$\mathfrak{cq}$: Cached quotients for fast lookups

Liam Eagen   Dario Fiore   **Ariel Gabizon**

13. januar 2023

Outline

- PCS/KZG review

Outline

- PCS/KZG review
- KZG shenanigans
  1. Committing to sparse polys.
  2. “cached quotients”

Outline

- PCS/KZG review
- KZG shenanigans
    1. Committing to sparse polys.
    2. “cached quotients”
- Lookups
    1. Motivation
    2. log derivative protocol[Eagen, Haböck,..]
    3. $\mathfrak{cq}$

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment $\mathbf{cm}(\mathbf{f})$ to polynomial.

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment $\mathbf{cm}(\mathbf{f})$ to polynomial.
- Later Verifier can choose value $\mathbf{i} \in \mathbb{F}$.

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment $\mathbf{cm}(\mathbf{f})$ to polynomial.
- Later Verifier can choose value $\mathbf{i} \in \mathbb{F}$.
- Prover sends back $z = \mathbf{f}(\mathbf{i})$; together with proof $\mathbf{open}(\mathbf{f}, \mathbf{i})$ that z is correct.

Polynomial commitment schemes [KZG, 10]

- Prover send short commitment $\mathbf{cm}(f)$ to polynomial.
- Later Verifier can choose value $i \in \mathbb{F}$.
- Prover sends back $z = f(i)$; together with proof $\mathbf{open}(f, i)$ that z is correct.

KZG give us PCS with commitments and openings are practically 32-48 bytes.
Notation: $[x] = x \cdot g$ where g generator of (an additive) elliptic curve group.

KZG10:

$\mathbf{srs} := [\mathbf{1}], [\boldsymbol{x}], \ldots, \left[\boldsymbol{x}^{\mathbf{d}}\right]$, for random $\boldsymbol{x} \in \mathbb{F}$.

KZG10:

srs $:= [\mathbf{1}], [\boldsymbol{x}], \ldots, \left[\boldsymbol{x}^{\mathbf{d}}\right]$, for random $\boldsymbol{x} \in \mathbb{F}$.

cm$(\mathbf{f})$ $:= [\mathbf{f}(\boldsymbol{x})]$

KZG10:

srs $:= [\mathbf{1}], [x], \ldots, \left[x^{d}\right]$, for random $x \in \mathbb{F}$.

cm$(f) := [f(x)]$

open$(f, i) := [h(x)]$, where $h(X) := \frac{f(X)-f(i)}{X-i}$

KZG10:

srs $:= [\mathbf{1}], [x], \ldots, \left[x^{d}\right]$, for random $x \in \mathbb{F}$.

cm$(f) := [f(x)]$

open$(f, i) := [h(x)]$, where $h(X) := \frac{f(X) - f(i)}{X - i}$

verify$(\mathbf{cm}, \pi, z, i)$:

$$e(\mathbf{cm} - [z], [\mathbf{1}]) \stackrel{?}{=} e(\pi, [x - i])$$

Shenanigan $\#1$: Committing to sparse polys

notation: parameters $n \ll N$, $d := N - 1$.

$\mathbb{V} = \left\{\omega, \ldots, \omega^N\right\} \subset \mathbb{F}$ subgroup of size N.

Shenanigan #1: Committing to sparse polys

notation: parameters $n \ll N$, $d := N - 1$.

$\mathbb{V} = \{\omega, \ldots, \omega^N\} \subset \mathbb{F}$ subgroup of size N.

Say $A \in \mathbb{F}_{<N}[X]$ is n-sparse if has at most n non-zeroes on $\mathbb{V}$.

Shenanigan $\#1$: Committing to sparse polys

notation: parameters $n \ll N$, $d := N - 1$.

$\mathbb{V} = \{\omega, \ldots, \omega^N\} \subset \mathbb{F}$ subgroup of size N.

Say $A \in \mathbb{F}_{<N}[X]$ is n-sparse if has at most n non-zeroes on $\mathbb{V}$.

For $i \in [N]$, denote $A_i := A(\omega^i)$
$L_1(X), \ldots, L_N(X)$ - Lagrange basis of $\mathbb{V}$ -
$(L_i)_j = 0$ when $i \neq j$.

Committing to sparse polys

From $\mathbf{srs} := [1], [x], \ldots, [x^d]$, we can precompute in $O(N \log N)$ operations the KZG commitments of $\mathbb{V}$'s Lagrange Base:
$\mathbf{srs}_L := \{[L_1(x)], \ldots, [L_N(x)]\}$

Committing to sparse polys

From **srs** $:= [1], [x], \ldots, [x^d]$, we can precompute in $O(N \log N)$ operations the KZG commitments of $\mathbb{V}$'s Lagrange Base:
srs$_L := \{[L_1(x)], \ldots, [L_N(x)]\}$

Now for n-sparse $A(X)$ of degree$<N$ compute

$$\mathbf{cm}(A) = [A(x)] = \sum_{i \in [N], A_i \neq 0} A_i \cdot [L_i(x)]$$

Shenanigan **#2**: “Cached quotients” method

Scenario: $\mathsf{T}(X) \in \mathbb{F}_{<\mathbf{N}}[X]$ preprocessed poly.
$\mathsf{Z}_{\mathbb{V}}(X)$-vanishing poly of $\mathbb{V}$.
Input: $\mathfrak{n}$-sparse $A(X) \in \mathbb{F}_{<\mathbf{N}}[X]$.

Shenanigan #2: "Cached quotients" method

Scenario: $\mathrm{T}(X) \in \mathbb{F}_{<N}[X]$ preprocessed poly.
$Z_{\mathbb{V}}(X)$-vanishing poly of $\mathbb{V}$.
Input: n-sparse $A(X) \in \mathbb{F}_{<N}[X]$.

V has $\mathbf{cm}(A)$. Want to prove to V that:
$Z_V(X)$ divides $A(X)\mathrm{T}(X)$ *using* $O(n)$ *prover operations*.

“Cached quotients” method

There exists quotient $\mathbf{Q}_A(\mathbf{X})$ such that $\mathbf{A} \cdot \mathbf{T} \equiv \mathbf{Z}_V \cdot \mathbf{Q}_A$.

“Cached quotients” method

There exists quotient $\mathbf{Q}_A(\mathbf{X})$ such that
$A \cdot \mathbf{T} \equiv \mathbf{Z}_V \cdot \mathbf{Q}_A$.

We’ll compute $[\mathbf{Q}_A(x)]$ in $\mathbf{O}(n)$ operations:

“Cached quotients” method

There exists quotient $\mathbf{Q}_{\mathbf{A}}(\mathbf{X})$ such that $\mathbf{A} \cdot \mathbf{T} \equiv \mathbf{Z}_{\mathbf{V}} \cdot \mathbf{Q}_{\mathbf{A}}$.

We’ll compute $[\mathbf{Q}_{\mathbf{A}}(\mathbf{x})]$ in $\mathbf{O}(\mathbf{n})$ operations:

preprocessing: For each $\mathbf{i} \in [\mathbf{N}]$, compute $[\mathbf{Q}_{\mathbf{i}}(\mathbf{x})]$ such that for some $\mathbf{R}_{\mathbf{i}}(\mathbf{X}) \in \mathbb{F}_{<\mathbf{N}}[\mathbf{X}]$

$$\mathbf{L}_{\mathbf{i}}(\mathbf{X}) \cdot \mathbf{T}(\mathbf{X}) = \mathbf{Q}_{\mathbf{i}}(\mathbf{X}) \cdot \mathbf{Z}_{\mathbb{V}}(\mathbf{X}) + \mathbf{R}_{\mathbf{i}}(\mathbf{X})$$

“Cached quotients” method

There exists quotient $Q_A(X)$ such that $A \cdot T \equiv Z_V \cdot Q_A$.

We’ll compute $[Q_A(x)]$ in $O(n)$ operations:

preprocessing: For each $i \in [N]$, compute $[Q_i(x)]$ such that for some $R_i(X) \in \mathbb{F}_{<N}[X]$

$$L_i(X) \cdot T(X) = Q_i(X) \cdot Z_{\mathbb{V}}(X) + R_i(X)$$

Also precompute $[Z_{\mathbb{V}}(x)], [T(x)]$

“Cached quotients” method

After preprocessing, prover can compute

$$[Q_A(x)] = \sum_{i \in [N], A_i \neq 0} A_i \cdot [Q_i(x)]$$

“Cached quotients” method

After preprocessing, prover can compute

$$[Q_A(x)] = \sum_{i \in [N], A_i \neq 0} A_i \cdot [Q_i(x)]$$

Verifier can check:

$$e([A(x)], [T(x)]) = e([Q_A(x)], [Z_{\mathbb{V}}(x)])$$

“Cached quotients” method

After preprocessing, prover can compute

$$[Q_A(x)] = \sum_{i \in [N], A_i \neq 0} A_i \cdot [Q_i(x)]$$

Verifier can check:

$$e([A(x)], [T(x)]) = e([Q_A(x)], [Z_{\mathbb{V}}(x)])$$

In algebraic group model[FKL] can prove this is sound.

Lookup protocols

Constraints vs Lookups

Example: Check $\mathbf{0 \leq x \leq 2^n - 1}$

Constraints vs Lookups

Example: Check $0 \leq x \leq 2^n - 1$

Constraint approach: $\mathcal{P}$ sends $x_0, \ldots, x_{n-1}$ Proves

- $\forall i, x_i \in \{0, 1\}$
- $\sum_i x_i 2^i = x$.

Constraints vs Lookups

Example: Check $0 \leq x \leq 2^n - 1$

Constraint approach: $\mathcal{P}$ sends $x_0, \ldots, x_{n-1}$ Proves

- $\forall i, x_i \in \{0, 1\}$
- $\sum_i x_i 2^i = x$.

Requires $n + 1$ "gates".

Lookup approach

Preprocess table $\mathbf{T} = \{\mathbf{0}, \ldots, \mathbf{2^n - 1}\}$ Devise protocol to check $x \in \mathbf{T}$.

Lookup approach

Preprocess table $\mathsf{T} = \{\mathbf{0}, \ldots, \mathbf{2^n - 1}\}$ Devise protocol to check $x \in \mathsf{T}$.

Thm-informal [Arya..plookup]: Check can be done in amortized $\mathbf{O}(\mathbf{1})$ constraints per check, when have $\mathbf{O}(|\mathsf{T}|)$ checks.

Lookup approach

Preprocess table $\mathsf{T} = \{\mathbf{0, \ldots, 2^n - 1}\}$ Devise protocol to check $x \in \mathsf{T}$.

Thm-informal [Arya..plookup]: Check can be done in amortized $\mathbf{O}(\mathbf{1})$ constraints per check, when have $\mathbf{O}(|\mathsf{T}|)$ checks.

Thm [Caulk..$\mathfrak{cq}$]: Can be done in $\mathbf{O}(\mathbf{1})$ constraints without need of amortization!

The question in polynomials:

Preprocessed: $\mathbb{V} \subset \mathbb{F}$ subgroup of size $\mathbf{N}$. $\mathbf{H} \subset \mathbb{F}$ subgroup of size $\mathfrak{n}$. $\mathbf{T} \in \mathbb{F}_{<\mathbf{N}}[\mathbf{X}]$.

The question in polynomials:

Preprocessed: $\mathbb{V} \subset \mathbb{F}$ subgroup of size $\mathbf{N}$. $\mathbf{H} \subset \mathbb{F}$ subgroup of size $\mathbf{n}$. $\mathbf{T} \in \mathbb{F}_{<\mathbf{N}}[\mathbf{X}]$.

Input: $\mathbf{f} \in \mathbb{F}_{<\mathbf{n}}[\mathbf{X}]$. $\mathbf{cm}(\mathbf{f})$ given to $\mathbf{V}$.

The question in polynomials:

Preprocessed: $\mathbb{V} \subset \mathbb{F}$ subgroup of size N. $H \subset \mathbb{F}$ subgroup of size n. $\mathsf{T} \in \mathbb{F}_{<N}[X]$.

Input: $f \in \mathbb{F}_{<n}[X]$. $\mathbf{cm}(f)$ given to V.

Want to convince V that $f|_H \subset \mathsf{T}|_{\mathbb{V}}$ in $O(n)$ prover operations.

Log-derivative approach:

Lemma[Haböck]: $f|_H \subset T|_{\mathbb{V}}$ if and only if there exists $m(X) \in \mathbb{F}_{<N}[X]$ s.t. as rational functions

$$\sum_{i \in [N]} \frac{m_i}{X + T_i} = \sum_{a \in H} \frac{1}{X + f(a)}$$

Log-derivative approach:

Lemma[Haböck]: $f|_H \subset T|_{\mathbb{V}}$ if and only if there exists $m(X) \in \mathbb{F}_{<N}[X]$ s.t. as rational functions

$$\sum_{i \in [N]} \frac{m_i}{X + T_i} = \sum_{a \in H} \frac{1}{X + f(a)}$$

Strategy: check this identity at random $\beta \in \mathbb{F}$*.*

Main prover task: Compute polynomial $A(X)$ that interpolates RHS on $\mathbb{V}$, and prove it correct:

$$A_i = \frac{m_i}{\beta + T_i}, \forall i \in [N]$$

Main prover task: Compute polynomial $A(X)$ that interpolates RHS on $\mathbb{V}$, and prove it correct:

$$A_i = \frac{m_i}{\beta + T_i}, \forall i \in [N]$$

Can be done via the “KZG shenanigans” we described before.

Main prover task: Compute polynomial $A(X)$ that interpolates RHS on $\mathbb{V}$, and prove it correct:

$$A_i = \frac{m_i}{\beta + T_i}, \forall i \in [N]$$

Can be done via the "KZG shenanigans" we described before.

Must compute $[Q_A(x)]$ where

$$A(X)(\beta + T(X)) - m(X) = Q_A(X) Z_{\mathbb{V}}(X).$$