[

Screen Shot 2023-06-16 at 10.23.38 AM

1028×360 10.5 KB

](https://forum.celestia.org/uploads/default/original/1X/839a5022c192a38bcaf4f73e10d1fab6b4a3240e.png)

Overview

We at Fairblock are building pre-execution privacy solutions with an initial focus on the Cosmos ecosystem.

In particular, we believe that our current architecture lends itself extremely well to the shared sequencer idea.

Using threshold encryption, as well as a decentralized set of validators, users are able to submit encrypted transactions for many rollups to Fairblock.

The set of validators then commit to the ordering and once an ordering is finalized, decrypt the transactions.

This approach provides pre-execution privacy to users ensuring frontrunning protection and censorship resistance.

The following are great resources for an overview of shared sequencing:

- Sharing a Sequencer Set by Separating Execution from Aggregation
- Rollups Aren't Real - by Jon Charbonneau - Jon's Newsletter

The Idea

Our Cosmos SDK app chain FairyRing

is a decentralized network of nodes responsible for aggregating decryption key shares used in threshold identity-based encryption (IBE).

At a high level, our solution leverages IBE where an encryption key can be derived from a single master public key and some ID (in this case, the height of our blockchain).

Currently, each validator is responsible for submitting their verifiable keyshare via a transaction on our chain.

Once a set threshold of keyshares have been received for a particular blockheight, a decryption key can be released allowing the encrypted transactions to be revealed.

Currently, the design has a one block delay, meaning the encrypted transactions submitted in block n

will be revealed in block n+1

.

We are optimistic that this delay can be removed completely, with the use of vote extensions in ABCI++.

To post encrypted transactions to fairyring

, the flow for a user would be the following:

In Block n

:

1. User first generates a signed rollup transaction payload.
2. The transaction payload is encrypted using a hybrid encryption scheme. To encrypt, the master public key is fetched from fairyring

along with the current block number n

(the target height will be n+1

).

1. User submits a transaction to fairyring

which contains the encrypted ciphertext.

1. fairyring

finalizes the ordering of encrypted transactions.

In Block n+1

:

1. Validators submit keyshares for the current block (n+1

)

1. If t+1

shares are received in the current block, a decryption key is generated by aggregating the keyshares submitted.

1. The encrypted transactions in block n

are decrypted and batched.

1. A relayer then relays the decrypted blocks from fairyring

to a DA layer (i.e Celestia).

[

image

1904×1322 289 KB

](https://forum.celestia.org/uploads/default/original/1X/4d2738663d1902649bbb8ad62c6e4ed8233f1f59.png)

Benefits

1. Pre-execution privacy

: * Protects the user from reordering of transactions on the sequencer level.

- Additionally prevents frontrunning attacks based on transaction contents.
- Protects the user from reordering of transactions on the sequencer level.
- Additionally prevents frontrunning attacks based on transaction contents.
- Censorship Resistance

: * A decentralized set of sequencers can reduce the risk of censoring particular addresses.

- Can prevent censorship based on transaction contents: i.e. all bids above a certain amount.
- A decentralized set of sequencers can reduce the risk of censoring particular addresses.
- Can prevent censorship based on transaction contents: i.e. all bids above a certain amount.
- Chain-agnostic

: Because we are only responsible for posting sequenced transaction data to Celestia, it is straightforward to provide this service most chains out of the box.

It is up to the rollups themselves to parse the transaction information and execute them.

1. Inherited benefits discussed here of a decentralized validator set
2. Other interesting applications including randomness generation services that can be provided using this architecture.

MEV

We acknowledge that MEV can be a major source of revenue for rollups.

There are multiple approaches to addressing MEV including auctions (like Skip or Flashbots) and encryption (like Fairblock).

We believe that the decision of how to deal with MEV should be up to the rollup.

We want to provide users with additional flexibility of pre-execution privacy and are open to a hybrid approach of having both auctions for blockspace and encrypted transactions.

Note also that this design can support both plaintext and encrypted transactions.

An ordering has already been established with the finalization of block n

. However, the block can only be relayed once all the transactions in the block have been revealed.

Thus the relayer must wait until block n+1

to relay the block to the DA layer.

This can be thought of as a soft-commitment at block n

of inclusion, even before the DA layer finalizes the block.

But as discussed earlier, this latency may be removed through vote extensions and ABCI++.

We are really excited to build and grow with the Celestia community and really appreciate the great support and feedback that we've got so far. Would love to hear your thoughts, ideas, and feedback!