GPE-PoS: A Fair and Sybil-Resistant Proof of Stake Consensus

Mingyue Zhang[1], Ming Liu[1], Xiang Ding[2], Yilei Wang[1], Guangshun Li[1]*

[1] School of Computer Science, Qufu Normal University, China
[2] College of Humanities and Tourism, Rizhao Polytechnic, China
zmy_qfnu@126.com, LiuMing2210@163.com, clove123@163.com,
wang_yilei2000@163.com, guangshunli@qfnu.edu.cn

Abstract

Consensus mechanisms are algorithms that ensure the security and stability of blockchain networks by achieving agreement and verifying transaction integrity. Proof of Stake (PoS) stands as a widely acknowledged consensus algorithm, wherein the privilege to validate transactions is predicated upon participants' stakes. However, long-term use of PoS may lead to wealth concentration among certain nodes, potentially undermining the network's fairness and security. Therefore, we propose the Group-Polynomial-based Election Proof of Stake (GPE-PoS) consensus mechanism. GPE-PoS involves categorizing nodes, calculating polynomial values for each group, encrypting these values using Paillier encryption, and then allocating validation rights based on comparisons of polynomial values based on polynomial value comparisons to enhance system fairness. The fairness of the system is further fortified against Sybil attacks, which undermine its security and fairness, through the incorporation of digital certificates within GPE-PoS, thereby verifying participant identities. Simulation results confirm that GPE-PoS successfully maintains fairness and security in blockchain systems.

Keywords: Proof of Stake consensus, Fairness, Security, Sybil attack, Paillier encryption

1 Introduction

Blockchain technology is a revolutionary innovation that is gradually transforming our traditional economic, social, and technological models [1]. The consensus mechanism is the fundamental principle and algorithm employed in blockchain networks to achieve agreement and validate transactions. It effectively addresses the issue of trust in distributed systems and ensures the consistency and security of data [2]. In conventional consensus mechanisms, such as the Proof of Work (PoW) approach, resource input or computing power is often relied upon. By contrast, the Proof-of-Stake (PoS) consensus mechanism presents a more eco-friendly and efficient alternative [3]. PoS demands significantly fewer computing resources than PoW, leading to substantial reductions in energy usage and maintenance costs [4]. However, as accounting rights accumulate, there is a risk of exacerbating wealth inequalities, leading to a "rich get richer" scenario and ultimately undercutting the fairness of the blockchain system [5].

As is well known, Sybil attack is a widely recognized security threat in blockchain systems [6-7]. In a PoS consensus network, Sybil attackers can fabricate a large number of virtual identities, and allow them to possess a greater amount of tokens and consequently influence the consensus process [8-10]. Attackers can influence network decisions by controlling a majority of nodes, selectively validating or rejecting transactions during block validation, and potentially manipulating transaction records [11].

To tackle the challenges mentioned above, we introduce the Group-Polynomial-based Election Proof of Stake (GPE-PoS) consensus mechanism. GPE-PoS enhances the fairness of the PoS consensus mechanism by grouping nodes and allocating block accounting rights within each group through a polynomial-based election process. To mitigate Sybil attacks, we have to make improvements to GPE-PoS by introducing digital certificates for participant identity verification. Additionally, we employ Paillier encryption to safeguard sensitive data during the block producers' selection, thereby boosting system security.

1.1 Related Works

In response to the centralization issue in the PoS consensus mechanism, the academic community has proposed several solutions. In papers [12-14], the stake-based proof mechanisms are replaced with credit-based, voting-based, and random-based consensus mechanisms, respectively. While these changes can diminish centralization in blockchain, evaluating credit and conducting statistical voting entail complexities and are vulnerable to fraud, manipulation, or erroneous information. Additionally, random selection may provide opportunities for malicious participants or attackers. To mitigate this concern, paper [15] introduces the DPoS consensus mechanism based on fuzzy sets, which adjusts the weights of participants based on their contributions to reduce the influence of cheating participants. However, it should be noted that this algorithm is computationally intensive and necessitates additional computational resources. In response to these challenges, Christian Badertscher et al. propose a composable proof method known as Ouroboros Genesis [16]. This method restricts

*Corresponding Author: Yilei Wang; Email: wang_yilei2000@163.com
DOI: https://doi.org/10.70003/160792642025072604005

participants' verification and accounting operations to specific time slots, effectively reducing resource consumption. However, the security of this algorithm relies on the majority of honest nodes in the network, making it unsuitable for networks under Sybil attacks. To address this issue, paper [17] presents an "Identity-Augmented" PoS algorithm aimed at mitigating the Sybil attack vulnerability present in conventional PoS algorithms. Nonetheless, introducing identity verification and stake distribution mechanisms could potentially lead to centralization tendencies and concentration of power.

1.2 Motivations and Contributions

Given the above issues, we aim to improve the Proof of Stake (PoS) consensus mechanism to enhance fairness and security in blockchain systems. In terms of fairness, we hope to provide mining opportunities to a broader range of stakeholders in the improved PoS algorithm. Regarding security, there are two key considerations. Firstly, the protocol should be capable of thwarting Sybil attacks and ensuring equitable recompense to potential attackers in the event of an attack. Secondly, in the process of selecting block producers, it is crucial to ensure that the participants' token holdings remain confidential to guarantee the secure operation of the consensus mechanism. With these motivations in mind, we propose GPE-PoS. The main contributions of this paper are as follows:

(1) We propose the GPE-PoS consensus mechanism as a solution to the centralization issue within PoS. The core idea is to group nodes based on their stake and calculate a polynomial value for each node within the group. Subsequently, block-producing rights are assigned based on these polynomial values. Through this grouping approach, the consensus mechanism provides more participants with the opportunity to compete for block-producing rights. this expansion ultimately broadens the distribution of block-producing power, consequently fostering fairness in blockchain consensus.

(2) We measure the fairness of the blockchain system using the Gini coefficient and identify that Sybil attacks pose a threat to the GPE-PoS mechanism by causing the system's Gini coefficient to increase from 0.1 to 0.9. As the Gini coefficient serves as an indicator of fairness within a system, this rise signifies a decline in fairness. Consequently, Sybil attacks not only compromise the fairness of the blockchain system but also compromise the overall security of the entire blockchain network.

(3) In the GPE-PoS consensus mechanism, we introduce an identity verification mechanism to identify illegitimate participants and thwart Sybil attacks. Simulation results demonstrate that the Gini coefficient of the blockchain system using the improved GPE-PoS mechanism stabilizes at around 0.3. This represents a notable improvement, as it signifies a 66.7% reduction in the Gini coefficient compared to pre-improvement levels, underscoring heightened fairness and enhanced security within the system.

1.3 Roadmap

The rest of the paper is organized as follows. Section 2 introduces the fundamental knowledge. Then, Section 3 outlines the details of the proposed solution. In Section 4, we describe the experimental details such as the operating environment and parameters used in the simulation experiments. Finally, in Section 5, we conclude the paper.

2 Preliminaries

2.1 Proof-of-Stake Consensus

The PoS consensus is a new consensus that distributes privileges based on the stake of digital currency, thereby establishing a fair and efficient consensus mechanism [18-20]. In PoS, participants can attain validator status by locking a specified amount of digital currency [21-24].

Suppose there is a set of participants $P = \{P_1, P_2, \ldots, P_n\}$, and each participant P_i holds a certain amount of cryptocurrency as their interest. Each participant P_i has an interest function V_i which represents the amount of cryptocurrency held by P_i. The goal of the consensus algorithm is to select a participant P_i as the next verifier, and this selection should be based on their interests. The probability of choosing the next producer should be proportional to the participants' equity. Assuming that the total equity is T, then the probability of P_i being selected as the next validator can be expressed as: $P_i = \frac{V_i}{T}$.

2.2 Paillier Encryption

Paillier encryption is a public key encryption algorithm proposed by computer scientist Pascal Paillier in 1999. It is based on computationally complex modular exponentiation and number theory problems, and it possesses provable security and homomorphic properties. Paillier encryption has been widely used for privacy protection and secure computation [25]. The core principle of Paillier encryption is based on solving the discrete logarithm problem with integers, including key generation, encryption, and decryption as its main steps.

(1) Key Generation

- Choose two large prime numbers, p and q. Then compute $n = p * q$.
- Calculate $\lambda = lcm(p-1, q-1)$, which is the least common multiple of $(p-1)$ and $(q-1)$.
- Select a random number g, such that $g^{\lambda} \bmod n^2 = 1$.
- The public key is (n, g), and the private key is (p, q).

(2) Encryption

- Assume the plaintext to be encrypted is denoted as m.
- Choose a random number r that satisfies $0 < r < n$ and gcd $(r, n) = 1$ (where gcd denotes the greatest common divisor).
- Calculate $c = (g^m * r^n) \bmod n^2$.
- The resulting ciphertext is denoted as c.

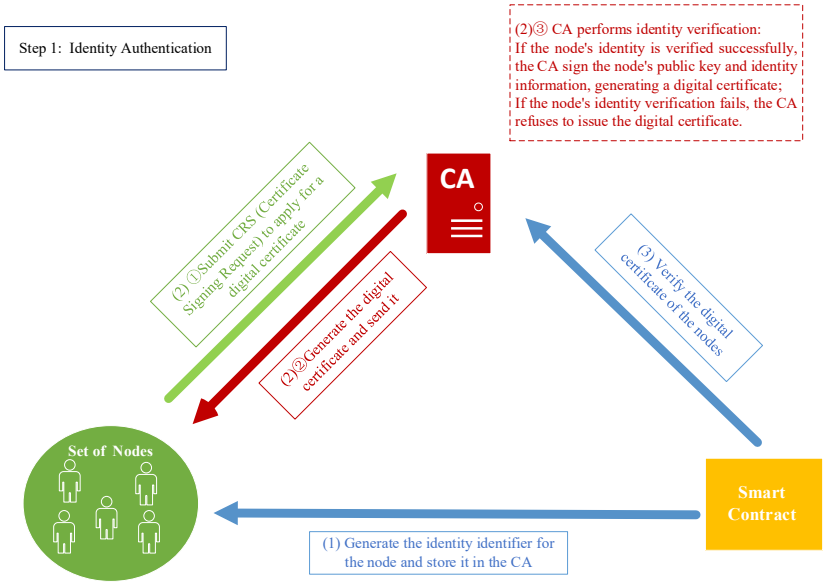


Figure 1. Identity authentication

(3) Decryption

- Assume the received ciphertext is denoted as c.
- Use the private key (p, q) to calculate

$$\frac{L(c^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)}, \text{ where } L(x) = \frac{(x-1)}{n}.$$

- The resulting decrypted plaintext is denoted as m.

The Paillier encryption scheme possesses the property of homomorphism, which means that when we perform multiplication on two ciphertexts, c_1 and c_2, obtained by encrypting two plaintexts m_1 and m_2 respectively, the decrypted result is equal to the product of the two plaintexts, that is $Dec(c_1 * c_2) = m_1 * m_2$.

3 GPE-PoS

3.1 Design Overview

We propose a remarkable enhancement to the PoS consensus mechanism, called Group-Polynomial-based Election Proof of Stake (GPE-PoS), as depicted in Figure 1 and Figure 2. The basic idea is to group the nodes in the network and allocate block generation rights to each group based on the computed polynomial results. However, experiments have indicated that the fairness of GPE-PoS can be compromised in the presence of a Sybil attack. Therefore, we distribute a unique and randomly generated identity identifier to each node as a basis for requesting a digital certificate from a Certificate Authority. In the process of competing for block generation rights, a smart contract diligently verifies the authenticity of the digital certificates belonging to participating nodes, effectively fortifying the system against Sybil attacks. Additionally, to prevent leakage of the polynomial values of participating nodes, we use Paillier encryption to encrypt the polynomial values and compare them using the method proposed in reference [26] to select the node with the maximum value for block generation rights.

3.2 Node Grouping

Algorithm 1 demonstrates how to group the participating nodes based on their stake. Initially, we calculate the number of nodes per group, groupSize, based on the desired number of groups, numGroups. Moving forward, we initialize an array, groups, to store the node IDs and corresponding stakes for each group. Next, we sort all the nodes using the sort function and store the sorted nodes in the list sortNodes. Lastly, we evenly distribute the nodes from sortNodes into the groups array in descending order, based on the groupSize, and return the result.

3.3 Selecting Block Producers

3.3.1 Polynomial Calculation

After grouping the nodes, the nodes within each group need to compete for block generation rights. Our method uses a polynomial calculation to determine the block generation rights. The calculation of the fifth-degree polynomial for each node depends on the node's stake, the number of times the node has generated blocks in the past, and parameter settings.

In Formula 1, the "Diff" value for each node is calculated as the weighted average of the node's stake minus the number of times it has been selected as a block producer in previous periods divided by the total number of selections. In Formula 2, the "Score" value for each node is calculated by substituting the "Diff" value from Formula 1 into the fifth-degree polynomial, where the parameters a[k] and a[0] are pre-set.

$$Diff[i] = stake[i] - \frac{blockCount[i]}{totalCount} * sW. \quad (1)$$

$$Score[i][j] = \sum_{k=1}^{k=5} a[k] * Diff[i][j]^k + a[0]. \quad (2)$$

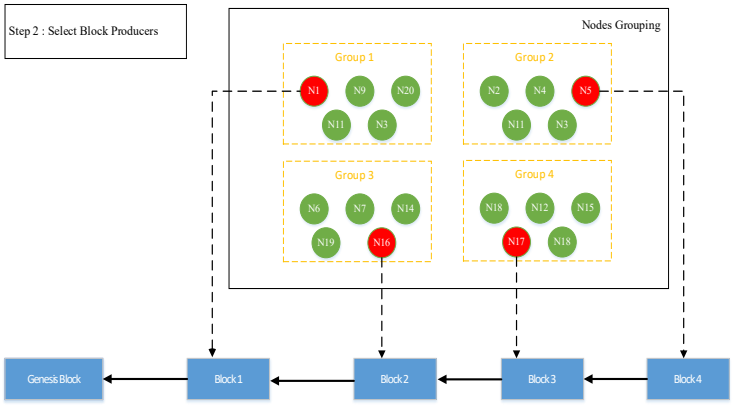


Figure 2. Select block producers

Algorithm 1. Node grouping algorithm

Input: Stakes, numGroups
Output: groups
//Number of nodes in each epoch

```
groupSize = length(stake)/numGroups
    //Calculate the size of each group and initialize
    the groups array
for i = 0 to numGroups-1 do
    groupList[i] = new array
end
//Assigning nodes to groups
nodeIndex = 0
for i = 0 to numGroups-1    do
    for j = 0 to groupSize-1   do
        group[i][j] = nodeIndex
        nodeIndex ++
    end
end
//Copy groups[0] to the sortedNodes array
sort.Slice(sortedNodes,func(i,j)>bool{
return stake[sortedNodes[i]]−>stake[sortedNodes
[j]]})
//Update the order of nodes within each grouping
nodeIndex = 0
for i = 0 to numGroup-1   do
    for j = 0 to groupSize-1 do
        Groups[i][j] = sortedNodes[nodeIndex]
        nodeIndex = nodeIndex + 1
    end
end
return groups
```

3.3.2 Comparing Polynomial Values

After calculating the "Score" values for each node, we select the node with the highest "Score" value as the block producer within each group. Given that the "Score" value plays a pivotal role in the competition for block generation rights, it is important to prevent other nodes from knowing the exact value. If the value is known, other nodes may try to modify their own "Score" values to increase their chances of being selected as block producers, thereby compromising the fairness of the entire blockchain system. In Algorithm 2 and Algorithm 3, we demonstrate how to compare the two values without revealing them.

3.3.3 Achieve Byzantine Consensus

Byzantine fault-tolerant algorithms are a class of algorithms used to achieve consensus in distributed systems with malicious or faulty nodes [27-29]. Figure 3 shows the basic process of a Byzantine fault-tolerant algorithm. In this paper, we achieve a consensus on the comparison results for each node, determine the block producer based on the consensus result, allocate block generation rights, and provide 5% of the transaction amounts rewards.

Algorithm 2. Compare integers

Input: ScoreA, ScoreB
Output: C
// Generate public and private keys for the Paillier encryption scheme

```
keyLength =2048
pk, sk = Paillier.GenerateKeyPair(keyLength)
// Encrypts the input integer
a_enc = pk.Encrypt(ScoreA)
b_enc = pk.Encrypt(ScoreB)
// Generate and encrypt random integers
r = randomInt(pk.N)
r_enc = pk.Encrypt(r)
// Performs ciphertext operations
c_enc = a_enc – b_enc +r_enc
// Decryption Comparison Results
C = sk.Decrypt(c_enc)
return C
```

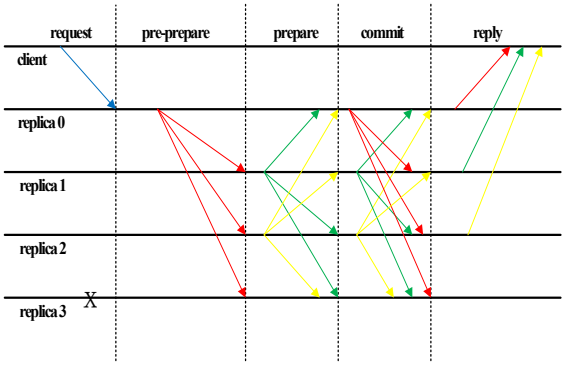


Figure 3. PBFT protocol

Algorithm 3. Select block producers

Input: groups, stake
Output: blockProducers
// *Initialize the lists*

```
blockProducers = new an empty array
Score = new an empty array
//Calculate the Score value
Score = ComputePolynomial (groupDiff , r)
maxIndex = 0
maxP = 0.0
for j, p in enumerate(polynomial) do
    if p>maxP then
        maxIndex = j
        maxP = p
    end
end
selectedProducerIndex = groups[i][maxIndex]
// Reaching the Byzantine Consensus
for index in groups[i] do
    comparisonResult =
    compareIntegers(stake[index])
    if comparisonResult == -1 then
        selecteProducerIndex = index
    end
end
// Determining Block Producers
blockProducers.append (selectedProducerIndex)
return blockProducers
```

3.4 Resisting Sybil Attack

In GPE-PoS, we have designed a node authentication mechanism called Node-Authentication. The process of authentication involves the following steps, as shown in Figure 1:

(1) Generating an Identity Identifier: In the context of smart contracts, a 16-bit binary string serves as the identity identifier for each node, which is then distributed and stored within a Certificate Authority (CA). Each node autonomously generates a key pair, with the public key used for encrypting the identity identifier and verifying digital signatures, while the private key is employed for decrypting the identity identifier and generating digital signatures.

(2) Applying and Distributing Digital Certificates: Nodes utilize their private key to generate a Certificate Signing Request (CRS), which contains relevant node information such as the encrypted identity identifier and public key. Subsequently, the CRS is then submitted to the CA for verification of the node's identity. After successful verification, the CA leverages its private key to digitally sign the node's public key and identity information, thus generating a digital certificate.

(3) Verifying Digital Certificates: Following the issuance of digital certificates to each node, we need to verify the authenticity of the certificates. Since the digital certificates are retained within the CA, we invoke the function verifyDigitalCertificate() to compare the node's digital certificate information with the stored data in the CA. A match signifies the validity of the node's identity, ensuring that it is not a virtual identity fabricated by a malicious node.

(4) Rejecting Fake Identities: In the event of a mismatch in a node's digital certificate, the smart contract rejects the node's engagement in the consensus process, effectively thwarting the inclusion of false identities and mitigating the risk of Sybil attacks.

4 Simulations and Results

4.1 Simulation

Now, we are conducting experimental simulations on the Proof-of-Stake (PoS) based Group Polynomial Election (GPE) mechanism. We utilize the Gini coefficient to measure the fairness of the consensus mechanism. The Gini coefficient nearing 0 signifies a fairer distribution, while a value approaching 1 indicates greater inequality. By enhancing the GPE-PoS mechanism and conducting a comparative analysis of the Gini coefficients before and after the modifications, we aim to evaluate the security enhancements of the refined consensus mechanism.

To simulate a normal blockchain network, we define 10,000 nodes and generate a random stake value for each node, representing their respective weights. We assume that the entire network is fully connected, allowing for peer-to-peer communication between any two nodes. We set up 100 epochs, where the consensus mechanism runs for 100 epochs, producing one Gini coefficient, and visualize them to provide a more intuitive representation of the changes in fairness. Subsequently, we simulate competition among nodes and evaluate the security of the consensus mechanism under different parameter settings. Utilizing the Go programming language, we simulate Sybil attacks on the GPE-PoS consensus.

4.2 Results and Evaluation

We present the simulation results in Figure 4, Figure 5, and Table 1. In Figure 4, we compare the Gini coefficients

of the original PoS and the unimproved GPE-PoS. The Gini coefficient of the original PoS gradually decreases over epochs, indicating an improvement in fairness that tends to stabilize. We disregard the Gini coefficients of GPE-PoS for the first 20 epochs because the stake distribution among nodes in the network is uneven during this period, making it unable to calculate the Gini coefficient. From the 20th epoch onwards, the Gini coefficient of GPE-PoS shows a decreasing trend and stabilizes around 0.1 in the 100th epoch. This indicates that compared to the original PoS, GPE-PoS reduces the Gini coefficient by around 0.5, making it relatively fair.

In Figure 5 and Table 1, we demonstrate the ability of GPE-PoS to resist Sybil attacks and provide the Gini coefficient variations of the blockchain system over 100 epochs. In Figure 5, "Sybil_attack" represents the change in Gini coefficient when launching a Sybil attack on GPE-PoS, while "Defect_sybil_attack" represents the change in Gini coefficient when launching a Sybil attack on the improved GPE-PoS. In Table 1, we calculate the average Gini coefficient over the 100 epochs. "PoS" represents the original Proof-of-Stake consensus mechanism, denoted with '*' as there is no grouping in this consensus. "GPE-PoS" represents the system under Sybil attacks, and "SY_GPE-PoS" represents the improved system resilient to Sybil attacks.

Since GPE-PoS uses a grouping selection method, the number of groups (numGroup) has an impact on the allocation of block-producing rights. Combining Figure 5 with Table 1, Figure 5(a) represents that when numGroups is 10, the Gini coefficient stabilizes at around 0.9 and the average value is 0.9347 after a Sybil attack on the system. This indicates a highly uneven distribution of benefits in the network and low fairness and security. As the number of groups increases, we can see that the Gini coefficient represented by "sybil_attack" gradually decreases, and fairness improves. This is because the more groups there are, the lower the probability of virtual nodes being selected as block producers. From Figure 5(a) to Figure 5(c) corresponding to SY-GPE-PoS in Table 1, we can see that the Gini coefficient represented by "Defect_sybil_attack" remains stable and the mean value is maintained at around 0.33, which is far lower than that represented by GPE-PoS. In summary, the improved GPE-PoS can resist Sybil attacks and enhance the fairness and security of the system.

Table 1. Mean values of Gini coefficients under different numbers of groups

Scheme	numGroups	Mean Gini
PoS	*	0.7888
GPE-PoS	10	0.9347
	15	0.6360
	20	0.4857
SY_GPE-PoS	10	0.3342
	15	0.3358
	20	0.3374

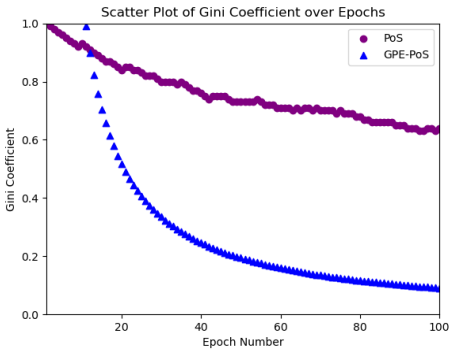
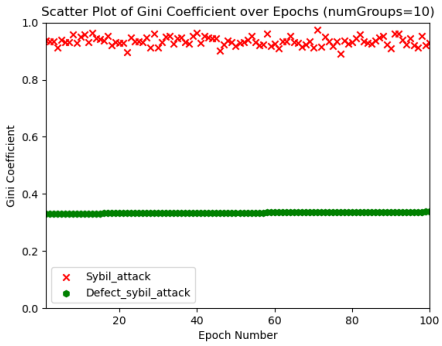
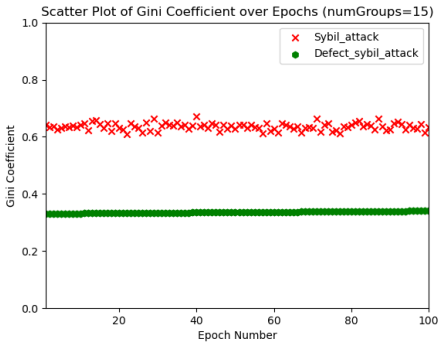


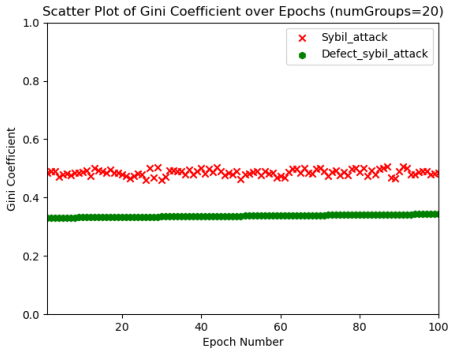
Figure 4. Gini coefficient under PoS and GPE-PoS



(a) Gini coefficient when numGroups=10



(b) Gini coefficient when numGroups=15



(c) Gini coefficient when numGroups=20

Figure 5. Gini coefficient under different numbers of groups

5 Conclusion

In PoS consensus, participants engage in the consensus process by token staking, which can result in the concentration of wealth and compromise the decentralization principle of blockchain. In this paper, we propose a PoS consensus based on polynomial grouping, aimed at enhancing system fairness by categorizing nodes into distinct groups and selecting block producers from within these groups. However, Sybil attacks can break through this consensus mechanism, and increase the Gini coefficient above 0.4, thereby undermining system security. To address this vulnerability, we enhance the GPE-PoS protocol by implementing digital certificate verification for node identity authentication to resist Sybil attacks. Additionally, we employ Paillier encryption to secure critical consensus data while ensuring equitable interest distribution. Through simulation, our findings demonstrate that surpasses traditional PoS in fairness, with lower Gini coefficients, thereby fortifying its resilience against Sybil attacks. Future research endeavors will focus on refining the performance and scalability of GPE-PoS, as well as exploring diverse applications of this innovative consensus mechanism.

Acknowledgement

This study is supported by the Foundation of National Natural Science Foundation of China (Grant No.: 62072273, 72111530206, 61962009); The Major Basic Research Project of Natural Science Foundation of Shandong Province of China (ZR2019ZD10); Natural Science Foundation of Shandong Province (ZR2019MF062); Shandong University Science and Technology Program Project (J18A326); Guangxi Key Laboratory of Cryptography and Information Security (No: GCIS202112); Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2019BD-KFJJ009); This work was supported by the Key-Area Research and Development Program of Guangdong Province (No. 2020B0101130015).

References

[1] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, Y. Bian, Blockchain Security: A Survey of Techniques and Research Directions, *IEEE Transactions on Services Computing*, Vol. 15, No. 4, pp. 2490-2510, July-August, 2022.

[2] Y. Fang, Z. Zhou, S. Dai, J. Yang, H. Zhang, Y. Lu, A Parallel Virtual Machine for Smart Contract Execution and Validation, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 35, No. 1, pp. 186-202, January, 2024.

[3] I. Rosu, F. Saleh, Evolution of Shares in a Proof-of-Stake Cryptocurrency, *Management Science*, Vol. 67, No. 2, pp. 661-672, February, 2021.

[4] J. Li, Z. Zhou, J. Wu, J. Li, S. Mumtaz, X. Li, H. Gacanin, S. Alotaibi, Decentralized On-Demand Energy Supply for Blockchain in Internet of Things: a microgrids approach, *IEEE transactions on computational social systems*, Vol. 6, No. 6, pp. 1395-1406, December, 2019.

[5] J. Zhang, S. Zhong, T. Wang, H. C. Chao, J. Wang, Blockchain-based systems and applications: a survey, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 1-14, January, 2020.

[6] D. Chulerttiyawong, A. Jamalipour, Sybil Attack Detection in Internet of Flying Things-IoFT: A Machine Learning Approach, *IEEE Internet of Things Journal*, Vol. 10, No. 14, pp. 12854-12866, July, 2023.

[7] M. Platt, P. McBurney, Sybil Attacks on Identity-Augmented Proof-of-Stake, *Computer Networks*, Vol. 199, Article No. 108424, November, 2021.

[8] Y. Liu, J. Liu, M.-V. Salles, Z. Zhang, T. Li, B. Hu, F. Henglein, R. Lu, Building Blocks of Sharding Blockchain Systems: Concepts, Approaches, and Open Problems, *Computer Science Review*, Vol. 46, Article No. 100513, November, 2022.

[9] Z. Liu, X. Zhang, L. Lao, G. Li, B. Xiao, DBE-voting: A Privacy-Preserving and Auditable Blockchain-Based E-Voting System, *IEEE International Conference on Communications*, Rome, Italy, 2023, pp. 6571-6577.

[10] S. Joshi, R. Li, S. Bhattacharjee, S. K. Das, H. Yamana, Privacy-Preserving Data Falsification Detection in Smart Grids using Elliptic Curve Cryptography and Homomorphic Encryption, *IEEE International Conference on Smart Computing*, Helsinki, Finland, 2022, pp. 229-234.

[11] S. Zhang, J. H. Lee, Double-Spending with a Sybil Attack in the Bitcoin Decentralized Network, *IEEE transactions on Industrial Informatics*, Vol. 15, No. 10, pp. 5715-5722, October, 2019.

[12] X. Han, Y. Yuan, F. Y. Wang, A Fair Blockchain Based on Proof of Credit, *IEEE Transactions on Computational Social Systems*, Vol. 6, No. 5, pp. 922-931, October, 2019.

[13] G. Sun, M. Dai, J. Sun, H. Yu, Voting-based Decentralized Consensus Design for Improving the Efficiency and Security of Consortium Blockchain, *IEEE Internet of Things Journal*, Vol. 8, No. 8, pp. 6257-6272, April, 2021.

[14] R. Pass, E. Shi, Fruitchains: A Fair Blockchain, *Proceedings of the ACM symposium on principles of distributed computing*, Washington, DC, America, 2017, pp. 315-324.

[15] G. Xu, Y. Liu, P. W. Khan, Improvement of the DPoS Consensus Mechanism in Blockchain Based on Vague Sets, *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 6, pp. 4252-4259, June, 2020.

[16] C. Badertscher, P. Gazi, A. Kiayias, A. Russell, V. Zikas, Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto, Canada, 2018, pp. 913-930.

[17] M. Drijvers, S. Gorbunov, G. Neven, H. Wee, Pixel: Multi-Signatures for Consensus, *29th USENIX Security Symposium*, Virtual Event, 2020, pp. 2093-2110.

[18] D. Liu, A. Alahmadi, J. Ni, X. Lin, X. Shen, Anonymous Reputation System for IIoT-Enabled Retail Marketing Atop PoS Blockchain, *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 6, pp. 3527-3537, June, 2019.

[19] J. Wang, H. Wang, Monoxide: Scale Out Blockchains with Asynchronous Consensus Zones, *16th USENIX symposium on networked systems design and implementation*, Boston, MA, America, 2019, pp. 95-112.

[20] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, Y. Li, Performance Analysis and Comparison of PoW, PoS and DAG based Blockchains, *Digital Communications and Networks*, Vol. 6, No. 4, pp. 480-485, November, 2020.

[21] M. Li, D. Hu, C. Lal, M. Conti, Z. Zhang, Blockchain-Enabled Secure Energy Trading With Verifiable Fairness

in Industrial Internet of Things, *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 10, pp. 6564-6574, October, 2020.
[22] J. R. Douceur, The Sybil Attack, *First International workshop on peer-to-peer systems*, Cambridge, MA, America, 2002, pp. 251-260.
[23] L. Xiao, L. J. Greenstein, N. B. Mandayam, W. Trappe, Channel-Based Detection of Sybil Attacks in Wireless Networks, *IEEE Transactions on Information Forensics and Security*, Vol. 4, No. 3, pp. 492-503, September, 2009.
[24] X. Yi, R. Paulet, E. Bertino, *Homomorphic Encryption and Applications*, Springer International Publishing, 2014.
[25] F. Bourse, O. Sanders, J. Traoré, Improved Secure Integer Comparison via Homomorphic Encryption, *Cryptographers' Track at the RSA Conference*, San Francisco, CA, America, 2020, pp. 391-416.
[26] M. Castro, B. Liskov, Practical byzantine fault tolerance and proactive recovery, *ACM Transactions on Computer Systems*, Vol. 20, No. 4, pp. 398-461, November, 2002.
[27] Y. Xiao, N. Zhang, W. Lou, Y. T. Hou, A Survey of Distributed Consensus Protocols for Blockchain Networks, *IEEE Communications Surveys & Tutorials*, Vol. 22, No. 2, pp. 1432-1465, Second Quarter, 2020.
[28] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, M. A. Imran, A Scalable Multi-Layer PBFT Consensus for Blockchain, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 32, No. 5, pp. 1146-1160, May, 2021.
[29] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, Y. Wang, PoBT: A Lightweight Consensus Algorithm for Scalable IoT Business Blockchain, *IEEE Internet of Things Journal*, Vol. 7, No. 3, pp. 2343-2355, March, 2020.

Biographies

Mingyue Zhang is currently studying for a master's degree at the School of Computer Science, Qufu Normal University, She received a bachelor's degree in network engineering from Qufu Normal University in 2018. Her research interests include blockchain technology and applications, machine learning, and information security theory.

Ming Liu is currently studying for a master's degree at the School of Computer Science. She received a bachelor's degree in computer science and technology from Qufu Normal University in 2017. Her research interests include blockchain technology and applications, machine learning, and information security theory.

Xiang Ding received her Bachelor's degree in computer English from Liaocheng University, China, and a Master's degree in computer science from Shandong Normal University, China, She is currently a full Lecturer with School of Computing English, Rizhao Polytechnic. She mainly engaged in research on network English. In recent years, she has published more than 15 academic papers in domestic authoritative journals and conferences.

Yilei Wang received her Doctoral degree in computer technology from Shandong University. She is currently a full Professor with School of Computing Sciences, Qufu Normal University. She mainly engaged in research on network security, blockchain and smart contracts. In recent years, she has published more than 30 academic papers in international authoritative journals and conferences.

Guangshun Li received the M.E. and Ph.D. degrees from Harbin Engineering University, China, in 2004 and 2008, respectively. He is currently a professor at the School of Computer Science, Qufu Normal University. He has published more than 70 academic papers. His research interests include network security, blockchain, UAV communication applications, and artificial intelligence.