}

Miner Network White Paper

Background

Client

To interact with 0chain a client needs to have a private key and a public key to create and receive transactions.

Transactions

A transaction is made up of seven important parts: to address, from address, data/code, transaction ID/nonce/timestamp, the chain address, and the signature.

Address formula:

Address = H(Public Key)

Hash formula:

h^T = H(to, from, data/code, transaction ID, chain address)

Signature formula:

$S^T = [h^T]_{clientTo}$

Transaction structure:

```
{
        to:
        data/code:
        from:
        transaction ID/nonce/timestamp:
        transaction hash:
        chain address:
        signature:
}
```

Blocks

Blocks have two main parts: the header and the body.

Header:

The header consists of the previous block in the chain's hash, a list of transaction hashes, and the block hash.

Block hash:

h^B = H(previous block's hash, h^{T0}, h,^{T1},…, h^{TN}).

Header structure:

```
{
        previous block hash:
        block hash:
        hash of transaction:
        hash of transaction:
        …
        hash of transaction:
}
```

Body:

The body of the block contains all the transactions, denoted as T, referenced in the list of transaction hashes in the header. The body also contains a list of miner signatures, denoted as S, used for the census protocol.

Body structure:

```
{
        T1
        T2
        …
        TT
        S1
        S2
        …
        SS
}
```

Miners

All miners in a network receive incoming transactions for their corresponding blockchain. Each miner verifies each incoming transaction by creating a hash of the transaction and comparing it to the hash value in the transaction. Then the miner uses the public key of the client who sent the transaction to verify the signature of the transaction is authentic. Once the transaction is verified it is added to the pool of transactions the miner will use to create or verify a new block.

Block Generation

To generate a block a miner selects as many transactions from the transaction pool within a certain time. For each transaction the miner verifies that the client who created the transaction has sufficient funds to cover the transaction. The miner then verifies each transaction signature with the client's public key, and makes sure the transaction ID/nonce/timestamp is valid. If the client has sufficient funds and the transaction is valid then the transaction is added to the block the miner is creating.

Once the miner has pooled all the transaction it can in the given timeframe then the miner creates both the header and the signature for the block. First to create the header the miner adds the previous block's hash to the header. Then for each transaction the miner adds the transaction hash to the header. Finally, the miner takes the previous hash and all the hashes of the transactions to create the hash for the block being created. Once the hash is made the miner uses its private key to sign the hash and add it to the body of the transaction.

Block Verification

To verify a block a miner looks at the block's head and verifies that the previous hash has been an option in the previous rounds. The miner then recreates the hash of the block with the previous hash and the transaction hashes. If the block's hash and the one generated by the miner matches then the miner verifies the signatures in the body of the block with the associated miners' public keys. If all the signatures are verified then the miner verifies all the transactions in the block, as described above.

Miner Network

The miner network is composed of a M x N matrix of miners with M primary miners and N = Backup Miners + Bench Miners + 1 Primary Miners. Each column of miners acts simultaneously either generating or verifying blocks in the appropriate timeslot. There are M timeslots, one for each column, and at any time one column will generate a block and M-1 columns will verify blocks.

Generate Block

At timeslot T, where T is ≥ 0 and < M, column T will generate a block. Individually, each miner generates a block as described above. Each miner then sends the block head to all the miners in T+1%M so they have the previous block hash (the hash of this block) and know which transactions not to include in the block they generate. Each miner in column T also sends the whole block to the miners in column T+2%M so they can verify the block.

Verify Block

When verifying a block each miner will have the number of backups plus one from the primary. The miner decides which one to verify by comparing the hashes of all the blocks sent to it and tallying the individual hashes. The hash with the highest tally is the block the miner will verify as described above. If there are no repeating hashes then the miner picks the block generated by the primary. When there are hashes that repeat the miner will take the signatures from all the blocks with the same hash and add them to the working block. The miner then sends the verified block to every miner in the T+1%M column.

Confirm Block

Once a block has gone through all M columns, the last set of miners broadcast their block to every miner in the network. Each miner takes all the blocks, which should have the same hash, and adds the miner signatures to the working block if they are not already on it and the block the miner is taking the signature from hash the same hash as the working one. Once all the signatures have been added the miner tallies the number of signatures and compares it to the quorum of the network. If the number of tallies is greater than the quorum and the previous hash of the block is equal to the current block hash the miner has then the miner adds the block to the chain.

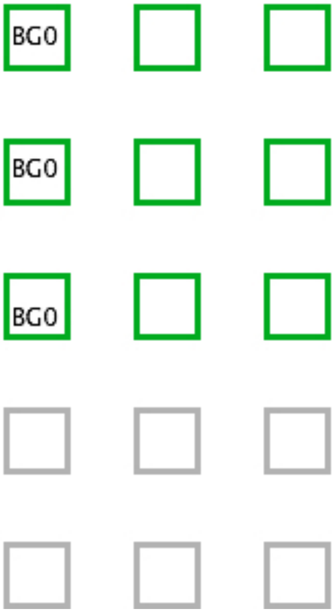
Example Round

BG = Block Generated

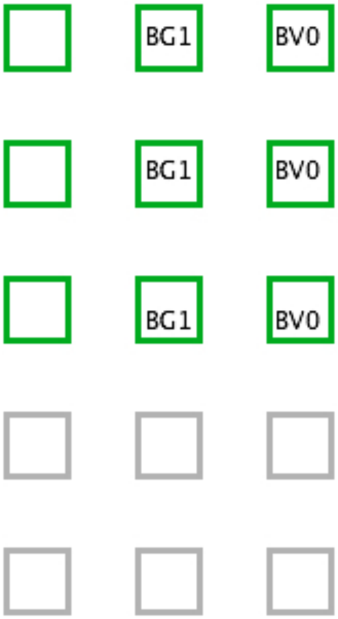
BV = Block Verified

Timeslot 0:

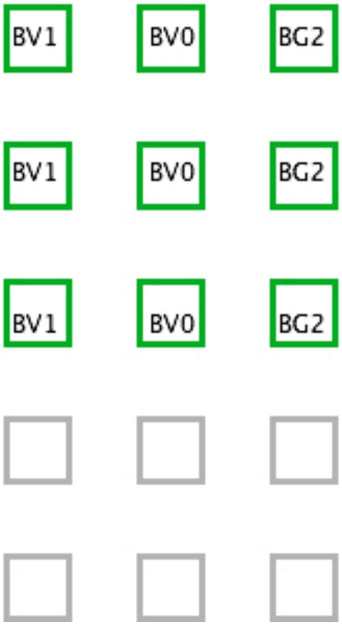
Miners in column 0 generate block 0. Miners from column 0 then send the block head to miners in column 1, and the whole block to miners in column 2.



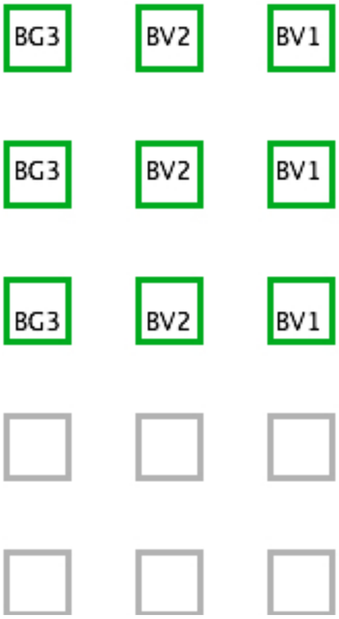
Timeslot 1:
The miners in column 1 generate block 1 and send the head of the block to the miners in column 2, and the whole block to column 0. Miners in column 2 verify block 0 and send the block to the miners in column 1.



Timeslot 2:
Miners in column 2 generate block 2 and sends the head to column 0, and the whole block to column 1. Miners in column 0 verify block 1 and send it to the miners in column 2. Miners in column 1 verify block 1 and broadcast it to all miners in the network to add it to the chain.



Timeslot 3:
Miners in column 0 generate block 3 and send the head to the miners in column 1, and the whole block to column 2. Miners in column 1 verify block 2 and send it to the miners in column 0. Miners in column 2 verify block 1 and broadcast it to all miners in the network to add it to the chain.


BG3
BV2
BV1
BG3
BV2
BV1
BG3
BV2
BV1