EECS 575: Advanced Cryptography
Fall 2022
Lecture 25

Paul Grubbs

paulgrub@umich.edu

Beyster 4709

Agenda for this lecture

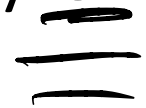
- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- ~~Interactive oracle proofs~~

Agenda for this lecture

- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- Interactive oracle proofs

Announcements

- Final exam due 12/7
- No hints on final – clarifying questions only. Sorry ☹



Agenda for this lecture

- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- Interactive oracle proofs

Proof systems until ~2012

Gen 1 of proofs

GMR Babai – Early 80s

Interactive

Info-theoretic

Limited / hard to use

↳ 3COL

NIZKs via Fiat-Shamir

Structured Reference string

No practical applications

≈ efficient degree-two relations in exponent

≠

Groth-Sahai proofs

Quadratic Arithmetic programs = Constant-size proofs

polynomial commitments

" KZG

Bitcoin/blockchain becomes more mainstream

Academic publications

Zcash

Zk contingent payments

Maxwell

Other applications zero-knowledge middlebrows

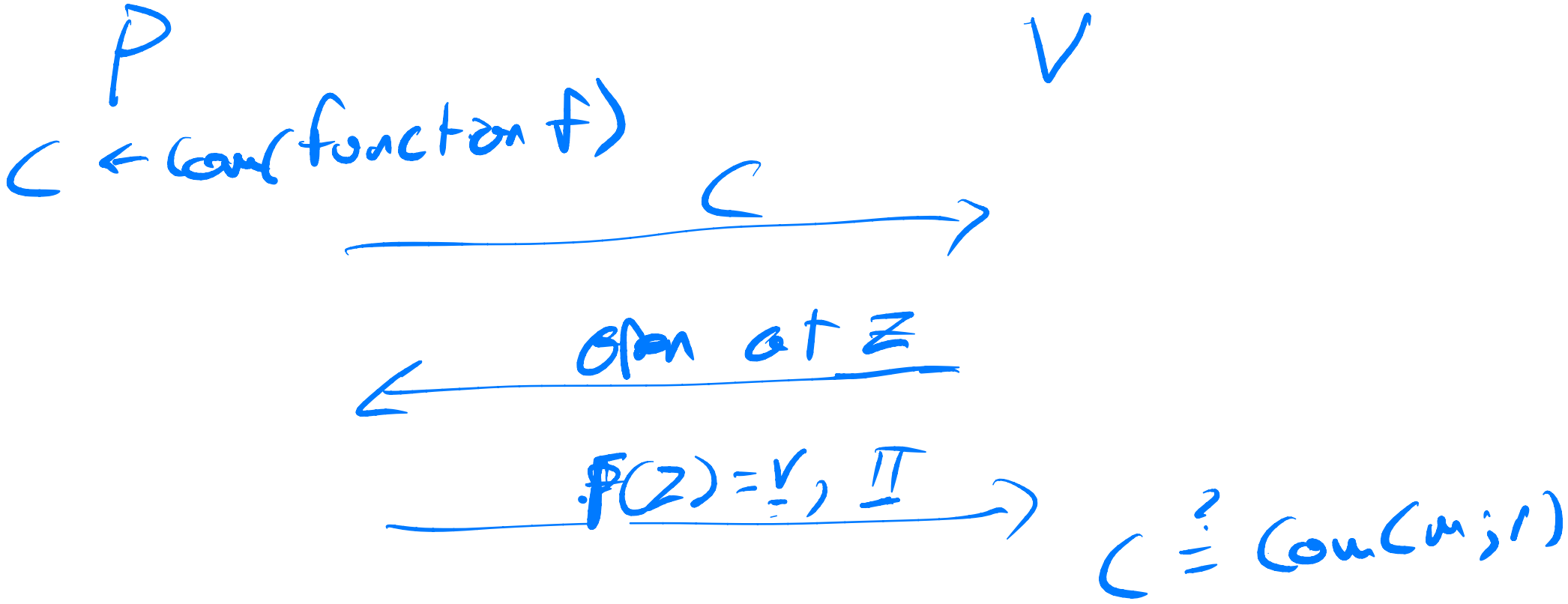
Agenda for this lecture

- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- Interactive oracle proofs

Polynomial commitment (PC) syntax

Polynomial commitment (PC) ~~syntax~~

Intuition



Polynomial commitment (PC) ~~syntax~~

Background

$p(X) = a_n X^n + \cdots + a_1 X + a_0$

$n+1$ elements of $\mathbb{F}_q$

$p(z) = a_n z^n + \quad a_1 z + a_0 \in \mathbb{F}_q$

$\mathbb{F}_q[X]$ tons of structure!

Fund. Thm. Algebra

$p(X)$ degree n

$\Rightarrow$ at most n roots

$p(r) = 0$

Extended euclidean algorithm

polynomial GCD

Div-mod:

$p(X) = w(X)\, q(X) + r(X)$

$\forall\, p(X) \in \mathbb{F}_q[X]$, deg n,

$\Pr[p(r) = 0 : r \leftarrow \mathbb{F}_q] \leq \frac{n}{q}$

Polynomial commitment (PC) syntax

Lemma: For poly $p(X) \in \mathbb{F}_q[X]$, $\deg(p) = n$,

$\forall z, v \in \mathbb{F}_q$,

$\exists\, w(X)$ degree $n-1$ s.t. $p(X) - v = w(X)(X-z)$

iff

$p(z) = v$

Proof:

$$p(X) = w(X)(X-z) + v$$

$$p(z) = \cancel{w(z)(z-z)} + v$$

Polynomial commitment (PC) syntax

- $\mathsf{Setup}(n)$: output pp (n is degree bound)
- $\mathsf{Commit}(pp, p)$: output c_p and opening r
- $\mathsf{Open}(pp, c_p, p, z, r)$: output π v s.t. $p(z) = v$
- $\mathsf{Verify}(pp, c_p, z, v, \pi)$: outputs 0/1

PC security

Evaluation Binding

PC is Eval binding if
$\forall p \neq 0$, $\forall \pi$, $\exists$ nuppt $\mathcal{A}$ s.t.

$$\Pr[\mathrm{EBind}^{\mathcal{A}}_{PC,p} = 1] \text{ non-negl}$$

$\mathrm{EBind}^{\mathcal{A}}_{PC,p}$:

$pp \leftarrow \mathrm{setup}(\lambda)$; $z \leftarrow \mathcal{F}_a$

$C_p, (\pi_1, v_1), (\pi_2, v_2) \leftarrow \mathcal{A}(p, z)$

Ret $\mathrm{Ver}(pp, C_p, \pi_1, z, v_1)$
$\wedge\ \mathrm{Ver}(pp, C_p, \pi_2, z, v_2)$
$\wedge\ v_1 \neq v_2$

- Hiding: as for regular commitments

Agenda for this lecture

- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- Interactive oracle proofs

Pairings background

Let G, G_t cyclic groups, order q

Map $e : G \times G \to G_t$ is bilinear if

$\forall u, v \in G$, $a, b \in [0, \ldots, q-1]$

$$e(u^a, v^b) = e(u, v)^{ab}$$

- Non-degenerate: $e(g, g) = 1_{G_t}$
- Eff. computable

G usually subgroup of elliptic curve over $\mathbb{F}_p$

G_t || || $\mathbb{F}_{p^k}$ $k =$ embedding degree

Leg $k=12$

Pairings background

Pairings allow checking mult in exponent

- $Com(x) = g^x$

C_x C_y C_z, can check $x+y=z$

$C_x \cdot C_y \stackrel{?}{=} C_z$

If G has pairing, check $x \cdot y = z$

$e(C_x, C_y) \stackrel{?}{=} e(g, C_z)$

if $x \cdot y = z$ then

$e(g^x, g^y) = e(g,g)^{xy} = e(g, C_z)$

Facts about polynomials

The KZG construction

- Setup(n): G, G_t order q
  - $\tau \leftarrow \mathbb{F}_q$ // g_0 // g_1
  - Ret $(g, g^{\tau}, g^{\tau^2}, \ldots, g^{\tau^n}) \in G^{n+1}$ — "Trusted" Setup
- Commit(pp, $P = (a_n, \ldots, a_1, a_0)$):
  - Ret $C_P = \prod_{i=0}^{n} (g_i)^{a_i}$
  - E.g. $n=1$: $a_1 x + a_0$; $C_P = g_1^{a_1} \cdot g^{a_0} = (g^{\tau})^{a_1} \cdot g^{a_0} = g^{a_1 \tau + a_0}$
- Open(pp, C_P, P, z, $r = \varepsilon$):
  - Compute $w(X)$ s.t. $P(X) - v = w(X)(X - z)$
  - Compute C_w as above
  - Ret C_w, $P(z)$
- Verify(pp, C_P, z, v, π):
  - Ret $e(\pi, g_1/g^z) \stackrel{?}{=} e(g, C_P/g^v)$

Coms, openings: $O(1)$

Ver $O(1)$

Polynomial division: Need FFT, $O(n \log^2 n)$

Correctness: $e(g^{w(\tau)}, g^{\tau}/g^z) = e(g, g)^{w(\tau)(\tau - z)}$

$e(g, C_P/g^v) = e(g, g^{P(\tau) - v}) = e(g, g)^{P(\tau) - v}$

The KZG construction

Eval binding: reduce to n-strong DH

Given KZG PP,

hard to compute $(z, g^{1/(z-z')})$

Agenda for this lecture

- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- Interactive oracle proofs

What is a computation, anyway?

- Boolean circuit:

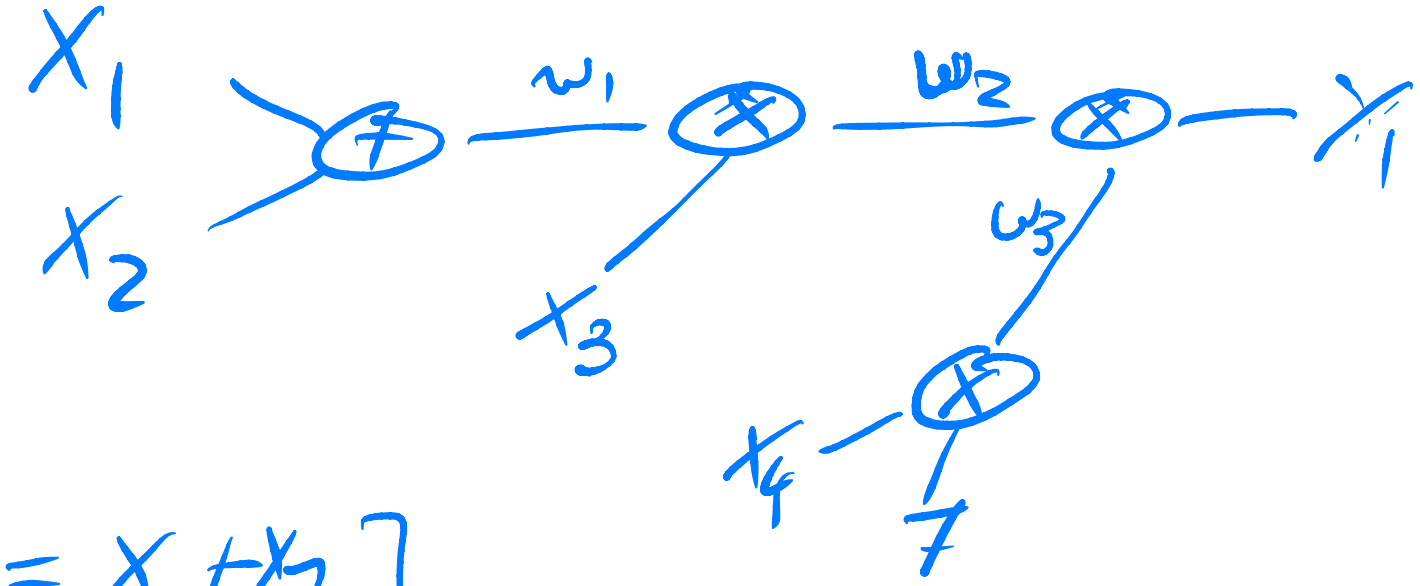
  DAG with in-deg two
  Each node XOR or AND
  wires labelled with
  output values

- Arithmetic circuits:

  DAG with in, +, ×
  Each node labelled with
  +/- in $\mathbb{F}_p$
  wires labelled with
  $\mathbb{F}_p$

  (Boolean: AC with $p=2$)

What is a computation, anyway?



$$w_1 = x_1 + x_2$$
$$w_2 = x_3 \cdot w_1$$
$$w_3 = x_4 \cdot 7$$
$$y_1 = w_2 \cdot w_3$$

→ Can we represent in just one constraint?

What is a computation, anyway?

SNARG : Succinct non-interactive argument

SNARK : " " " of knowledge

ZkSNARK

What is a computation, anyway?

Rank-one constraint

Define $z = (x, y, w, 1) \in \mathbb{F}^n$

For $a, b, c \in \mathbb{F}^n$, a rank-1 constraint

$$\langle a, z \rangle \cdot \langle b, z \rangle = \langle c, z \rangle$$

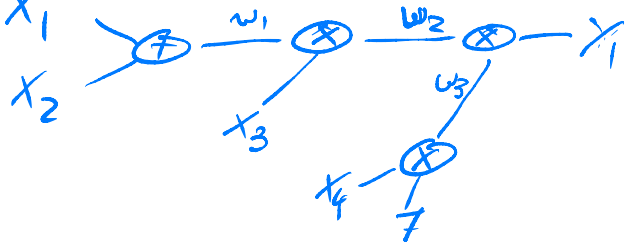
$\langle \vec{x}, \vec{y} \rangle = \sum_i x_i y_i$

Constraints on linear combinations of wire values

What is a computation, anyway?

To translate into R-1 constraints:

1. write down mul. gates
   - C_1: $w_2 = (x_1 + x_2) \cdot x_3$
   - C_2: $w_3 = 7 \cdot x_4$
   - C_3: $y_1 = w_2 \cdot w_3$
2. Define $z = (x_1, x_2, x_3, x_4, y_1, w_1, w_2, w_3, 1)$
3. For each constraint, write a, b, c vectors



C_1:

$c_1 = (0\,0\,0\,0\,0\,0\,0\,1\,0\,0)$ (w_2)

$a_1 = (1\,1\,0\,0\,0\,0\,0\,0\,0\,0)$ $(x_1 + x_2)$

$b_1 = (0\,0\,1\,0\,0\,0\,0\,0\,0\,0)$ (x_3)

C_2:

$c_2 = (0\,0\,0\,0\,0\,0\,0\,0\,1\,0)$ w_3

$a_2 = (0\,0\,0\,0\,0\,0\,0\,0\,0\,7)$ 7

$b_2 = (0\,0\,0\,1\,0\,0\,0\,0\,0\,0)$ x_4

C_3:

$c_3 = (0\,0\,0\,0\,1\,0\,0\,0\,0\,0)$

$a_3 = (0\,0\,0\,0\,0\,0\,0\,0\,1\,0)$

$b_3 = (0\,0\,0\,0\,0\,0\,0\,0\,0\,1)$

Mistake: All should have nine zeros. Sorry ☹

What is a computation, anyway?

Rank-1 constraint system

Given by three matrices $A, B, C \in \mathbb{F}^{m \times n}$

Satisfied for $z \in \mathbb{F}^n$ s.t.

$$Az \overset{\circ}{\cdot} Bz = Cz$$

element-wise multiplication

For RICS, A, B, C and inputs x, y, system is satisfiable if $\exists w$ s.t.

$z := (x, y, w, 1)$ satisfies RICS

Thm: RICS-SAT is NP-complete.

Why RICS? "fits" with tools we have for proofs

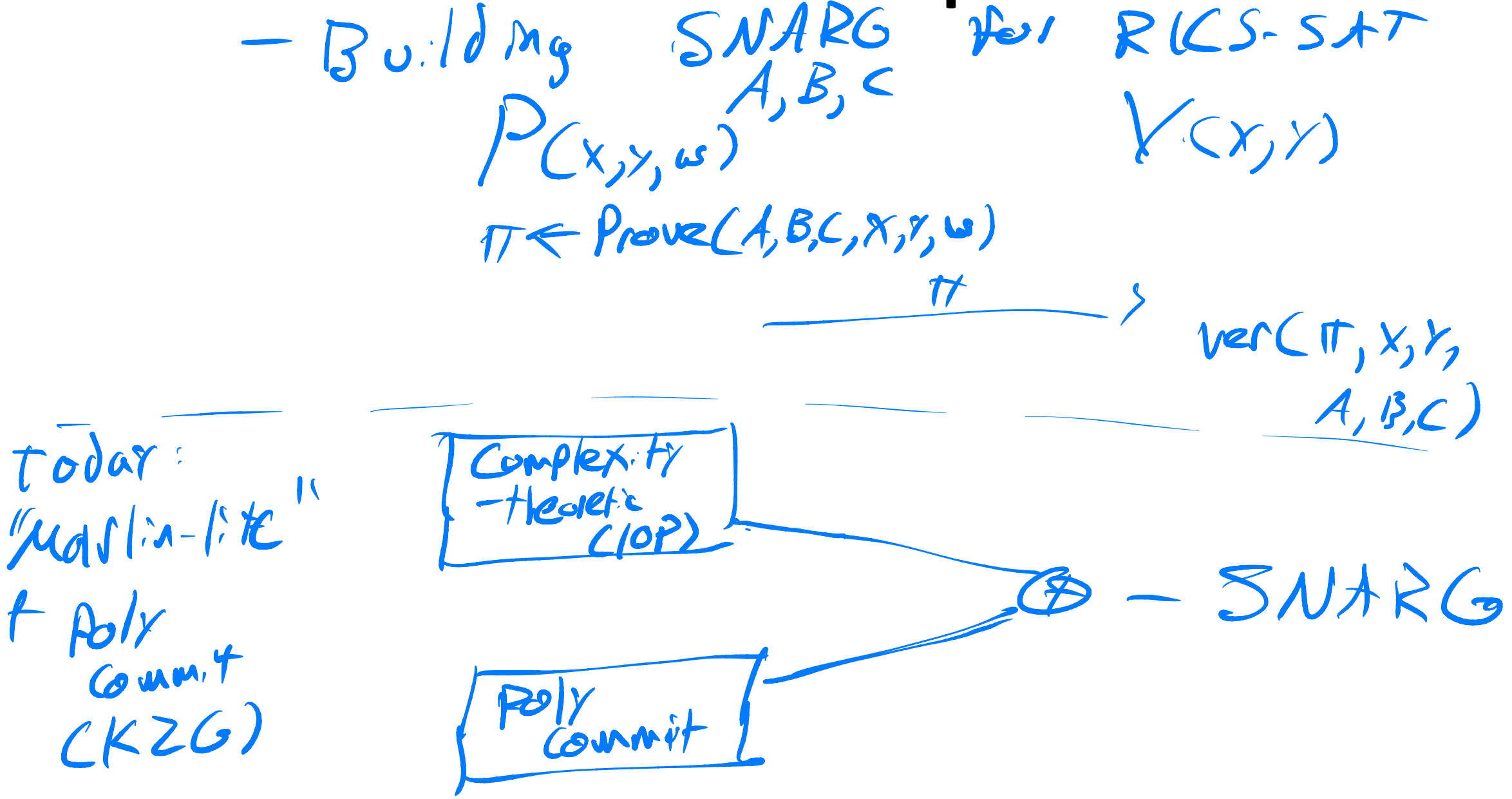
"Tools" = polynomial protocols

"How to compile" to RICS? Non-trivial!

Agenda for this lecture

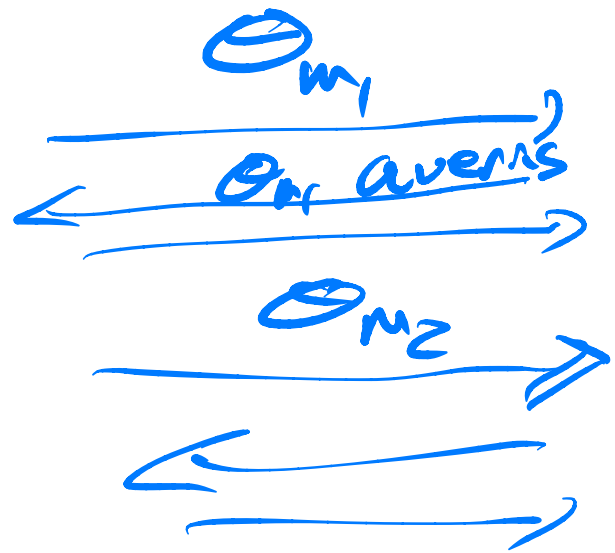
- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- Interactive oracle proofs

Interactive oracle proofs



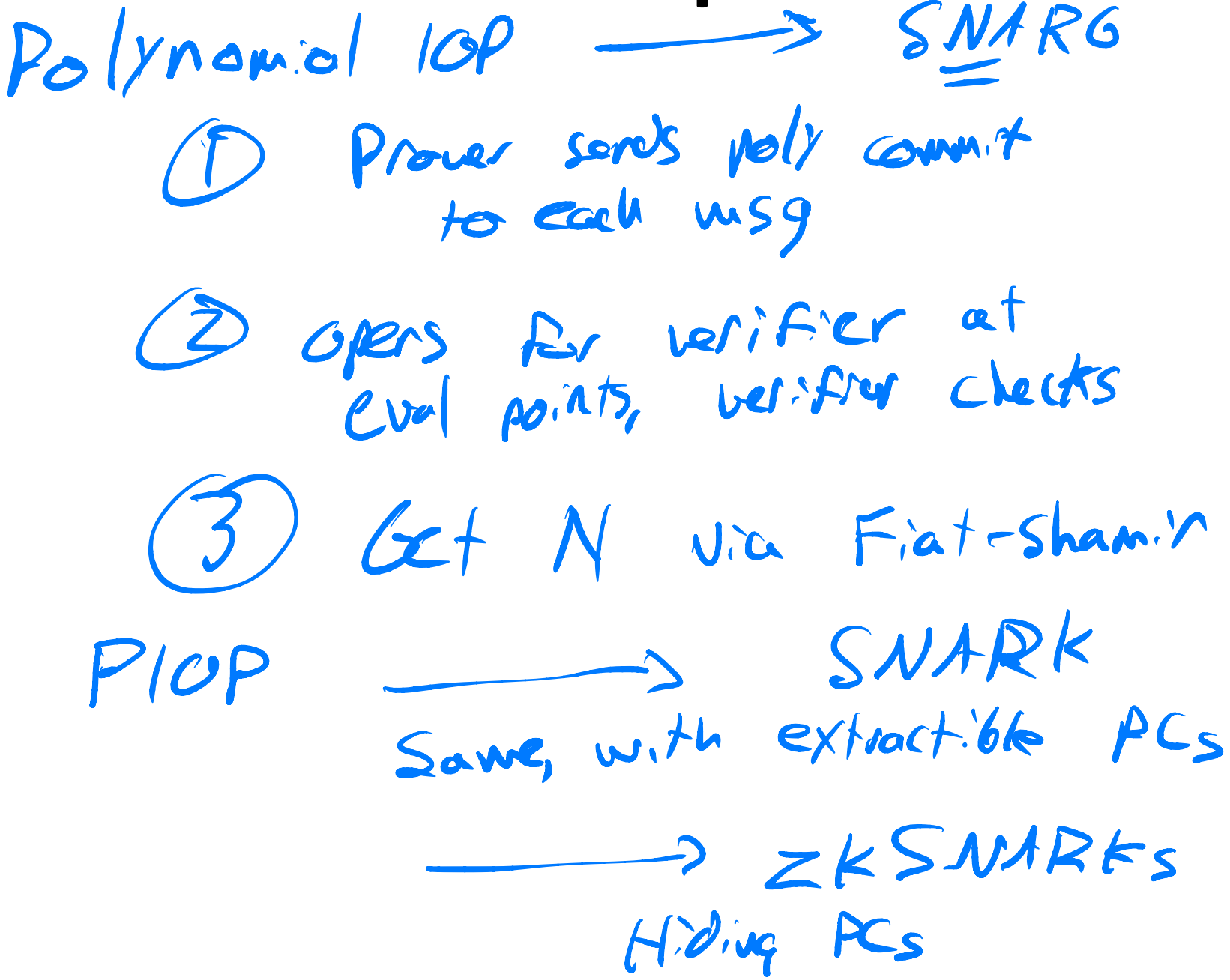
Interactive oracle proofs (IOP)

- generalization of IP + PCPs
- In each round, P "sends" oracle to its message. V can "query" message



Marlin-like IOP: public-coin, each P message consists of bounded-degree polys

Interactive oracle proofs



PP #1

Fact: Two distinct degree-d polys P, Q have $P(x) = Q(x)$ for at most d xs in $\mathbb{F}$

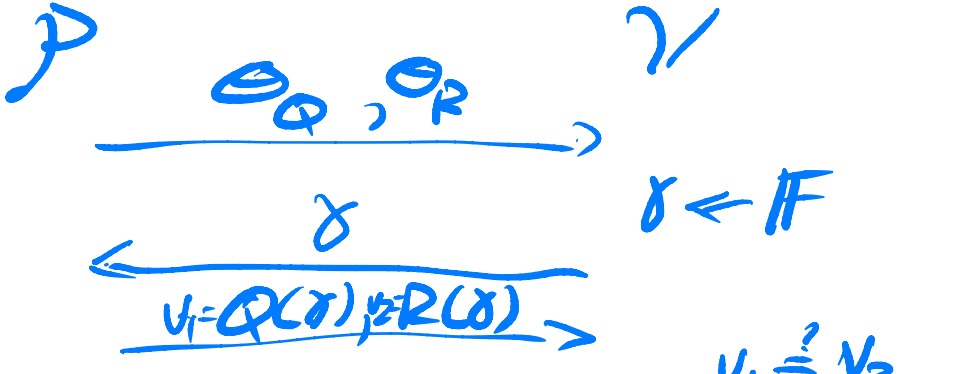
$(P-Q)(x) = 0$

$\Rightarrow P(x) = Q(x)$

Polynomial Equality (cont.)

P convinces V it has two degree $\le d$ polys

Protocol:



Completeness: obvious

Soundness: $\le \frac{d}{|\mathbb{F}|}$

PP #2 Polynomial vanishes on mult. subgroup

Let $H \subseteq \mathbb{F}$, $|H| = n$, h generator $H = \{h^1, \ldots, h^n\}$

Define $Z_H(X) = \prod_{i=1}^{n} (X - h^i)$ "vanishing poly" on H

Fact #2: Degree $\leq d$ poly $g(x)$ vanishes on H iff $\exists R$ of deg. $d-n$ s.t. $g(x) = R(x) Z_H(x)$

For $s \in H$: $g(s) = R(s) Z_H(s) = 0$

Protocol:

① $\mathcal{P}$ send g, R to $\mathcal{V}$

② $\mathcal{V}$ $\gamma \leftarrow \mathbb{F}$, queries $g(\gamma)$ and $R(\gamma)$, $Z_H(\gamma)$ (easy)

③ $\mathcal{V}$ $g(\gamma) \stackrel{?}{=} R(\gamma) Z_H(\gamma)$

Why?

Completeness: by Fact 2

Soundness: $\leq \frac{d}{|\mathbb{F}|}$

because no R exists, so LHS ≠ RHS ∴ reject

PP#3

Univariate sumcheck

For $H \subset \mathbb{F}$ mult subgroup and poly Q of degree d, Check

$$\sum_{a \in H} Q(a) = 0 \quad (1)$$

Fact 3: Eqn (1) holds iff $\exists$ polys R deg $\leq d-n$, S deg $< n-1$ s.t. $Q(X) = Z_H(X) R(X) + X S(X)$

Protocol:

1. $\mathcal{P}$ sends Q, R, S oracles
2. $\mathcal{V}$ sends $\gamma \leftarrow \mathbb{F}$, gets $Q(\gamma), R(\gamma), S(\gamma)$ and evals $Z_H(\gamma)$
3. $\mathcal{V}$ $Q(\gamma) = Z_H(\gamma) R(\gamma) + \gamma S(\gamma)$

Completeness: Fact 3

Soundness: $\leq \frac{d}{|\mathbb{F}|}$

Fix $A, B, C \in \mathbb{F}^{n \times n}$

Goal: For public (x, y), $\mathcal{P}$ convinces $\mathcal{V}$ $\exists w$ s.t.

$\vec{z} = (x, y, w, 1)$

Satisfies:

$A\vec{z} \circ B\vec{z} = C\vec{z}$

$H \subseteq \mathbb{F}, |H| = n$

Morlin-lite: IOP for R1CS-SAT

Idea: encode $\vec{z}, A\vec{z}, B\vec{z}, C\vec{z}$ as polys of deg. $n-1$

Define $\hat{z}(X)$ is unique deg $n-1$ poly s.t. $\hat{z}(h^i) = z[i], i \in [n]$

Likewise for $A\vec{z}, B\vec{z}, C\vec{z}$ get $\hat{z}_A, \hat{z}_B, \hat{z}_C$

Step 1: $\mathcal{P}$ send $\hat{z}, \hat{z}_A, \hat{z}_B, \hat{z}_C$ to $\mathcal{V}$

Then $A\vec{z} \circ B\vec{z} = C\vec{z} \iff \hat{z}_A(h^i)\hat{z}_B(h^i) = \hat{z}_C(h^i)$

$\Rightarrow (\hat{z}_A \hat{z}_B - \hat{z}_C)(X)$ vanishes on H

Use pp#2!

① P sends γ R s.t.

$$\hat{Z}_A \hat{Z}_B - \hat{Z}_C(x) = R(x) Z_H(x)$$

② γ queries $\gamma \in \mathbb{F}$, check

$$\underline{\hat{Z}_A(\gamma)\hat{Z}_B(\gamma)} - \hat{Z}_C(\gamma) = R(\gamma) Z_H(\gamma)$$

Soundness $\leq \frac{2^n}{|\mathbb{F}|}$

One problem: still need to check consistency of $\hat{Z}_A$ $\hat{Z}_B$ $\hat{Z}_C$ oracles with Z, A, B, C

Remainder of protocol: loop for doing that