EECS 575: Advanced Cryptography
Fall 2022
Lecture 25

Paul Grubbs

paulgrub@umich.edu

Beyster 4709

Agenda for this lecture

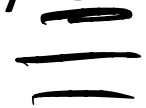
- Announcements
- Recap from last time
- Polynomial commitments (PCs)
- PCs from pairings (KZG)
- Representing computations as constraints
- ~~Interactive oracle proofs~~

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

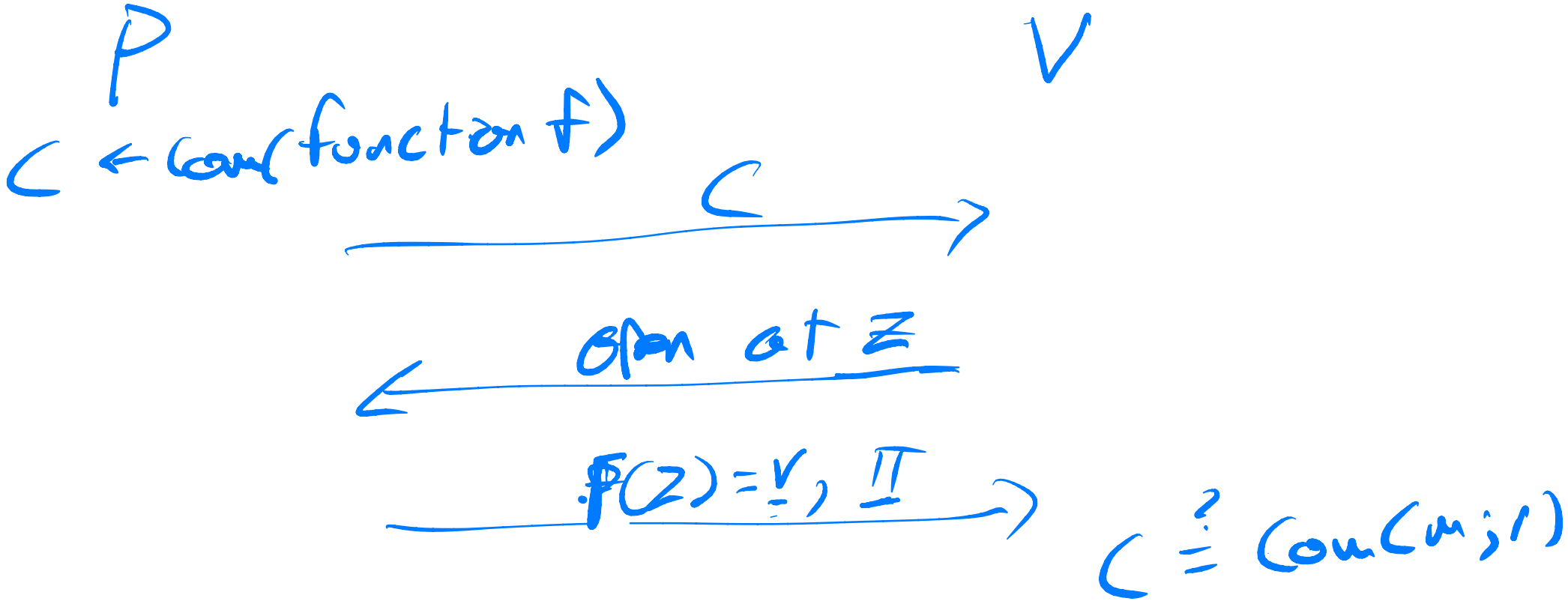
Agenda for this lecture

- 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?

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