Background

The security of sharding depends on “windback”. When validators extend the collation header chain for a given shard, windback means downloading and verifying a few collation bodies from the immediate past. Vitalik estimated that we’d want a windback length of about 25. (Side question: why 25?)

SPV mining (“blind” mining) is basically when there’s little to no windback. It is bad (c.f. the BIP66 Bitcoin incident), and particularly so in the context of sharding.

Slightly more abstractly winding back means checking two things:

1. Collation (real-time) data availability

: Usually checked (locally with respect to your network access) by attempting to download to relevant collation bodies.

1. Collation validity

: Usually checked (locally with respect to your consensus rules) by executing the transactions in the collation body.

The above two checks are essential to security but seem to go against scalability. The reason is that downloading collation bodies and executing transactions both scale linearly

, whereas we need something sublinear for scalability.

Enforcing validity

SNARKs/STARKs solve (in theory at least) the validity problem. By requiring validators to publish a succint and quick to verify proof of validity in the collation header, validity can be enforced sublinearly. This is very neat, but does not

help with data availability. Collation bodies from the immediate past can simultaneously be valid and unavailable. Even immediate block creation can be outsourced to a third party (e.g. a briber) and the newly-created block withheld from the validator.

Enforcing availability

Enforcing availability is less trivial. One approach is “proofs of custody”. A proof of custody is a cryptographic guarantee that some identity (in our case, a validator) had full access to some piece of data (identified by its hash) when the proof was produced. (Signatures schemes don’t usually work because they are based on digests which are usually outsourceable.)

Assuming the existence of proofs of custody, we can require validators to publish a proof of custody for the concatenation of all the collation bodies in the windback period, which would prove availability.

Some progress was made on proofs of custody in a paper by Pavel Kravchenko and Vlad Zamfir. Their construction does not work for our purposes because the proof size is linear (and concretely too large), and the proof is interactive (although they note it can be made non-interactive with the Fiat-Shamir heuristic).

Proof of custody

We present below a new proof of custody. It uses SNARKs/STARKs and is non-interactive, constant space, and constant time to verify.

Let B

be the collation bodies for which a validator needs to prove direct (non-outsourceable) custody. Let P

be the validator’s private key. Split B

into an array of small chunks (e.g. 32-byte chunks) so that B

is the concatenation of B[0], ..., B[n]

.

The idea is to build a digest D

for B

which heavily “incorporates” P

. Because P

cannot be shared without compromising the validator’s deposit, computing D

is non-outsourceable. The specific digest construction we suggest is D = \textrm{SHA3}(B[0]\oplus P) \hspace{1mm}\oplus\hspace{1mm} ... \hspace{1mm}\oplus\hspace{1mm} \textrm{SHA3}(B[n] \oplus P)

.

To prove availability, the following is included in the collation header:

1. The digest D
2. A SNARK/STARK that proves D

faithfully corresponds to B

and P

(using zero-knowledge for privacy of P

)