TL; DR

: Polynomial commitment schemes can provide proof of custody-like schemes that prove custody of data in a non-interactive way.

Background

In order to avoid the “honest but lazy” validator problem, validators have to compute a “proof of custody” that shows that they actually possess a copy of the data that is being signed. The current proof of custody construction mixes an ephemeral secret only known to the validator with the data and computes one bit. Later, validators have to reveal their ephemeral secret and everyone can check the custody bit was computed correctly.

This construction has two disadvantages that can be addressed using pairing-based polynomial commitment schemes:

1. It is interactive, requiring an on-chain challenge game
2. It is not aggregatable, so proof size is proportional to the number of validators in an attestation

Number 2 is not a huge problem as attestations only grow one bit per validator.

One scheme that does not have both of these disadvantages can be constructed using recursive zk-SNARKS. However this is currently not practical due to any such system requiring millions of constraints, so it is extremely expensive. As shown below, using the lighter machinery of polynomial commitments we can create a system that is still expensive, but can actually be computed within an epoch and could be very well doable given an elliptic curve operation ASIC that is currently in discussion (the most important component in a SNARK ASIC).

Outline

I will present two schemes. The first is based on the well known Kate commitment scheme [1]. Its main problem is that it is not aggregatable. Scheme two derives from the “balancing of polynomials” idea that several polynomial commitment schemes (and also SNARKs) use, but is not directly related to a polynomial commitment scheme and only really useful for the proof of custody. It is aggregatable.

Bot schemes do not completely prevent outsourcing, however while the cost to the secret owner is O(n)

elliptic curve multiplications where n

is the size of the data, the cost to an outsourcing provider would be O(n^2)

, which makes it very impractical to outsource.

Kate commitments

The Kate commitment scheme is a way to commit to a polynomial f(x)=\sum_{i=0}^n f_i x^i

and decommitting a value y=f(r)

(i.e. giving y

and a witness that proves that it is the evaluation of the previously committed polynomial at r

). The witness is a single group element.

Let e: G_1 \times G_2 \rightarrow G_t

be a pairing of two elliptic curves G_1

and G_2

with generators G \in G_1

and H \in G_2

(I use additive notation for group operations). We need a trusted setup that provides s^i G

for i=0,\ldots, n

and s H

where n

is the degree of the polynomial to be committed to and s

is a random value not known to anyone. Then a commitment to a polynomial consists of

\displaystyle C = \sum_{i=0}^n f_i s^i G \text{ .}

To evaluate this polynomial at x=r

, compute the polynomial

\displaystyle h(x) = \frac{f(x) - f(r)}{x-r} = \sum_{i=0}^n h_i x^i \text{ .}

The witness for the evaluation f(r) = y

then consists of

\displaystyle \Pi = \sum_{i=0}^n h_i s^i G

which can be checked using the pairing equation

\displaystyle e(\Pi, (s - r) H) = e(C - y G, H) \text{ .}

The first scheme for a non-interactive proof of custody simply uses a Kate commitment: Let f(x)

be a polynomial that represents the data for which the proof of custody should be computed, for example by setting f_i

to the data blocks. Then the validator uses their validator secret for r

. Note that neither r

nor y=f(r)

have to be published in order for this to work, but only r H

and y G

, making this scheme perfectly safe for the validator. Also both r

and the complete data are needed to compute the proof of custody efficiently (however, see the section on outsourcing on how an outsourcing scheme is possible that uses n^2

instead of n

elliptic curve multiplications).

This scheme is based on the well-known Kate commitment scheme and proves that a non-interactive proof of custody is possible without the heavy machinery of full-blown pairing-based zk-SNARKS. However, it is not yet aggregatable, and the pairing on the left-hand side of the pairing equation makes aggregation impossible as it has a validator-specific term on each side of the pairing. Scheme two below is a somewhat more experimental idea that is aggregatable.

Scheme two

This is not directly a polynomial commitment scheme, but based on similar ideas. Let’s assume that our trusted setup consists of s^i G

for i=0,\ldots, n

and s^i H

for i=0,\ldots, 2n

and in addition to their public key rH

validators commit to a “multiexponential public key” M_r = \sum_{i=0}^n r^i s^i H

(see below on how such a commitment can be checked). Commit to the polynomial f

using C

as before. Then define the proof of custody as

\displaystyle \Pi' = \sum_{i=0}^{2n}\left(\sum_{j+k=i} r^j f_k \right)s^i H

which can be checked using the pairing equation

\displaystyle e(C, M_r) = e(G, \Pi') \text{.}

Note that all the pairing equations are linear in the validator-specific terms, namely M_r

and P

. So this proof of custody is aggregatable, meaning it is constant sized (however the constant is one elliptic group element, which is quite substantial).

Commiting to the “multiexponential public key”

As a precondition for being able to use this scheme, we need a way for a validator to prove that their commitment M_r = \sum{i=0}^n r^i s^i H

is correct, i.e. the coefficients are indeed the powers of their private key. It can be checked using the geometric series identity by additionally providing rsG

and r^{n+1} G

. Then

\displaystyle e(G - rsG, M_r) = e(G, H) \cdot e(-r^{n+1}, s^{n+1} H)

because

\displaystyle (1 - rs) \sum_{i=0}^n r^i s^i = 1 - r^{n+1} s^{n+1} \text{ .}

By comparing coefficients in s^n

it can be shown that this is sufficient to prove M_r = \sum_{i=0}^n r^i s^i H

, even if the validator is allowed to cheat when providing r^{n+1} G

(they have no other option to make the equation work).

Trusted setup

One obvious disadvantage compared to interactive proof of custody schemes is the trusted setup that is required. However, a nice property of both schemes is that the only required trusted setup are monomial terms s^i G

and s^i H

. This means that the “updateable” setup that is used for Sonic SNARKs is applicable [2].

Outsourcing

The idea of the proof of custody is to make outsourcing difficult without giving away some secret that makes the validator slashable. As we want to make the proof non-interactive, we are directly using the validator secret and not some ephemeral secret, as in the interactive schemes. Actually both of these schemes suffer from one obvious way to oursource them: A validator can give away r^j s^i G

for 0\leq i, j \leq n

. This allows anyone to compute the proof of custody in their behalf. However, in order to do this, n^2

elliptic group multiplications have to be performed, versus just n

for a validator computing it using their own secret. Outsourcing is thus prohibitively expensive using this scheme.

Elliptic curve multiplication ASICs can probably make these schemes practital.

Performance

The dominant component of scheme one, using Kate commitments, is elliptic group operations. Computing the commitment

and proof will cost 2n

elliptic curve multiplications, and with each BLS12-381 multiplication costing ca. 1 ms, at n=2^{16}

will cost 120 s.

Scheme two involves 3n

elliptic curve multiplications, however, there are also n^2

curve-order field multiplications necessary. These are much cheaper than EC multiplications but will probably dominate at this order. It is probably not practical at this point.

MPC-friendliness

I don't know yet how to make either scheme efficiently computable if r

is Shamir-shared. However, if it is additively shared, there is a way to allow this in the second scheme due to it being aggregatable, by allowing a validator to commit to several "multiexponential public keys" M_{r_0}, \ldots, M_{r_m}

where r = \sum_{i=0}^m r_i

. Then the proof of custody would be checked against \sum_i M_{r_i}

rather than M_r

.

Possible application to data availability proofs

One interesting consequence of polynomial commitment schemes is that they make it very easy to prove that a polynomial is of a low degree. Indeed by limiting the number of polynomials in the trusted setup, it can be made impossible to commit to a higher-degree polynomial at all. This can potentially be used for data availability proofs [3].

[1] https://www.iacr.org/archive/asiacrypt2010/6477178/6477178.pdf

[2] https://eprint.iacr.org/2019/099.pdf

[3] https://arxiv.org/pdf/1809.09044.pdf