Polynomial Commitments from Pairings, KZG

Alexander Lindenbaum

Notes are from Chapters 15 of [Tha22].

1 Cryptographic Background

Polynomial commitments from pairings is based on cryptographic assumptions, similarly to the random oracle model assumption that hash functions are collision resistant.

1.1 Decisional Diffie-Helman Assumption

Let $\mathbb{G}$ be a cyclic group with generator g. The Decisional Diffie-Helman (DDH) assumption states that, given g^a and g^b for a, b chosen uniformly and independently from $|\mathbb{G}|$, the value g^{ab} is computationally indistinguishable from a random group element. That is, it is hard for any polytime adversary to distinguish $\langle g, g^a, g^b, g^{ab}\rangle$ from $\langle g, g^a, g^b, g^c\rangle$ for a uniformly randomly chosen c, except with negligible probability.

If discrete logarithm was efficient in $\mathbb{G}$ then DDH assumption is false in that group. On input: $\langle g, g_1, g_2, g_3\rangle$, compute discrete logarithms a, b, c of g_1, g_2, g_3 in base g. Then check if $c = a \cdot b$. This succeeds with non-negligible probability. Hence, DDH is a stronger assumption than the hardness of DL.

Computational Diffie-Helman (CDH) is a close relative, stating that no efficient algorithm can compute g^{ab} from $\langle g, g^a, g^b\rangle$. This is a weaker assumption than DDH, because being able to solve CDH implies solving DDH. There are groups in which CDH is believed to hold but DDH does not.

1.2 Pairing-friendly Groups and Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_t$ be two cyclic groups of the same order. A map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ is *bilinear* if for all $u, v \in \mathbb{G}$ and $a, b \in \{0, \dots, |\mathbb{G}| - 1\}$, $e(u^a, v^b) = e(u, v)^{ab}$. e is also *non-degenerate* if it does not map all pairs in $\mathbb{G} \times \mathbb{G}$ to the identity element $1_{\mathbb{G}_t}$. If e is an efficiently computable, non-degenerate bilinear mapping, then e is called a *pairing*.

$\mathbb{G}$ and $\mathbb{G}_t$ are isomorphic, but not necessarily equivalent from a computational perspective. The elements of both groups could have different representations and are computed very differently.

Note that for any group $\mathbb{G}$, if $\mathbb{G}$ is equipped with an efficiently computable symmetric pairing, then DDH does not hold. This is because one can check if $e(g, g_3) = e(g_1, g_2)$. In the case where $g_3 = g^{a,b}$, this check will always pass by bilinearity. If g_3 is a random element, this check will almost always fail by non-degeneracy.

1.3 Intuition for Bilinear Maps

Additively homomorphic commitment schemes allow parties to perform addition "under commitments." Bilinear maps convey the same power, but for multiplicative-homomorphism. Let us think of $g^{m_i} \in \mathbb{G}$ as a commitment to m_i (if m_i is chosen at random, the commitment is computationally hiding if DL is hard in $\mathbb{G}$). The bilinear map always any party, given commitments g_1, g_2, g_3 to check whether the values m_1, m_2, m_3 inside the commitments satisfy $m_3 = m_1 \cdot m_2$. This is because $e(g^{m_1}, g^{m_2}) = e(g^{m_3}, g)$ if and only if $m_3 = m_1 \cdot m_2$.

The power to perform a single "multiplication check" of commited values is enough to obtain a polynomial commitment scheme! Why? Because for any degree-D univariate polynomial p, the assertion "$p(z) = v$" is equivalent to the assertion that there exists a polynomial w of degree $\leq D-1$ such that

$$p(X) - v = w(X) \cdot (X - z).$$

This equation can be probabilistically verified by evaluating the polynomials on both sides at a randomly chosen point τ. The committer can commit to p by sending a commitment c_3 to $m_3 = p(\tau)$, then convince a verifier of the equation by sending a commitment c_2 to $m_2 = w(\tau)$. If the verifier can compute a commitment c_1 to $m_1 = \tau - z$ on its own, then the verifier can use the bilinear map to check that $m_3 = m_1 \cdot m_2$. This approach works if the prover does not know τ, otherwise an adversary could choose a w such that the general polynomial equation does not hold, but only at the point τ. KZG: is a specification of this model of a polynomial commitment scheme.

2 KZG and Trusted Setups

KZG commitments are due to Kate, Zaverucha, and Goldberg in [KZG10]. The major advantage to KZG is that commitments and openings consist of only a constant number of group elements. Another advantage is that we can commit to several univariate polynomials at once! A downside is that it requires a *structured reference string* (SRS) that is as long as the number of coefficients in the polynomial being commited to. This string should be computed and made available to any party that wishes to commit to a polynomial, but is also a trapdoor and must be discarded for security. Such a scheme that requires hiding of an SRS operates in what is called a *trusted setup.*

Interestingly, Dory is transparant. Dory has longer pre-processing time, but no trapdoors are produced. So we have a tradeoff here.

2.1 A binding scheme

Let e be a bilinear map pairing groups $\mathbb{G}, \mathbb{G}_t$ of prime order p, and $g \in \mathbb{G}$ is a generator, and D an upper bound the degree of the polynomials we would like to support commitments to. The SRS is this: first, choose an integer $\tau \in \{1, \ldots, p-1\}$ at random. The SRS equals $\langle g, g^\tau, g^{\tau^2}, \ldots, g^{\tau^D}\rangle$. The value τ is a trapdoor and must be discarded, otherwise it breaks binding.

To commit a polynomial q over $\mathbb{F}_p$, the committer sends $c = g^{q(\tau)}$ (or claims to be equal to that). The committer does not know τ (τ is generated in the trusted setup) but is still able to compute $g^{q(\tau)}$ using the SRS and additive homomorphism. If

$$q(Z) = \sum_{i=0}^{D} c_i Z^i,$$

then

$$g^{q(\tau)} = \prod_{i=0}^{D} \left(g^{\tau^i}\right)^{c_i},$$

which the committer can compute.

To open the commitment at input $z \in \{0, \ldots p-1\}$, to some value v (to prove that $q(z) = v$), the committer computes a witness polynomial

$$w(X) = (q(X) - v)/(X - z),$$

and sends a value y claimed to equal $g^{w(\tau)}$. Again, $g^{w(\tau)}$ can be computed since w has degree $\leq D$. The verifier will check that

$$e(c \cdot g^{-v}, g) = e(y, g^\tau \cdot g^{-z}).$$

Analysis and Correctness If the committer is honest, then we have

$$e(g^{q(\tau)-v}, g) \stackrel{?}{=} e(g^{w\tau}, g^{\tau - z}),$$

which is true by bilinearity. For binding, suppose $q(z) \neq v$. Passing the verifier's check requires computing $g^{w(\tau)}$ where $w(X) = (q(X) - v)/(X - z)$ which is now not a polynomial in X. The intuition is that the SRS gives enough information to compute any degree $\leq D$ polynomial at τ, which is not enough information to compute a rational function where you divide by $\tau - z$.

Let's make this intuition formal. We need a cryptographic assumption, namely the *D-strong Diffie-Hellman (SDH) assumption.* Given the SRS that consists of the generator of g raised to all powers of τ up to power-D, there is no efficient algorithm $\mathcal{A}$ that outputs a pair $(z, g^{1/(\tau - z)})$ except with negligible probability. The SDH assumption in $\mathbb{G}$ implies DL is hard in $\mathbb{G}$, because if discrete logs are easy to compute then τ can be efficiently computed from g^τ, then $g^{1/(\tau - z)}$ is easy to compute. So we believe that the SDH assumption holds.

If binding was not true and a committer can compute $g^{w(\tau)} = g^{((\tau - v)/(\tau - z)}$, then they break SDH by subtracting exponents through division.

2.2 Extractable Schemes

An *extractable* polynomial commitment scheme is one in which for every efficient committer adversary $\mathcal{A}$, there is an efficient algorithm E (depending on $\mathcal{A}$) that produces a degree-D polynomial p explaining all of $\mathcal{A}$'s answers to evaluation queires. In such a scheme, if $\mathcal{A}$ is able to successfylly answer evaluation query z with v, then $p(z) = v$. Since E is efficient, it cannot know anything more than $\mathcal{A}$ does, and E knows p by outputting it. Then $\mathcal{A}$ must "know" a polynomial p that $\mathcal{A}$ is using to answer evaluation queries.

Once the prover sends a KZG commitment, it is bound to some function. This means that for each input query z there is at most one v that the committer can successfully answer the query with. But it does not establish that the function bound to is a degree-D polynomial at all. To make this polynomial commitment scheme extractable rather than just binding, it must be modified. Here is one way:

In the modified scheme, the SRS is doubled in size. For τ and α chosen at random from $\mathbb{F}_p$, the SRS consists of

$$\langle (g, g^{\alpha}), (g^{\tau}, g^{\alpha\tau}), (g^{\tau^2}, g^{\alpha\tau^2}), \ldots, (g^{\tau^D}, g^{\alpha\tau^D}) \rangle,$$

adding exponents also raised to the power of α.

The *Power Knowledge of Exponent* (PKoE) assumption states that for any polytime $\mathcal{A}$ given access to this SRS, whenever the algorithm outputs two group elements $g_1, g_2 \in \mathbb{G}$ such that $g_2 = g_1^{\alpha}$, the algorithm must "know" coefficients $c_1, \ldots, c_D$ such that $g_2 = \prod_{i=1}^{D} g^{c_i \cdot \tau^i}$. The idea is that it is easy to compute (g_1, g_2) with $g_2 = g_1^{\alpha}$ by letting g_1 equal any product of quantities in the first half of the SRS raised to constant powers,

$$g_1 = \prod_{i=0}^{D} g^{c_i \tau^i},$$

and let g_2 be the results of applying the same operations to the second half of the SRS. But PKoE says that this is the only way an efficient party is capable of computing such a pair.

Now, the commitment to q is the pair $(g^{q(\tau)}, g^{\alpha \cdot q(\tau)})$. To open a commitment $c = (U, V)$ at $z \in \mathbb{F}_p$ to value y, committer computes $w(X) = (q(X) - v)/(X - z)$ and sends $w(\tau)$. The verifier checks two equations:

$$e(U \cdot g^{-v}, g) = e(y, g^{\tau} \cdot g^{-z}),$$
$$e(U, g^{\alpha}) = e(V, g).$$

Extractability follows from that if there's another commitment an adversary could send, that gives another efficient algorithm for producing a pair (g_1, g_2).

Thaler writes that assumptions like PKoE which are not based in falsifiability are controversial. Yet researchers and practitioners are confident that it holds, and use it for SNARK schemes.

References

[KZG10] Aniket Kate, Gregory M Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In *Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16*, pages 177–194. Springer, 2010.

[Tha22] Justin Thaler. Proofs, arguments, and zero-knowledge. *Foundations and Trends in Privacy and Security*, 4(2–4):117–660, 2022.