Proof of Reputation: A Reputation-based Consensus Protocol for Blockchain Based Systems

Qianwei Zhuang
Software College
Northeastern University
Shenyang, Liaoning, China
zhuangqianlong@foxmail.com

Yuan Liu
Software College
Northeastern University
Shenyang, Liaoning, China
liuyuan@swc.neu.edu.cn

Lisi Chen
Inception Institute of Artificial Intelligence
Abu Dhabi, UAE
chenlisi.cs@gmail.com

Zhengpeng Ai
Software College
Northeastern University
Shenyang, Liaoning, China
im@aizhengpeng.cn

ABSTRACT

A consensus mechanism serves as a basic and essential component of a blockchain based system (e.g. an IoT system) and constructs the decentralized trustworthiness between the decentralized nodes/things in the system network. The design of the consensus mechanism strongly impacts the blockchain system performance, including transaction capacity, scalability, and security. The Traditional consensus mechanisms (i.e., PoW and PoS) have encountered great challenges and require further investigation. In this work, a proof of reputation consensus mechanism is proposed, where the reputation of a node is constructed based on its asset, transaction activity, and consensus participation. In the proposed mechanism, a new block is generated by the leader node with the highest reputation and the new block is validated and confirmed through the reputation based voting. The rewards from generating the new block are then divided among the validators in proportion to their reputation values.

CCS Concepts

•**Networks** → **Network protocol design; Peer-to-peer protocols; Security protocols;**

Keywords

Consensus Mechanism; Reputation Model; Blockchain based Applications

1. INTRODUCTION

In distributed blockchain systems, the consensus mechanisms are protocols to ensure that all the nodes (participators) are synchronized with each other and agree on which transactions are legitimate to be added in their commonly maintained ledger, where the data structure is in the form of a block chain [14, 24]. For example, the proof of work (PoW) mechanism in the Bitcoin system is a consensus mechanism where nodes with more computational resources are more likely to win the chance to write a new block [15]. Similar consensus mechanisms include the proof of state (PoS) in Ethereum [6], practical Byzantine false tolerant mechanism (PBFT) in Hyperledger Fabric [2], etc. These consensus mechanisms are crucial for a blockchain system to sustain their decentralization feature with the guarantee of the security against malicious attacks.

However, the consensus mechanism in a blockchain system has become a bottleneck in improving its transaction capacity, scalability, and fault tolerance. For instance, PoW in Bitcoin has been widely criticized for its low transaction capacity at the scale of 7 transactions per second and up to 25 transactions per second by tuning protocol parameters without jeopardizing consensus safety [5]. Besides the unsatisfied transaction capacity, high energy consumption also limits the application of PoW-based blockchain systems. A single Bitcoin transaction (April 2019) consumes the equivalent electricity that can power 14.02 U.S. households for one day on average [7]. To address the above performance limitations of PoW consensus mechanism, many alternative new block proposing mechanisms have been investigated by avoiding computation-intensive mining, such as proof of state (PoS), proof of authority (PoA), and proof of elapsed time (PoET), etc. In this paper, we aim to explore a new consensus mechanism based on the computational trust and reputation theory without the cost of unsustainable energy consumption.

Reputation can be defined as the rating of a member's trustworthiness by others [11]. In peer-to-peer networks, reputation systems are used to drive the ability of each participant to trust each other and promote successful interaction [19]. Reputation based on historical behavior is the cornerstone of many trust systems. For example, merchants in the market build their reputation through long-term fair trade to win the trust of customers [18]. Compared with proof of work and proof of stake, proof of reputation is a more effective and efficient consensus mechanism that enables us to

IECC ' 19 July 7–9, 2019, Okinawa, Japan

 ISBN 978-1-4503-7177-3/19/07…$15.00

DOI: https://doi.org/10.1145/3343147.3343169

generate blocks through reputation. The reasons are two folds. First, the waste caused by workload proof can be avoided. Second, reputation consensus can create a good network environment and restrain the malicious behavior of nodes. In this paper, we propose a reputation-based consensus protocol called proof of reputation (PoR). Nodes in a peer-to-peer system build their high reputation by actively participating in system transaction consensus to gain cooperation with other nodes [25][26]. In PoR, the nodes with high reputation values are eligible to participate in the consensus process so as to achieve their rewards. The network can be synchronized to a consensus status with low latency at the cost of linear communication complexity. Our main contributions are summarized as follows.

- A reputation model is proposed to constructing the reputation for each distributed node through considering its asset, transaction value and consensus participation.
- A new reputation-based consensus protocol is designed where the validating nodes vote for a new block generated by the most reputable node. A block structure, consisting of transaction sub-block and reputation sub-block is designed.
- An incentive mechanism for successfully generating a new block is designed, where a certain fixed amount of rewards are divided by all the nodes participating the consensus process, and each node can achieve the rewards proportionally to their reputation.

The rest of this paper is organized as follows: in Section 2, we review the earlier work related to our work. In Section 3, We formalized the consensus problem and the threat models, as well as outline design of the proposed PoR protocol. The design details of the protocol are presented in Section 4. In Section 5, we conducted a security analysis. Finally, Section 6 concludes this work.gorous descriptions or explanations of such commands.

2. RELATED WORK

Since Nakamoto proposed Bitcoin in 2009 [15], a large number of similar altcoins have been proposed. By February [21], 2019, the number of cryptocurrencies available on the Internet has exceeded 2017 and the number is still growing. Bitcoin has the largest market capitalization, accounting for about 53%. The blockchain serves as the core of Bitcoin. In recent years, blockchain technology has shown tremendous potential to revolutionize the global financial ecosystem [21]. Not only in the economic arena, but blockchain technology can also be applied in politics, healthcare, society, and science fields [22], and has the potential to become the next major disruptive technology.

Blockchain is a decentralized public digital ledger that records data on a large number of distributed nodes. A growing list of organized records, called blocks, is cryptographically linked to each node. Each of these blocks contains the cryptographic hash of the previous block, timestamp, and transaction data [16]. Using this emerging technology, you can create an untrustworthy environment for distributed applications. Most blockchain applications rely on miners to mine to create blocks, while issuing virtual currency as an incentive for miners, such as Bitcoin. Blockchain systems can be classified into permissioned, permissionless, and protected (between permissioned and permissionless) networks.

The consensus mechanism is a key component of a blockchain system, which are mainly classified into two types, proof-of-X (PoX) based consensus and Byzantine fault tolerant (BFT) based consensus. PoX based consensus is usually used in the permissionless network, achieved by combining a series of cryptographic techniques, and incentive mechanism design. PoX based consensus provide significantly better support for network scalability, where anyone can join and leave the consensus group at any time, but at the cost of lower throughput. The BFT based consensus is usually used in the permissioned network, relying on a semi-centralized consensus framework and a higher messaging overhead to provide immediate consensus finality and thus high transaction processing throughput [23]. However, due to the higher communication complexity, the set of participants running consensus is small.

Since the use of proof-of-work (PoW) in Bitcoin, PoX based consensus has caused extensive research interests. Proof-of-stake, which was first proposed in a Bitcoin community forum, has been proposed to reduce the difficulty of creating blocks by the size of currency age, thus reducing the computational waste of PoW. Proof-of-space proposes to use physical storage resources instead of computing power in the proof of work mechanism [13][17]. Proof-of-reputation is tried to use in RepuCoin. They measured the reputation of the users based on their behaviours and proposed a reputation-based weighting scheme consensus to overcome the computation cost in PoW. However, their systems have some serious flaws. Their reputation scores are based on the total amount of valid work over time. Accumulated reputation scores and committee-based consensus may lead to serious monopoly problems. If users with high reputation collude with each other, they will double the consumption attack. CertChain proposes a consensus based on reliability level and incentive mechanism considering economic benefits and improper behavior [4]. However, they are designed to be tailored to the certification authority (CA), not to scalability issues in permissioned network. Another related work is a proof of reputation protocol published recently[10], where a reputation ledger for permissioned blockchain systems and the node with the highest reputation value can generate new blocks. Even though we have the similar abbreviations, we have major differences. First of all, our work does not limited to the permissioned blockchain systems where the nodes are distributed and their architecture can be pemissioned, protected or public. Secondly, we have proposed a novel block architecture which records reputation and transactions, where the reputation ledger in [10] has the same block design with transactions. Thirdly, the verification of new blocks in our paper is based on a voting mechanism instead of signature verification.

The BFT consensus protocol mainly include PBFT [3], Zyzzyva [12]. The PBFT agreement reaches a consensus by voting and can tolerate one-third of the Byzantine nodes. However, since the communication complexity of the protocol is n^2, the scale of the consensus node set is limited. Zyzzyva is a speculative-based BFT protocol that reduces password overhead and increases peak throughput for demanding workloads compared to traditional state machine replication. However, a recent analysis suggests that Zyzzyva has a security risk [1].

3. THE MAIN COMPONENTS OF THE PROPOSED MODEL

In this section, we construct the main components in the proposed model, including the system network, threat model, as well as the overview of the procedures in PoR.

3.1 System Network

We make an assumption that the network is partial synchronous, which is the same as that in Bitcoin [8]. Specifically, the connections between the honest nodes are well established, and the transmission time between them is within a predefined and negligible constant Δ. Once any user broadcasts any message, the rest of the honest nodes will receive the message within the bounded delay Δ. Such timing and connectivity assumption is implicitly used in Bitcoin. Moreover, to simplify the proposed model, we assume that the distributed network is an idea network in terms of reliable connection and low-latency broadcast channel. Actually, the relief of these assumptions is open challenges for many distributed systems, which are beyond the scope of this work.

3.2 Threat Model

Although we assume that nodes in a peer-to-peer network can communicate reliably, the network can still be damaged by selfish mining behaviors, malicious attacks, and node failures. The nodes controlled by a Byzantine adversary can produce arbitrary malicious behavior such as deviating from the protocol or remaining silent. The Byzantine nodes can also collude with each other to act out complex behaviors such as sending invalid information. We consider a threat model that includes three attacking strategies, as listed as follows.

- Simple Attack: an attacker performs adversary behavior continuously.
- Camouflage Attack: a malicious node pretends to be an honest node most of the time. When its reputation value reaches a high level, it occasionally attacks the system.
- Sybil Attack: an attacker user uses a single node to forge multiple identities in a P2P network, thereby bringing network redundancy and reducing system security.

The motivation of an attacker to take malicious strategy is to achieve more rewards, which is also an assumption in our analysis. Each node has the opportunity to participate in the consensus process. In a certain consensus mechanism, if a node always achieves low rewards and high costs, the node has no incentive to attack the system. In other words, our system will not consider the irrational attacking strategy. However, the proposed mechanism should also sustain robustness against a marginal proportion of irrational attackers.

3.3 Block Structure

In our consensus mechanism, the reputation of each node is modeled and also requires to be recorded and agreed by all the other nodes. Logically speaking, our system requires another data chain to store the node reputation. In our system, we redesigned the block structure so as to affiliate the proposed design, which is presented as in Figure 1.

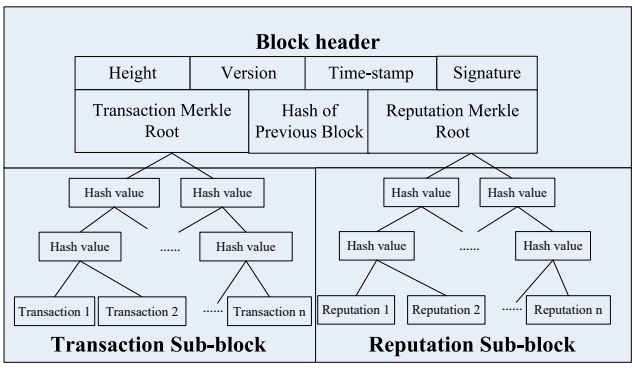

Figure 1. The Block Data Structure in Our System

Specifically, a block is divided into three parts: the block header, transaction sub-block, and reputation sub-block. The block header contains a version number, block height, time-stamp, hash of the previous block, fee, and signature of the block creator, transaction Merkle root, and reputation Merkle root, and each content is explained in Table 1. The transaction sub-block consists of the transactions, which are organized in the form of a Merkle tree [20], and the root hash of this Merkle tree is included in the block header. The reputation sub-block consists of the reputation of the voters invalidating the block as well as their voting information which is also organized by a Merkle tree and their root hash is contained in the block header.

Table 1. The Explanation of The Content in A Block Header

No.	Content in block header	Explanations
1	Height	the index of the block
2	Version	consensus protocol serial number in generating the block
3	Time-stamp	a non-repeated random nonce generator
4	Hash of Previous Block	the hash of the previous data block
5	Transaction Merkle Root	the root hash of the Merkle tree in organizing the transaction list
6	Reputation Merkle Root	the root hash of the Merkle tree in organizing the reputation list
7	Signature	the encryption outcome or hash created based on the private key of the block creator.

It is worthy to note that the hash of the previous block is the hash of the whole block including the transaction sub-block and reputation sub-block.

In the transaction sub-block, the transaction list is ordered according to the time when the transactions happen. Sim-

ilarly, in the reputation sub-block, the reputation list is organized in an acceding order. Each record in the reputation list contains a voter's public key, the consensus reputation value of the voter, the vote message. Since the reputation of a node will be updated once the node's status (the assets, transaction behavior, consensus participation) changes, it is time-consuming to record the updated reputation value considering the node behavior in this block. In our design, nodes' reputation will be updated once the new block is confirmed in the network, whilst, the block generation process will continue to proceed to the next block height based on the latest consensus reputation values. Once the reputation updation is completed, the updated reputation is then used for the block next to the current processed block. The detailed design is presented in Section 4.

3.4 Identity and Mortgage

Identity and Transactions: A user can register an identity by creating a pair of keys based on an asymmetric encryption algorithm, such as RSA. The public key serves as the identity of the user. Any pair of nodes can conduct a transaction by transferring a certain amount of values through encrypting the values and the receiver node's public key by the owner's private key.

In our system, When a node behaves maliciously, it is not enough to just reduce its reputation as a punishment. In our design, We allow users to exchange their real money into the system currency, to serve as a mortgage. The amount of mortgage or system currency is then converted to one of the three reputation values of the node by considering the age (in the unit of the system time) of the mortgage. The formulation of converting system currency into reputation value is as follows:

$$R_s(S_i, t) = \alpha \log(S_i k), \tag{1}$$

where S_i denotes the number of system currency owned by the node i ($S_i > 1$), t represents the time when the node holds the system currency, $S_i t$ denotes the currency age of the currency S_i, and α is the conversion factor indicating the proportion of the currency age converted to the reputation value ($0 < \alpha < 1$). We believe that long-term holders of a large amount of system currency are more likely to be credible and are less motivated to misbehave, compared with short-term holders.

In comparison to PoS where wealth is the only measurement of credibility, PoR incorporates the logarithmic formulation that reduces the increasing rate of reputation value as the wealth mount grows up. Consequently, middle-wealth people also have a good opportunity to gain high credibility as well.

4. PROOF OF REPUTATION MECHANISM

4.1 Procedure Overview

The core idea of PoR is to ensure that each node has an agreed reputation ledger, recording the reputation value of each node. Nodes with high reputation publish block to the peer-to-peer network through proof of reputation. To achieve this purpose, we propose to address the following three technical problems:

Problem 1: How to keep the reputation ledgers consensus? Bitcoin applies a proof of work protocol, which enables all nodes to agree on each block in the blockchain. In the PoR protocol, the content of each block is divided into a transaction sub-block and a reputation sub-block as shown in Fig. 1. The protocol is based on a two-chain architecture - a transaction chain and a reputation chain. The calculation of the reputation of an identity is determined by the transaction data in the transaction sub-block and the behavior of nodes. Nodes require to agree on the transaction and reputation block. Given the behavior of all nodes through the consensus process, all nodes can calculate and reach a consensus on the reputation scores.

Problem 2: How to generate blocks through proof of reputation? The answer to this question is similar to Bitcoin. Bitcoin uses the PoW protocol. The first node that solves the math problem gets the right to publish the block, and others verify the block. In the PoR protocol, the node with the highest reputation value generates the block and publishes it. Nodes with higher reputation value votes for the block. The node verifies the block by the number of votes.

Problem 3: How to motivate nodes to publish blocks? We motivate nodes to publish block through rewards and reputation values. Reputation values are different from coins and cannot be spent or transferred. High reputation means that nodes are willing to maintain the security of the system. They are more likely to be rewarded. Nodes in the network need to be registered with their public keys with private keys stored locally so that nodes in the network can identify each other. A node's reputation value will be agreed upon in the network. If a node behaves maliciously, its reputation changes will be recorded by the whole network.

In each round, the PoR protocol runs the following 3 steps:

Step 1: Leader selection and building block. The node with the highest reputation is selected as leader. The leader collects transactions and packages them into transaction blocks. Then the leader will broadcast the transaction block together with its signature.

Step 2: Reputation-based consensus: We call the top 20% nodes in the reputation list high-reputation nodes. High-reputation nodes verify the received transaction block and signatures. If the transaction is legal and the signature is correct, they will vote on the transaction block. When a node receives votes from most high-reputation nodes, it will append the transaction blocks to the blockchain.

Step 3: Reputation Updation. The reputation scheme guarantees the feasibility of the first two steps. In PoR, there are four factors that affect the reputation value of a node: historical transactions, currency age, participation in consensus, and illegal behavior of nodes. Trading with low-reputation nodes may reduce the reputation value of nodes. The node calculates the reputation value through the established reputation model. Considering that the network is partially synchronized, the behavior information received by all nodes will be consistent. All nodes can reach a consensus on reputation scores.

We proceed to present the detailed design of the proof of reputation (PoR).

4.2 leader selection and building block

This section presents our leader selection method and how the blockchain is stored.

4.2.1 Leader selection

We select the node with the highest reputation value in the reputation list as the leader. We consider the following two aspects in the process of the leader selection.
Security: We believe that nodes with higher reputation values are more willing to be responsible for system security. The reason is that system security is inextricably related to their wealth.
Incentive: The leader will get more rewards in a round of consensus. Thus, we assume that every node wants to be a leader.

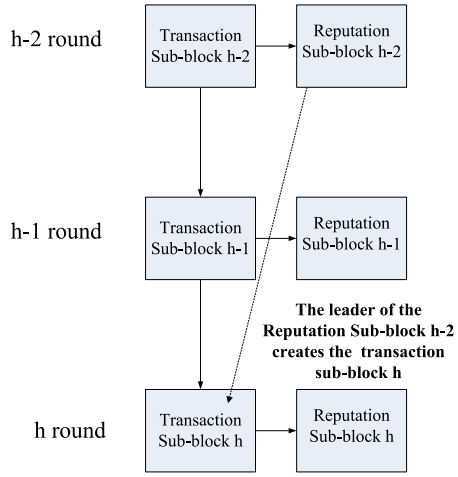

Figure 2. Leader Selection Process

As shown in Figure 2. In round $h-1$, reputation list $h-1$ denotes the latest reputation value of network nodes when consensus for transaction block $h-1$ is completed. Because we assume that the network is partially synchronized, in the h-1 round, the network agrees on block $h-1$ within time T. In round h, the node with the highest reputation in the reputation list $h-2$ acts as the leader to create a new block. We can choose the leader in the reputation list $h-1$ to create the trading block h. In this case, we have to wait at least time T to create the next transaction block. To reduce the waiting time, we choose the leader of the reputation list $h-2$ to create the next transaction block.
The leader creates transaction block by collecting legitimate transactions. Then the transaction block is signed by leader and broadcast to the network.

4.2.2 *Storage of Blockchain*

When the nodes agree on the created blocks, they need to store the block on the blockchain. The structure of the blockchain is shown in Fig. 3.
Our structure corresponds to the leader selection method. The reputation block initial represents the initial reputation of the network node. The transaction block is created by the leader in the reputation block. The first block is special when the system is started. In round 0, the transaction sub-block 0 is created by the reputation block initial. After the block is appended to the blockchain, the reputation sub-block 0 is calculated and saved. In round 1, the transaction block 1 is created by the reputation block initial. After the block in round 1 is appended to the blockchain, the

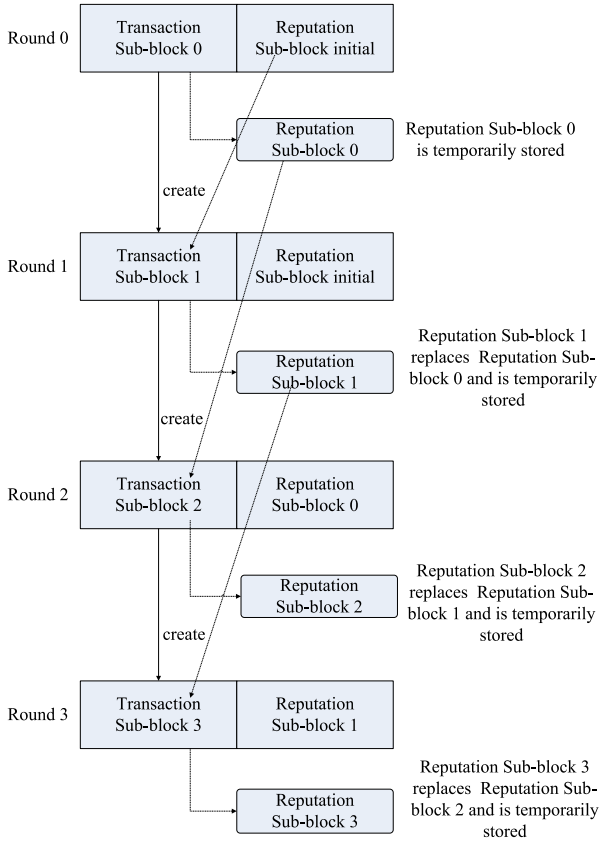

Figure 3. The Storage Structure of The Blockchain

reputation block 1 is calculated and the reputation blocks 0 and 1 are temporarily saved. In round 2, the transaction sub-block 2 is created by the reputation block 0. After the block in the round 2 is appended to the blockchain, the reputation block 2 is calculated and the reputation blocks 1 and 2 are saved.

4.3 Reputation-based Consensus

The detailed Reputation-based consensus process is described as in Algorithm 1. The algorithm is divided into 2 phases - verification and voting phase and submission phase. We regard the top 20% of the nodes in the reputation list as the high reputation nodes.
In the verification and voting phase, after the node receives the transaction block, it verifies the transaction sub-block. If the transaction block is illegal or the transaction block is not from the leader, the high reputation nodes (the top 20% of the reputation list nodes) will send a vote against. Otherwise, the high reputation nodes send a consent vote. If it is not a high reputation node, we do nothing. The structure of the voting message is shown in Figure 4. Here, *type* represents the type of vote, which is divided into consent and objection, *Blockhash* represents the hash value of the transaction block, and *Signature* is the sender's signature.
In the submission phase, a node collects the message and accumulates the reputation value of the nodes that send the consent vote and object vote. When the total reputation value of the nodes that send the consent vote is greater than the total reputation value of the 2/3 high reputation nodes,

Algorithm 1 Reputation-based Consensus

Input: *tranction*block
Parameter: node $i \in (1, ..., n)$
set S = top 20% of reputation list nodes
T = The total number of reputation values for the top 20% of nodes
Output: *transaction*block validity

verification and voting phase:
receive(*transaction*block)
if !verify(*transaction*block.signature
||!verify(*transaction*block.header)
||!verify(*transaction*block.tranction) **then**
  if $i \in S$ **then**
    broadcast($msg\langle objection, blockhash, signature\rangle$)
  end if
else
  if $i \in S$ **then**
    broadcast($msg\langle consent, blockhash, signature\rangle$)
  end if
end if
submission phase:
while receive(msg) **do**
  if $msg.type == consent$ **then**
    sum_c = acc($msg.signature.reputation\ value$)
    if $sum_c > 2/3T$ **then**
      return submission
    end if
  else if $msg.type == objection$ **then**
    sum_o = acc($msg.signature.reputation\ value$)
    if $sum_o > 2/3T$ **then**
      return not submission
    end if
  end if
end while

the block is appended to the blockchain. When the total reputation value of the nodes that send the objection vote is greater than the total reputation value of the 2/3 high reputation nodes, the block is invalid. Leader's reputation value will be reduced.

The reasons that we choose the top 20% of reputation list as voters are two folds. First, according to the Tareto Principle (also known as the 80-20 rule) [9], 80% of the reputation value is concentrated on 20% of the nodes. These 20% of nodes have the majority of decision-making power. Second, 20% of nodes broadcast message at the same time. The communication complexity of the algorithm is reduced to $O(0.2n^2)$. This ensures that the messages received by the nodes are agreed in a short time.

Since the high reputation value node may also have a failure or evil situation, we define that when 2/3 of the total reputation value of the high reputation nodes is reached, the block can be submitted.

4.4 Reputation Updation

We consider three aspects to measure the reputation of a node - the currency age of the node, social interaction of node, and the regularity of participation in the consensus. Node socialization is determined by the number of friends a node has, the frequency of interaction with friends, the reputation of friends and the size of transactions.

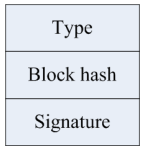

Figure 4. **The Data Structure of A Voting Message**

When the block is submitted to the blockchain, we update the reputation value of the network nodes based on the transaction block and the voting message. In this section, we describe the design of a reputation scheme, including the calculation of reputation scores and the generation of the reputation block. Our proposed reputation scheme can improve the security and incentive property of the PoR.

1. Reputation Score Calculation: At the end of the credibility consensus, the reputation score can be calculated by all nodes based on the transaction block, the currency age of the currency, the activity degree of participating in the consensus. The formulas are as follows:

$$R_i = \beta_1 R_s(S_i, t) + \beta_2 R_a(A, V) + \beta_3 R_c(N_i) \tag{2}$$

where

$$R_s(S_i, t) = \alpha \log(S_i t) \tag{3}$$

$$R_a(A, V) = \sum_{k=1}^{j} A_k \log(V_k) S(k) \tag{4}$$

$$R_c(N_i) = \gamma_1 N_{ic} T_{total} - \gamma_2 N_{ie} T_{total} \tag{5}$$

Where β_i is the weight. R_s denotes the value of reputation converted by the age of currency. R_a denotes the degree of social activity of a node. R_c denotes the contribution of consensus.

In Eq.(2), S_i represents the currency owned by the node i. t denotes the time of holding currency. The logarithmic formula can reduce the reputation gap caused by the gap between rich and poor.

In Eq. (3), j indicates the number of friends who have conducted transactions with node i. A_k is the weight function for each transaction and is positively related to the transaction value. The scaling factors $S(k)$ are used to increase or decrease the reputation value of a node. When the k-node reputation value is positive, $S(k)$ is a positive number. When the k-th node's reputation value is negative, $S(k)$ is a negative number. The result is that trading with a node with a high reputation value will increase its reputation value. Trading with a node with a negative reputation value will reduce its reputation value. Therefore, a node does not like to trade with nodes with low reputation, so as to prompt every node to actively in maintaining high reputation status.

In Eq.(5), We believe that if a node participates in the system consensus frequently, it will be more credible. When the high reputation node sends the correct voting message, the reputation value will increase slightly. When a malicious message is sent, the reputation value of the node is greatly reduced to become a negative number. N_{ic} denotes the correct voting frequency of the node in recent time. N_{ie} denotes the frequency of erroneous voting oft the node in recent times. T_total denotes the total number of transaction

values generated over a recent period of time. γ_1 and γ_2 are weight parameters. γ_2 is much larger than γ_1.
2. Reputation Block Generation: As shown in Figure 2, the node votes on the transaction block of the round h. After the voting is completed, the voting message collected and the updated list of reputation values are combined into a reputation block.

4.5 Incentive Mechanism

What are the benefits of a high reputation? In order to encourage nodes to maintain high credibility, the system gives monetary incentives to successful participants in the consensus. When a new block is generated, the system generates a fixed amount F of system currency and allocates these currencies to the nodes participating in the consensus in the form of transactions, which is inspired by the generation of most of the cryptocurrency, such as Bitcoin.
The allocation of the system currency is as follows.

$$Rd_i = \frac{R_i}{\sum_{k \in S} R_k} F \tag{6}$$

The gained system currency will be further accounted in updating nodes' reputation according to Eq.(3)

5. SECURITY ANALYSIS

Given the design of the protocol PoR mechanism, we now provide a security analysis for how PoR prevents potential threats. Malicious high-reputation nodes may be attacked by malicious voting. According to our reputation updation model, sending malicious voting messages can greatly reduce the reputation value of nodes. If a node's reputation value is reduced to a negative number, there is no user willing to conduct transactions with the nodes trading with a node with a negative reputation will reduce its reputation.
Malicious nodes can increase their reputation value by frequent transactions with each other. The logarithmic function of reputation value in Eq.(3) is used to control that when a low-reputation node trades with multiple nodes, it can not significantly improve its reputation value. This effectively avoids the possibility of trying to increase reputation by adding malicious fake nodes.
If the total reputation value of the high reputation value node accounts for 80% of the total network reputation value, the malicious reputation value that PoR can tolerate is $0.8(1-2/3)R_{total}$. R_{total} represents the total reputation value of all network nodes.

6. CONCLUSION

In this work, we have proposed a reputation-based consensus mechanism, PoR. In PoR, the consensus process in a round, consisting three steps: leader selection and building block, reputation-based consensus, reputation updation, Where we choose the node with the highest reputation value as the leader to create the block. Some nodes with high reputation values vote to reach consensus on the block. The selection of a leader improves the throughput of the protocol. The voting consensus of the high reputation node guarantees its security.

7. ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation for Young Scientists of China under Grant No.61702090 and No. 61702084; the Natural Science Foundation of Liaoning Province of China under Grant No.20170540319, No.201602261; and the Fundamental Research Funds for the Central Universities under Grant No. N162410002, N161704001.

8. REFERENCES

[1] I. Abraham, G. Gueta, D. Malkhi, and J.-P. Martin. Revisiting fast practical byzantine fault tolerance: Thelma, velma, and zelma. *arXiv preprint arXiv:1801.10022*, 2018.
[2] E. Androulaki, A. Barger, and V. e. a. Bortnikov. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15, 2018.
[3] M. Castro and B. Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002.
[4] J. Chen, S. Yao, Q. Yuan, K. He, S. Ji, and R. Du. Certchain: Public and efficient certificate audit based on blockchain for tls connections. In *IEEE INFOCOM Conference on Computer Communications*, pages 2060–2068, 2018.
[5] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125, 2016.
[6] P. Daian, R. Pass, and E. Shi. Snow white: Robustly reconfigurable consensus and applications to provably secure proofs of stake. In *Iacr*, pages 1–64. 2017.
[7] digiconomist. Bitcoin energy consumption index. https://digiconomist.net/bitcoin-energy-consumption, 4 2019.
[8] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2):288–323, 1988.
[9] H. Fawcett. *Manual of political economy*. Macmillan and Company, 1883.
[10] F. Gai, B. Wang, W. Deng, and W. Peng. Proof of reputation: a reputation-based consensus protocol for peer-to-peer network. In *International Conference on Database Systems for Advanced Applications*, pages 666–681, 2018.
[11] M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, pages 144–152, 2003.
[12] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong. Zyzzyva: speculative byzantine fault tolerance. In *ACM SIGOPS Operating Systems Review*, volume 41, pages 45–58, 2007.
[13] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz. Permacoin: Repurposing bitcoin work for data preservation. In *2014 IEEE Symposium on Security and Privacy*, pages 475–490, 2014.
[14] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun. A review on consensus algorithm of blockchain. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572, 2017.

[15] S. Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[16] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. *Bitcoin and cryptocurrency technologies: A comprehensive introduction*. Princeton University Press, 2016.

[17] S. Park, K. Pietrzak, A. Kwon, J. Alwen, G. Fuchsbauer, and P. Gazi. Spacemint: A cryptocurrency based on proofs of space. In *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC). Springer*, 2018.

[18] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. In *The Economics of the Internet and E-commerce*, pages 127–157. Emerald Group Publishing Limited, 2002.

[19] A. A. Selcuk, E. Uzun, and M. R. Pariente. A reputation-based trust management system for p2p networks. In *IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004.*, pages 251–258, 2004.

[20] M. Szydlo. Merkle tree traversal in log space and time. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 541–554, 2004.

[21] A. Tapscott and D. Tapscott. How blockchain is changing finance. *Harvard Business Review*, 1(9), 2017.

[22] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123, 2016.

[23] M. Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International workshop on open problems in network security*, pages 112–125, 2015.

[24] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou. A survey of distributed consensus protocols for blockchain networks. *arXiv:1904.04098v1*, pages 1–27, 2019.

[25] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[26] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on parallel and distributed systems*, 18(4):460–473, 2007.