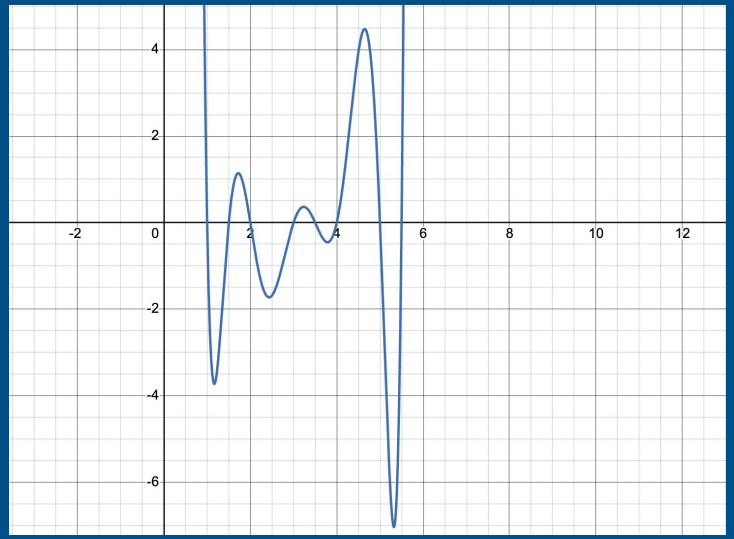
January 2025

Succinct Zero-Knowledge Proofs

Using Polynomial Commitments



Elizabeth van Oorschot

1. Zero-knowledge proofs
2. Commitments
3. KZG Polynomial commitment
4. A simple example
5. PLONK & applications

What is a zero-knowledge proof?

The verifier learns no new information while interacting with the prover

To formalize this, we use a simulator

Completeness

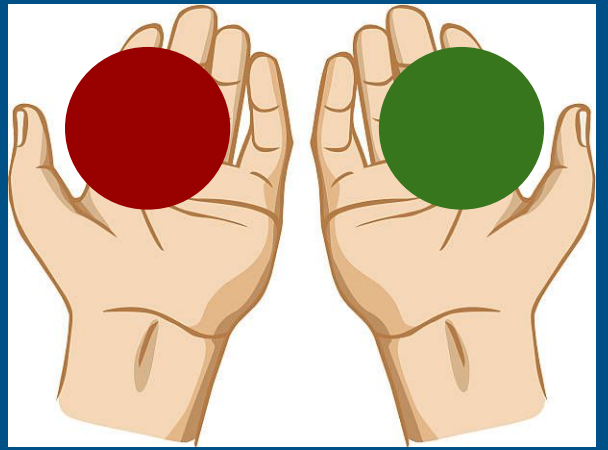
For any statement for which the relation holds, we can produce a proof of it that is accepted with probability 1

Soundness

For any statement for which the relation does not hold, no proof can be produced that is accepted with greater than negligible probability

Is this possible?

Let’s see an example.



https://www.istockphoto.com/illustrations/two-hands-open

zk-SNARK

zero-knowledge Succinct Non-interactive ARgument of Knowledge

Succinctness

Sub linear proof size and verifier time

→ comes at cost of increased prover time

Non-interactive

Can we get rid of the verifier sending challenges?

→ Fiat-Shamir transform

Fiat, Shamir 1987

Commitments

Like putting a bit into an envelope and sealing it, to be opened at a future date

Binding

The commitment can only be opened to a single value

Hiding

Before being opened, no information about the value committed to can be obtained

Feldman Commitments

Consider committing to z with g^z (g a generator of a finite group)

Open by publishing z

Unconditionally binding, computationally hiding (if the discrete log problem is hard in the group)

→ but what if we commit to the same value twice?

Pedersen Commitments

Instead, use $g^z h^r$ with $h = g^y$ some unknown y and r random

Open by publishing z and r

→ random r prevents it from being deterministic

→ $h = g^y$ preserve binding

Computationally binding, unconditionally hiding

Kate, Zaverucha, Goldberg 2010

KZG polynomial commitments

How can we commit to some polynomial $p(x)$?

Commit to $p(\tau)$

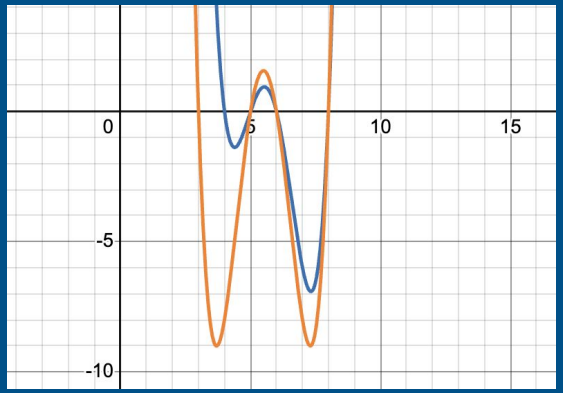
So the commitment is $g^{p(\tau)}$

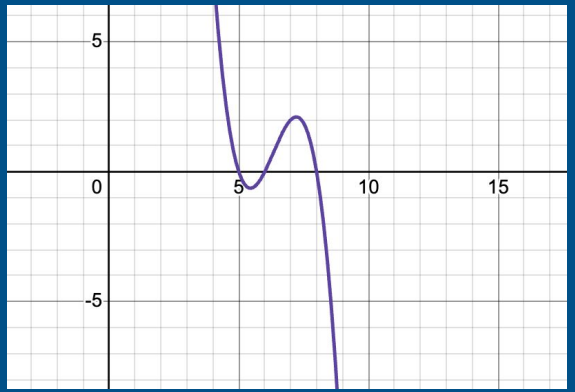
Is this enough information to commit to a polynomial?

→ Schwartz-Zippel lemma ⇒ if 2 polynomials are equal at a random point they are equal with probability at least $1 - \frac{d}{|\mathcal{S}|}$

Hiding comes from the discrete log assumption and binding from the t-Strong Diffie-Hellman assumption

… so long as no one knows τ





Kate, Zaverucha, Goldberg 2010

Set-up

A Structure Random String (SRS) must be generated and public:

$$\langle g^{(\tau^0)}, g^{(\tau^1)}, g^{(\tau^2)}, g^{(\tau^3)}, \ldots, g^{(\tau^d)} \rangle$$

If anyone knows τ, then they can break the binding and hiding so we require a trusted set-up

→ this can be distributed across multiple participants such that only one being honest ensures secrecy of τ

Commit

Now, to commit to a polynomial simply raise each element in the SRS to the corresponding coefficient of p(x)

And multiply all the terms together to get $g^{p(\tau)}$

Publish this (single group element!) as the commitment

Kate, Zaverucha, Goldberg 2010

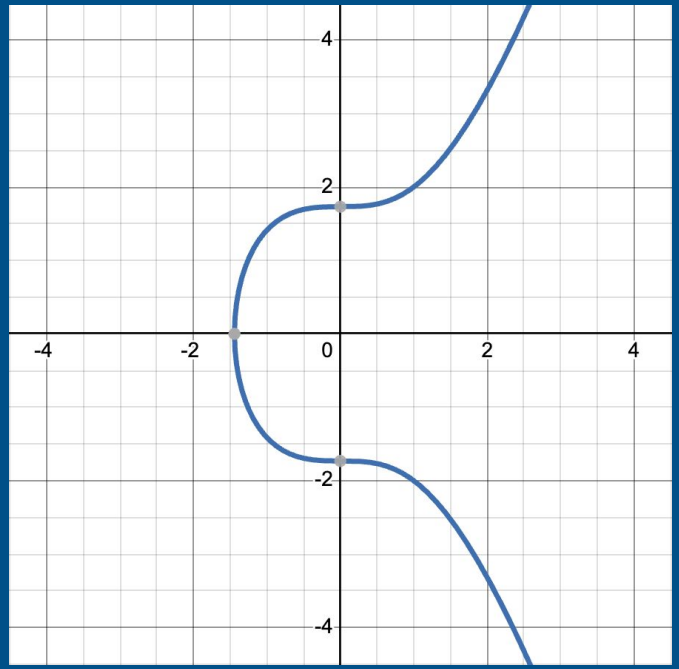
Homomorphic Properties

Addition: $g^{p(\tau)}\, g^{q(\tau)} = g^{p(\tau)\ +\ q(\tau)}$

Multiplication?

$$e : G_1 \times G_2 \rightarrow G$$

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



elliptic curve BN254

Kate, Zaverucha, Goldberg 2010

Opening at a root

If z is a root of p(x), then $Q(x) = \frac{p(x)}{x - z}$ is polynomial

So open at a root by publishing z and a commitment to Q(x) = $g^{Q(\tau)}$

→ prover wants to check:

$$p(\tau) = Q(\tau)(\tau - z)$$

→ and does so using the commitments:

$$e(g^{p(\tau)}, g) \stackrel{?}{=} e(g^{Q(\tau)}, g^{\tau - z})$$

Opening at an arbitrary point

Define s(x) = p(x) - z and are the previous method to prove s(x) has a root at z

Kate, Zaverucha, Goldberg 2010

One more note

What we have so far is like the Feldman commitment

As before, we can get unconditional hiding (like the Pedersen commitment)

An example: entry-wise addition of two array

We show $Arr_1[i] + Arr_2[i] = Arr_3[i]$ for all i

Polynomial representations

We interpolate the arrays so that the entries are encoded as the y-coordinates of univariate polynomials with x-coordinates as a multiplicative group of order κ with generator ω

We want to prove:

$$\mathsf{Arr}_3[i] = \mathsf{Arr}_1[i] + \mathsf{Arr}_2[i] \text{ for } i \text{ from 0 to } n-1$$

In polynomials:

$$\text{For all } X \text{ from } \omega^0 \text{ to } \omega^{\kappa-1}\text{: } \mathsf{Poly}_{\mathsf{Arr}_3}(X) = \mathsf{Poly}_{\mathsf{Arr}_1}(X) + \mathsf{Poly}_{\mathsf{Arr}_2}(X)$$

We rearrange to equal zero:

$$\mathsf{Poly}_{\mathsf{Vanish}}(X) = \mathsf{Poly}_{\mathsf{Arr}_1}(X) + \mathsf{Poly}_{\mathsf{Arr}_2}(X) - \mathsf{Poly}_{\mathsf{Arr}_3}(X) = 0$$

van Oorschot, Deng, Clark 2024

Polynomial representations

The previous equation holds for all x in our multiplication group (but not outside of it)

To show this, we define another polynomial:.

$$Q(X) = \frac{\mathsf{Poly}_{\mathsf{Vanish}}(X)}{X^{\kappa}-1}$$

The denominator is the minimal vanishing polynomial on multiplication group, so if it cleanly divides the numerator, the numerator must also vanish on the multiplicative group

By rearranging, we get:

$$\mathsf{Poly}_{\mathsf{Zero}}(X) = \mathsf{Poly}_{\mathsf{Vanish}}(X) - Q(X) \cdot (X^{\kappa} - 1) = 0$$

a polynomial that is zero on the whole domain. By proving that this is the zero polynomial, we prove the desired relation.

van Oorschot, Deng, Clark 2024

Working with Commitments

The prover publishes a KZG commitment to the polynomials for each array, as well as Q(x)

Then they generate a random challenge, ζ, (by Fiat-Shamir transform and hashing) and publish ζ, as well as an opening at this point for each of the 4 polynomials

To verify the proof, the verifier computers the following:

$$Y_{\mathsf{Vanish}} = \mathsf{Poly}_{\mathsf{Arr}_1}(\zeta) + \mathsf{Poly}_{\mathsf{Arr}_2}(\zeta) - \mathsf{Poly}_{\mathsf{Arr}_3}(\zeta)$$

$$Y_{\mathsf{Zero}} = Y_{\mathsf{Vanish}} \ - Q(\zeta) \cdot (\zeta^{\kappa} - 1)$$

And to verify the constraints hold, the prover checks

$$Y_{\mathsf{Zero}} \overset{?}{=} 0$$

And if this holds then with overwhelming probability (by Schwartz-Zippel lemma) the prover can be confident that Arr1 + Arr2 = Arr3

van Oorschot, Deng, Clark 2024

PLONK

Proving the evaluation of an arithmetic circuit

& Applications

A lot of current interest is fueled by blockchain applications

→ zk-roll ups, smart contracts, proof of solvency

But lots of broader uses too!

→ computational integrity

→ voting systems, auctions

→ authentication

Gabizon, Williamson, Ciobotaru 2024

References

Lindell, 2021

How To Simulate It – A Tutorial on the Simulation Proof Technique

https://eprint.iacr.org/2016/046.pdf

Fiat, Shamir, 1987

How to Prove Yourself: Practical Solutions to Identification and Signature Problems

Lecture Notes in Computer Science. Vol. 263. Springer Berlin Heidelberg. pp. 186–194

Kate, Zaverucha, Goldberg, 2010

Constant-Size Commitments to Polynomials and Their Applications

https://www.iacr.org/archive/asiacrypt2010/6477178/6477178.pdf

van Oorschot, Deng, Clark, 2024

Plonkbook: A Handbook for Poly-IOP Gadgets

https://www.plonkbook.org

Gabizon, Williamson, Ciobotaru, 2024

PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge

https://eprint.iacr.org/2019/953.pdf