A Scalable and Secure Consensus Scheme Based on Proof of Stake in Blockchain

Fayuan Zhu[1], Jie Yin[1(✉)], Lei Liu[2], Jie Feng[2], and Zhangquan Wang[3]

[1] State Key Laboratory of ISN, School of Telecommunications Engineering, Xidian University, Xi'an 710126, China
yinjie0003@stu.xidian.edu.cn
[2] Shaanxi Key Laboratory of Blockchain and Secure Computing, Xi'an 710071, China
[3] The 601th Institute, the 6th Academy, China Aerospace Science and Industry Corporation, Hohhot 010076, China

Abstract. As a decentralized database, the public blockchain has broad application prospects in many fields such as finance, healthcare, and supply chain, so it is getting more and more attention. The current mainstream public blockchain protocol based on Proof of Work (PoW) cannot be applied to various extendable application scenarios with performance requirements due to performance bottlenecks. Proof of Stake (PoS) circumvents the performance bottleneck of PoW by utilizing equity instead of computing power. However, ordinary PoS protocols still have security problems for they are vulnerable to nothing-at-stake attacks, and rely on third parties to support dynamic availability. In this paper, we propose a novel scalable and secure PoS consensus scheme to support the application of public blockchain in various extendable scenarios. We classify nodes in different states and perform node state transitions through different protocols. Combining stake mechanism with consensus schemes and using dynamic stake proportion table to support dynamic stake scenario, we propose a block compression method to reduce the communication consumption of consensus. We propose a chain selection rule based on Verifiable Random Function (VRF) nonce and longest chain rule, which supports dynamic availability without third parties. In addition, we prove the security of our scheme and analyze its performance at general security threats. Finally, experimental results show that our scheme have better system performance and better scalability under the same condition.

Keywords: Blockchain · Consensus · PoS · Chain selection · Scalability

1 Introduction

Consensus mechanism is the core and cornerstone of public chain. PoW has become the prevailate consensus mechanism of public chain because of its simplicity, practicality and provable security, such as Bitcoin [1] et al. However,

H.-N. Dai et al. (Eds.): BlockSys 2021, CCIS 1490, pp. 287–301, 2021.
https://doi.org/10.1007/978-981-16-7993-3_22

energy consumption and performance barrier make it hard to apply to various extendable scenarios. PoS [2] is an energy-efficient public chain consensus framework, it uses stake to replace computing power for leader election. PoS overcomes the shortcomings of PoW and is a popular alternative of it. However, PoS chain is more susceptible to nothing-at-stake attacks from malicious nodes, so its security cannot be ascertained.

Related work about PoS protocol is mainly based on Byzantine Fault Tolerance (BFT) and the longest-chain-rule [17]. BFT-based PoS protocol [3,4] selects candidates to form a committee based on nodes' stake and guarantees the security of the protocol by BFT, therefore increases the communication consumption of the consensus. In addition, the scalability of the protocol is also limited by BFT. The longest-chain-rule-based PoS protocol [5–9] uses stake proportion for leader election and longest-chain rule for block synchronization. It avoids the scalability limitation of the front and its communication consumption is low. However, its security cannot be guaranteed, and nodes are vulnerable to bribery attacks [4], and it relys on a third party to support dynamic availability, which reduces the decentralized property of public chain.

In this paper, we propose a novel scalable and secure PoS consensus scheme which has an independent equity mechanism and does not rely on third parties to satisfy dynamic availability. We utilize a block compression method to enhance the efficiency of consensus, which increases the system performance of public blockchain. Meanwhile, it has a provable security under the standard blockchain attributes and could defend other threats.

Our contributions are as follows:

1) We propose a novel blockchain protocol based on PoS, which combines stake mechanism and consensus scheme, applies dynamic stake proportion table to support dynamic stake scenario, and utilizes a block compression method to reduce the consumption of the block propagation in a distributed network, which could improve the system performance efficiently.
2) We present a chain selection rule based on VRF nonce and longest-chain rule, and then propose a block synchronization protocol based on it, which can support dynamic availability without trusted third parties, further strengthen the decentralization attribute, and effectively circumvent single point effects.
3) We propose a scalable and secure PoS consensus scheme with an independent stake mechanism, which has a stronger provable security and better scalability. It has been proven to meet the security attributes of the blockchain, and the proportion of stake fault tolerance of the protocol also performs well.

The rest of this paper is as follows. The tools and model assumptions will be introduced in the next section. Section 3 illustrates the system model and detailed design of the scheme. Security analysis is given in Sect. 4, and Sect. 5 presents the experiments and results. Section 6 introduces the related works and Sect. 7 gives the conclusion and discussion.

2 Tools and Model Assumptions

In this Section, we introduce the VRF and KES tools used in our scheme, and explain the clock model, communication model and security model assumptions, and finally give our design goals, which could better expound our scheme.

Tools. We use two tools to improve the security of our scheme.

Verifiable Random Function (VRF). VRF [11] is a non-interactive verifiable nonce algorithm based on an asymmetric encryption algorithm, which guarantees the uniformity, verifiability and security of the generated nonce. It has two participants: prover uses secret key SK and a random seed to calculate VRF nonce and proof, verifiers use the homologous verifiable key VK and other public information to verify it. VRF was applied to PoS scheme design in [3,6–8,16].

Key Evolving Signature schemes (KES). KES [12] is a forward signature scheme, which periodically generates a new key pair based on the original key and destroys the old key. It can effectively prevent malicious nodes from modifying the transaction information of stored blocks, and ensure that the security of data.

Models. Public chain is an open and distributed network and its clock is uncertain. To avoid other interferences, we have 3 models below:

Clock Model. We assume the consensus time (the time of each round of the block generation protocol) is determinate, and the consensus node could get the same current timestamp when its network is normal.

Network Model. The communication between nodes is partially synchronized, that means messages sent by nodes can arrive within the maximum delay Δt_{delay}, and the messages are invalid when their reachable time beyond Δt_{delay}.

Security Model. The security threats of our scheme mainly come from Byzantine [11] nodes. The biggest security threat is nothing-at-stake attack [4], which malicious nodes could expand various chains at a extremely low cost to make chain diverges. Bribery attacks is also a inevitable problem, we assume that it's only feasible to few honest nodes.

Goals. Based on the tools and models above, our PoS consensus scheme should have dynamic availability, scalability, and provable security.

Dynamic Availability. The public chain is an open network that nodes can join and exit at any time. Unsynchronized honest nodes could synchronize the longest chain dominated by honest nodes by executing our protocol.

Scalability. The communication consumption of distributed networks increases rapidly with the increase of the number of nodes, our scheme needs to enhance the efficiency of consensus and have a high system performance to make public chain more scalable.

Provable Safety. our scheme should have provable security under the models above, which means it meets the three standard security attributes of the blockchain chain growth, common prefix and chain quality.

3 Scalable Consensus Scheme Based on Proof-of-Stake

In this Section, we specifically introduce our PoS consensus scheme. We classify the nodes in the network into 3 types: new nodes, unsynchronized nodes and synchronized nodes according to their states. Then we propose three protocols: node registration, block generation and block synchronization, which node could execute them for state transition. In order to combine stake mechanism with our scheme, we initialize the stake value in node's registration transaction in *NodeRegistration* protocol. We propose a candidate block compression method in *BlockGeneration* protocol to reduce the communication consumption of consensus. In *BlockSynchronization* protocol, we use the longest-chain-rule and VRF nonce to propose a new chain selection rule, so that our scheme supports dynamic availability without third parties.

3.1 System Model

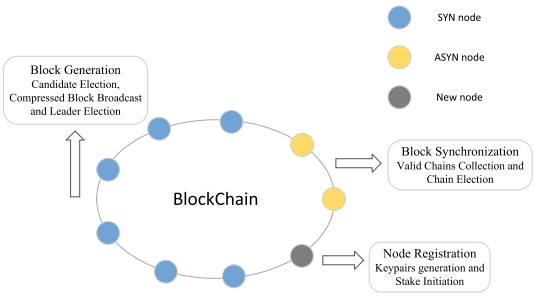

Fig. 1. Overview of system model.

Figure 1 is an overview of our model, there is a peer-to-peer network in our system model, it includes three types of new nodes: synchronized nodes, and unsynchronized nodes. And our scheme involves three protocols: *NodeRegistration*, *BlockGeneration* and *BlockSynchronization*. Synchronized nodes have synchronized to the latest block and have the longest chain, while unsynchronized

nodes doesn't synchronize to the latest block due to network failures or other additions. Only synchronous nodes can participate in the consensus to compete to become the leader to generate new blocks.

NodeRegistration protocol includes two stages: key pair generation and equity initialization. *BlockGeneration* protocol includes three stages: candidate election, compressed block broadcast and leader election. *BlockSynchronization* protocol includes two stages: legal chain collection and chain selection.

3.2 Scheme Design

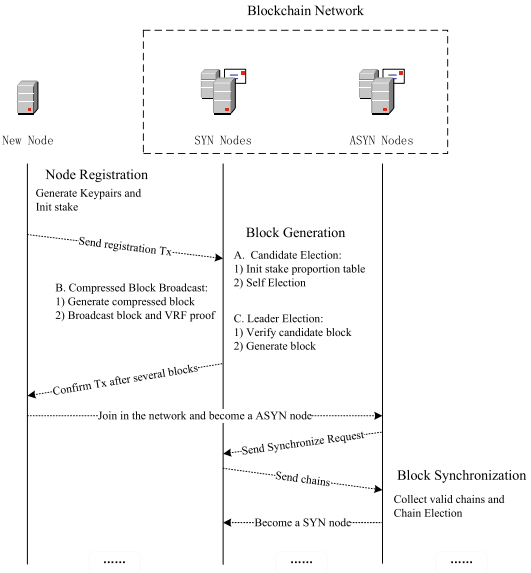

Fig. 2. Scheme overview.

Scheme Overview. Figure 2 is the overview of our view, the state of the nodes in the system is as dynamic as the real scene, and can transform to another state by executing the homologous protocol, or keep the original state:

1) The new node executes the node registration protocol to join the network. After the transaction is confirmed, the node registers successfully and becomes an unsynchronized node.
2) The unsynchronized node executes the block synchronization protocol to become a synchronized node. After the synchronization is completed, it become a synchronized node.
3) The synchronization node executes the block generation protocol to maintain the synchronization state. If the execution is successful, it will remain synchronized state, otherwise it will become an unsynchronized node.

Node Registration. In the original PoS protocol, stake is the token in the incentive mechanism, which is negative to the development of public chains. We separate the stake mechanism from the incentive mechanism and make it a independent part of our PoS consensus scheme. For each new node N_i, it performs the following initialization processing respectively.

Init. Generate Keypairs and Init Stake. N_i generate KES keypairs PK_i^{kes} and SK_i^{kes} to sign and verify the node information to ensure the authenticity of it. Then it uses random input 1^k and $KeyGen(*)$ to generate VRF keypairs $VK_i^{vrf}$and SK_i^{vrf}. Nodes use SK_i^{vrf} to generate nonces and proofs for candidate election, use VK_i^{vrf} to verify the proof to ensure the unforgeability of VRF nonces. N_i inits its stake value w_i^{init}.

Send Registration Transaction. N_i sends the registration transaction with w_i^{init}, its PK_i^{kes}, VK_i^{vrf} and other public information to the consensus network. The node registers successfully when the new block containing the transaction is generated and confirmed.

Node Stake. The changes of nodes' stake will record in a transaction and store in the block, stake is valid only when it's recorded in the blockchain. We set the stake mechanism as a open setting in this paper, which makes it more flexible.

3.3 Block Generation

We use stake proportion table to support dynamic stake scenario with our stake mechanism in this part. And the communication of broadcasting messages in large-scale distributed networks is costly, we use a block compression method to reduce communication consumption and increase the efficiency of consensus. Then we use VRF nonce to select the leader, which is impartial and secure.

For each participant node N_i with local chain $Chain_l^{local}$, and its chain length is $l(>1)$ after r rounds. The latest block is

$$Block_l = \left(\left(root_l^{trans} | nonce_l | Other\right) | BlockBody_l\right) \tag{1}$$

for $r+1th$ round, our consensus performs below:

Candidate Election. Every node in our scheme could compete to be the leader, we use self election to save communication consumption. It's divided into 2 stages:

1) Generate Node Stake Proportion Table. N_i reads stake transactions from $Chain_l^{local}$ and calculates the stake value w_i^l of each node, then generates the node stake proportion table W_l^{local}. For the stake proportion s_i^l of each node is $s_i^l = \frac{w_i^l}{\sum_{j=1}^{n} w_j^l}$.

2) Self Election. Consensus node uses $nonce_l$ from $Block_l$, consensus round $r+1$, nonce $index$ and SK_i^{vrf} to generate $nonce_i^{r+1}$ and $proof_i^{r+1}$, then uses $nonce_i^{r+1}$ and its stake proportion s_i^l for candidate election, the condition is $nonce_i^{r+1} \leq s_i^l \times 2^{len}$, len is the binary length of VRF nonce. The node meets the condition becomes a candidate and executes the next protocol.

Compressed Block Broadcast. In a general blockchain, transaction data is actually broadcast twice, the first time is in the transaction broadcast stage, for the second time in the block broadcast stage. Different with PoW, the hash calculation time of the PoS protocol is almost 0, and the block broadcast time is a big factor of consensus efficiency.

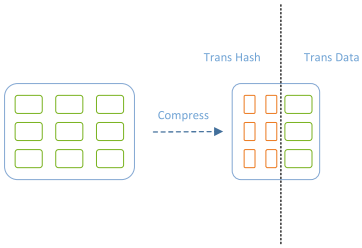

Fig. 3. Candidate block compression.

1) Generate compressed block. For the received transaction, node N_i first verifies its validity, then saves the valid one in $TransPool_i$. The transactions that have been broadcast are deposited in $Trans_i^{broadcast}$, others are deposited in $Trans_i^{local}$. The candidate node only packs the broadcast transaction's unique identifier (such as transaction hash). Figure 3 shows the diminution of the block size with the same number of transactions after compression.

2) Broadcast block and VRF proof. After the candidate node generates the compressed candidate block, it packages them as

$$Candidate_i^{l+1} = \left(Trans_i^{l+1} | \left(nonce_i^{l+1} | proof_i^{l+1} | VK_i^{vrf} | Other\right)\right) \quad (2)$$

then broadcast it to other nodes in the network.

Leader Election. *i*) Verify Messages. For each candidate message $Candidate_{j(j\neq i)}^{l+1}$, node N_i firstly verifies the validity of the VRF nonce $nonce_j^{r+1}$, and then verify the validity of candidate according to W_l^i. If both the results are true, N_i adds it to candidate pool $Candidates_i^{l+1}$. *ii*) Generate Block. If the number of candidates in $Candidates_i^{l+1}$ is more than 1, N_i chooses the one with the smallest VRF *nonce* as leader, and generates a new block based on it; if it is 0, then no block will be generated. The next round will begin at the end of this consensus.

Empty Block. Assume that a total of n participates in one round of consensus, and the stake of node N_i is s_i. It has been proved that the VRF *nonce* is uniform and satisfies the law of distribution in [3]. Therefore, it can set the probability of node N_i self-elected successfully is s_i when n is big enough. The probability of generating empty block θ_{empty} when a node only generates one nonce is

$$\theta_{empty} = \prod_{i=1}^{n} (1 - s_i) \tag{3}$$

it have $\sum_{i=1}^{n} s_i \leq 1$, when n is big enough, $\prod_{i=1}^{n} (1 - s_i) \leq \prod_{i=1}^{n} \left(1 - \frac{1}{n}\right)$ and $\lim_{n \to \infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$ thus

$$\theta_{empty} = \prod_{i=1}^{n} (1 - s_i) \leq \prod_{i=1}^{n} \left(1 - \frac{1}{n}\right) = \frac{1}{e} \tag{4}$$

which means the phenomenon of generating empty blocks is inevitable.

Block Synchronization. In the original PoS protocol, it obtains dynamic availability through "checking point", which decreases the decentralized attribute of the blockchain. In Block Synchronization protocol, honest nodes only need genesis block and current timestamp and then they could synchronize to the valid chain, which supports dynamic availability without a third party.

Chain Selection Rules. i) If there are two chains with different lengths, choose the longer one; *ii*) if their length is equal, choose the one with the earlier consensus round after their divergence point firstly; *iii*) If their round is the same, choose the one with the smaller VRF nonce of the block after the divergence point.

The chain selection rule is a special longest chain rule, which guarantees that the honest nodes will choose the same longest chain. Although malicious nodes may successfully attacks in new blocks, honest nodes would still dominate the longest main chain while their stake proportion is big enough.

We assume that the local chain of node N_i is $Chain_local$ and the current timestamp is T. The protocol steps are as follows:

1) Calculate the consensus round. The node N_i get the initial timestamp t from genesis block, and then uses current timestamp T to calculate the current consensus round R.

2) Collect legal chains. N_i requests other nodes for their local chain $Chain_i$ and verifies its validity. Firstly, it verify whether the consensus round is smaller than R, then the validity of genesis block, and finally verify the legality of the chain and block. All valid chains are collected into $Chains_i^{syn}$.

3) Chain selection. For each $Chain_i$ from $Chains_i^{syn}$, N_i uses chain selection rules to select one chain as its new local chain.

Besides the chain length, we also use consensus round R and VRF *nonce* to upgrade our Chain Selection Rule. That is to ensure that the node would select only one chain among multiple chains. Any honest node can finally synchronize the longest chain dominated by honest nodes with executing the block synchronization protocol. Therefore, our scheme can support dynamic availability when the stake proportion of malicious nodes is less than a certain value.

4 Security Analysis

In order to ensure the feasibility of our scheme, we firstly prove its security under a standard security model, then we analyze the stake fault tolerance proportion of the protocol and its performance under other security threats.

4.1 Security Attributes

Definition. Blockchain has three key security attributes: chain growth [13], common prefix [14,15] and chain quality [14]. These three attributes are used for security analysis in PoS protocols such as [6–9,16]. We also use them to analyze our protocol's security. We firstly give their definitions and then prove that our protocol satisfies them under our assumptions and security model.

Definition 1. *Chain growth. There is a PoS protocol Pos with a set of participants* P_**. The chain growth properties are described as follows: there is a parameter* $\sigma \in R$*, for any honest participant* P_i *with local chain* C_i *and chain length* L_i *when the round of consensus is* R_i*, and honest participant* P_j *with local chain* C_j *and chain length* L_j *when the round of consensus is* R_j*. when* $P_i, P_j \in P^*, R_i > R_j$*, there is* $L_i - L_j \geq \sigma(R_i - R_j)$ *in the execution of Pos.*

Definition 2. *Common prefix. There is a PoS protocol Pos with a set of participants* P_**. The public prefix attributes are described as follows: there is a parameter* $k \in N_+$*, for any honest participant* P_i *with local chain* C_i *and chain length* L_i*, and honest participant* P_j *with local chain* C_j *and chain length* L_j*. When* $P_i, P_j \in P_*, L_i > L_j$*, there is* $Chain_j\,[\neg k] \subseteq Chain_i$ *in the execution of Pos.*

Definition 3. *Chain quality. There is a PoS protocol Pos with a set of participants* P_**. The chain quality attribute is described as follows: there are two parameter* $\mu \in R\,(0 < \mu < 1)\,, L \in N_+$*, for any honest participant* P_i *with local chain* C_i *and chain length* L_i*. When* L_i *is large enough, the proportion of blocks generated by honest nodes in* C_i *is at least* μ *in the execution of Pos.*

Proof. In order to prove that our scheme satisfies these three security attributes, we have tree preconditions below: *i*) The stake proportion α of all honest nodes is greater than that of all malicious nodes β, it means $\alpha = \lambda\beta, \lambda > 1$; *ii*) Honest nodes generate a every r rounds on average, $r = e/(e-1), r \in R$, e is the natural exponential; *iii*) Protocol executes in the $NetworkModel$ of Sect. 2.

We assume that there is a honest node N_i with local chain $Chain_i$ and chain length l_i when the round of consensus is r_i, a honest node N_j with local chain $Chain_j$ and chain length l_j when the round of consensus is r_j, and a node N with local chain $Chain_{local}$ and chain length l.

Proof. Chaingrowth. In the worst case, the malicious node does not generate new blocks. The stake proportion of honest nodes is α, and on average at least one block is generated every $r' = e^{\alpha}/\left(e^{\alpha} - 1\right)$ round, then there is $l_i - l_j \geq \left(r_i - r_j\right)/r'$. It's obvious that $1/r' \in R$, so it satisfies the chain growth attribute.

Proof. Commonprefix. For honest nodes N_i and N_j,$l_i > l_j$, we know that their genesis blocks are generated when the blockchain is initialized, so at least $Chain_j\,[\neg k] \subseteq Chain_i$. According to the chain selection rules, we know that all honest nodes will eventually choose a chain as the main chain. So we have $Chain_j\,[\neg l_j - h] \subseteq Chain_i, h > 1$. so it satisfies the common prefix attribute.

Proof. Chainquality. The chain length l of the honest node's local chain $Chain_{local}$ will be long enough after the protocol has been executed for a long time. And the probability of a consensus node being elected as a leader to generate a new block is positively correlated with its stake proportion. Since we assume that the stake proportion of honest nodes is α, that of malicious nodes is β, and $\alpha = \lambda\beta, \lambda > 1$. We proved that the worst-case inflation rate [9] of malicious nodes will eventually converge to a certain value $ratio^{full-greedy}$ in next segment, then the proportion of blocks generated by honest nodes $\mu' \approx \alpha/\left(\alpha + \beta \times ratio^{full-greedy}\right) = \lambda/(\lambda + ratio^{full-greedy})$. we have $0 < \mu' < 1$ when $\lambda > 1$, so it satisfies the chain quality attribute.

4.2 Security Threats

Stake Fault Tolerance Proportion. The PoS protocol is vulnerable to nothing-at-stake attacks because its low computation consumption. Malicious nodes can expand on multiple chain branches of a consensus at a very low cost. We use the amplification ratio [9] to analyze the stake fault tolerance proportion of our protocol. Similarly, we have the following analysis:

For the chain $Chain'$ with length l' and consensus rounds r', the stake proportion of the malicious nodes $N_{adversary}$ is *beta*, and the max probability of generating a block each round is also β. Let $f\,(t,l)$ be the number of chains with chain length $l' + l$ and consensus time $r' + t$ which extends from $Chain'$, we have $f\,(t,l) = f\,(t-1,l) + f\,(t,l-1)\,\beta = \binom{t}{l}\beta^l = \frac{t!}{l!(t-l)!}\beta^l$. we know $t! \approx \sqrt{2\pi t}t^t$ when t is large enough, combine (10) and $t = rl$ we have $f\,(t,l) \approx \frac{\sqrt{2\pi rl}(rl)^{rl}}{\sqrt{2\pi l}l^l\sqrt{2\pi(t-l)}(t-l)^{t-l}} = \sqrt{\frac{r}{2\pi(r-1)l}}\left[\frac{r^r}{(r-1)^{r-1}}\beta\right]^l$, and we know that $f\,(t,l) \geq 1$ when l is big enough, then combine $t = rl$ we have $\frac{r^r}{(r-1)^{r-1}}\beta > 1 \Rightarrow l < \left[\frac{r}{r-1}\right]^{r-1}\beta t$. we know that l is the length of $Chain'$ with full-greedy [9] strategy extends after time t, so the amplification ratio is $ratio^{full-greedy} < \left[\frac{r}{r-1}\right]^{r-1}$.

When malicious nodes implement the full-greedy strategy and honest nodes not, our stake fault tolerance proportion $\rho = \frac{1}{1+ratio^{full-greedy}}$.

Prediction Window. The results of [17] show that it needs higher stake proportion to guarantee security when the prediction window is larger, because it's more vulnerable to bribery attacks (BA) [4]. The random seed of generating VRF nonce will change in each round in our protocol, so our prediction window

size is 1, which can be regarded as unpredictable when ignoring network delays, so it defends BA well.

Table 1. Secure attributes of PoS consensus schemes.

	Ourobros	Algorand	Nakamoto-Pos	Our scheme
Window size	κ	$\Theta(\kappa)$	1	1
Faulr tolerance	50%	33%	27%	36%/42%/46%
Threat of BA	High	Medium	Low	Low

Table 1 shows the comparison of our scheme and other schemes. κ is a security parameter in [5–7] and its value is 2160. The stake fault tolerance proportion is about 36% in our scheme when there is only one nonce generated in each round, it would be 42% while there is 2 and 46% while it's 3.

5 Experiments and Results

In our scheme, the candidate block broadcast time is the main part of the block generation protocol. Therefore, we implemented the prototype of the candidate block compression module and used the Bitcoin network in [16] to simulate it. Te block header size is 100B, the size of ordinary transaction is 256B, and the transaction hash size is 32B. Our simulation runs on a quad-core machine (Intel core i5-4590, 3.3 GHz, 8 GB RAM), and evaluated the average time of 100 block broadcasts on 75% and 90% of a 3000-node Bitcoin simulation network while the transaction compression proportion is 100%, 75% and 50%, 0% represents the original PoS scheme. We mainly evaluate the relationship between the $BlockSize$, $BlockBroadcastTime$ and $SystemPerformance$ and the transaction compression proportion.

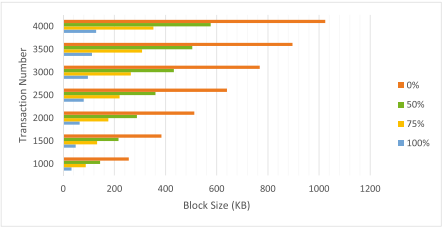

Fig. 4. Block size with different tx numbers and compression proportions.

Block Size. Figure 4 is the block size of the number of transactions ranges from 1000 to 4000. When the compression proportion is 100%, 75% and 50%, the size of block is from 32 KB to 128 KB, 88 KB to 352 KB, 144 KB to 576 KB, it's about 13%, 35%, and 56% of the original one. And it will be more efficient when the transaction size is larger than the ordinary ones.

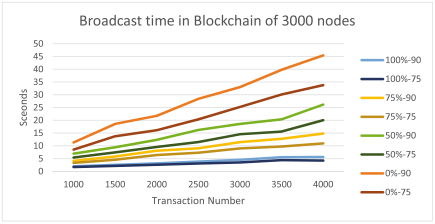

Fig. 5. Broadcast time with different number of transactions.

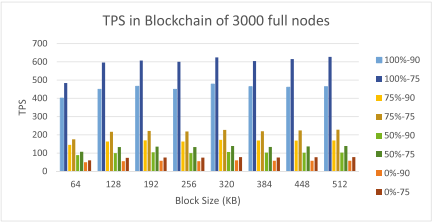

Fig. 6. System performance with different tx sizes.

Block Broadcast Time. Figure 5 is also the block broadcast time of the number of transactions ranges from 1000 to 4000. The time of block broadcasting to 75% nodes is about 15%, 35%, and 57% of original one; while 90% is about 14%, 33%, and 56%, and it grows faster with lower compression proportion. In addition, it takes about more than 26–34% of time to broadcast the block to 90% nodes than 75% when other conditions are the same.

System Performance. Figure 6 is the system performance of the block size ranges from 64 KB to 512 KB. When the compression proportion is 100%, 75% and 50%, TPS (transactions per second) is approximately 400–630, 140–230 and 90–140. It's about 8 times, 2.9 times, and 1.8 times of the ordinary PoS schemes. In addition, the TPS of the block broadcast to 90% of the network nodes is about 24% lower than that of 75% when other conditions are the same.

Due to the message delay of the distributed network, even the performance of the blockchain system of a 100% compressed block has an upper limit, but compared to the ordinary PoS system, the TPS of our solution is significantly improved, and there are also some optimized aspects. In general, the block compression method we proposed can reduce the size of the block under the same number of transactions and save the communication consumption of block broadcasting, it indeed improves the TPS of public chain.

6 Related Work

Review the work of the current public chain consensus protocols, Algorand [3] uses the VRF and BFT mechanisms in the PoS protocol for the first time. Hotstuff [4] optimizes the BFT protocol, and uses a pipeline method to optimize the

protocol execution to improve efficiency. Libra [18] is a more robust and efficient consensus mechanism designed based on Hotstuff. However, the scalability of these schemes is limited by BFT, and the consensus consumption will increase rapidly when the number of nodes in the network increases. Snow White [5] firstly proposes the framework of the provably secure PoS protocol, and analyzed the security of the protocol in an open network environment. Ouroboros [6–8] divides the longest chain PoS protocol into two stages, epoch and slot, which makes the stake fault tolerance proportion close to 50%. However it's vulnerable to bribery attacks. [9] proposes a new security property *chainsoundness* of blockchain. [17] discusses the impact of predictability on security of PoS protocols. These schemes are mainly concerned the security instead of the system performance of the protocol [19,20].

7 Conclusion and Discussion

In this paper, we propose a scalable and provably secure PoS protocol to support the application of public chains in extendable scenarios. We propose an independent stake mechanism that binds stake to nodes with transactions, and uses stake proportion table to support dynamic stake scenario. We propose a method of candidate block compression to decrease the communication consumption of block broadcast. We propose a new longest-chain rule, which supports the dynamic availability without a third party. In addition, we prove the security of our protocol and analyze its stake fault tolerance proportion. Our experimental results show that our communication consumption is lower with the same number of transactions, and the system performance is indeed better than other solutions. Furthermore, we plan to increase the stake fault tolerance proportion of this scheme, and prove its security in a more realistic and complex network environment. On the other hand, we will continue to research the public chain consensus schemes that have higher system performance.

Acknowledgments. This work is supported by the National Key Research and Development Program of China (2020YFB1807500), in part by Guangdong Basic and Applied Basic Research Foundation (2020A1515110496, 2020A1515110079).

References

1. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2009). https://bitcoin.org/bitcoin.pdf
2. Quantum Mechanic. Proof of Stake (2011). https://bitcointalk.org/index.php?topic=27787.0
3. Gilad, Y., Hemo, R., Micali, S., et al.: Algorand: scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles, pp. 51–68. ACM (2017). https://doi.org/10.1145/3132747.3132757

4. Brown-Cohen, J., Narayanan, A., Psomas, A., et al.: Formal barriers to longest-chain proof-of-stake protocols. In: Proceedings of the 2019 ACM Conference on Economics and Computation, pp. 459–473 (2019). https://doi.org/10.1145/3328526.3329567
5. Bentov, I., Pass, R., Shi, E.: Snow white: provably secure proofs of stake. IACR Cryptology ePrint Archive 2016:919 (2016)
6. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: a provably secure proof-of-stake blockchain protocol. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 357–388. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_12
7. David, B., Gaži, P., Kiayias, A., Russell, A.: Ouroboros Praos: an adaptively-secure, semi-synchronous proof-of-stake blockchain. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10821, pp. 66–98. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_3
8. Badertscher, C., Gaži, P., Kiayias, A., et al.: Ouroboros genesis: composable proof-of-stake blockchains with dynamic availability. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 913–930 (2018). https://doi.org/10.1145/3243734.3243848
9. Fan, L., Zhou, H.S.: A scalable proof-of-stake blockchain in the open setting (or, how to mimic Nakamoto's design via proof-of-stake). Cryptology ePrint Archive, Report 2017/656 (2017). Version 20180425:201821
10. Micali, S., Rabin, M., Vadhan, S.: Verifiable random functions. In: 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039), pp. 120–130. IEEE (1999). https://doi.org/10.1109/SFFCS.1999.814584
11. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**(3), 382–401 (1982). https://doi.org/10.1145/3335772.3335936
12. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 431–448. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_28
13. Kiayias, A., Panagiotakos, G.: Speed-security tradeoffs in blockchain protocols. IACR Cryptology ePrint Archive 2015:1019 (2015)
14. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: analysis and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 281–310. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_10
15. Pass, R., Seeman, L., Shelat, A.: Analysis of the blockchain protocol in asynchronous networks. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10211, pp. 643–673. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_22
16. Gervais, A., Karame, G.O., Wüst, K., et al.: On the security and performance of proof of work blockchains. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 3–16. ACM (2016). https://doi.org/10.1145/2976749.2978341
17. Bagaria, V., Dembo, A., Kannan, S., et al.: Proof-of-stake longest chain protocols: security vs predictability. arXiv preprint arXiv:1910.02218 (2019)
18. Bano, S., Baudet, M., Ching, A., et al.: State machine replication in the Libra Blockchain (2020). https://developers.libra.org/docs/state-machine-replication-paper. (Consulted on December 19, 2020)

19. Liu, L., Feng, J., Pei, Q., et al.: Blockchain-enabled secure data sharing scheme in mobile edge computing: an asynchronous advantage actor-critic learning approach. IEEE Internet Things J. **8**(4), 2342–2353 (2021). https://doi.org/10.1109/JIOT.2020.3048345
20. Feng, J., Yu, F.R., Pei, Q., et al.: Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach. IEEE Internet Things J. **7**(7), 6214–6228 (2020). https://doi.org/10.1109/JIOT.2019.2961707