This post continues the exploration of a core theme in the sharding category around the separation of ordered data (called “logs”) and state. We detail a protocol-level collation proposal mechanism where logs benefit from short-term private lookbehind. Short-term private lookbehind is meant to be a general approach to curb short-term adaptive attacks (censorship, collusion, bribes, blackmail, discouragements, …).

Progress has been made designing private proposal mechanisms (see here for a scheme with private lookbehind). As noted by @iddo the privacy of these mechanisms extends only as far as the collation header, not the collation body

:

Cryptoeconomic "ring signatures"

attackers can always censor the block that was created if they don’t like the contents of the block

We seek to address that concern.

Construction

For the purpose of simplicity we assume log shards where collation bodies consist fully of logs (the scheme extends more generally). To propose a collation body B

a validator does the following:

1. Keeps B

secret and broadcasts the time-lock encryption E(B)

of B

with a time-lock set to LOG_LOOKBEHIND_PERIODS

1. Includes a short zk-proof in the collation header that E(B)

is the faithful time-lock encryption of a collation body B

that produces a collation root matching the collation header

(An alternative to time-lock encryption is to have a cryptoeconomic scheme where the validator is highly incentivised to make the decryption key available onchain within a certain time period, similar to the fair exchange without a third party construction.)

Discussion

The above construction allows for logs to benefit from short-term lookbehind privacy, i.e. the content of the logs is not immediately publicly disclosed. (Notice that users may have immediate private knowledge of the inclusion of the logs they care about, especially in the context where logs can be included in exchange for out-of-band compensation.)

By their nature, adaptive attacks may be a cat-and-mouse game between attackers and defenders (like ad blocking). Individual applications can setup their own mitigations, but having a global strategy feels

like a big step in favour of defenders. Combining private lookbehind for both collation headers and collation bodies feels effective if LOG_LOOKBEHIND_PERIODS

is set large enough to cover the time to reach some decent level of finality.

Notice the construction only naturally applies to logs (as opposed to transactions) because of the tight state coupling between transactions.