UJ crypto course, Lecture 1: the KZG scheme

Ariel Gabizon
Zeta Function Technologies

April 18, 2024

1 The KZG polynomial commitment scheme - an informal intro

The idea of polynomial commitment schemes is that a prover $\mathbf{P}$ can send a short commitment to a large polynomial $f \in \mathbb{F}_{<d}[X]$; and later open it at a point $z \in \mathbb{F}$ chosen by the verifier. $\mathbf{P}$ can also construct a proof π that the value he sends is really $f(z)$ for the f he had in mind during commitment time.

2 The KZG commitment scheme:

Prequisites: Given integer security parameter λ. We assume access to

- Groups $\mathbb{G}$,$\mathbb{G}_t$ of prime order r written additively with $\lambda^{\omega(1)} < r \leq 2^\lambda$
- randomly chosen generator $g \in \mathbb{G}$ and generator $g_t \in \mathbb{G}_t$.
- Denote by $\mathbb{F}$ the field of size r. A bi-linear pairing function $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ such that $\forall a, b \in \mathbb{F}$, $e(a\cdot, b \cdot g) = g_t^{a \cdot b}$

  For $c \in \mathbb{F}$, we use the notation $[c] := c \cdot g$.

Remark 2.1. *How do we actually construct pairings?? They were first constructed by André Weil, where $\mathbb{G}$ is taken to be the degree zero divisor class group of an algebraic curve - which is the same as the so-called Jacobian of the curve. In crypto we always use elliptic curves which are a special case, and where this group is isomorphic to the curve itself! These pairings were part of Weil's proof of what's called the Riemann hypothesis for curves over finite fields/Algebraic function fields. Much later, Victor Miller showed a practical algorithm to compute them (the "Miller loop").*

2.1 The scheme:

- **setup:**: Generate $\mathsf{srs} = [1], [x], \ldots, [x^d]$,
  for random $x \in \mathbb{F}$.

- **Commitment to $f \in \mathbb{F}_{<d}[X]$:**
  $\mathsf{cm}(f) := [f(x)]$

- The $\mathsf{open}(\mathsf{cm}, z, s; f)$ protocol - proving $f(z) = s$
  1. **P** computes $h(X) := \frac{f(X)-f(i)}{X-i}$
  2. **P** sends $\pi = [h(x)]$.
  3. **V** outputs acc if and only if

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

3 The Algebraic Group Model

We wish to capture the notion of an Algebraic Adversary that can generate new group elements by doing "natural" operations on the set of group elements he already received.

The intuition is that when the discrete log is hard, the group elements look random to (an efficient) adversary, and thus there's no other way for him to produce "useful" group elements.

More formally,

Definition 3.1. *By an* SRS-based protocol *we mean a protocol between a prover* **P** *and verifier* **V** *such that at before the protocol begins a randomized procedure* setup *is run, returning a string* $\mathsf{srs} \in \mathbb{G}^n$.

In our protocols, by an algebraic adversary $\mathcal{A}$ *in an SRS-based protocol we mean a* $\mathsf{poly}(\lambda)$*-time algorithm which satisfies the following.*

- *Whenever* $\mathcal{A}$ *outputs an element* $A \in \mathbb{G}$*, it also outputs a vector* v *over* $\mathbb{F}$ *such that* $A = \sum_{i \in [n]} v_i \mathsf{srs}_i$.

4 Knowledge Soundness of the KZG scheme

We say a PCS has *Knowledge soundness in the Algebraic Group Model* if, there exists an efficient algorithm E such that any algebraic adversary $\mathcal{A}$ has probability at most $\mathsf{negl}(\lambda)$ to win the following game:

1. $\mathcal{A}$ outputs a commitment cm.
2. E outputs a polynomial $f \in \mathbb{F}_{<d}[X]$
3. $\mathcal{A}$ outputs $z, s \in \mathbb{F}, \pi \in \mathbb{G}$.
4. $\mathcal{A}$ takes part of **P** in $\mathsf{open}(\mathsf{cm}, z, s)$.

5. $\mathcal{A}$ wins iff

   (a) $\mathbf{V}$ outputs acc in the end of open.

   (b) $f(z) \neq s$.

We use the following extension of the discrete-log assumption:

Definition 4.1. *The Q-DLOG assumption for $\mathbb{G}$ says that: For any efficient algorithm A, the probability that given $[1], [x], \ldots, \left[x^Q\right]$ for random $x \in \mathbb{F}$ outputs x is $\mathsf{negl}(\lambda)$*

Lemma 4.2. *Assuming the d-DLOG for $\mathbb{G}$, KZG has knowledge soundness in the algebraic group model*

Proof. We must define the extractor E: When $\mathcal{A}$ outputs cm, since it's algebraic, it also outputs $f \in \mathbb{F}_{<d}[X]$ such that $\mathsf{cm} = [f(x)]$. E will simply output f. Let $\mathcal{A}$ be some efficient algebraic adversary.

We will describe an algorithm B following the game between E and $\mathcal{A}$ such that whenever $\mathcal{A}$ wins the game, B finds x!.

During open when $\mathcal{A}$ sends π, it also sends $h \in \mathbb{F}_{<d}[X]$ such that $\pi = [h(x)]$. Assume we're in the case that $\mathcal{A}$ won the game. This means that $\mathcal{A}$ sent z, s and π such that $\mathbf{V}(\mathsf{cm}, z, s, \pi) = \mathsf{acc}$. This means that $e(\mathsf{cm} - [s], [1]) = e(\pi, [x - z])$ In our case this is the same as $e([f(x)] - [s], [1]) = e([h(x)], [x - z])$ This implies

$$f(x) - s = h(x)(x - z)$$

Define the polynomial

$$p(X) := f(X) - s - h(X)(X - z)$$

We claim that $p(X)$ *can't* be the zero polynomial: If it was, in particular $p(z) = 0$, and then $f(z) - s = h(z)(z - z) = 0$, which means $f(z) = s$ - but $\mathcal{A}$ won the game so $f(z) \neq s$!

So $p(X)$ isn't the zero polynomial in the case that $\mathcal{A}$ won. We claim that in this case we can find x:

Note that we can compute the coefficients of p, since $\mathcal{A}$ sent the coefficients of f and h.

Since $\mathbf{V}$ accepting means that $p(x) = 0$. Thus, we can factor p, and x will be one of its roots. We can check for each root γ of p if $\gamma = x$ by checking $[\gamma] = [x]$.

□