future internet

Article

Hybrid Consensus Algorithm Based on Modified Proof-of-Probability and DPoS

Baocheng Wang [1,*], Zetao Li [1] and Haibin Li [2]

[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China; lzt953822835@gmail.com
[2] Department of Computer and Software Engineering, Shandong College of Electronic Technology, Jinan 250200, China; lihaibin@sdcet.cn
* Correspondence: wbaocheng@ncut.edu.cn

Received: 16 June 2020; Accepted: 22 July 2020; Published: 24 July 2020

Abstract: As the core of blockchain technology, the consensus algorithm plays an important role in determining the security, data consistency, and efficiency of blockchain systems. The existing mainstream consensus algorithm is experiencing difficulties satisfying the needs of efficiency, security, and decentralization in real-world scenarios. This paper proposes a hybrid consensus algorithm based on modified Proof-of-Probability and Delegated Proof-of-Stake. In this method, the work of block generation and validation is, respectively, completed by the nodes using the modified Proof-of-Probability consensus algorithm and Delegated Proof-of-Stake consensus algorithm. When a transaction occurs, the system sends several target hash values to the whole network. Each modified Proof-of-Probability node has a different sorting algorithm, so they have different mining priorities. Every time a hash is decrypted by a modified Proof-of-Probability node, the modulo operation is done to the value of nonce, which is then compared with the expected value given by the supernode selected by the Delegated Proof-of-Stake nodes. If they are not the same, the Proof-of-Probability node enters the waiting time and the other Proof-of-Probability nodes continue to mine. By adopting two consensus algorithms, the malicious nodes must control more than 51% of the nodes that adopt the two consensus algorithms, at the same time, to effectively attack the system, that is, they must have more than 51% of the computing power and more than 51% of the tokens. This not only increases the cost of malicious attacks, but also reduces waste of computing power. In addition, the efficiency of the DPoS algorithm makes up for the deficiency of the PoP algorithm in system efficiency, and the mining behavior based on probability in the PoP algorithm also significantly weakens the ability of supernodes in the DPoS algorithm to conduct monopoly behavior or other malicious behaviors. In a word, the combination of the two algorithms makes the system perform better in terms of security, system efficiency, and decentralization.

Keywords: blockchain; consensus algorithm; hash; transaction; voting

1. Introduction

As a distributed shared ledger and database [1], blockchain has the characteristics of openness, transparency, decentralization, and anti-tampering. With the above characteristics, blockchain is widely used in the field of finance and the Internet of Things. Among them, a rising number of blockchain-based technical implementations for smart city (e.g., references [2–4]) have been proposed. The above application scheme makes full use of the decentralization of blockchain. Decentralization is realized by consensus algorithms [5], the most representative of which [6] are PoW [7], PoS [8], and DPoS [9].

PoW ensures the safety and fairness of the system through computing power competition among nodes, but it also causes a lot of unnecessary computing power and electric power waste. In addition, with the expansion of large mines and mining pools, it is difficult for individual users to obtain benefits, which obviously violates the original intention of blockchain. In contrast, PoS does not completely rely on computing power to compete for the right to keep accounts. The difficulty of node mining and the probability of success are determined by the coin age, which effectively avoids the waste of resources and increases the cost of attacking the whole system, because the node needs to have more than 51% of tokens to carry out effective attacks. However, it also makes the system less reliable and fair. On the one hand, nodes with more tokens tend to obtain more revenue and bring more revenue in the future, which leads to a monopoly situation in PoS similar to that in PoW. On the other hand, because PoS does not completely rely on computing power to compete for bookkeeping rights, it does not consume any additional resources for the node to try to fork the blockchain, while other nodes will try to mine on all forks without any loss in pursuit of profit maximization. As a result, PoS is prone to fork. DPoS, on the other hand, continuously selects a few nodes to replace most nodes to produce blocks, which improves the production efficiency of blocks, reduces the maintenance cost of the network, and enables common nodes to have a greater opportunity to gain benefits. However, it also sacrifices some decentralization. The PoP [10] consensus algorithm weakens the advantage of nodes with high computing power in probability by introducing false hash and waiting time mechanism, but its definition of true hash is not rigorous enough, which leads to the fact that the final selection result of true hash may not be unique, resulting in the risk of the system being forked.

As can be seen from the above, a single mainstream consensus algorithm is often difficult to fully satisfy the needs of real-world scenarios (e.g., security, decentralization, and efficiency). Therefore, we hope to design a consensus algorithm to make it have the advantages of the mainstream consensus algorithms and make up for their shortcomings as much as possible. This paper proposes a hybrid consensus algorithm based on the modified PoP consensus algorithm and DPoS consensus algorithm to make up for the shortcomings of single consensus algorithms. In this algorithm, the work of block generation and verification is given to the nodes adopting the modified PoP consensus algorithm and DPoS consensus algorithm, respectively. As a result, malicious nodes must have more than 51% computing power and more than 51% tokens in the system to control the whole system and carry out effective attacks. This significantly increases the cost of malicious attacks. Among them, the modified PoP algorithm allocates the true hash and many false hashes together to the blockchain network when a transaction occurs. Each node has a different hash sorting algorithm, so the decryption order is different. Every time a node decrypts a hash, the modulo operation is done to its nonce value and is compared with the expected value given in advance by the supernode in the DPoS nodes to preliminarily verify the validity of the block. If they are not equal, the verification fails, and the PoP node enters the waiting time, and the other PoP nodes continue to mine. The supernode can select one more expected value for the next successful mining. After the verification is successful, block validity will be further verified by other supernodes, and finally, a block will be successfully generated and verified. The introduction of the waiting time mechanism and expected value mechanism greatly weakens the advantages of high computing power nodes, thus, alleviating excessive computing power competition and ensuring fairness; at the same time, it also ensures the generation speed of blocks to a certain extent. In theory, our proposed consensus algorithm has higher block generation efficiency and security than the traditional blockchain system.

The rest of this paper is organized as follows: the second section summarizes the related work, the third section explains the process and details of the hybrid consensus algorithm, the fourth section analyzes the security and liveness of the algorithm, and the fifth section evaluates the performance of the algorithm. The sixth section summarizes the whole paper and proposes the future research direction and goal.

2. Background and Related Work

This section will introduce the mainstream blockchain consensus mechanisms, such as PoW, PoS, DPoS, and the original PoP consensus algorithm in detail.

2.1. Proof-of-Work

The concept of PoW was first proposed by Cynthia Dwork and Moni Naor in their 1993 academic paper [7], while the term proof-of-work was proposed in the 1999 paper [11] published by Markus Jakobson and Ari Juels. Later, in his 2008 paper [12], Nakamoto took proof-of-work as the consensus algorithm of bitcoin. The success of bitcoin had a great impact on cryptocurrencies, and PoW has become one of the mainstream consensus algorithms.

The block structure in bitcoin is shown in Figure 1. It consists of version number, previous block hash value, Merkle root, timestamp, difficulty, nonce value, and transaction data. Due to blockchain being a decentralized distributed system, each node has the right to record transactions and package them into blocks. In order to ensure that the blocks are orderly and effective, a method is needed to make each node fairly compete for bookkeeping rights. The algorithm flow is shown in Figure 2. By giving a target hash value, the algorithm requires that the block hash value calculated by the node is less than the target value [13]. The nonce value plays a decisive role in the calculation process. Since the information in the block is known and fixed except for nonce value, we can only guess the nonce value randomly if we want to calculate the block hash value smaller than the target hash value. At the same time, the hash function is irreversible and sensitive to the input value, so no node can cheat. Nodes usually need to go through trillions of attempts to discover the nonce value. The first node to discover the nonce value is regarded as a successful miner, and obtains the cryptocurrency from the system as a reward, and its block will be regarded as effective and published on the blockchain. With the increasing difficulty of mining and the emergence of mining pools, it is almost impossible for individuals with limited resources to succeed in mining. The mining pool concentrates the computing power of multiple nodes to mine and distributes the income to each node according to the computing power proportion after successful mining. Obviously, the emergence of mining pools and mines makes it more difficult for other personal nodes to mine successfully, which obviously goes against the original intention of blockchain. In addition, the computing power of the four largest mining pools once exceeded 50% of the total network, which greatly increased the possibility of a 51% attack [14] and seriously weakened the security of bitcoin.

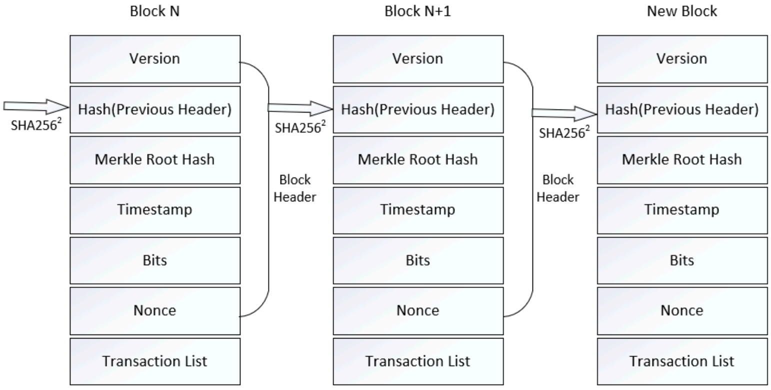

Figure 1. Bitcoin block structure.

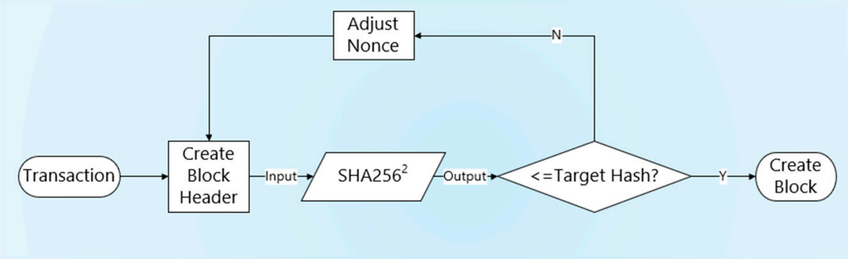

Figure 2. Algorithm flow.

Other problems with bitcoin include resources waste and inefficiency. In order to compete for bookkeeping rights, miners have to keep mining. With the increasing difficulty of mining, the miners will invest more computing resources, which not only wastes a lot of computing resources, but also leads to the increasing power consumption of mining. A recent survey found that bitcoin mining consumes 5 gigawatts of electricity per day. This is slightly less than 1% of the world's total electricity consumption, equivalent to about 4 million households. However, although the price of bitcoin has dropped by about 75%, the computing power of the bitcoin system is rising, which has reached 54 eh/s at present, meaning that the mining cost may be higher than the mining revenue in the future, making the prospect of bitcoin not optimistic. At the same time, in order to prevent hard fork and make block data difficult to be tampered with, bitcoin transaction needs to wait for six blocks to be confirmed before it can be completed. In addition, block capacity, block generation speed, and other factors further limit the transaction speed of bitcoin, which makes it difficult to put bitcoin into practical use on a large scale.

2.2. Proof-of-Stake

Based on the existing problems of PoW, PoS was first adopted in the Peercoin launched in 2012. Instead of requiring the certifier to perform a certain amount of computation, the proof of stake requires the certifier to provide a certain amount of ownership of cryptocurrency. The mechanism of proof-of-stake is that when creating a new block, the miner needs to create a "currency right" transaction, which will send some coins to the miner according to the preset proportion. According to the proportion and time of token owned by each node, and according to the algorithm, the proof-of-stake mechanism can reduce the mining difficulty of nodes, thus, speeding up the search for random numbers. This consensus algorithm can shorten the time needed to reach a consensus, but in essence, it still needs the nodes in the network to mine. Therefore, the PoS algorithm does not fundamentally solve the problem of low transaction efficiency and poor scalability of the PoW algorithm in real-world scenarios. In addition, for those nodes that hold a large number of tokens, they can gain tokens more easily than other nodes, which further consolidates the monopoly position of a few nodes [15]. As a latecomer in the blockchain system, it is almost impossible to surpass the earlier nodes in the number of coins and coin age. This also makes a large number of nodes tend to hold their own tokens rather than transacting. These defects make the whole system unfriendly to new nodes, which is not conducive to the expansion of the system.

Obviously, the shortcomings of PoW and PoS will eventually render the system centralized. Therefore, a more stable and efficient DPoS algorithm is proposed.

2.3. Delegated Proof-of-Stake

DPoS (Delegated Proof of Stake) is the fastest, most effective, most decentralized, and most flexible consensus mechanism among all consensus algorithms. The algorithm flow is shown in Figure 3. DPoS uses the right of stakeholders to approve votes to solve consensus problems in a democratic and fair way. All network parameters, from cost estimation to block spacing and transaction size, can be adjusted by selected representatives. The principle is to let each holder vote, resulting in a certain number of Representatives, which are verified and accounted for by these supernodes on behalf of the holder; the rights of these supernodes are equal. DPoS is like the board of directors voting. The coin holders cast a certain number of supernodes. The selected nodes generate blocks in turns, according to the established schedule. If a supernode fails to exercise its power properly (such as generating blocks), it will be removed, and the network will select a new supernode to replace it. All representatives will receive 10% of the transaction fee contained in the average level block for reward. The deterministic choice of block producers allows transactions to be confirmed in an average of one second. By protecting all participants from unnecessary logical checks, DPoS greatly reduces the number of nodes participating in verification and bookkeeping compared with the PoW and PoS algorithms. DPoS algorithm greatly improves the efficiency and can reach the consensus verification at the second level [16]. It does not have complete decentralization, but has weak centralization. However, the design of the DPoS algorithm does not guarantee that there must be sufficient real block producers. Due to a person or an entity may control multiple nodes, the whole system may be substantially monopolized by one entity. For example, in LBTC (lightning bitcoin), half of the nodes were controlled by the f2pool. At the same time, the governance power and economic interests of supernodes are too centralized. If they collude, they will further form a giant monopoly, which is at odds with the blockchain idea. In addition, there are many difficulties for the system to deal with the nodes. Community election cannot effectively prevent the emergence of some destructive nodes in time, which causes security risks to the network. At the same time, in the case of a small number of network nodes, the supernodes elected are not representative.

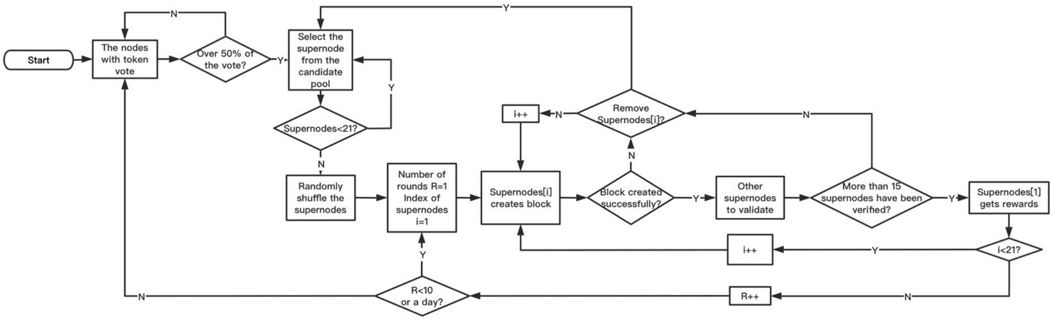
Start
The nodes with token vote
Over 50% of the vote?
Select the supernode from the candidate pool
Supernodes<21?
Randomly shuffle the supernodes
Number of rounds R=1 Index of supernodes i=1
Supernodes[i] creates block
Block created successfully?
Remove Supernodes[i]?
i++
Other supernodes to validate
More than 15 supernodes have been verified?
Supernodes[1] gets rewards
i<21?
i++
R++
R<10 or a day?
Y
N

Figure 3. DPoS (Delegated Proof of Stake) algorithm flow.

2.4. *Proof-of-Probability*

The overall structure of the PoP consensus algorithm is shown in Figure 4. In this method, each node has a different hash sorting algorithm, mixing a true hash and many false hashes together. The true hash is not determined in advance, but depends on a random number, n, that is, if a node calculates a qualified hash value, and its nonce value is expressed as a binary number, and its nth bit is 1, then, the current target hash is determined as a true hash. Otherwise, it is a false hash. Obviously, this method makes it impossible for any node to know in advance whether the hash calculated by itself is a true hash until it calculates the qualified nonce value. The first node that decrypts the true hash is regarded as the successful miner and can create blocks and obtain rewards. If the hash is a false hash, the node needs to wait for a period of time before it can start to decrypt the next hash. This is in order to limit excessive computing power competition. Through this method, the advantage of the high computing power node is weakened and the probability of successful mining of the general node is increased. However, the way to determine the true hash is not rigorous, which will lead to the selection result of the true hash being not unique, and then, cause fork. The specific algorithm flow is shown in Figure 5.

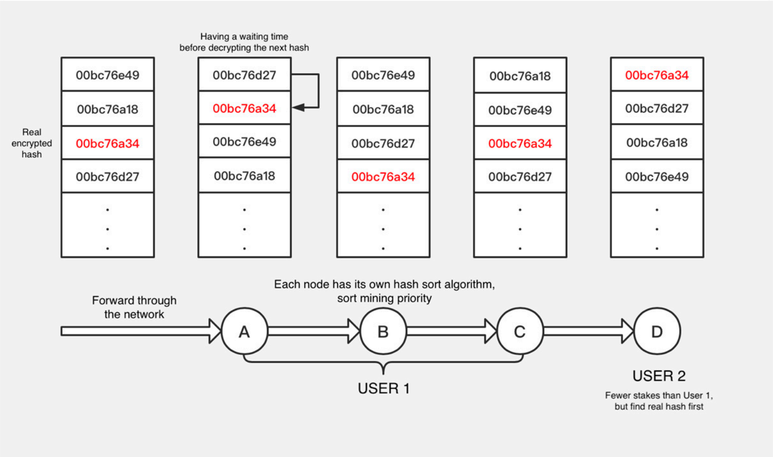

Figure 4. Overall structure of the PoP (Proof-of-Probability) algorithm.

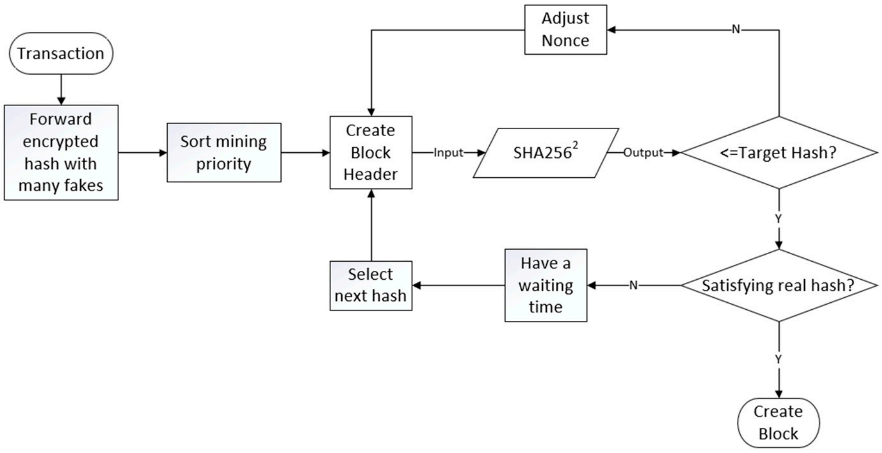

Figure 5. PoP (Proof-of-Probability) algorithm flow.

2.5. Requirement Analysis

Based on a review of the literature, we consider that a good blockchain consensus algorithm should satisfy as many of the following requirements as possible.

Bookkeeping right: Fair competition of bookkeeping rights is the basis of blockchain system security and fairness. For the mining-based consensus algorithm, it needs to solve mathematical problems to compete for accounting rights, and the higher difficulty helps to create fair competition. Difficult mathematical problems that are difficult to solve but easy to verify can make the system difficult to be cheated, and the system can withstand a large number of nodes. For other types of consensus algorithms, it is not easy to guarantee security and fairness at the same time.

Resource saving: Low computing power consumption and electric power consumption can not only reduce the waste of resources, but also reduce the operating cost of the node, which can effectively avoid the situation that the cost is too high or even exceeds the income. In addition, it also helps to deploy lightweight devices with low computing power in the blockchain.

High efficiency: In order to be used in a wider range of real-world scenarios, higher system efficiency (e.g., transaction throughput) is necessary.

Decentralization: As the core feature of blockchain technology, the decentralized degree of consensus algorithm determines the security (e.g., resistance of single point of failure) of the blockchain system and the ability to prevent monopoly and protect user privacy.

Scalability: Due to the number of nodes in the blockchain being not fixed, the efficiency of the blockchain system should not decrease significantly with the increase in the number of nodes.

Punishment mechanism: In order to maintain the security and stability of the system, the system should have a punishment mechanism to stop malicious behavior in time, and it is best to remove the malicious nodes from the network.

Resistance to temporary fork: In order to maintain the uniqueness of blockchain data, the consensus algorithm should be able to avoid or effectively deal with the situation of fork. This paper only discusses the fork caused by two nodes producing blocks at the same time.

Resistance to various attacks: To establish a secure network environment, the consensus algorithm should make the system resistant to various attacks (e.g., 51% attack, distributed denial of service attack, selfish mining attack and traditional sybil attack).

3. System Model and Algorithm

Through the analysis of the above requirements and the mainstream consensus algorithm, we consider that the demand that a single consensus algorithm can meet is very limited, so we try to combine the two consensus algorithms to make up for their shortcomings.

The basic concept design of the algorithm proposed in this paper will be given in Section 3.1; the specific algorithm flow and details will be given in Section 3.2.

3.1. Design

Through the analysis of various mainstream consensus algorithms, it is not difficult to realize that there is a trade-off among three desirable properties, namely decentralization, efficiency and resources saving, and security. For example, although PoW is highly decentralized and safe, it is inefficient and consumes a lot of resources. Although DPoS is efficient, it sacrifices security and decentralization in part. Therefore, using a single consensus algorithm to control the whole block generation process is difficult to achieve satisfactory results. Therefore, this paper comprehensively considers the advantages and disadvantages of PoP and DPoS, and uses them in the process of mining and block verification, in order to improve efficiency as much as possible under the premise of ensuring security.

In the original PoP algorithm, the true hash is not determined in advance, but depends on a random number, n, that is, if the nonce value of a hash is represented by a binary number and the nth bit is 1, then, the current target hash is determined as the true hash. However, this way of determining the true hash is not rigorous, the result may not be unique, and there is a greater risk of fork. Therefore, in view of this defect, the following improvements are proposed for the PoP:

Before each round of mining, the supernode of DPoS needs to choose an expected value between 0 and 9. Every time the PoP node decrypts a hash, it modules the nonce value to 10. The reason for choosing 10 is that we want to make the shortest time for successful verification of a block equal to that of an existing mainstream blockchain system (e.g., bitcoin system) to ensure its practicability. If the selected value is too small, it cannot effectively restrain multiple nodes with high computing power, and if the value is too large, the system's efficiency will be affected too much. If the result is the same as the expected value selected by the DPoS supernode, it is considered to have decrypted the true hash. The initial verification of the block is successful.

According to the number of tokens that DPoS nodes hold, they vote to select supernodes to make decisions instead of themselves. In our proposed algorithm, the token acquisition method is basically the same as that of the PoS-based blockchain system, that is, nodes need to purchase to obtain tokens. By introducing a sublinear function $Y = \sqrt[2]{X}$ (Y is the corresponding number of votes and X is the number of tokens held by the node), this limits the voting weight of nodes with many tokens. All the voted nodes in each round form a candidate pool, from which 72 nodes with the most votes are selected as supernodes. The reason why 72 nodes are selected as supernodes is that in the BitShares project, which adopts the DPoS consensus mechanism, the number of supernodes is set to 101. However, according to the observations of community members, there are not so many ideal nodes that can be voted for, so we choose a relatively small number of 72 nodes. If the total number of nodes is less than 72, all nodes will automatically become supernodes. After scrambling the sequence, a block is successfully verified in turn. Blocks that are not generated in sequence are considered invalid. If there is a supernode that does not act or does evil, it will be replaced by a new supernode selected from the candidate pool according to the number of votes. If the number of supernodes is less than 72, no new supernode will be added after the malicious supernode is removed until new nodes join the system. The bits value is dynamically adjusted by the bit adjustment algorithm [12] to ensure that a hash is decrypted every minute. In fact, the time to decrypt the hash can be dynamically controlled according to specific needs. The supernode can specify an expected value for the first time. If it is not correct, the PoP node which decrypted the hash will enter the waiting time. The supernode can specify an additional expected value the next time a hash is decrypted, and so on. Ideally, each block that meets the expected value can quickly pass the verification of more than half of the supernodes, so it can ensure that a block can be generated for up to ten minutes. Every time a block is successfully created, 60% of the total revenue is allocated to the successful mining PoP node and 20% of the revenue is allocated to the DPoS supernode of the current verification block. This 20% of the revenue is allocated to the general DPoS nodes voting for the supernode.

3.2. Main Algorithm

The algorithm flow of the proposed consensus algorithm is shown in Figure 6.

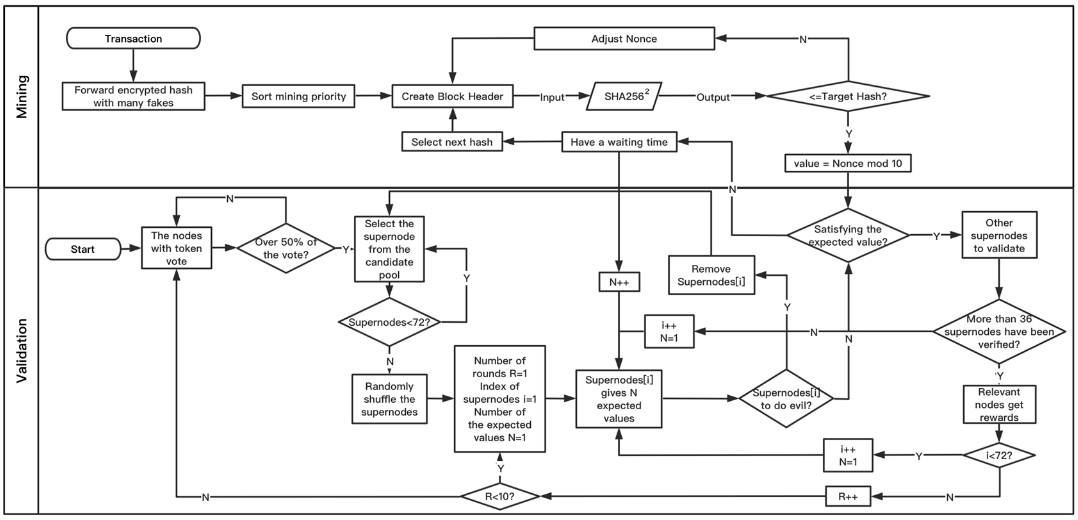
Mining
Transaction
Forward encrypted hash with many fakes
Sort mining priority
Create Block Header
Input
SHA256²
Output
<=Target Hash?
Adjust Nonce
N
Y
value = Nonce mod 10
Select next hash
Have a waiting time
Validation
Start
The nodes with token vote
Over 50% of the vote?
Y
N
Select the supernode from the candidate pool
Supernodes<72?
Y
N
Randomly shuffle the supernodes
Number of rounds R=1 Index of supernodes i=1 Number of the expected values N=1
Supernodes[i] gives N expected values
N++
Remove Supernodes[i]
i++ N=1
Supernodes[i] to do evil?
Y
N
Satisfying the expected value?
N
Y
Other supernodes to validate
More than 36 supernodes have been verified?
N
Y
Relevant nodes get rewards
i<72?
Y
i++ N=1
N
R++
R<10?
Y
N

Figure 6. Main algorithm.

1. When the transaction occurs, the whole network publishes a true hash and many false hashes.
2. Each PoP node uses its own hash sorting algorithm to give the mining order.
3. The PoP node creates the block header and exhausts the nonce value for mining.
4. If a hash is decrypted, the nonce value is compared with the expected value given by the supernode of DPoS. If they do not match, the PoP node needs to enter the waiting time (the waiting time depends on the number of PoP nodes. If the number of nodes is less than 10, the waiting time is 1 min. If the number of nodes is greater than or equal to 10, the node needs to wait until a block is successfully verified, i.e., at most wait for ten minutes) and mining should be continued by other PoP nodes. When the next hash is decrypted, the current supernode can specify one more expected value, until the tenth time, when ten expected values are specified; it must be consistent. If it is consistent, it shall be submitted to other supernodes for further verification. If more than half of the supernodes pass the verification, it shall be deemed as successful mining (it should be noted that the nodes that generate blocks and pass all validation should also enter the waiting time to prevent the same high computing power node from generating blocks continuously). Otherwise, this PoP node will enter the waiting time, and other PoP nodes will continue mining. After a block is generated and verified successfully, the next supernode will verify when the next hash is decrypted.
5. DPoS supernodes are elected through calculating the corresponding votes of each DPoS node by using the sublinear function according to the number of tokens they own (not one vote for one person, which increases the cost of controlling votes, and is not linearly related to the number of tokens, which limits the voting weight of nodes with many tokens). If more than half of the tokens have voted, 72 nodes with the most votes will be selected as supernodes from the candidate pool.
6. After randomly disrupting the order of supernodes, each supernode validates the block by giving the expected value in turn. If the node does something bad, it is removed from the system by voting, and a node is selected from the candidate pool to replace it. After ten rounds, they vote again to select the supernodes.

4. Analysis of Algorithm

Due to the consensus algorithm proposed in this paper combining PoP and DPoS, this makes the algorithm have the advantages of both algorithms. We will analyze the algorithm from two aspects of security and liveness. The security and liveness analyses are as follows.

4.1. Security Analysis

In addition to the advantages of tamper proof and traceability of the traditional consensus algorithm, the security of the algorithm is mainly reflected in: supernodes election security, mining security, and system security.

4.1.1. Supernodes Election Security

DPoS nodes vote for supernodes by holding tokens, not by one node one vote, and the voting end condition is more than 50% of the total token holding nodes having voted, which makes the election process not susceptible to general Sybil attack. At the same time, by introducing formula (1), the number of votes held by each node is not linearly related to the number of tokens held, which effectively limits the nodes with many tokens and avoids monopoly.

$$Y = \sqrt[2]{X} \tag{1}$$

4.1.2. Mining Security

By introducing a random expected value and waiting time, and using the bit adjustment algorithm [12] to adjust the speed of the hash decryption, even if the malicious node has a very high computing power, it cannot decrypt the true hash faster than other nodes and cannot guarantee stable block generation speed. At the same time, even if the high computing power node decrypts the hash, according to formula (2), it can be concluded that the average probability of each result being the same as the expected value given by the supernode is only 55% at most, otherwise it will enter the waiting time and continue mining by other nodes. This greatly weakens the advantages of high computing power nodes probabilistically and avoids excessive computing power competition.

$$\overline{P_n} = \begin{cases} \frac{1}{10(n-1)} \sum\limits_{i=1}^{10} P_i, n < 10 \\ \frac{1}{10} \sum\limits_{i=1}^{10} \frac{P_i}{n-i+1}, n \geq 10 \end{cases} \tag{2}$$

4.1.3. System Security

Based on the above reasons and the work of generating blocks and verifying blocks being respectively allocated to two types of nodes, even if malicious nodes have more than 51% of the computing power and 51% of the tokens, they may not be able to attack the system effectively. Therefore, the hybrid consensus mechanism proposed in this paper has higher attack costs and significantly reduces the risk of attacks. Separating the block generation and block verification and assigning them to the two types of nodes can resist various traditional attacks to some extent, such as Byzantine failures [17], the eclipse attack [18], selfish mining attack [19], and other double-spend attacks [20]. At the same time, it can resist DDoS attack [21] as well as the bitcoin system can. In addition, through relatively stable income, high attack cost, and uncertainty of probability, collusion attack is restrained from motivation. The above measures make the nodes in the consensus algorithm proposed in this paper tend to do no evil in terms of cost and benefits (the interests of the two types of nodes responsible for block generation and verification are irrelevant). Compared with the traditional blockchain system, this increases the robustness of the system.

4.2. Liveness Analysis

The algorithm proposed in this paper is mainly reflected in the following aspects: the completion of supernode election in a limited time, the low consumption of supernode verification work, the successful mining of PoP node in a limited time, and the stable block generation efficiency of the system.

Due to the supernode having to complete the election within a limited time or when the number of votes reaches the standard, this can ensure that the supernode election ends in time. As the supernode gives the expected value in advance, and the verification of the correctness of hash calculation results can be completed by other supernodes immediately, block verification only takes a short time. The introduction of the bits dynamic adjustment algorithm and expected value mechanism limits the upper and lower bounds of block generation speed. The fastest is about 1 min, and the slowest is about 10 min. Therefore, the algorithm has good liveness.

5. Evaluation and Results

We evaluate the algorithm design extensively on the local cluster. First of all, the blockchain system is implemented simply by python programming language, and the consensus mechanism is modified according to the design. The main functions, such as supernodes election, PoP nodes mining, dynamic addition of nodes, supernode verifying blocks, and submitting them to the blockchain are simply realized. Then, we evaluate the supernodes election time, transaction throughput, and scalability of the system. Finally, we compare the experimental data with the mainstream algorithm. In order to facilitate programming and comparison, the environment configuration and some parameters are uniformly given a random value of the same range.

The implementation of node communication in the algorithm is realized through Python socket.

5.1. Supernodes Election

As a hybrid algorithm, the modified DPoS algorithm used in the block verification work is basically the same as the traditional DPoS algorithm when electing nodes. The only difference is the number of supernodes that are elected. We set the number of supernodes of the hybrid algorithm as 72, and the number of supernodes of the DPoS algorithm as 21 and 101. The supernodes election time is calculated by the time stamp returned by the system. After several rounds of experiments, 25 of them were randomly selected. The experimental results are shown in Figure 7.

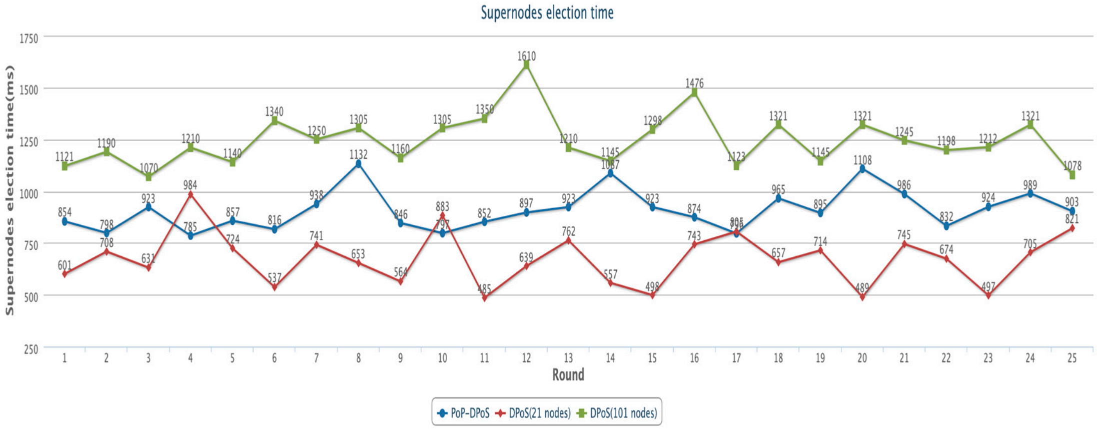

Figure 7. Supernodes election time.

According to the experimental data, in the same network configuration, the election time of this algorithm and the DPoS algorithm are generally in the same order of magnitude, that is, it is basically only affected by the network delay and the number of nodes.

5.2. Transaction Throughput

Throughput is a key indicator of the efficiency of consensus algorithms. By using the bits dynamic adjustment algorithm and expected value mechanism, the algorithm can ensure that its throughput is roughly the same as that of bitcoin in the worst case, while avoiding excessive computing power competition. In the process of system implementation, for comparison, we recorded the throughput under the same premise. Then, randomly we selected 20 rounds of all experiments. The experimental results are shown in Figure 8. From this, we can conclude that the transaction throughput of the algorithm is better than the PoW algorithm in general, and its average transaction throughput is about three times that of the PoW algorithm.

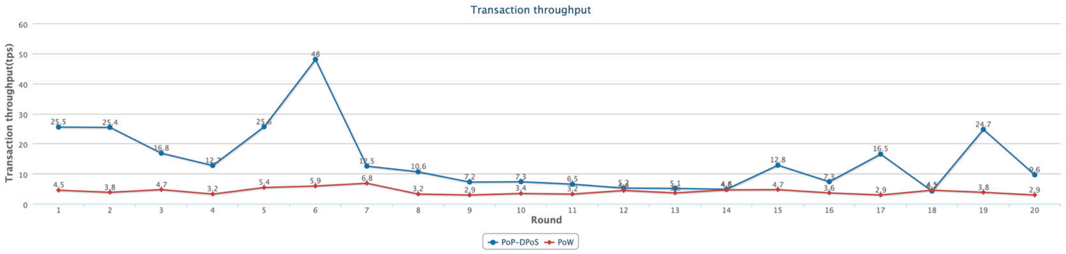

Figure 8. Transaction throughput.

It should be noted that, because the expected value mechanism operates based on probability, the throughput of the system in real-world scenarios is uncertain and floating randomly within a fixed range.

5.3. Scalability

The algorithm proposed in this paper implements a multi-node consensus process. In theory, the algorithm will have good scalability. In order to verify the scalability of the algorithm, we designed an experiment to explore the relationship between the number of nodes and transaction throughput. The experiment compares the transaction throughput of the blockchain system based on this algorithm and the PoW algorithm under the conditions of 18 nodes, 36 nodes, 72 nodes, and 144 nodes. In each case, we did 20 sets of experiments and took the average of the throughput. The experimental results are shown in Figure 9.

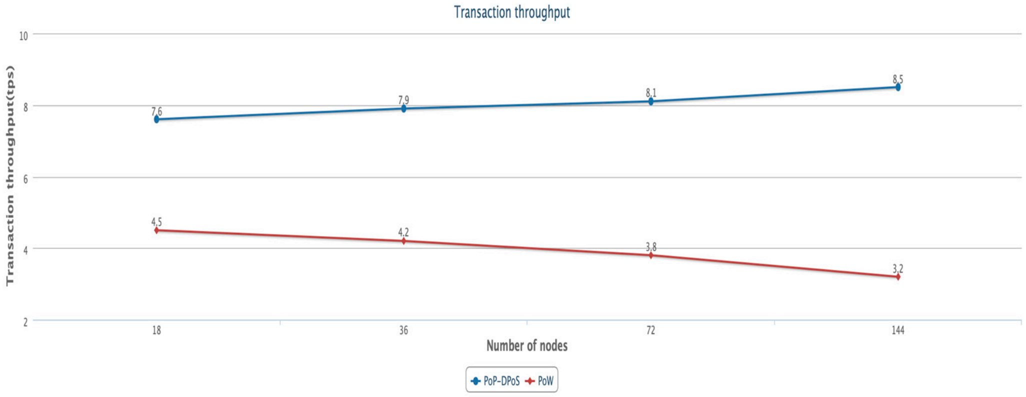

Figure 9. Scalability.

It can be seen from the above figure that the PoW algorithm's throughput will decrease with the increase in nodes, which is usually caused by the increase in the difficulty of hash calculation and the increase in consensus time. In contrast, the algorithm proposed in this paper has relatively low throughput only in the case of an increase in the number of elections caused by a small number of supernodes. As the number of nodes increases, its throughput will gradually increase and stabilize, that is, the throughput of the algorithm is not easily affected by the increase in the number of nodes. In general, under the same number of nodes, the throughput of this algorithm is basically higher than that of the PoW algorithm.

After algorithm analysis and performance evaluation, we explain how our proposed consensus algorithm meets all the requirements mentioned in Section 2.5.

Bookkeeping rights: In the consensus algorithm proposed in this paper, due to the introduction of the waiting time mechanism and the expected value mechanism, even if the node with high computing power first calculates the nonce value, it has very small probability of matching the expected value, and then, it will enter the waiting time. This significantly weakens the advantages of high computing power nodes, so that other general nodes have a greater opportunity to obtain bookkeeping rights. This effectively avoids excessive computing power competition. At the same time, due to the special setting of the expected value mechanism, block generation efficiency can also be guaranteed, that is, a block can be generated in one minute at the fastest time and in ten minutes at the slowest time (the bitcoin system generates a block every ten minutes). Therefore, the proposed algorithm can provide fair and efficient bookkeeping rights competition.

Resource saving: Due to the algorithm providing fair and reasonable bookkeeping rights competition, the opportunity for nodes to obtain bookkeeping rights is not linearly related to its computing power, so nodes will not excessively pursue high computing power, which obviously avoids excessive computing power competition and reduces the waste of computing power and electric power. Thus, the running cost of nodes is reduced. After further optimization, the algorithm can also be deployed on some lightweight devices. Therefore, the proposed algorithm has the characteristics of resource saving.

High efficiency: By introducing the expected value mechanism, block generation speed is significantly improved compared with the bitcoin system. In the worst case, it has the same generation speed as the bitcoin system, and in the best case, it is ten times as fast as the bitcoin system. In fact, the bits dynamic adjustment algorithm can shorten the mining time and further improve block generation speed. Therefore, the efficiency of this algorithm is better than that of the bitcoin system.

Decentralization: In PoW and other mining-based consensus algorithms, the competition for bookkeeping rights seems fair, but it will be monopolized by high computing power nodes. In a consensus algorithm such as DPoS, the supernodes obtained by election can be used for bookkeeping. Although the competition of computing power is avoided and the system efficiency is improved, they have almost all the power (i.e., the power of block generation and verification), which to a certain extent, sacrifices decentralization and is easy to form a new monopoly. Thus, by combining PoP and DPoS, we can not only compete for bookkeeping rights fairly through proper mining, but also weaken the power of DPoS supernodes (i.e., the power to generate blocks is handed over to PoP nodes), and only block verification can be carried out through them. Block verification by a small number of nodes inherits the high efficiency of DPoS to a certain extent, and effectively avoids the occurrence of monopoly. Therefore, the proposed consensus algorithm can provide good decentralization for the blockchain system.

Scalability: In the blockchain system based on the PoW algorithm, the throughput usually decreases with the increase in the number of nodes. As time goes on, excessive computing power competition will lead to the increasing difficulty of target hash calculation, and it will take more time for the system to reach a consensus due to the increase in the number of nodes. In contrast, the proposed algorithm has a relatively low throughput only when the number of supernodes is small, which leads to the increase in election times. With the increase in the number of nodes, its throughput will gradually improve and tend to be stable. Due to block verification being only carried out by a small number of DPoS supernodes, it takes less time for the blockchain system to reach a consensus, that is, the throughput of the algorithm is not easily affected by the number of nodes. Therefore, with the same number of nodes, the throughput of the proposed algorithm is basically higher than that of the PoW algorithm.

Punishment mechanism: In the blockchain system represented by bitcoin, there is usually no good supervision and punishment mechanism, which means malicious nodes will not be punished even if they carry out malicious attacks, which obviously increases the security risk of the blockchain system. In the proposed algorithm, DPoS supernodes are still monitored by other DPoS nodes after being elected. Once a supernode is found to have malicious behavior, other nodes will vote to remove the supernode from the system. In addition, the algorithm also provides a similar punishment mechanism for PoP nodes. Once a PoP node has malicious behavior, it will be immediately removed by voting. Therefore, the proposed algorithm has a basic punishment mechanism.

Resistance to temporary fork: In most blockchain systems such as the bitcoin system, once two nodes generate blocks at the same time, it will generally cause a short-term blockchain fork. Other nodes will continue mining on the first received blockchain until one blockchain is longer than the other, and the short blockchain will be replaced. In the proposed algorithm, the method to deal with fork is the same as bitcoin. The difference is that our algorithm makes it more difficult for blockchain system to fork than the bitcoin system. In our blockchain system, when a node generates a block, it needs to further meet the expected value before it can pass block verification. The minimum success rate is 10% and the average success rate is 55%. Therefore, when two nodes generate blocks at the same time, they also need to meet the expected value before fork occurs. By calculation, we can see that the minimum success rate is 1%, and the average success rate is 30.25%. Therefore, the proposed algorithm can make the blockchain system much more difficult to fork than the general blockchain system.

Resistance to various attacks:

(1) 51% attack: For the blockchain system based on a single consensus algorithm, malicious nodes usually only need 51% of the computing power or 51% of the token in the system to launch 51% attacks. In the proposed algorithm, due to the combination of two consensus algorithms, the malicious node must have 51% of the computing power and 51% of the token in the system to launch an effective malicious attack, which greatly increases the cost of malicious attacks and improves the security of the system.

(2) DDoS attack: In fact, blockchain technology itself can well resist traditional DDoS attacks. This is because the traditional centralized system will absorb a lot of spam information, which may lead to DDoS attacks, and the nature of DDoS attacks makes it almost impossible to achieve the bandwidth needed to process these data. A decentralized blockchain platform allows users to rent their bandwidth, which can be pooled, thus, greatly increasing the amount of data processing and greatly reducing the risk of DDoS attacks. At the same time, for the proposed algorithm, due to the use of the waiting time mechanism and the expected value mechanism, it is easy to verify the validity of the request, so a large number of malicious requests will be easily rejected, and at the same time, force the malicious nodes to enter the waiting time, which significantly reduces the frequency of malicious attacks, and then, enables the blockchain system to effectively resist DDoS attacks.

(3) Selfish mining attack: In the general blockchain system, malicious nodes may choose not to disclose their own generated blocks, but to continue mining on their own branch. When the length of the public branch exceeds that of the private branch, the malicious node will publish the private branch to make the mining of the honest node invalid. However, in the proposed algorithm, due to the introduction of the waiting time mechanism and expected value mechanism, block generation only depends on computing power competition to a small extent, and more depends on probability. Therefore, malicious nodes usually do not have the ability to create private branches. In addition, even if a malicious node can generate blocks earlier, it must be disclosed, compared with the expected value, and matched to be considered as a valid block. Therefore, the malicious node can't hide blocks and guarantee their validity. Moreover, it will be regarded as malicious behavior to disclose a blockchain containing several invalid blocks later.

Then, the malicious node will be removed by voting. Therefore, the proposed algorithm can effectively prevent a selfish mining attack.

(4) Sybil attack: In the blockchain system, a single malicious node may disguise as multiple nodes to control the whole network and conduct malicious behavior. In the proposed algorithm, due to the combination of PoP and DPoS, no matter what kind of node the malicious node wants to disguise as, it needs to pay a price. That is, if the malicious node disguises as a PoP node, it must have certain computing power and carry out mining. Only if the node mines successfully can it be regarded as an effective node. If it is disguised as a DPoS node, it must have a certain number of tokens to participate in voting. Obviously, the cost of a Sybil attack by a malicious node is as high as that of 51% attack. Therefore, the proposed algorithm can also better resist Sybil attacks. It is worth mentioning that this algorithm can be further improved to better resist Sybil attacks. For example, adding an authentication mechanism can make use of third-party trusted organizations such as Oracle for authentication, but this will sacrifice the decentralization of the system and the anonymity of nodes. Therefore, a guaranteed algorithm can also be used to ensure that any new node must be guaranteed by the trusted node in advance, which can be used for identity authentication under the premise of ensuring anonymity.

To sum up, the comparison between our consensus algorithm and other consensus algorithms is shown in Table 1.

Table 1. Comparison between our proposal with other consensus algorithms.

Feature	PoP-DPoS (Proposed)	PoW	PoS	DPoS	PoP
Bookkeeping right	✓✓✓	✓	✓	✓✓	✓✓
Resource saving	✓✓	×	✓	✓✓✓	✓
High efficiency	✓✓	×	✓✓	✓✓✓	✓
Decentralization	✓✓✓	✓✓	✓✓	✓	✓✓✓
Scalability	✓✓✓	✓	✓✓	✓✓✓	✓✓
Punishment mechanism	✓✓	×	✓✓	✓✓	×
Resistance of temporary fork	✓✓	✓	✓	✓✓✓	✓
Resistance of various attacks	✓✓✓	✓	✓✓	✓✓	✓

6. Discussion

This paper proposes a hybrid consensus algorithm based on modified PoP and DPoS. By adopting two different consensus algorithms for block generation and verification, the system combines the advantages of the two consensus algorithms. The superiority of high computing power nodes is weakened by the fake hash and the waiting time mechanism introduced in PoP, and excessive computing power competition is avoided. DPoS greatly improves the efficiency of block generation and verification, but it also sacrifices decentralization to some extent. Each node can choose the appropriate work according to its own situation, and obtain more stable income. Most importantly, this method increases the cost of a 51% attack. An attacker needs to have both 51% of the computing power and 51% of the tokens to effectively attack the system. It effectively prevents various double-spend attacks. The optimization of the waiting time mechanism and the bit adjustment algorithm mentioned in this method will be the main research directions in the future. Due to the introduction of a sublinear function, the voting weight of some nodes holding a small number of tokens is more than that of a single node holding a large number of tokens, which in another aspect, leads to the relatively weak resistance of the consensus mechanism to attacks similar to Sybil attacks [22]. Related improvement measures will also be the main work in the future. In addition, the further management of the transaction by the PoP node can resist a dusting attack. We will also compare the performance of this method with other consensus mechanisms in extreme cases.

Author Contributions: Writing—original draft, B.W.; writing—review and editing, Z.L.; resources, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research (Project NO. 110052971921/005/021) was funded by "The Fundamental Research Funds for Beijing Universities".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shen, X.; Pei, Q.Q.; Liu, X.F. Survey of block chain. *J. Netw. Inf. Secur.* **2016**, *2*, 11–20.
2. Hosen, S.; Singh, S.; Sharma, P.K.; Ghosh, U.; Wang, J.; Ra, I.H.; Cho, G.H. ABlockchain-Based Transaction Validation Protocol for a Secure Distributed IoT Network. *IEEE Access* **2020**, *8*, 117266–117277. [CrossRef]
3. Malik, A.A.; Tosh, D.K.; Ghosh, U. Non-Intrusive Deployment of Blockchain in Establishing Cyber-Infrastructure for Smart City. In Proceedings of the 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA, 10–13 June 2019; pp. 1–6. [CrossRef]
4. Singh, P.; Nayyar, A.; Kaur, A.; Ghosh, U. Blockchain and Fog Based Architecture for Internet of Everything in Smart Cities. *Future Internet* **2020**, *12*, 61. [CrossRef]
5. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals. Problem. *ACM Trans. Program. Lang. Syst.* **1982**, *4*, 382–401. [CrossRef]
6. Conti, M.; Kumar, E.S.; Chhagan, L.; Ruj, S. A survey on security and privacy issues of bitcoin. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3416–3452. [CrossRef]
7. Dwork, C.; Naor, M. Pricing via Processing, Or, Combatting Junk Mail. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 1992; Springer: Berlin, Germany, 1992; pp. 139–147.
8. King, S.; Nadal, S. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake [EB/OL]. 5 March 2012. Available online: https://peercoin.net/assets/paper/peercoin-paper.pdf (accessed on 15 May 2018).
9. Dantheman. DPOS Consensus Algorithm—The Missing White Paper [EB/OL]. 2017. Available online: https://steemit.com/bitshares/@.testz/bitshares-history-delegated-proof-of-stake-dpos (accessed on 15 November 2019).
10. Sungmin, K.; Joongheon, K. POSTER: Mining with Proof-of-Probability in Blockchain. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS' 18), Incheon, Korea, 4–8 June 2018; Association for Computing Machinery: New York, NY, USA; pp. 841–843.
11. Tromp, J. Cuckoo Cycle: A memory bound graph-theoretic proof-of-work. In *Financial Cryptography and Data Security*; BITCOIN 2015; Springer: Berlin, Germany, 2015; pp. 49–62.
12. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical Report. Available online: http://bitcoin.org/bitcoin.pdf (accessed on 21 October 2019).
13. Antonopoulos, A.M. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*; O' Reilly: Sebastopol, CA, USA, 2014; p. 5760.
14. BTC.com. Mineral Pool Statistics [EB/OL]. 13 May 2018. Available online: https://btc.com/stats/pool?pool_mode=day (accessed on 13 May 2018).
15. Houy, N. It will cost you nothing to 'kill' a proof-of-stake. *Cryptocurrency* **2014**, *34*, 1038–1044.
16. Xia, Q.; Zhang, F.J.; Zuo, C. Review for consensus mechanism of. cryptocurrency system. *Comput. Syst. Appl.* **2017**, *26*, 1–8.
17. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In Proceedings of the OSDI, New Orleans, LA, USA, 22–25 February 1999; Volume 99, pp. 173–186.
18. Nayak, K.; Kumar, S.; Miller, A.; Shi, E. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. In Proceedings of the IEEE European Symposium on Security & Privacy, Saarbrücken, Germany, 21–24 March 2016; IEEE: Piscataway, NY, USA, 2016.

19. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin Mining is Vulnerable. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 3–7 March 2014; pp. 436–454.
20. Karame, G.; Androulaki, E.; Capkun, S. Two bitcoins at the price of one? Double-spending attacks on fast payments in bitcoin. In Proceedings of the Conference on Computer & Communications Security, Raleigh, NC, USA, 16–18 October 2012; pp. 906–917.
21. Mirkovic, J.; Robinson, M.; Reiher, P.; Oikonomou, G. *Distributed Defense against DDOS Attacks*; University of Delaware CIS Department Technical Report CIS-TR-2005-02; University of Delaware: Newark, DE, USA, 2005.
22. Douceur, J.R. The Sybil Attack. In Proceedings of the Peer-to-Peer Systems, First International Workshop (IPTPS), Cambridge, MA, USA, 7–8 March 2002; Springer: Berlin, Germany, 2002.