Abstract.

A new type of atomic swap on PoS chains with a merkle patricia tree, which can be completed without communications like submitting a preimage of blankhash.

This eliminates the lock time after atomic swap transactions of both sides are set.

It is possible because a storage state with a merkle patricia tree and its signs by validators can be proven by Turing complete verifiers on PoS chain, and these validators cannot be evil as far as Slasher's punishment exists. Voters(validators) of PoS don't do anything special, just do voting as usual.

Main usages are cross chain swap and cross shard transaction.

A simulation and detailed explanations are in codes here.

Introduction.

Generally speaking, atomic swap is said to be slow. This derives from locking time of HTLC. If the second tx setter can force the first tx to be executed, there is no locking time within trustless atomic swap.

('lock time' is waiting for submitting hash-preimage of the setter of the first tx)

Traditional Atomic Swap in nutshell

(Alice=A, Bob=B)

1. A broadcasts payment tx1 to B with condition hash-preimage X is required to be submitted (Hash(X) is revealed)
2. B broadcasts payment tx2 to A with condition hash-preimage X is required to be submitted, as a reaction to 1. B does not know X,but copies Hash(X)
3. A submits and reveals X to tx2 on network and gets paid B's coin.
4. B knows and submits X as a reaction of 3. B can get paid as well.
5. If time limit expired as a failure of 3 or 4 ,one gets paid back.

Main Concept of NoCommunication Atomic Swap

(the native currency is ETH for instance)

1. A broadcasts payment tx1 on PoS-chain1 to B with a condition that the storage merkle hash of A's ETH balance and World-StateX(value) is signed by PoS voter rich-E ,rich-T,rich-H.

(if the voting is signing to World State in PoS voting)

(when a voting sign is on a block hash, it'll be also fine)

1. B broadcasts normal tx (tx2 on PoS-chain2) to pay to A.
2. PoS voter rich-E ,rich-T,rich-H sign the block(massage is World-State?).
3. B submit votes for block (all signs to world-StateX by rich-E ,rich-T and rich-H), the storage merkle hash of A's account state, the merkle branch hashes which collect path to world-State from the account state.

Then B can get paid A's coin.

(A does not submit anything after contract setting. B can execute tx1 as soon as the payment is done)

The simulation and detailed explanations are in codes here.

GitHub

### leohio/no_communication_atomic_swap_simulation

Explaining NoCommunication Atomic Swap. Contribute to leohio/no_communication_atomic_swap_simulation development by creating an account on GitHub.

The Requiries

(1)In PoS voting, voters sign to the world state or the block hash, then finally can be proven to have sign-relationship to the change of balance A's ETH.

(2)Slasher Protocol is implemented to prevent double voting. Without this, voter can be malicious about this swap by signing(voting) to a wrong hash.

Direct Usage

This can get along with traditional atomic swap (HTLC). Then B can execute tx1 with merckle hash operation, on the other hands, A can execute it by submitting preimage X either. When time expired, A and B can get paid back as well.

Then this has either of

(1) atomic swap without waiting while locking time

(2) safety when time expires

Cross Shard Transaction

Each shard has a state root of accounts within it, then cross shard transaction is essentially same of cross chain transaction between PoS-mpt chains.

NoCommunication Atomic Swap is useful for cross shard tx.

When an execution of a contract is set after the swap condition gets satisfied, Atomic Swap payment will be Atomic Swap Execution across shards.

Problem and Risk

The hack occurs when all the assigned validator sign another block to revert atomic swaps.

It’s really difficult to happen. But if this is used at cross shards,a small shard can make this problem.

Data availability is also a problem. Bob has to submit merkle branches, so this needs more historical nodes which provide such data.