Voting blockchain for High Security NFT

V1.2
OCT 2021
Emaan Institute management and Science
Computer Science dep.
Dean Prof. Dr. Hiro Takahashi
Abstract

• NFT secure blockchain architecture confirming the ownership of transaction and its implementation of secure payment method to avoid mistake of transfer is proposed.

• Escrow account confirms its transaction by J Node using unique voting method.

• Participant node can obtain mining reward by voting work, This is a motivation of generation of NFT blockchain, continuously.
NFT contents
NFT

• NFT is a unit of data stored on a blockchain digital ledger to represent items such as photos, videos, audio, and other intellectual property.

• NFT certifies a digital asset to be unique, it is not interchangeable.

• Current blockchain store the transaction from each node but it is required more high security of ownership.
Issues

- Higher transaction gas fee
- Latency time
- Risky token transfer
- Electricity cost of PoW mining
Proposed secure NFT blockchain
Secure NFT (NFTSBL)

- NFTSBL (Non-Fungible Token) secure blockchain using voting function to establish consensus for each transaction and avoid mistake of coin transfer by escrow account on selected node which is called “J Node”.

- NFTSBL is minable original Blockchain.

- The escrow account refers and judges the transaction between sender to receiver with selected multiple number of witness nodes.

- Eventually, escrow account stores the status of the final judgment of transaction and it transfers NFT token from sender to receiver wallet if transaction was confirmed.
Voting for consensus of NFTSBL

J node With Escrow account

If A + B + C = true + True + Not True
Then, release token to Receiver wallet

NFTSBL selects multiple nodes for consensus of transaction.
NFTSBL also selected J node randomly for judgment and holds escrow account.
Whenever transaction was confirmed, then release the transaction token to receiver node.
Voting Blockchain sample code

```solidity
function voter_registration(address _ip_address) public {
    // for bytes conversion of msg.sender and _ip address
    uint _size = abi.encodePacked(msg.sender).length + bytes(_ip_address).length;
    bytes memory _data = new bytes(_size);
    uint counter = 0;
    for (uint i = 0; i < abi.encodePacked(msg.sender).length; i++) {
        _data[counter] = abi.encodePacked(msg.sender)[i];
        counter++;
    }
    for (uint i = 0; i < bytes(_ip_address).length; i++) {
        _data[counter] = bytes(_ip_address)[i];
        counter++;
    }
    require(!voters[keccak256(_data)]);
    voters[keccak256(_data)] = true;
}

function vote_casting(address _ip_address, string _category) {
    // for bytes conversion of _ip address and _category
    uint _size = bytes(_ip_address).length + bytes(_category).length;
    bytes memory _data = new bytes(_size);
    uint counter = 0;
    for (uint i = 0; i < bytes(_ip_address).length; i++) {
        _data[counter] = bytes(_ip_address)[i];
        counter++;
    }
    for (uint i = 0; i < bytes(_category).length; i++) {
        _data[counter] = bytes(_category)[i];
        counter++;
    }
    urlData[keccak256(_data)].voteCount++;
    emit votedEvent(keccak256(_data));
}
```
J node

• J node is voting transaction confirmation node.

• Randomly selected, and receives the transaction confirmation status from multiple nodes.

• J node decides whether the transaction is true or not by decision making formula.

• If sum of transaction result is more than 75% (threshold value) then, then the token of transaction is released.
Escrow account

• Escrow account on J-Node helps in avoiding the mistake of token transfer to the wrong party.

• J node keeps the token until the transaction in the escrow account is confirmed by selected multiple nodes.

• Upon confirmation of transaction is true, J node releases the token to the receiver’s wallet and complete this task.
Escrow account decision making

\[ D = \sum_{i=1}^{n} (R_i) - \theta j \]

\( \theta j \) is bias factor of judgment condition if any NFT content has comment to deal.
There are seven nodes were selected and six node replied True answer but one node was non true answer, Therefore accuracy of this transaction is 6/7 which is 85.71%. It is accepted as True transaction if threshold value was set under 75%.

<table>
<thead>
<tr>
<th>User Type</th>
<th>Node</th>
<th>Reply Signal</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTX buyer</td>
<td>A</td>
<td>T</td>
<td>True</td>
</tr>
<tr>
<td>NFT Provider</td>
<td>B</td>
<td>T</td>
<td>True</td>
</tr>
<tr>
<td>other User</td>
<td>C</td>
<td>NT</td>
<td>True</td>
</tr>
<tr>
<td>other services</td>
<td>D</td>
<td>NT</td>
<td>True</td>
</tr>
<tr>
<td>other User</td>
<td>E</td>
<td>NT</td>
<td>True</td>
</tr>
<tr>
<td>other services</td>
<td>F</td>
<td>T</td>
<td>Not True</td>
</tr>
<tr>
<td>other User</td>
<td>G</td>
<td>NT</td>
<td>True</td>
</tr>
</tbody>
</table>
Voting Mining (PoW)

• The motivation of maintain NFTSBL blockchain, it has a function of PoW mining which is using voting of confirmation for each transaction.

• PoW and reward distribution are defined by NFTSBL algorithm. It selects multiple nodes for confirm a transaction node randomly.

• If the consensus was established, then all of selected nodes are received the reward of token.

• Thus, all participating nodes can get the reward opportunity if they reply the transaction confirmation within a time period.
NFTSBL network generates the reward of confirmation of transaction. And distribute the coin to the all of participant of transaction confirmation nodes.

Consensus Reward

Selected Node for consensus

A + B + C + J node are receiver the reward coin from NFTSBL.
Latency time enhancement

• To achieve low latency NFT transaction consensus, kernel level cache on node memory was implemented.

• Environment of test computer system is 1) Kernel Cache Size: 1 GB with write back policy, 2) Platform: Windows Ultimate 64-bit Edition, 3) Blockchain utilized: mix, nilu & pirl blockchain, 4) Time measurement by stopwatch. Three different blockchain 275 MB, 500 MB and 1 GB were selected.
Conclusion

• NFTSBL (Non-Fungible Token) secure blockchain using voting function with escrow account on J node is proposed.

• The system architecture was shown along with reward distribution policy. The concept of escrow account to avoid mistake of coin transfer was proposed also.

• To enhance the confirmation time, we had implemented kernel level cache.

• The evaluation of latency time in NFT blockchain was shown at implementation session. It shows around 50% reduction of time of blockchain generation at 1024MB size packet.
Any question, please contact to us.

hiro@dts-1.com