Two-Way Pegged Sidechains On Ergo (ErgoHack Report)

kushti, soysor

November 16, 2023

Abstract

In this report we outline some ways to implement pegged sidechains on Ergo with smart contract powered trustless transfers, where both main and side chains are based on UTXOs and support ErgoTree (in its current form, at the same time, sidechain may support more features). In such setting, transfer from main-chain to sidechain and back can be done via on-chain contract, then security issues are reduced to sidechain consensus security only.

1 Introduction

A sidechain [1] is a secondary blockchain connected to the main blockchain with a two-way peg, which is defined in [1] as "mechanism by which coins are transferred between sidechains and back at a fixed or otherwise deterministic exchange rate". Pegged sidechains enable ERG (native cryptocurrency of the Ergo blockchain) and other Ergo blockchain assets to be transferred between multiple Ergo-like blockchains. Sidechains can be considered as playgrounds for experimental features (for example, a sidechain may support shielded transactions or other privacy-preserving techniques), or scalability solution, allowing to off-load transaction from the main chain to sidechains.

In this work we consider what should a sidechain support and what data about sidechain is needed on the Ergo blockchain in order to have trustless (smart contracts powered) two-way peg, with providing prototype contracts for peg-related transfers.

1.1 Ergo blockchain details

We provide relevant Ergo blockchain related information relevant to our constructions here. Prior knowledge of UTXO accounting model, Bitcoin [2], data structures (and many other topics involved in cryptocurrency design) is needed.

- Ergo has UTXO set commitments, so set of unspent outputs after processing transactions in a block is committed via authenticated AVL+ tree, with tree digest included into a block header.

- Last headers as well as some fields from upcoming header can be read from a contract.

2 Sidechain Data on the Mainnet

In this section we consider how sidechain data can be stored on the mainchain. A sidechain progress can be represented in the simplest form as a tuple (h, T_h, U_h, C_h), where h is a sidechain height, T_h are state changes (transactions) done in h, U_h is digest of AVL+ tree built on top of UTXO set after processing the changes T_h, C_h is AVL+ tree which contains all the previous sidechain states in form of key-value pairs $h \rightarrow hash(h||T_h||U_h||C_{h-1})$.

First of all, we need to write this minimal data somewhere, to read in contracts later. Currently, we can think about two options:

- there's extension section of a block, which contains key $\rightarrow$ value pairs, and root of a Merkle tree built upon the pairs is included into a block header (and can be read by a contract). We can include different sidechains here, or even a tree built on top of sidechains. The only issue with this approach is that Merkle trees are not supported in ErgoTree/Script yet
- we can just put sidechain data into a box identified by some NFT. Then data can be read via read-only input.

2.1 Sidechain Consensus

There could be different option how sidechain can make a progress

- merged mining - PoW fields (nonce, solution) to be added to the sidechain data tuple defined in Section 2, and written on the Ergo blockchain. Then sidechain header would be Ergo block header + extension Merkle path or path to box in UTXO set tree + sidechain tuple with PoW data. Security: subset of Ergo miners (so honest majority amongst them)
- if sidechain data is in a box, we can make the box spendable for an Ergo block miner (by using miner pubkey from context like done in emission contract). Security: subset of Ergo miners (so honest majority amongst them). There could be other options here (multisig etc)

3 Transfers

In this section we consider how to transfer ERG from main chain to sidechain, to get sERG tokens on the sidechain, how how to transfer sERGs back, getting ERG on the main chain back.

We assume setting where mainchain data is known to sidechain miners (validators) and so can be used in contracts. It seems this requirement can be relaxed further though (by using SPV relay hard-coded or done via a contract).

There are some assumptions we have for simplifying contracts:

- sidechain has no storage rent
- sidechain consensus rules require that during input script validation, for context variable # 125 only mainchain; state digest can be provided

Then contracts-powered transfers from main-chain to side-chain and back are done as follows:

1. user sends E ERG to a contract which is allowing to unlock them when there is a box on the sidechain with at least E sERG locked forever (with $false$ contract in simplest case, when there's no storage rent on the sidechain); unlock may happen after enough main-chain confirmations.
2. as mainchain state digest can read in sidechain contracts (in our case, via context variable # 125), sidechain contract holding all the sERGs initially, may unlock them when main-chain box is in the UTXO state for enough side-chain confirmations.

4 Security

As transfers are based on contracts, sidechaining security is reduced to consensus. We consider options from Section 2.1 here.

In case of merged mining, so PoW puzzle written to the sidechain data, sidechain will be mined by subset of Ergo main-chain miners, then consensus security is about honest majority of a subset of Ergo miners.

Similarly, if every block every miner (and only miner) can publish sidechain data, consensus security is trivially also about honest majority of a subset of Ergo miners.

5 Evaluation

kushti notes : Fill with tests data

6 Further Work

In this section we sketch possible extensions for proposed sidechain construction.

Trustless Crosschain Transfers via Sidechains It is possible to use Ergo sidechains for exchanges with other cryptocurrencies. We consider Bitcoin as an example. Sidechain in this case will have SPV relay support for Bitcoin, and also support for Bitcoin Merkle trees and transactions parsing. Then it is possible to unlock sBitcoins on sidechain on presenting proof-of-lock on Bitcoin sidechain. Still, no options for two-peg sidechains with Bitcoin blockchain known yet.

Sidechain Evolution at Superblock Speed It seems to be doable to have a merged-mined sidechain, for example, which progress is done at faster pace than main-chain blocks. For that, it is needed to tie sidechain data updates with subblocks.

References

[1] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, vol. 72, pp. 201–224, 2014.

[2] "Bitcoin whitepaper," *Interventions*, 2021.