CSE 575: Advanced Cryptography
Fall 2024
Lecture 22

Paul Grubbs

paulgrub@umich.edu

Beyster 4709

Agenda for this lecture

- Announcements
- Recap: arithmetizing Plonkish constraints
- Gate check
- Wiring check
- Zero test PIOP
- Product check PIOP

Agenda for this lecture

- Announcements
- Recap: arithmetizing Plonkish constraints
- Gate check
- Wiring check
- Zero test PIOP
- Product check PIOP

Announcements

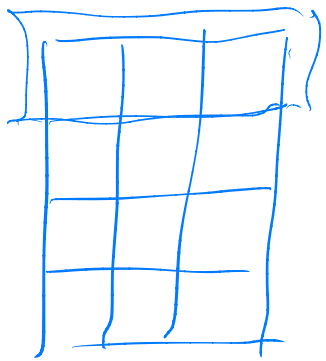
- HW4 is out, due 12/6
- Rest of class: no typeset lecture notes.
  - Can get extra credit for typesetting

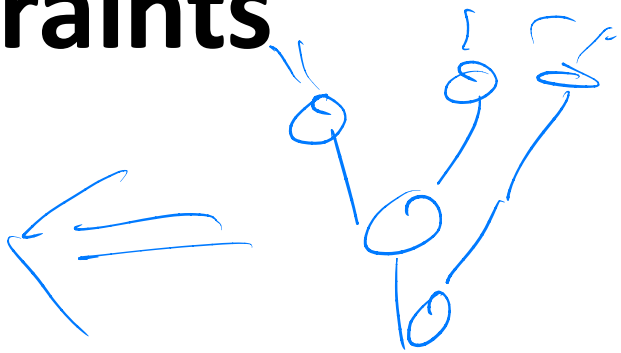
Agenda for this lecture

- Announcements
- Recap: arithmetizing Plonkish constraints
- Gate check
- Wiring check
- Zero test PIOP
- Product check PIOP

Arithmetizing Plonkish constraints

Transcript T





$T(\omega^{3i}) =$ left input of gate i

$T(\omega^{3i+1}) =$ right input of i

$T(\omega^{3i+2}) =$ output of i

$T(\omega^{-j}) =$ pub input j

Arithmetizing Plonkish constraints

Four checks on T

(1) T encodes correct pub inputs

(2) all gates correct

(3) all wiring correct (permutation check)

(4) output is 1 $3n-1$

$\vdash T(\omega^{3n-1}) = 1$

Arithmetizing Plonkish constraints

Pub inputs I

(1) Imagine we have V s.t.

$$V(\omega^{-i}) = \text{pub input } i$$

Public inputs are correct in T if

$$(*) \quad T(\omega^{-i}) - V(\omega^{-i}) = 0 \quad \forall\, i = 1, \ldots, |I|$$

Can use "zero test" to check $(*)$

Agenda for this lecture

- Announcements
- Recap: arithmetizing Plonkish constraints
- Gate check
- Wiring check
- Zero test PIOP
- Product check PIOP

$: array of gate ops

Gate check

(Note: S is preprocessed)

$S(\omega^{3\ell}) = 1$: if gate ℓ is MUL

$S(\omega^{3\ell}) = 0$: if " " " ADD

$$S(\omega^{3i})\,T(\omega^{3i})\,T(\omega^{3i+1}) + (1 - S(\omega^{3i}))\,(T(\omega^{3i}) + T(\omega^{3i+1})) - T(\omega^{3i+2}) = 0$$

for $i = 0, \ldots, n-1$

Use zero Test to check (see later)

Agenda for this lecture

- Announcements
- Recap: arithmetizing Plonkish constraints
- Gate check
- Wiring check
- Zero test PIOP
- Product check PIOP

Wiring check

σ of wire indexes

Let P be poly that permutes ω according to σ

$$P: \omega^i \to \omega^{\sigma(i)}$$

Intuition: $\forall i,\ T(\omega^i) = T(P(\omega^i))$

Need to do in different way for efficiency

Note: P also preprocessed

~~[illegible]~~ Plonk PIOP

$P(S, P, X, W, pp)$:

$V(\ell_S, \ell_P, X)$:

compute trace T,
interpolate it

ℓ_T ⟶ build PI poly v

⟵ PIOPs of ℓ_T for checks:
1. pub input
2. gate
3. wiring
4. output ⟶

Agenda for this lecture

- Announcements
- Recap: arithmetizing Plonkish constraints
- Gate check
- Wiring check
- Zero test PIOP
- Product check PIOP

Vanishing polynomial of a subset

Let Ω be a subset of $\mathbb{F}$ of size K

Vanishing Polynomial Z_Ω of Ω is unique monic deg-K poly

$$Z_\Omega(X) = \prod_{a \in \Omega} (X - a)$$

If Ω is cyclic subgroup of order K:

$$Z_\Omega(X) = X^K - 1$$

why?

$\{\omega^0, \omega^1, \ldots, \omega^{K-1}\}$ $\longleftrightarrow$ mult order K, so

$(\omega^i)^K = 1$, $1 - 1 = 0$

Zero test Polynomial IOP

Ω has order k in $\mathbb{F}$

$f(1) = 0, f(\omega) = 0, \ldots f(\omega^{k-1}) = 0$

Key idea:

$f \equiv 0$ on Ω if Z_Ω divides f

$f = q Z_\Omega$

$P(f)$:

Compute q s.t. $f = q Z_\Omega$

$\mathcal{V}$

$\xrightarrow{l_f, l_q}$

$r \leftarrow \mathbb{F}$

$\xleftarrow{\text{query } l_f, l_q \text{ at } r}$

$r_1 = f(r)$

$r_2 = q(r)$

$\xrightarrow{r_1, r_2}$ Rej

Completeness

Soundness:

f doesn't vanish on Ω. for any q, $(f - q Z_\Omega)(X)$ has deg $k-1$, so $\leq \frac{k}{|\mathbb{F}|}$

$r_1 \stackrel{?}{=} r_2 \frac{Z_\Omega(r)}{r^k - 1}$

Agenda for this lecture

- Announcements
- Recap: arithmetizing Plonkish constraints
- Gate check
- Wiring check
- Zero test PIOP
- Product check PIOP

Product check Polynomial IOP