A study on an energy-effective and secure consensus algorithm for private blockchain systems (PoM: Proof of Majority)

1st Jun-Tae Kim
Dept. of Computer Science & Engineering
Konkuk University
Seoul, Republic of Korea
tae8579@konkuk.ac.kr

2nd Jungha Jin
Dept. of Computer Science & Engineering
Konkuk University
Seoul, Republic of Korea
drake75@konkuk.ac.kr

3rd Keecheon Kim
Dept. of Computer Science & Engineering
Konkuk University
Seoul, Republic of Korea
kckim@konkuk.ac.kr

***Abstract*— We present an energy-efficient, security-enhanced consensus algorithm to compensate for the security weaknesses that may arise when a node that generates a block is vulnerable in a closed blockchain environment. Consensus algorithms, which are usually used in blockchain environments, use a large amount of hash to create a new block, producing significant energy waste. The consensus algorithm that is proposed in this paper reduces energy waste by generating new blocks in a single operation according to the defined rules of all miner nodes that can generate blocks and converging them into blocks with many identical block values, forming a blockchain.**

Keywords— blockchain, consensus algorithm, PoM: Proof of Majority, reducing energy, private (closed) blockchain system

I. INTRODUCTION[1]

Blockchain technology involves the decentralization of existing centralized database recording and management. In this technology, multiple nodes that participate in a P2P network collectively manage transactions that are recorded by verifying and storing data. The benefit of joint record management by all nodes is that they can ensure data reliability and security and reduce maintenance costs for central servers.

Blockchain members have all transaction ledgers for the transaction. In a public (open) blockchain environment, where many unspecified parties participate, all the nodes that participate in the blockchain autonomously maintain the same transaction ledger through a consensus algorithm. In a private (closed) blockchain environment, where only authorized nodes can participate, either a block creation node is designated and operated under management or the same transaction ledger is maintained by using an agreement algorithm. Typical consensus algorithms that are used in blockchains include Proof of Work (PoW) and Proof of Stake (PoS). However, such a scheme in private (closed) blockchain environments is inefficient because of energy waste and performance problems that arise from the consumption of computing power.

In this paper, we propose a consensus algorithm that focuses on reducing energy waste, improving performance, and providing high security in a private (closed) blockchain environment.

II. CONSENSUS ALGORITHM

A. Proof of Work (PoW)[2][3]

The task proof agreement algorithm is a way for miner nodes to obtain the right to create blocks by proving their work with the hash algorithm to find a specific target. The target value is defined as Hash(B) ≤ M/D. The block B that satisfies the target value is found by using the D value for the degree of difficulty and M for the maximum value of difficulty D(2^256-1). A number of miners perform numerous operations to find a hash value that satisfies a certain target value. Only the first miner node that finds the target value will have the right to create the block. The difficulty of the target value depends on how the system is designed with the blockchain. Typically, the difficulty level of Bitcoins is adjusted to the point in time at which 2016 blocks are generated so that a block can be generated at an average cycle of approximately 10 min.[4]

B. Proof of Stake (PoS)[5]

The equity agreement algorithm is an alternative to the work proof method, which requires excessive computing power, and is a method of granting block-generation authority based on the shares that are held by the miner node. The hash function in the equity-demonstration method is defined as Hash(Hash(Bprev),A,t) ≤ bal(A)M/D. Bprev means a previously generated block, A means an account (Address), and t means a generation time (Timestamp). Additionally, bal (A) means the equity that is owned by account A, and D and M mean the same level of difficulty and maximum value of D as defined in the job proof, respectively. Block B, which is the target value, is affected by the stakes that are owned by A and the difficulty level. Having many stakes facilitates problem solving. The Proof of Stake (PoS) method can shorten the block-generation cycle and reduce energy waste. However, problems may arise regarding the initial coin distribution and attempts to converge into a single blockchain. Additionally, a node that has an initially large share that is based on ownership shares has advantages in terms of obtaining the right to create blocks, creating fairness problems.

III. PROOF OF MAJORITY (POM) CONSENSUS ALGORITHM

In this chapter, we propose a Proof of Majority (PoM) algorithm, which is a consensus algorithm that can be used

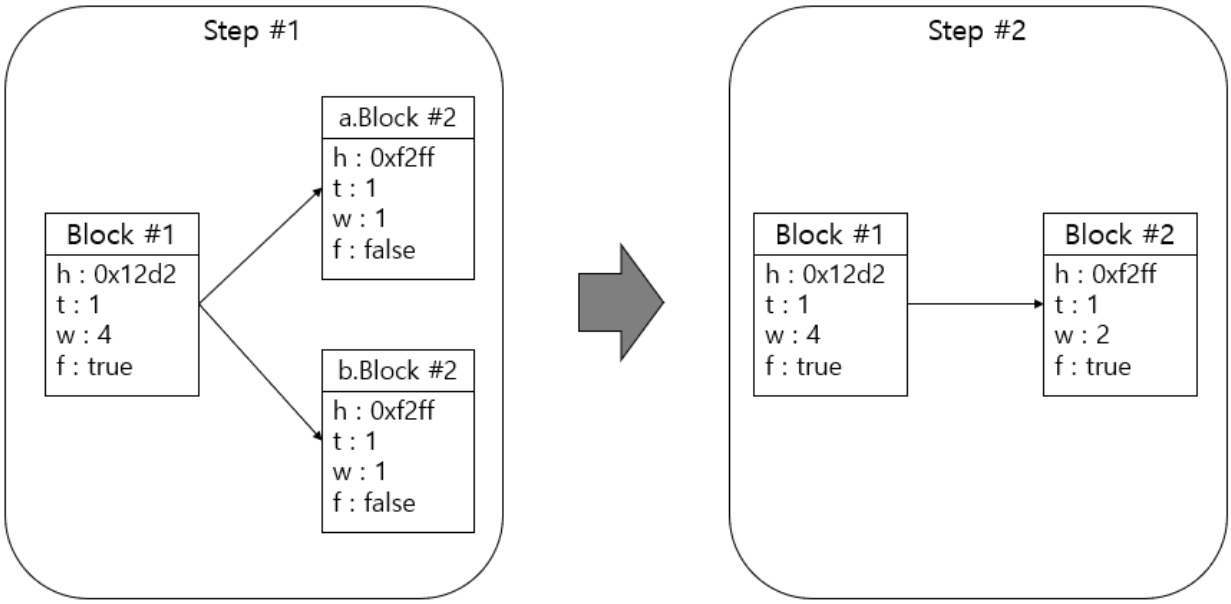

Fig. 1. Merging process when finding the same block

in a private (closed) blockchain environment to reduce energy waste and improve block-generation performance.

A. Issues

By using a consensus algorithm in a private (closed) blockchain environment, the following problems can be solved:[6]

- When all the nodes participate in the block creation, the specified block-creation node behaves abnormally.
- Problems may also arise when a block-generating node is vulnerable to security, such as generating an abnormal block by tricking an unauthorized node into a normal node.[7]

B. Proof of Majority (PoM)

The PoM consensus algorithm is a method in which all the block-generation nodes that are involved in a blockchain network simultaneously create and converge blocks into a blockchain with many identical blocks. The header structure of a blockchain with the PoM consensus algorithm is shown in Table 1 below.

The weight in the block header is incremented by one each time the same block is found and not included in the hash input when the block is created. Finalization is also not included in the hash input for block creation. The finalization is true for a fully validated block; otherwise, it is false. The miner node can only generate blocks if it owns a block whose finalize value is true. The PoM consensus algorithm creates and links new blocks in the following order:

- (1) The miner node whose finalize value of the current block header is true prepares the new block by generating a Merkle tree and sorting the hash values that correspond to each transaction in ascending order.

TABLE I. BLOCK'S HEADER STRUCTURE

Designation	Size (bytes)	Explanation
Version	4	Version of blockchain
Previous Block Hash	32	Hash value of previous block header
Merkle Root	32	Merkle root hash value of the transaction that consists of the Merkle tree
Timestamp	4	Block creation time
Weight	4	Number of identical blocks
Finalize	4	Whether block creation is confirmed

- (2) The root of the generated Merkle tree is included in the block header, a new block is created, and the block is transferred to all the connected miner nodes.
- (3) The miner nodes that received all the blocks are verified to see if each block is normal; when the same block is found, the weight of the block header is incremented by one, as shown in [Fig. 1].
- (4-1) If only one block (final block) is left, as shown in [Fig. 1], the finalize value of the block header is changed to true and the block is passed to all the miner nodes.

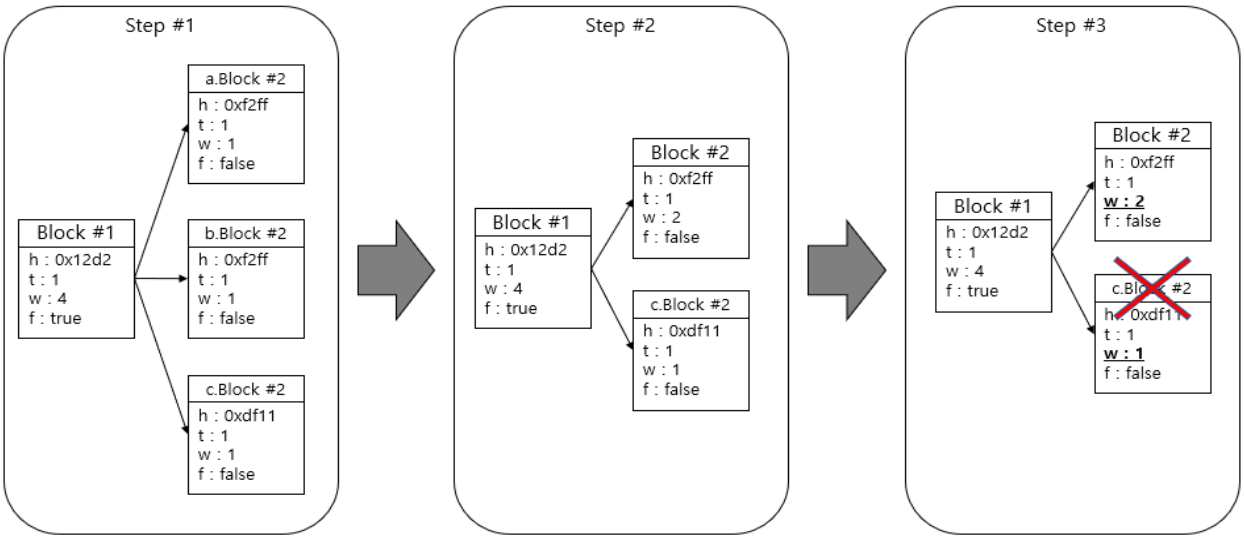

Fig. 2. Expansion of the merging process when the same block is found

- (4-2) As shown in [Fig. 2], if other blocks (a fork phenomenon where two or more child blocks with the same parent block are created) with more than two values are present in step #2, the final block is selected according to the conditions in TABLE II and then re-executed from step #3.

TABLE II. POM CONSENSUS ALGORITHM BLOCK-SELECTION CRITERIA

Priority	Block Selection Criteria
1	Block whose finalize value is true
2	Block with a high weight value
3	Block with a large number of transactions
4	Blocks with a fast timestamp

When a fork event occurs in the PoM consensus algorithm, any remaining blocks, except those blocks whose finalize value is true, are removed. If the finalize values of all the blocks are false or if two or more blocks are true, the block with the highest weight value (the same block is generated) is given priority. If a comparison is impossible, the number of transactions in the corresponding block is compared. If determining the number of transactions is impossible, the block that was created first is preferentially converged into one blockchain. [Fig. 3] shows the process of converging into a single blockchain.

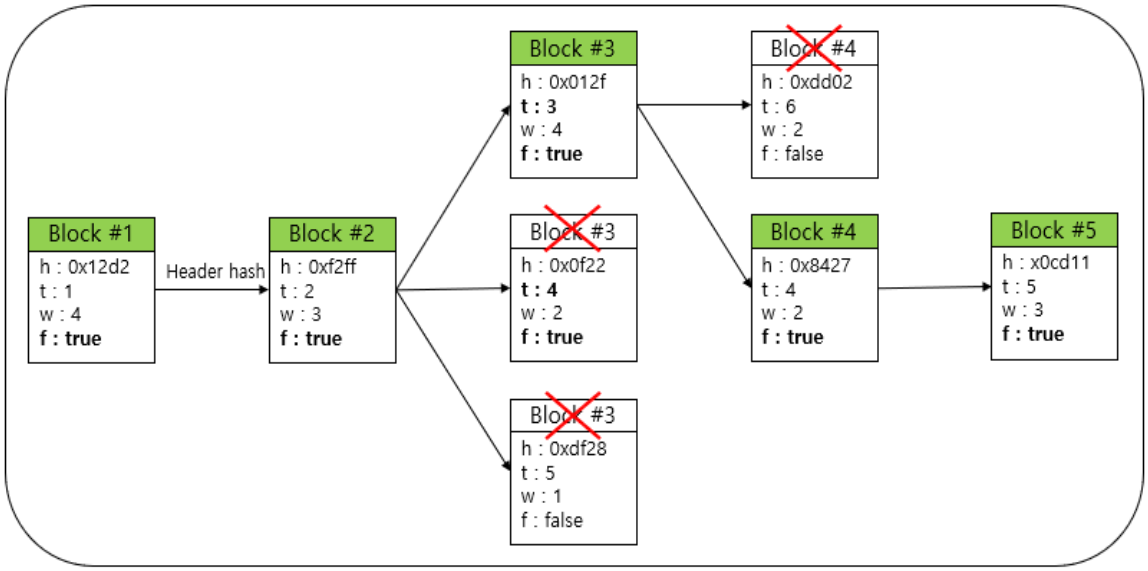

Fig. 3. Process of converging into a single blockchain

IV. CONCLUSIONS

Consensus algorithms that are used in public (open) blockchain environments consume a large amount of computing power to perform reckless hash calculations, raising energy waste and performance problems. Such issues are inefficient for use in private (closed) blockchain environments. In this paper, we proposed a Proof of Majority consensus algorithm that is available in a controlled, private (closed) blockchain environment. This algorithm does not require reckless hash calculations, which can solve problems regarding energy waste and performance. Additionally,even with a controlled, trusted, closed environment, all miner nodes participate in maintaining a single blockchain, thereby addressing potential security issues that could arise from the vulnerability of the block-generated nodes.

As a future study method, we intend to verify the performance of the Proof of Majority consensus algorithm that was proposed in this thesis. For this purpose, we intend to perform a variety of verification simulations, such as controlling the time of creation of large miner-node environments and creating abnormal blocks to verify efficiency and performance.

V. ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2014-0-00547, Development of Core Technology for Autonomous Network Control and Management) and Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2017M3C4A7083678).

REFERENCES

[1] Electronics and Telecommunications Research Institute, “Blockchain and Consensus Algorithm”, 2018.

[2] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008.

[3] M. Castro and B. Liskov, “Practical Byzantine Fault Tolerance”, 1999.

[4] https://bitcoin.org/en/developer-reference#block-chain, “Bitcoin Developer Reference”.

[5] BitFury Group, “Proof of Stake Versus Proof of Work White Paper”, 2015.

[6] Tendermint Wiki, “Byzantine Consensus Algorithm”, 2017.

[7] Grand View Research, “Blockchain Technology Market”, 2017.