ERGO

May 2024

Alexander Chepurnoy
(aka kushti)

Trustless Bitcoin relay on Ergo

ErgoHack VIII report

Introduction

What is a trustless Bitcoin relay?

- Minimal Bitcoin client done as a contract on Ergo
- Validating Proof-of-Work, storing valid headers and building best chain
- Similar to SPV (simple payment verification) Bitcoin clients
- Validating a Bitcoin transaction and passing it to other contracts

  (if the transaction is valid)

Use Cases

- Trustless Bitcoin hashrate derivatives on Ergo

  (i.e. Bitcoin miners can do contracts on Bitcoin hashrate on Ergo, to hedge against possible hashrate increase)

- Making Ergo a smart layer for Bitcoin (Ergo contracts can do actions based on Bitcoin transactions and data written on Bitcoin, e.g. in OP_RETURN fields, that is how RGB/RGB++ etc operate)

- Trustless bridging from Bitcoin to Ergo (but how to send back ?)

- Trustless cross-chain DEXes

History

- First Bitcoin relay was launched on Ethereum (supported by Consensys now)
- Was too expensive, often behind Bitcoin chain
- Nervos recently utilized relay to build RGB++

Scope of the ErgoHack submission

- Get relay contract done, which is validating incoming headers, builds commitment to their database and choosing best chain out of all the headers
- Do validation for a Merkle proof of Bitcoin transaction inclusion into a block with enough confirmations
- Show feasibility of the approach
- Publish contracts along with detailed description
- Outline further research

Implementation

Relay contract

- Holds digest of two trees: one for best headers and their heights,
  for client applications, and one for all the headers, as well as tip's block id and cumulative work
- Efficient switch during forking as all-headers tree contains digests of header-chains,
  so to switch from one best chain to another only best headers chain digest rewriting is needed
- Bitcoin PoW validation
- Relies on Sigma 6.0 features (nbits decoding), could be bypassed maybe

Bitcoin transaction inclusion validation contract

- Validates that some transaction was included into a block with enough confirmations
- Provided with transaction bytes, relay contract (as read-only input), and two proofs
- One proof is for proving that a header in the relay’s contract best-chain has enough confirmations
- Another is Merkle proof for transaction inclusion into a block with the header from the first proof
- Pretty efficient

Further R&D

Further steps

- Make a transaction parser, for concrete goals for starters (like reading OP_RETURN data from concrete output)
- Make application on top of concrete parsers
- Consider economic model for relay (incentives for submitting blocks)

Build on it!

Free foundation to build on

- https://github.com/ross-weir/ergohack-sidechain
- All the code is in public domain, can be freely used
- So come and build smart layers for Bitcoin on Ergo

Thank You

https://ergoplatform.org

https://twitter.com/chepurnoy

https://t.me/ergoplatform