Proof of Efficiency

David Schwartz and Jordi Baylina, Polygon Hermez

A new consensus mechanism for zk-rollups

We at Polygon Hermez are currently working on the zkEVM implementation, and this challenge has required us to research and develop a new consensus mechanism for a decentralized L2 protocol.

It leverages the experience of the existing Proof-of-Donation in v1.0, designed to build the first decentralized zk-rollup and support the permissionless participation of multiple coordinators in order to produce batches in L2.

We are still considering several options and improving this protocol for v2.0 (zkEVM), but we are happy to share our ideas with the community and receive feedback.

Background

In zk-rollups, the challenge of decentralizing is huge and it has been difficult to find a good solution so far. This happens because protocols such as PoS have some issues on L2 and there's a need to get production of zk validity proofs (which are very computation intensive for the prover) with high performance so the network can keep its service level. Assigning the right to produce a batch (L2 block) to any random validator does not guarantee that.

Proof of donation/Proof of Burn (PoD/PoB) is based on a decentralized auction model to get the right to produce batches in a specific timeframe. In this case, the economic incentives are set up so the validators need to be very efficient in order to be competitive, and it represents a big improvement.

One problem with this model is that for a specific time, the network is controlled by a single actor which could potentially be malicious and even if there are ways to mitigate the impact, it's difficult to avoid zero impact on the service level, especially on the bootstrapping phases.

On the other hand, the auction protocol is very costly and complex for coordinators/ validators, while at the same time only the most effective will be rewarded. It's difficult to automate for them and the complexity of the predictions is high, since the auction requires bidding some time in advance.

Another problem with the previous protocol is the effectiveness for selecting "the best" operator that converges to a "winner takes it all" model. This doesn't allow operators with slightly less performance to compete. The consequence is that operators in control of the network became very centralized with the censorship resistance limitations that this situation produces.

New requirements

The new protocol aims to cover the key properties that such a consensus model for a L2 zk-rollup requires:

- Permissionless access to produce L2 batches
- Efficiency as key to network performance
- Avoid control from any single party
- Protection from malicious attacks
- Total validation effort proportional to the value in the network

Proof-of-Efficiency (PoE) model

The protocol of creation of batches consists of a two-step model that splits activities between different parties. The first party to participate is the Sequencer and the second one is the Aggregator.

Sequencers

In this model, the sequencers are parties that collect L2 transactions from users and so they select and pre-process a new L2 batch in the network by sending a L1 TX with the data of all selected L2 TXs. Anyone can be a sequencer, it's a permissionless role consisting of a gateway to the network.

The interesting thing is that these proposed batches will be recorded in a L1 transaction for a zk-rollup model (or in a different data availability network in the case of a Validium).

This batch proposal happens when the sequencers decide to do so based on the incentives they have:

- one potential being the economic value of transactions in their pools.
- or the service level that they need to fulfill with their users (fees could vary accordingly since they will be requested by the sequencer).

To propose a new batch to the network, sequencers will need to pay the gas fees of L1 network to produce a TX with all the batch transactions data, and the protocol defines an additional fee in $MATIC token that will need to be deposited. This way, there is an incentive for sequencers to propose valid batches with valid transactions.

The batch fee will be variable depending on the network load, this will be calculated from a parameter called , automated from the protocol smart contract.

The batches, in the format of L1 transactions with information in the CALLDATA, will be used as the data availability for the L2 network and any new permissionless node will be able to synchronize the state, reconstructed also from this information.

Once mined, these data availability L1 transactions define the L2 TXs that will be executed and the specific order. This creates a deterministic new state, which can be computed as a virtual future state by network nodes.

Of course, this new state will be settled when the validity proof of a new state (the ZKP) is generated and mined in L1. This corresponds to the second part of the protocol.

Aggregators

Of course one of the main advantages of the zk-rollups is the fast finality of transactions that validity proofs provide. This protocol tries to enhance the effectiveness of these proofs.

The aggregators are the parties that participate in a permissionless way in the consensus protocol of Proof-of-Efficiency.

In this mechanism, the right to create the validity proof of a new state of the L2 (and of course, collect part of the fees in the txs) is earned simply by being the first aggregator to do it.

It works in the following way: the batches proposed by the Sequencers in L1 are sorted by their appearing position in the L1 and contain the transaction data. The PoE smart contract will accept as valid the first validity proof that updates to a new valid state including one or more of the proposed batches.

The aggregators need to define their objectives to trigger proof generation and run the race based on their own strategy.

For example, if there are batches with few TXs included, some aggregators may find it not interesting to produce a proof until there is more value and produce a proof that includes the change of state of N proposed batches. Other aggregators may have a different strategy.

For the aggregators that are late to the race, the smart contract will execute with a Revert if the proof sent is not proposing a new state, checked with the merkle tree hash of the overall state database. So, not being the first comes at the cost of proof generation but most of the gas fees are recovered.

Of course, the proof will exist only if the aggregator has processed the proposed batches correctly, meaning they have an order and all

of them need to be processed. It’s a mechanism similar to the “Force tx” implemented in Polygon Hermez v1.0, in that case useful to avoid censorship.

This mechanism avoids control of a single party and many of the potential attacks, since any Sequencer can propose a batch, but there is a cost on it.

And the Aggregators have the option to participate in a permissionless manner too, but if they don’t do it, there will be a moment when the economic value will be interesting for someone to do it.

In our case, the Polygon Hermez network would launch a Boot Aggregator backing up that there is a new validity proof at a specific frequency during the bootstrapping phase.

Fees will be distributed in the following way:

- Fees from L2 TXs will be processed and distributed by the same aggregator that creates the validity proofs.
- All TXs fees will be sent to the corresponding Sequencer of each batch.
- The deposited fees from Sequencers to create a batch will be sent to the Aggregator that included this batch into a validity proof.

Conclusions

The PoE consensus mechanism is ideated to solve some of the challenges of decentralized and permissionless validators in L2 for zk-rollups.

It defines a two-step model where it enables:

- Permissionless Sequencers as benefited participants in the protocol, also as a source of scalability of the network.
- A data availability model perfectly compatible with Volition (zk-rollup and Validium) schemas which could enable different tiers of service for users.
- The calculation of a “virtual” state from the data availability and a “final” state based on the validity proofs. This architecture can save a lot of cost for decentralized zk-rollups by settling validity proofs frequency based on different criteria, but not as the only solution to confirm transactions.
- Space for permissionless Aggregators as the agents to perform the specialized task of cryptographic proof generation, expected to be costly for zkEVM protocols. It provides a very simple and straightforward model for them to manage their incentives and returns.
- Native protection against L2 network problems such as attacks from malicious actors or technical problems of selected validators.
- Incentives model to maximise the performance of the network finality.