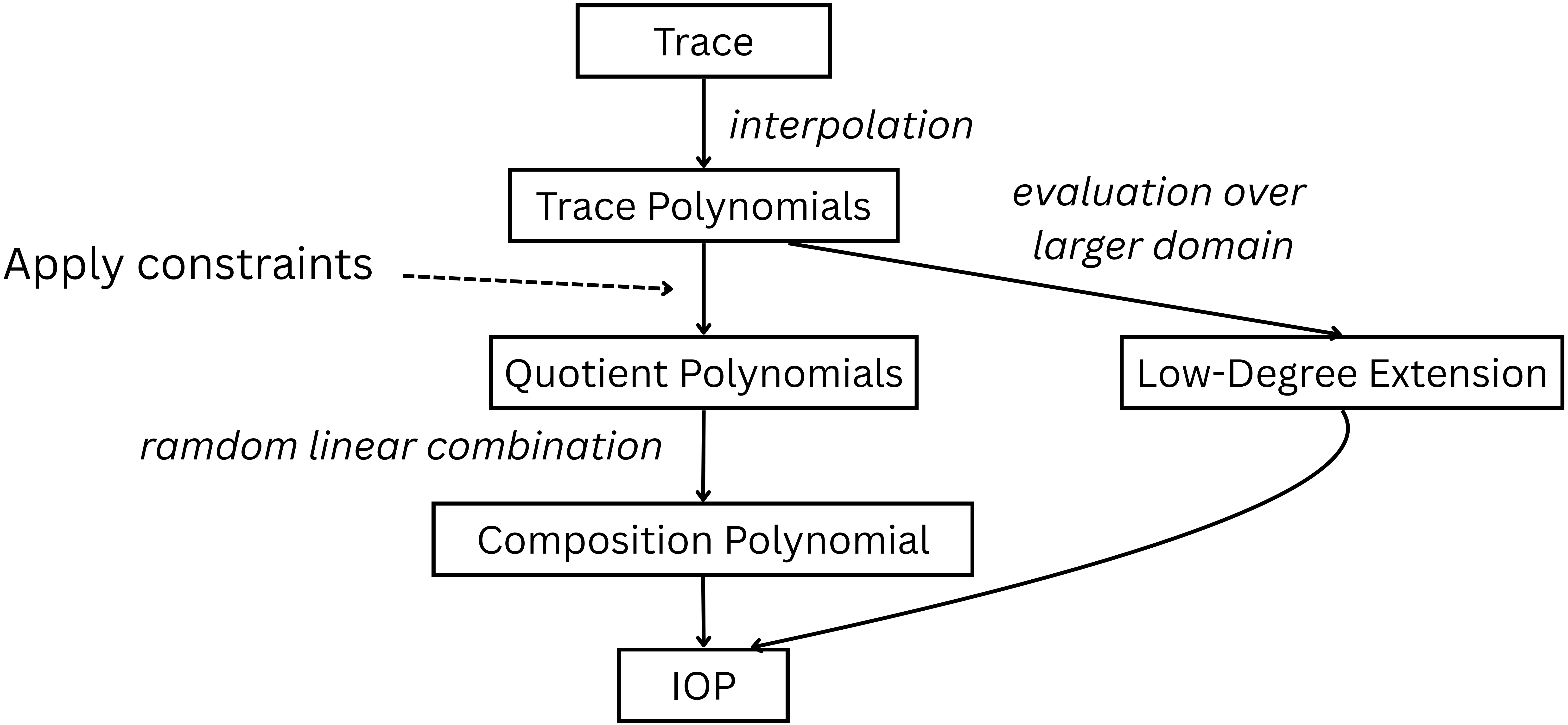
$$\text{is vanishing polynomial for } x \in \{g^0, \ldots, g^{100}\}$$

3. STARKs Flow



Source: Anatomy of STARK - aszepieniec

3. STARKs Flow



4. STARKs Components

- Trace
- Trace Polynomial
- Low Degree Extension (LDE)
- Constraints
- Quotient Polynomials
- Composition Polynomials
- FRI

4.1 Wide Fibonacci Example

Wide Fibonacci Sequence:

- $a_0 = 1$
- $a_1 = 1$
- $a_{n+2} = a_{n-1}^2 + a_n^2 \pmod{p} \quad , \forall n \geq 0$

With $p = 2^{31} - 2^{27} + 1 = 2.013.265.921$ (Baby Bear prime)

Prove: The Wide Fibonacci number at index 1022 is $a_{1022} = 1.969.673.408$.

4.1 Wide Fibonacci Example

Write this problem in Computational Integrity statement style:

$$f(\mathbb{w}, \mathbb{x}) = \mathrm{WideFibo}_p(index = 1022, a_0 = 1, a_1 = 1) - 1.969.673.408 = 0$$

- instances : $\mathbb{x} = \{a_0 = 1, a_1 = 1, a_{1022} = 1.969.673.408, p = 2.013.265.921\}$
- witness : $\mathbb{w} = \{a_2, a_3, \ldots, a_{1021}\}$

Prove: we "know" the correct witness $\mathbb{w}$ satifies function f with instances $\mathbb{x}$, such that $f(\mathbb{w}, \mathbb{x}) = 0$.

4.2 Trace

Sequence of value that represent the correct computation

In the Wide Fibonacci Example, the trace is:

$$\mathrm{Trace} = \{a_0, a_1, a_2, \ldots, a_{1022}\}$$

4.3 Trace Polynomials

The polynomial form of trace

1. Choose **Multiplicative Subgroup** size n

$$\begin{cases} n \text{ is power of } 2 \\ n >= \text{size of Trace} \end{cases}$$

$$n = 1024 = 2^{10} \geq |Trace| = 1023$$

$$H = \{h^0, h^1, \ldots, h^{1023}\}, \quad |H| = 1024$$

2. Create n points (xi,yi) with x-coordinate is element in Multiplicative Subgroup and y-coordinate is element of Trace (padding if necessary)

$$\{ (h^0, a_0), (h^1, a_1), \ldots, (h^{1022}, a_{1022}), (h^{1023}, 0) \}$$

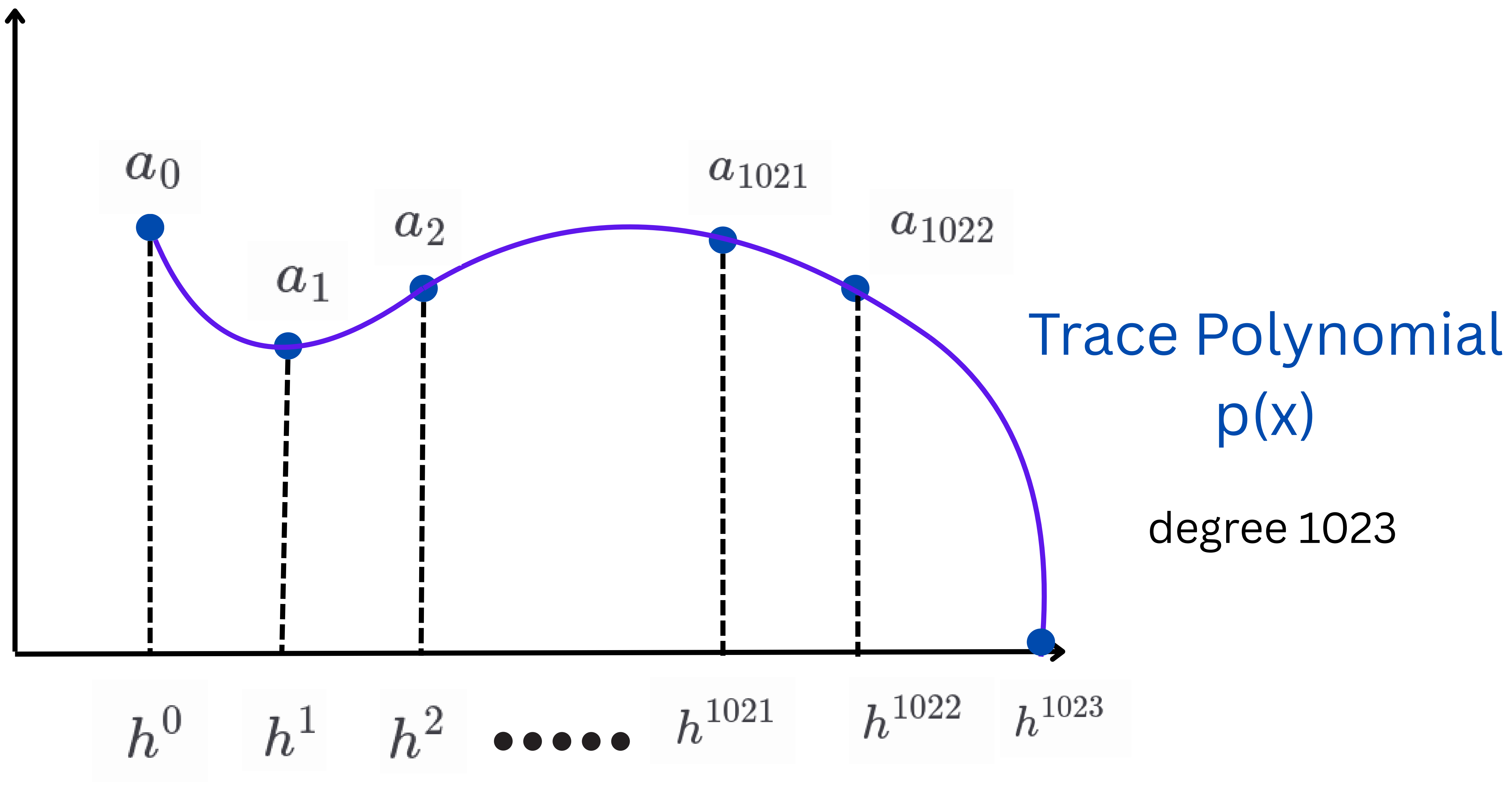
padding

FFT

3. Using Polynomial Interpolation to find the **Trace Polynomial degree n-1** pass through all these points

4.3 Trace Polynomials

x	p(x)
h^0	a_0
h^1	a_1
$\vdots$	$\vdots$
h^{1021}	a_{1021}
h^{1022}	a_{1022}
h^{1023}	0



4.4 FFT

Evaluation View → **FFT** → **Coefficient View**

x	p(x)
h^0	a_0
h^1	a_1
$\vdots$	$\vdots$
h^{1021}	a_{1021}
h^{1022}	a_{1022}
h^{1023}	0

$$p(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \cdots + c_{1023} \cdot x^{1023}$$

$$p = [c_0, c_1, c_2, \ldots, c_{1023}]$$

Complexity: O(NlogN)

4.4 FFT

We have:

$$p(x) = (c_0 + c_2x^2 + \ldots + c_{1022}x^{1022}) + x(c_1 + c_3x^2 + \ldots + c_{1023}x^{1022})$$

$$\Leftrightarrow p(x) = p_0(x^2) + x \cdot p_1(x^2) \quad (*)$$

where $p_0 = [c_0, c_2, \ldots, c_{1022}]$, $p_1 = [c_1, c_3, \ldots, c_{1023}]$ are polynomials with degree 511

From (*) =>

$$p_0(x^2) = \frac{p(x) + p(-x)}{2}$$

$$p_1(x^2) = \frac{p(x) - p(-x)}{2x}$$

4.4 FFT

In Multiplicative Subgroup $H = \{h^0, h^1, h^2, \ldots, h^{n-1}\}$, with $|H| = n$

We have: $h^n = h^0 = 1$

For some $0 \leq i < \frac{n}{2}$, let $x = h^i$ and $y = h^{\frac{n}{2}+i}$, we have:

$$x^2 = h^{2i} \quad \text{and} \quad y^2 = h^{n+2i} = h^{2i}$$

We see: $x^2 = y^2$, because $x \neq y \Rightarrow x = -y$

This mean: $\forall x \in H, \exists(-x) \in H$

Consider 2-to-1 map:

$$\begin{aligned} \pi: \quad H \quad &\to H_1 \\ (x, -x) &\mapsto x^2 \end{aligned}$$

We have new Multiplicative Subgroup $H_1 = \{h^0, h^2, h^4, \ldots, h^{n-2}\}$, with $|H_1| = \frac{n}{2}$

4.4 FFT

Problem 0:

Given Evaluation View

size 1024

x	p(x)
h^0	a_0
h^1	a_1
$\vdots$	$\vdots$
h^{1021}	a_{1021}
h^{1022}	a_{1022}
h^{1023}	0

Find Coefficient View

$$p(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \cdots + c_{1023} \cdot x^{1023}$$

$$p = [c_0, c_1, c_2, \ldots, c_{1023}] \longrightarrow \text{size 1024}$$

We can write: $p(x) = p_0(x^2) + x \cdot p_1(x^2)$

Implies:

$$\boxed{\begin{aligned} p_0(x^2) &= \frac{p(x) + p(-x)}{2} \\ p_1(x^2) &= \frac{p(x) - p(-x)}{2x} \end{aligned}}$$

4.4 FFT

Problem 1:

Given Evaluation View

$y = x^2$	p_0(y)
h^0	$\frac{a_0 + a_{512}}{2}$
h^2	$\frac{a_1 + a_{513}}{2}$
$\vdots$	$\vdots$
h^{1022}	$\frac{a_{511} + a_{1023}}{2}$

size 512

Find Coefficient View

$$p_0(y) = c_0 + c_2 \cdot y + c_4 \cdot y^2 + \cdots + c_{1022} \cdot y^{511}$$

$$p_0 = [c_0, c_2, c_4, \ldots, c_{1022}] \longrightarrow \text{size } 512$$

We can write: $p_0(y) = p_{00}(y^2) + y \cdot p_{01}(y^2)$

Implies:

$$\boxed{\begin{aligned} p_{00}(y^2) &= \frac{p_0(y) + p_0(-y)}{2} \\ p_{01}(y^2) &= \frac{p_0(y) - p_0(-y)}{2y} \end{aligned}}$$

4.4 FFT

Problem 10:

Given Evaluation View

size 1

z	p_0..0(z)
u	v

Find Coefficient View

$$p_{0..0}(z) = c_0$$

$$p_{0..0} = [c_0] \longrightarrow \text{size 1}$$

=> We find c_0 = v after 10 rounds

(similar for another coefficient)

4.5 Low Degree Extension (LDE)

Evaluation of Trace Polynomial(s) over larger domain

1. Choose **Multiplicative Subgroup** size 2^Bxn (with B is blowup factor)

$$W = \{\omega^0, \omega^1, \ldots, \omega^{8191}\}, \quad |W| = 8192$$

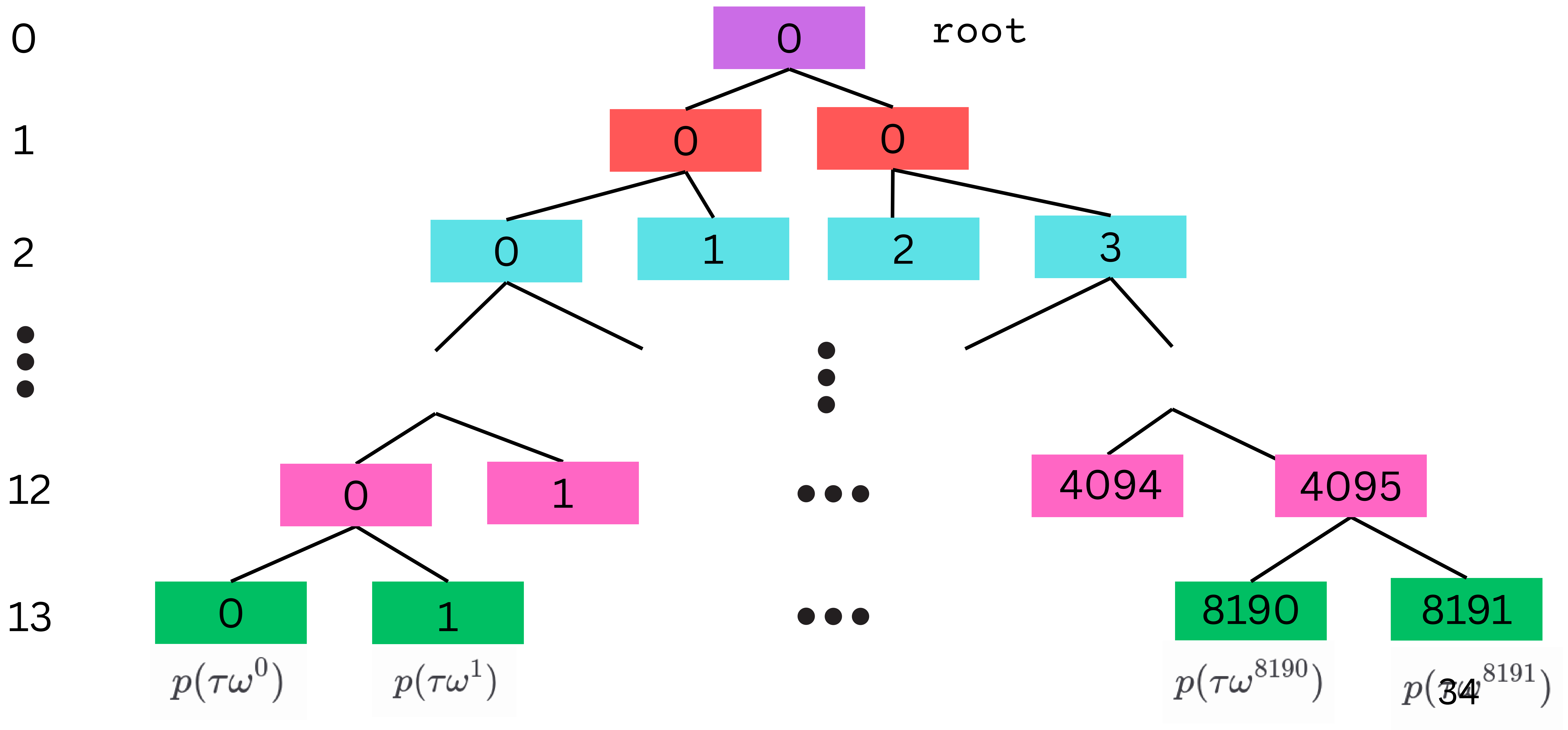
2. Create **Coset** of this Multiplicative Subgroup

$$\text{Choose } \tau \in \mathbb{F}_p : \quad D = \{\tau\omega^0, \tau\omega^1, \ldots, \tau\omega^{8191}\}, \quad |D| = 8192$$

3. Evaluate Trace Polynomial(s) over **Coset**

$$LDE = \{p(\tau\omega^0), p(\tau\omega^1), \ldots, p(\tau\omega^{8191})\}$$

4.6 Merkle Commitment on LDE



4.7 Constraint

Initial Statement		Constraint on $p(x)$
$a_0 = 1$	$\longrightarrow$	$p(x) = 1, \text{with } x = h^0$
$a_{1022} = 1.969.673.408$	$\longrightarrow$	$p(x) = 1.969.673.408, \text{with } x = h^{1022}$
$a_{n+2} = a_{n+1}^2 + a_n^2$	$\longrightarrow$	$p(h^2 x) = p(hx)^2 + p(x)^2, \text{with } x \in \{h^0, \ldots, h^{1020}\}$

boundary constraint

transition constraint

4.8 Quotient Polynomials

Change constraint to polynomial form:

$$(1) \implies q_0(x) = \frac{p(x) - 1}{x - h^0} \quad \text{is a polynomial}$$

$$(2) \implies q_1(x) = \frac{p(x) - 1.969.673.408}{x - h^{1022}} \quad \text{is a polynomial}$$

$$(3) \implies q_3(x) = \frac{p(h^2x) - p(hx)^2 - p(x)^2}{(x - h^0)\dots(x - h^{1020})}$$

$$= \frac{\Big(p(h^2x) - p(hx)^2 - p(x)^2\Big)(x - h^{1021})(x - h^{1022})(x - h^{1023})}{x^{1024} - 1} \quad \text{is a polynomial}$$

4.9 Composition Polynomial

Receive random number from verifier

$$Random = \{\alpha_0, \alpha_1, \alpha_2\}$$

Take random linear combination of quotient polynomials

$$CP(x) = \alpha_0 \cdot q_0(x) + \alpha_1 \cdot q_1(x) + \alpha_2 \cdot q_2(x)$$

Composition Polynomial

$$\boxed{CP(x) \text{ is polynomial} \implies q_0(x), q_1(x), q_2(x) \text{ is polynomial (with high probability)}}$$

4.10 Short Summary

From the **Trace** represent the computation

↓ FFT

Interpolate into **Trace Polynomials**

↓ *apply constraint*

Apply constraint on Trace Polynomials get **Quotient Polynomials**

↓ *random linear combination*

Composition Polynomial

Interpolate into **Trace Polynomials** → iFFT →

Evaluate on larger domain → **Low Degree Extension**

↓ build merkle tree

Merkle Commitment of LDE

At this point, if CP is polynomial, then the statement is true with high probability

4.11 How to prove CP is polynomial?

Goal: Prove CP is a polynomial

Do: Prove CP is **close** to a **low-degree polynomial**

Low-degree polynomial: polynomial p with degree < n (n defined by problem)

Close: a function f **agree** with polynomial p in almost points over domain D

i.e., f(x) = p(x) in 90% points of D

FRI *Fast Reed-Solomon Interactive Oracle Proofs of Proximity*

A tool to prove "a function f is close to a polynomial of low degree"

4.12 FRI

FRI protocol

```
Loop until trivial case:
    • receive random β
    • apply FRI operator
    • commit
Prover send result of last round and
commitment of all rounds to Verifier
```

intitution

Prove function f is close to a polynomial of degree < n

$\downarrow$ `apply FRI operator`

Prove new function f’ is close to a new polynomial of degree < n’

Smaller Problem

4.12 FRI

- get random β

Assume cp(x) is a polynomial

- split cp(x) to even and odd power (like FFT)

$$cp(x) = cp_{even}(x^2) + x \cdot cp_{odd}(x^2)$$

- create new polynomial

$$cp_1(y) = cp_{even}(y) + \beta \cdot cp_{odd}(y)$$

$$cp(x) = 3x^5 + 4x^4 + 2x^3 + 5x^2 + 6x + 7$$

$$cp_{even}(x^2) = 4x^4 + 5x^2 + 7$$

$$x \cdot cp_{odd}(x^2) = 3x^5 + 2x^3 + 6x$$

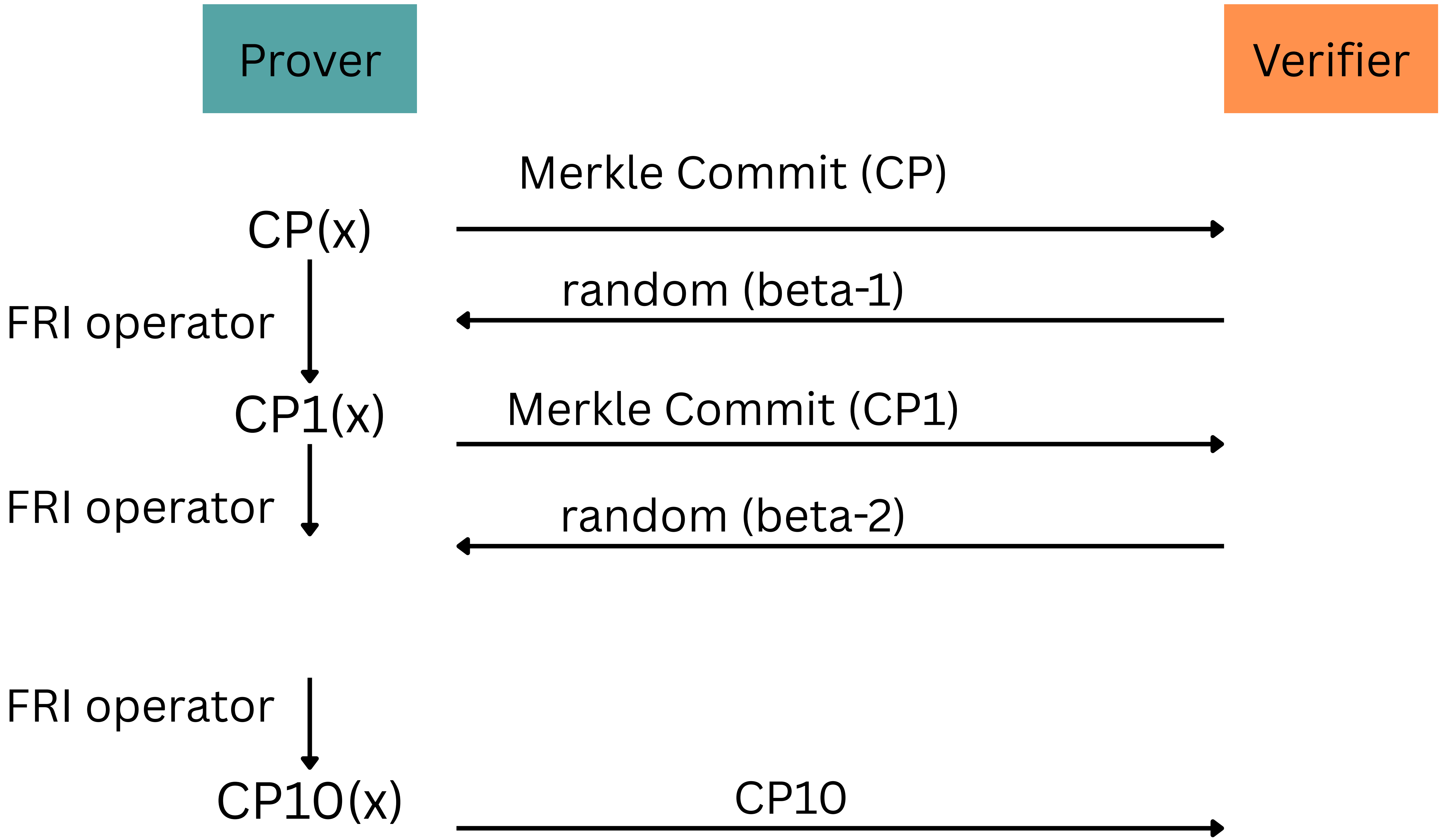
$$cp_{even}(y) = 4y^2 + 5y + 7$$

$$cp_{odd}(y) = 3y^2 + 2y + 6$$

$$cp_1(y) = (4 + 3\beta)y^2 + (5 + 2\beta)y + (7 + 6\beta)$$

4.12 FRI

Commit Phase



4.12 FRI

Query Phase

Prover

Verifier

proof = { cp(q) + path
cp(-q) + path
cp1(q^2) + path
cp1($-q^2$) + path
... }

query at q

proof

Verify:

$$isValid(CP_i(q), path, commit(CP_i))$$

$$CP_{i+1}(q^{2(i+1)}) = \frac{CP_i(q^{2i}) + CP_i(-q^{2i})}{2} + \beta_i \cdot \frac{CP_i(q^{2i}) - CP_i(-q^{2i})}{2q^{2i}}$$

query at q’

proof’

●●● query at q’’’

proof’’

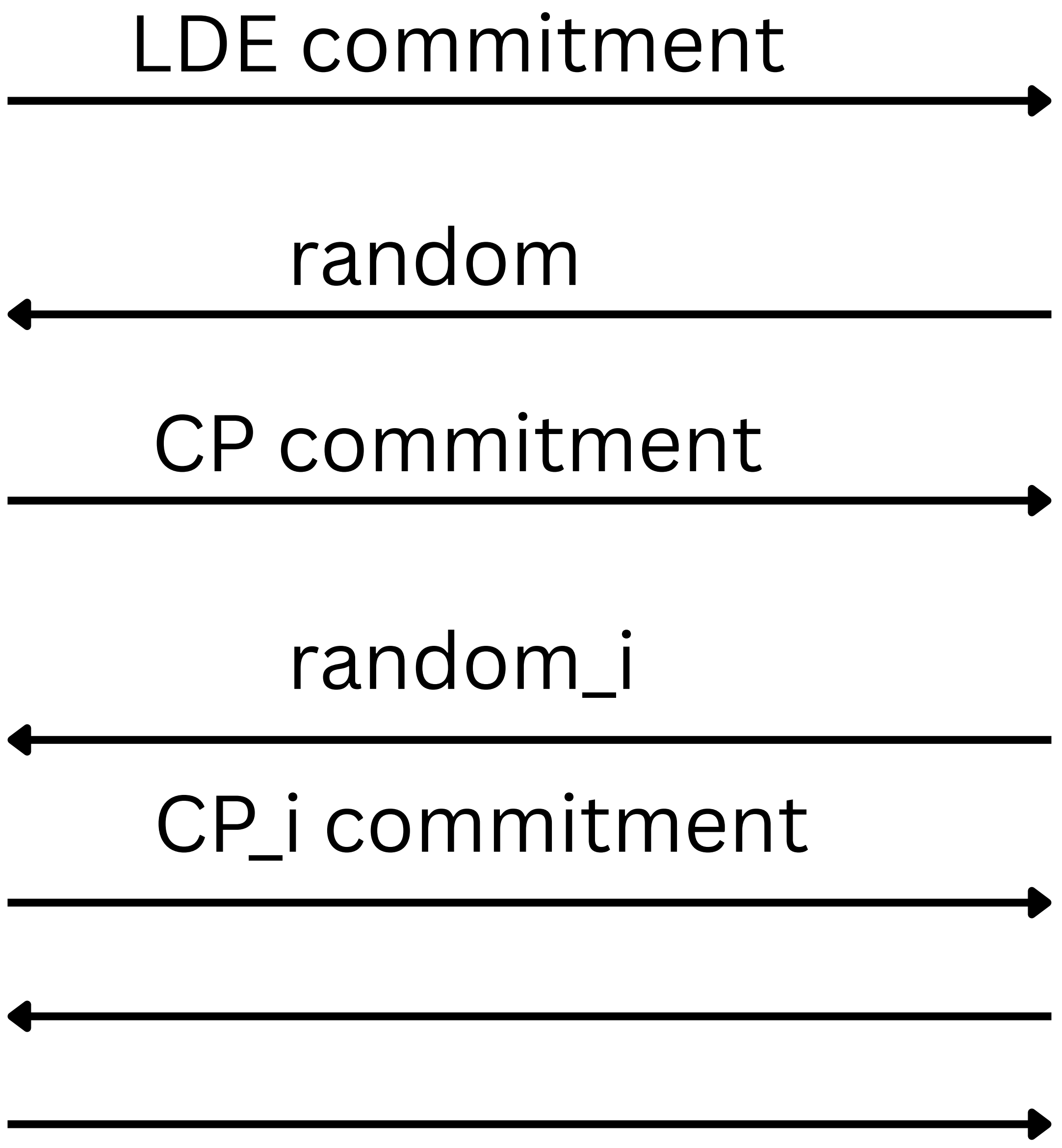
5. STARK Protocol

Commit Phase

Prover

Verifier

1. Interpolate Trace Poly
2. LDE of Trace
3. Quotient Polys
4. Composition poly
5. FRI commit phase



5. STARK Protocol

Query Phase

Prover

Verifier

6. Query phase

Query at q

proof = { p(q) + path
p(hq) + path
p(h^2 q) + path
cp(q) + path
cp(-q) + path
cp1(q^2) + path
cp1(-q^2) + path
… }

Verify:

proof

$$isValidMerklePath(\dots)$$

$$cp(q) = \alpha_0 p(q) + \alpha_1 p_1(q) + \alpha_2 p_2(q)$$

$$CP_{i+1}(q^{2(i+1)}) = \frac{CP_i(q^{2i}) + CP_i(-q^{2i})}{2} + \beta_i \cdot \frac{CP_i(q^{2i}) - CP_i(-q^{2i})}{2q^{2i}}$$

•••

Quizzes

1. "S" in STARK stands for?
   a. Scalable
   b. Succint
   c. Safe
   d. Secure

2. Polynomial interpolation used in STARK is:
   a. FFT
   b. Lagrange Interpolation
   c. Newton Interpolation
   d. Hermite Interpolation

3. Why need multiplicative subgroup in STARK?
   a. Because it helps performing FFT
   b. Because it easy to choose
   c. Because it is secure
   d. Because hashing element in multiplicative subgroup is faster than others

Quizzes

4. What Low Degree Extension (LDE) in STARKs does?
a. Extend evaluation of polynomial to a larger domain
b. Build Merkle Tree commitment
c. Choose coset of trace domain
d. Check that the prover has constructed a low-degree polynomial

5. What Low Degree Testing in STARKs?
a. Extend evaluation of polynomial to a larger domain
b. Build Merkle Tree commitment
c. Choose coset of trace domain
d. Check that the prover has constructed a low-degree polynomial

Recommend Resources

- Vitalik's STARK posts: part I, part II, part III
- Anatomy of STARK
- STARK 101
- Note on STARK arithmetization
- Starknet blogs