tl;dr

Computational requirements for ZK provers can lead to centralization of ZK rollups, and make them

vulnerable to censorship, in particular, from the government.

This can be avoided by asking users to do ZK proofs on transactions they submit.

Unfortunately when a user submits a transaction, transaction ordering and state roots are not yet known. This can be addressed by asking the users to do proofs on previously ordered transactions, and paying them in ZKPOW token, which can later used to submit transactions to the chain.

Description

A decentralized crowdsourced ZK rollup can consists of the following components:

- a decentralized sequencer, which is itself a PoS EVM chain, which has an additional feature of the ZKPoW token described below
- submission agents that post ordered transactions from the sequencer to the ETH main net. Each node in the sequencer chain can also be a submission agent. Each submission agent can be allocated a time slot to post to the mainnet.
- user browser plugins performing ZK proofs

The transaction processing then works as follows:

- A user submits a transaction to the sequencer, paying gas fees in ZKPoW token
- The transaction is ordered, committed to the chain and processed by the EVM in the usual way. At this point the ZK proof does not yet exist, but before/after state roots are already known
- The unproven committed transaction is then assigned psueudorandomly to one of active browser plugins for processing. The plugin will perform the computation and submit the proof back to the sequencer chain, so that it can get included on chain.
- For every successful computation, the user plugin is paid in ZKPOW.
- If the proof is not submitted within the required period of time, the user is penalized a fixed amount of ZKPoW. This can include going negative on user ZKPoW balance. Another plugin is then assigned to the computation.
- user plugins can mark themselves active/inactive by issuing heart beat transactions on the sequencer chain. Only active plugins are assigned work and penalized. The ZKPoW bounty can increase if the number of active plugins drops.
- In order to prevent the sequencer from getting stuck, the bounty paid for a given transaction will increase with time the transaction stays unproven.
- When there are enough contiguous proven transactions to form the next sequencer block, the proofs are aggregated in a similarly crowdsourced fashion
- when a fully proven block is formed on the sequencer, it is submitted to Eth mainnet by a submission agent