Wisp: Cross-Rollup-Communication Protocol

Daniel, Architect at LimeChain (blockchain development company) and part of LimeLabs - R&D Divison.

Abstract

The following aims to describe an enshrined Cross-Rollup-Communication protocol for data transfer between rollups, completely aligned with Ethereum's rollup-centric future and supporting the Ethereum community.

The draft paper elaborates on the economic incentives for the actors participating in the protocol, presents a CRC message flow, and reviews the security and scalability implications of the protocol.

How it works

Wisp is (1) an on-chain SNARK-based light client and (2) a verification mechanism for the storage of a rollup. The on-chain light client makes sure that the destination rollup can trust and reason about a specific execution state root at a specific height of Ethereum L1. Based on this root, smart contracts can reason about the inclusion (or not) of a certain piece of information inside any rollup anchoring with Ethereum L1. The way that the data inclusion reasoning happens will be specific for each source rollup.

The proposed system includes relayers as actors who transfer data from a source rollup into a destination rollup. A successful data transfer requires:

1. Ethereum executionStateRoot

posted on the Destination Rollup

1. Merkle Inclusion Proof (from Ethereum L1) of the root

of the Source Rollup

1. Merkle Inclusion Proof (from Source Rollup) of the storage

slots that must be proven and for the Destination rollup to verify the integrity of the data transfer.

Proving the L1 Execution State Root

The CRC protocol incorporates an on-chain light client that follows theEthereum Sync Protocol and updates its head through the usage of ZK-SNARKs. The ZKP proves that the majority of the SyncCommittee has signed a given block header.

Proving the Rollup State Root

The root

of the Source rollup is posted on the Rollup's L1 Contract address. Merkle Inclusion Proof of the storage key holding the Source Rollup state is provided to the CRC contract on the destination network. Using the executionStateRoot

already proven from the last step, the contract verifies the state root of the source rollup.

Proving the Data to be transferred

Merkle Inclusion Proof of the storage key holding the data inside the Source Rollup is provided to the CRC contract on the destination network. Using the already proven source rollup state, the contract verifies the raw data that must be transferred.

Alpha Version

There is a live alpha version of the protocol that uses a SNARK similar toProof-of-Consensus to prove the L1 Execution State Root (step 1).

- Draft Paper - Introducing Wisp - a Cross-Rollup Communication Protocol
- Demo Application - https://demo.wispprotocol.com/
- Docs - https://docs.wispprotocol.com/

How is this different from other initiatives?

- Ethereum rollup centric
- Wisp is specifically focused on the Ethereum ecosystems and its rollups. It recognizes the nuances of the rollup-centric vision of Ethereum and is not designed nor intended to become a “cross-chain” initiative.
- Open-source public good.

A cross-rollup communication protocol should be 1) open-source (non-negotiable), 2) public good and ideally 3) built in the open with contributions (or at least input) from different teams. A public good does not exclude having a sustainable revenue stream, but it does exclude rent-seeking behaviour, centralization and optimizing for profit (rather than impact).

- Security.

Absolutely crucial. The ideal CRC solution must provide security beyond crypto-economics and incentives. A preferable approach here would step on the security of L1 Ethereum and complement that with additional cryptography (zk proofs). Wisp does this through SNARKs rather than economical incentives.

- Decentralization.

There is no multi-sig controlling a bridge. Anyone can participate as a relayer in the Wisp protocol. No actor is special or permissioned - anyone can assume any of the protocol roles. The protocol’s decision-making should also decentralize over time if it becomes a key part of the ecosystem.

- Neutrality.

The protocol should facilitate interoperability in the Ethereum ecosystem and avoid servicing certain rollups or applications at expense of others.

An always-open invitation to join and contribute

Wisp is intended to be completely permissionless and built-in public. We’ve modelled our approach by the work of the Flashbots initiative - being a public good and completely in line with Ethereum. For Wisp to be permissionless and neutral, it would require multiple diverse parties to join the initiative. Below are some top-of-mind ways to join and contribute.

Feedback and support

We are still early in the development and hope to get feedback from the Ethereum community and the Ethereum thought leaders. Any critical feedback and improvement suggestions are welcomed and appreciated. Feel free to comment here or reach out in discord.

A shortlist of topics to further explore and collaborate

Here are some unexplored or underoptimized aspects of Wisp. We would love to see collaborators and suggestions in these or any other aspects of the protocol.

- Fast-tracking Ethereum finality
- how not to need to wait 12 minutes for block finality
- Dealing with rollups finality
- how to deal with the (not)finalized state of a rollup.
- Optimizing and combining the state relay proofs
- this could mean completely moving away from Circom and Groth16 if needs be.
- Optimizing the multiple Merkle inclusion proofs
- for the Ethereum execution root or the storage inclusion in a rollup
- Moving away from the sync protocol committee and basing on the wider validator set
- is this needed and beneficial?

Supporting rollups

We would love to support all rollups. At the moment we support Optimism Bedrock-style rollups. We’ve explored several other rollups but would need closer collaboration with the roll-up teams in order to support them. This is mainly due to

differences in the state management of most ZK rollups. We would like to invite any interested rollups to get in touch - we would love to align with you and add as many rollups as possible.

Building on top of Wisp

Protocol, without applications on top of it, is worth nothing. We've started exploring building sample applications on top of it (much like the demo one). If you are interested in being a cross-rollup app developer please get in touch. We would love to make it so that is super convenient and easy for your dapp to live multi-rollup.