INPUT | OUTPUT

IOHK | SCOREX

ROLLERCHAIN, A BLOCKCHAIN WITH SAFELY PRUNEABLE HISTORY

SCOREX BLOG

SEPTEMBER 07, 2016 ALEXANDER CHEPURNOY 8 MIN READ 0 COMMENTS

When you starting a Bitcoin node it is downloading all the transactions for more than 7 years in order to check them all. People are often asking in the community resources whether it is possible to avoid that. In a more interesting formulation the question would be “can we get fullnode security without going from genesis block”?

The question becomes even more important if we consider the following scenario. Full blocks with transactions are needed only in order to update a minimal state, that is, some common state representation enough to check whether is arbitrary transaction is valid(against the state) or not. In case of Bitcoin, minimal state is unspent outputs set (we call this state*minimal* as a node could also store some additional information also, e.g. historical transactions for selected addresses, but this information is not needed to check validity of an arbitrary transaction). Having this state (with some additional data to perform possible rollbacks) full blocks are not needed anymore and so could be removed.

In Bitcoin fullnodes are storing all the full blocks since genesis without a clear selfish need. This is the altruistic behavior and we can not expect nodes to follow it in the long term. But if all the nodes are rational how a new node can download and replay the history?

The proposal recently put on Arxiv trying to solve the problems mentioned with a new Proof-of-Work consensus protocol. I very lightly list here modifications from Bitcoin, for details please read the paper.

1. State is to be represented as an authenticated data structure(Merkle tree is the simple example of such a data structure) and a root value of it is to be included into a blockheader. It is the pretty old idea

already implemented in Ethereum(and some other coins).
2. We then modify a Proof-of-Work function. A miner is choosing uniformly **k** state snapshot versions out of last **n** a (sufficiently large) network stores collectively. In order to generate a block miner needs to provide proofs of possession for all the state snapshots. On a new block arrival a miner updates **k+1** states, not one, so full blocks (since minimal value in **k**) are also needed.

Thus miners store a distributed database of last **n** full blocks AND state snapshots getting rewards for that activity. A new node downloads blockheaders since genesis first (header in Bitcoin is just 80 bytes, in Rollerchain 144 bytes if a hash function with 32 bytes output is chosen). Then it could download last snapshot or from **n** blocks ago, or from somewhere in between. It is proven that this scheme achieves fullnode-going-from-genesis security with probability of failure going down exponentially with "n" (see Theorem 2 in the paper). Full blocks not needed for mining could be removed by all the nodes. They can store them as in Bitcoin but we do not expect it anymore.

The RollerChain fullnode is storing only sliding window of full blocks thus storing disk space, also less bandwidth is needed in order to feed new nodes in the network and so bandwidth saved could be repurposed for other tasks.

May 17, 2016 by alex.chepurnoy@iohk.io

Announcing Ergaki - A performant, public bulletin board for voting and auctions

The first Scorex-based testnet, Lagonaki, combines the Permacoin consensus protocol implementation with a simple, Nxt-like payments module. After Lagonaki, the next Scorex-based testnet will be *Ergaki*, a block chain system that will be used as a public and performant bulletin board for various protocols including voting and auctions.

May 17, 2016 by alex.chepurnoy@iohk.io

Ergaki, the Next Scorex Testnet

A Scorex application is comprised if core, and Scorex itself is the core functions and module interfaces, and modules. The current testnet, Lagonaki, is made of Permacoin consensus protocol implementation and a simplest Nxt-like payments module.