PoO - Proof of Online Synchronization

Background

Proof Of Online is used as a supplement to the Block Withholding Attack (BWA), a type of 51% attack on the block chain.

Current Problems

If an attacker acquires more than 51% of the hash power of a block chain and intentionally creates a longer block in the offline state and then brings it online, a double spending attack is possible.

To prevent this problem, we have devised a way to prove that the block is online and synchronized.

51% Attack Scenario

If the attacker gets 51% of the hash power, the attacker can have his block without having to propagate it to the network.

The network synchronization algorithm of the block chain determines the block as a final block if a block with a height higher than the height of the node exists. It is designed to discard the blocks it holds and to accept higher blocks.

The attacker uses this point. In other words, after generating more blocks offline than the external nodes and using their coins through exchanges etc., they can open their own nodes to the network and return the coins they used to their previous state.

All PoW-based coins, including existing bit coins, are potentially at risk, and coins with a small PoW hash power are exposed to the attack. In fact, some coins have been attacked such as double payment of funds due to the attack, and recent examples of such attacks have been reported.

Contents of Technology

As a way to compensate for the above risk, it is an online proof method that prevents an attacker from holding a block longer than a certain length in an off-line state.

The basic logic of this technique is as follows

Transaction

Existing block-chain transactions can be divided into coin-based transactions and general transactions.

In a general transaction, there are a sender and a receiver, the address (public key) of the sender is stored in the input portion, and the address (public key) information of the receiver is stored in the output portion. The sender signs (encrypts) the contents of the transaction with the private key, and the receiver decrypts the transaction with the public key of the sender to check whether it is a transaction created by a person who actually holds the private key of the corresponding public key.

On the other hand, the Coinbase transaction (tx[0]) is the first transaction of the coin mined and does not require signature information because only the receiver exists. However, PoO block has a new type of transaction structure called CoinOnline because only specially selected verifiers can generate blocks and signature information is needed to confirm them.

PoO Block

The height % n == 0 block of the block chain is called the PoO block and always contains the signature signed by a particular signer. In the Qcity, every 10th block, that is, xxx00, xxx10, xxx20, ... , xxx90 becomes a PoO block.

A particular signer is an account owned by a foundation or organization that initially designs the block chain or operates the block chain, and is called a verifier. They sign the PoO block with the private key of the public address to provide that the block is synchronized to the network. All other nodes accept the block by confirming that the PoO block has been signed with the public address of the foundation.

Verifier Public key

In a typical block chain chain, it uses the SeedNode method, which records the operating nodes in the program source or registers with SeedDns as the top priority, making most nodes on the network a default connection and a starting point for other networks.

In this way, the Verifier Public Key List operated by the Foundation is recorded in the program source, and the block addition task for online verification grants the signing authority only to the public key of the verifier.

How it Works

Assume that there are verifier nodes (vn [0], vn [1], vn [2], vn [3]), which is a node with a hard-coded verifier public key set in the source, and general mining nodes n[0],n[1],n[2],n[3].

The verifier node and the mining node both receive information that a new block has been created and start preparing to make the next block. The first transaction in the block is a coin-based transaction that provides compensation for the generated block. It generates the hash as its receiver, selects the remaining transactions that are in the block, and calculates the hash value. If this value is smaller than the target value, Information is complete and can be distributed. However, if the new block to be generated corresponds to the tenth block, ordinary mining nodes can not generate a coin-based transaction with itself as the receiver. On the other hand, if one of the verifier nodes, vn[0], generates a block, puts its public key at the output of the coin-based transaction and signs (encrypts) In header. To add this information, we have added a field called VchBlockSig, which is called a coin online transaction because the structure of this transaction is different from a regular coin-based transaction.

The process of block reception is as follows. If it is the 10th block, verify that the public key recorded in the output of the first (coin online) transaction is in the list of verifier nodes. Then, it is confirmed that the data obtained by decoding the contents of VchBlockSig with this public key is identical to the information of the coin on-line transaction, thereby confirming that it is a PoO block signed by a verifier.

Proof Of Online

With this logic, an attacker can only have fewer than the number of blocks defined by %n, so even if he seizes the network hash power, he can not get enough time to do a malicious double spending attack. The miners of all blocks can be identified as the corresponding block with the seed node operated by the foundation. For this reason, the name of this algorithm is called Proof Of Online.

Agreement with the miners

Since the PoO block can only be signed by the operating foundation, there is not much opportunity for the miners, so setting the appropriate %n value is necessary. (If you set the signature to be checked every 20 blocks, the miner will lose the chance of mining by 1/20). The above compensation method does not pay mining compensation for% n blocks to eliminate reverse discrimination of miners.

Preparing for Risk

The public key address, which confirms that it is online, is provided by hardcoding by default. a later verifier is added or a transfer of authority is required, a new verifier is added with a majority of the co-signatures using the authority granted only to the verifiers. And this content is stored in the client's internal database on each node.

If all the nodes operating on the foundation are stopped, all block chains expect the corresponding block to be created, and other PoW blocks are not accepted and the whole block can be made unstable.

As a way to compensate for this, the %n block is designed to accept a normal block if the %n block exceeds twice the expected block time (1 * 2 minutes for Qcity).

In the worst case, even if the operation of the entire verifier node is stopped, the block chain is delayed in mining time of the designated block, and the entire operation is not obstructed.

The way of implementation

The operator of the foundation must always maintain PoO mining at nodes that have more than a half of the verifier public keys, and the implementation method is as follows.

The first transaction tx [0] of a general block is a CoinBase transaction that is a compensation for mining, while the tx [0] of a PoO block is a variation of this and is called a CoinOnline transaction.

	CoinBase	CoinOnline
input[0]	size()== 1, prevout == null	size()==1,prevout == null
output	size()>=1	size()>=1 && size()<=3 출력[0]==empty() 출력[1].scriptPubkey== One of the provided verifier public keys 출력[2].scriptPubkey = Newly provided public keys for verification
Validation Code	vin.size()==1 &&vin[0].prevout.IsNull()	vin().size()==1&& &&vin[0].prevout.IsNull() vout.size()>=1 && vout[0].isEmtpty() && vout.size()<=3 && vout[1].scriptPubKey == One of the provided verifier public keys && vout[1].nValue== Block.nFees&& vout[2].scriptPubKey == isvalid() &&vout[2].nValue== 0

(prevout : Information from previous transactions, vin : input portion, vout : output portion)

Since the CoinOnline transaction is different from the CoinBase transaction in general, the VchBlockSig field is added to the header area of the structure of the following general block chain to contain the verifier signature information of the PoO block.

version	0250000
previous block hash	3450b3d32g3257d...
Merkle root	8a239d32kfs34hg...
timestamp	35680433
bits	2451d126
nonce	4286321
VchBlockSig	[ ]
transaction count	53
CoinBase transaction	
transaction	
...	

Header

Body

[Structure of coinbase transaction]

version	0250000
previous block hash	3450b3d32g32...
Merkle root	8a239d32kfs34...
timestamp	35680433
bits	2451d126
nonce	4286321
VchBlockSig	32kfs34hg318...
transaction count	53
CoinOnline transaction	
transaction	
...	

Header

Body

[Structure of coin-online transaction]