Only 3 opecodes are needed to build DEX on Plasma Chain.

Leona Hioki (leona.hioki@laurus-school.com)

Soichi Kawai (skawai@math.kyoto-u.ac.jp)

Let us think about a fraud proof of general computings with a merkle tree ,and we think about an actual use case for instance.

Most of contract codes like EVM byte-codes are thought to be difficult to execute on Plasma chain. This hardness derives from the limitation that we cannot use Root Hash of state because only an operator has all value data which is the result of transactions.

This can be solve to a certain extent by recording all of execution process in a Merkle Tree. By this, we can prove which part of a process is precisely wrong at law cost, and minimize any fraud proof of general computings.

====Fraud Proof of General Computing====

Let us have a problem 1+1+1 and an evil executor returns "1+1+1=3+1=4".

We introduce opecode "ADD" which adds up left term.

And have a tree which leaves are,

(1,1,1,ADD),(2,1,ADD),(3,JUNK),(JUNK,JUNK)

This order means process of execution.

Even if the evil executor returns,

(1,1,1,ADD),(3,1,ADD),(4,JUNK),(JUNK,JUNK)

We can prove that the executor broadcasted tree includes

(1,1,1,ADD),(3,1,ADD) by hash values in the tree,

and also can prove that this part is wrong in the function of Plasma Contract.

This proof size scales at O(log(n)),so it can be used in any big contract execution.

NOTE: "JUNK" is just used to make a tree with 2^n leaves.

===Making DEX with this tree with some opecodes===

A merkle tree which includes opecodes can be set as a program, and can record the whole process of the execution of the program when input comes.

We will have the special merkle tree as an extension of that tree above.

The leftost branch just below the root hash is the hash of argument leaves (x1,x2,xn).

These arguments are not set when users of DEX send this to the operator with a transaction to make bid or ask.

The middle branch just below the root hash is a hash of memory leaves (m1,m2,…,mj),which record the value used in the process of the execution.

In other branches, opecodes, arguments leaves(like x1) and memory leaves (like m1) are set as leaves like (m2 x1 SUB).

The result of execution of branch will be inherited to the next right branch.

So, in a program branch "(9,3,SUB,m1),(m1,4,SUB,m2),(m2,1,SUB,m3)", m1 will be 9-3=6

and m2 will be 6-4=2, and m3 will be 1.

Like first case "1+1+1",this process and result can be proven with hash values in the tree by this structure either.

We need only 3 opecodes SUB, MUL,TX to make a DEX.

TX code branch is like (amount,from,to,currency,TX)

For example, we send 300 ETH to the operator with a tree of [300 ETH at rate 10 ETH/ETC] to get 3000 ETC. And one person can deal with amount smaller than 300 at rate higher that 10 ETH/ETC after he/she sends ETH. Operator sends the change to us.

The tree of this deal is described with leaves below,

(x1,x2,x3) //this means (amount,rate,address) arguments

(m1,m2,m3) // memory

(owner_address,operator_address)// our address

// program start

(300,x1,SUB,m1)

(10,x2,SUB,m2)

(x1,x2,MUL,m3)

(m3,operator_address,owner_address,"ETC",TX)//we get ETC

(m1,operator_address,owner_address,"ETH",TX)//this is change

(x1,operator_address,x3,"ETH",TX) // the other get ETH we set

Then, if any underflow in (m1,m2,m3), it’s a fraud transaction which can be easily proven with hash values in the tree.

Anyway, an operator executes and proves it with a tree, so he/she chose and match the transactions not to make wrong one.

if we or the operator want a condition like [transaction fee is 1ETH or 10 ETC for the DEX operator], we just add

(m1,1,SUB,m4),

(x1,10,SUB,m5)

We can extend DEX transactions in many ways by this tree.

===PROCESS===

(1) Alice set (price, amount, transaction fee) ,and make a exchange program with the merkle tree

(2) Alice sends a tx with UTXO (to Operator) and the exchange program tree.

(3) OP accepts it and broadcast this with OP’s sign.

(4) Bob react this, and sends a tx with UTXO (to OP)

(5) OP completes the exchange program tree and make UTXO to both of Alice and Bob as the result of the exchange.

===SECURITY REQUIREMENT===

Anyone can prove any fraud in this chain, but need to accuse one by one in the plasma contract on main chain. And every node needs to simulate the set transaction program written by tree structure before sending a real transaction.