Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6185910, 10 pages
https://doi.org/10.1155/2021/6185910

Research Article

Proof of Engagement: A Flexible Blockchain Consensus Mechanism

Yuntao Xu,[1] Xingyu Yang,[1] Jiale Zhang,[1] Junwu Zhu,[1] Maosheng Sun,[1] and Bing Chen[2]

[1]*College of Information Engineering, Yangzhou University, Yangzhou 225127, China*
[2]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*

Correspondence should be addressed to Jiale Zhang; jialezhang@yzu.edu.cn

Received 2 July 2021; Accepted 10 August 2021; Published 20 August 2021

Academic Editor: Weizhi Meng

Consensus mechanism plays an important role in blockchain. At present, mainstream consensus mechanisms include proof of work (PoW), proof of stake (PoS), and delegated proof of stake (DPoS). PoW, as is widely used in virtual currency, results in significant energy consumption; PoS and DPoS are proposed to reduce energy waste caused by PoW, but their disadvantage is that they tend to create Matthew Effect (ME): "the rich get richer." In order to balance the discourse power of new nodes and elder ones, this paper proposes a flexible consensus mechanism called proof of engagement (PoE), based on the activity and contribution of network nodes. We analyze the incentive compatibility of PoE from the perspective of mechanism design. In our simulation experiments, we tested the profit changes under PoW, PoS, and PoE. The results illustrate it is easier for new nodes to accumulate their profits under PoE than under PoW or PoS, so as to reduce the negative impacts of ME.

1. Introduction

Bitcoin [1], since it was proposed in 2008, has been considered the most successful application of blockchain. Its ability to work properly on a distributed system relies on the genius consensus algorithms, proof of work (PoW) [2]. Bitcoin is also commonly called Blockchain 1.0. Ethereum [3], also based on PoW, is called Blockchain 2.0 for its Turing-complete smart contract system.

Despite its widespread use, PoW still has several much-criticized problems [4, 5], and one of the most serious is its energy consumption. According to digiconomist.net [6], as of October 2020, Bitcoin has consumed electrical energy 74.38 TWh per year, which is comparable to the power consumption of Venezuela. The carbon footprint of Bitcoin has reached 35.33 Mt CO_2 per year, comparable to the carbon footprint of New Zealand. The reason for such a disappointing situation is that under PoW, network nodes need to run the SHA256 algorithm repeatedly until they successfully find the hash solution; then, they will be rewarded by many digital currencies; this process completely depends on the computing power of the devices. Another problem is the centralization of computing power. To increase the chances of getting a reward, the users either buy more powerful computing devices and keep them running at full capacity or join some huge mining pools [7], which causes a shift in the computing power from decentralized back to centralized [8] and greatly threatens the security of the blockchain network.

Proof of stake (PoS) [9] was originally designed to solve the energy consumption problem. Under PoS, the probability of getting a reward is affected not only by the computing power but also by the length of time a node holding the coins (coinage). Thus, to some extent, PoE reduces energy consumption and weakens the absolute control of the full-time miners and mining pools over the blockchain network. However, this easily leads to the Matthew Effect [10, 11]: the richer always gains more profits than those who are not that rich. Unfortunately, it will be hard for new users to gain their profits under PoS, and the discourse power will gradually be centralized in the hands of a few rich ones. The centralization problem is still not effectively solved; this reduces the incentive of the whole system.

To increase the incentive of the system and reduce the negative effect of ME, in the following we propose a new consensus mechanism named proof of engagement (PoE). A blockchain system under PoE is more like a work-based society, where nodes can accumulate their profits by contributing computing power to maintain the security and creating high-quality smart contracts to maintain the autonomy. In this way, new nodes will be able to gain a voice more easily; the flexibility of the whole system is also increased.

Our contributions. Our contributions are summarized as follows:

(i) We propose PoE, a flexible consensus mechanism for a smart contract system based on blockchain, which shows the ability to reduce the negative impacts of ME

(ii) We build the static and dynamic evaluation models of smart contracts and calculate the contract quality according to the evaluation results. We calculate the activity of a node based on its transaction volume and computing power contribution during recent periods of blocks. From the perspective of mechanism design, we analyze the incentive compatibility of PoE

(iii) We propose a method to study the flexibility of different consensus mechanisms. In our simulation experiments, we test the profit changes of those 3 nodes under PoW, PoS, and PoE. The results show that PoE is more flexible than PoW and PoS. At last, we discuss the main application directions of PoE

Paper organization. We organize the remainder of this paper as follows. Some related works are listed in Section 2; models and algorithms are presented in Section 3; the incentive compatibility of PoE is analyzed in Section 4; our simulation experiments and discussions are presented in Section 5; our conclusions are summarized in Section 6.

2. Related Work

Yuan et al. presented abstract models of PoW and PoS [12]. They modeled the rules of block production into simple inequalities.

PoW is modeled as follows:

$$F_{\text{diff}}(\text{blockheader}) \longrightarrow \text{SHA256}(\text{SHA256}(\text{blockheader})) < \frac{\text{MaxTarget}}{\text{diff}} \quad (1)$$

In this model, MaxTarget represents the maximum target value, and diff represents the degree of difficulty, which is used to control the production interval of each block.

PoS is modeled as follows:

$$\text{SHA256}(\text{SHA256}(\text{timestamp})) < \text{target} \times \text{CoinAge}. \quad (2)$$

In this model, CoinAge means the holding time of coins.

In recent years, researchers have been constantly improving or redesigning consensus mechanisms to make up for the deficiency of mainstream mechanisms and get adaption to different applications of blockchain.

To reduce the energy waste of PoW, various consensus mechanisms have been proposed. PoS [13, 14] is recognized as an excellent improvement. Besides, proof of luck (PoL) [15] and proof of elapsed time (PoET) [16], based on the trusted execution environments (TEE), are also practical alternatives. In such TEE, the node who becomes the bookkeeper is decided by the waiting time generated by a random number generator, according to a presupposed probability. The introduction of TEE greatly reduces energy consumption and improves the output efficiency of blocks. Proof of useful work (PoUW) [17], proposed in 2017, gets rid of the meaningless SHA256 operation in PoW and replaces it with valuable operations in the actual scene, such as computing orthogonal vector problem, 3SUM problem, and shortest path problem. PBFT [18] is completely different from the concept of PoW: rather than choosing a winner to lead but to make sure that everyone performs the same action. It does not need to consume a lot of computing power.

Several studies are committed to solving the centralization problem brought by PoW and PoS. Based on the combination of PoW and PoS, researchers proposed proof of burn (PoB) [19], proof of activity (PoA) [20], etc. PoB enforces the miners to send their coins to a specific address that cannot be found, that is, to compete for the bookkeeping right by "burning" their coins, which alleviates the Matthew Effect to a certain extent.

3. Model and Process

3.1. Generic Model. Consensus mechanisms like PoW and PoS are usually classified as the "proof-class" mechanisms, in which once the calculated value is within the target range, and a new block is produced. PoE is also a kind of proof-class mechanism. In order to model PoE, we first present the abstract definition of a consensus mechanism for the blockchain.

Definition 1. A consensus mechanism $M := <\mathbb{R}, \times B, f_M>$ is a triple, which consists of the following:

(i) A real number set $\mathbb{R}$

(ii) A blockchain $\times B = \{B_{[0]}, B_{[1]}, \cdots, B_{[k]}\}$

(iii) A mapping $f_M : \mathbb{R} \longrightarrow B_{[k+1]}$ of block production

Based on Definition 1, we present the generic model of the proof-class consensus mechanisms, as is shown in Figure 1. In Figure 1, the consensus mechanism extracts

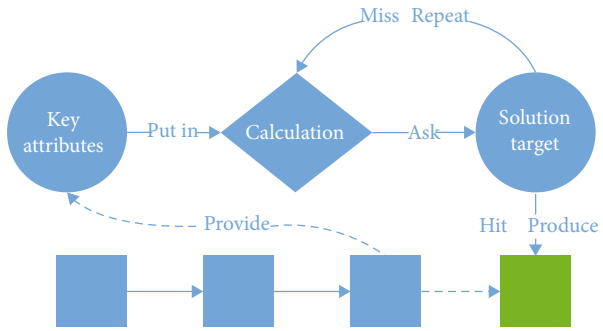

Figure 1: The generic model of a proof-class mechanism.

the key attributes from the existing blocks as the basis for the computation, and when the computation results in a certain target solution, a new block is produced.

Under a proof-class consensus mechanism M_p, a new block is produced if and only if f_{M_p} follows the inequality:

$$\text{Calculation}(\,) < \text{target} \times \text{attribute}(\,). \tag{3}$$

Calculation() represents the calculation for a solution, target represents the solution target, and attribute() represents the key attributes of network nodes for the "proof," such as computing power in PoW and coinage in PoS; note that some of the attributes are recorded in the blockchain ledger.

Main idea. Our goal is to design a consensus mechanism with the following:

(i) More flexible discourse power system

(ii) Incentives for smart contract creators

(iii) Preference support for different work areas

To achieve our goals, we introduce the contract quality and the node activity into PoE. Based on Definition 1, we present the generic model of PoE.

Definition 2. $\text{PoE} := <\mathbb{R}, \times B, I, \times C, f_{\text{PoE}}>$ is a quintuple, which consists of the following:

(i) A real number set $\mathbb{R}$

(ii) A blockchain $\times B = \{B_{[0]}, B_{[1]}, \cdots, B_{[k]}\}$

(iii) A set $I = \{i_1, \cdots, i_n\}$ of network nodes

(iv) A set $\times C = \times C_i \cup \times C_{-i}$ of smart contracts

(v) A mapping $f_{\text{PoE}} : \mathbb{R} \longrightarrow B_{[k+1]}$ of block production

Under PoE, a new block is produced if and only if f_{PoE} follows the inequality:

$$\text{Calculation}() < \text{target} \times \text{contractquality} \times \text{activity}. \tag{4}$$

Table 1: Parameters in PoE model.

Symbol	Description
Ω_i^a	Node i's engagement in work area a
C_i^a	Smart contract applied in work area a, created by node i
Q_C	Quality of smart contract C, $Q_C \in [0, 1]$
$\text{Sta}(C)$	Static evaluation value of smart contract C
$\text{Dyn}(C)$	Dynamic evaluation value of smart contract C
θ	Adjustable parameter to control the value of Q_C
H_i^a	Node i's activity in work area a
k	Block periods in a PoE-based blockchain
T_i^a	Node i's transaction volume in work area a
γ_i^a	Node i's computing power consumption in work area a
P_i	Node i's expected profits in a PoE-based blockchain

3.2. Calculation Model. Based on the generic model presented above, to be more specific, this section presents the calculation model of PoE. The parameters needed in our calculation model are shown in Table 1.

The engagement Ω_i^a reflects the discourse power of node i in work area a, which is calculated by the following:

$$\Omega_i^a = \left(1 + \sum Q_{C_i^a}\right) \times H_i^a. \tag{5}$$

The quality Q_C of smart contract C is as follows:

$$Q_C = \text{Sta}(C) \times \text{Dyn}(C) \times \theta. \tag{6}$$

To show the professionalism of a node, the quantity and quality of smart contracts it creates must be considered. As we know, a new smart contract needs to be locally validated by other nodes (miners). Not only its feasibility, the security and algorithm complexity of a contract can also be evaluated during validation. We propose a model to evaluate the static properties of a smart contract.

The static evaluation $\text{Sta}(C)$ reflects the code quality of smart contract C; it is related to some properties of the code, such as extensibility, reusability, readability, security, and scalability, and the calculation of its value is a complex software engineering problem, which is not discussed more in this paper because of the limitation of space. However, $\text{Sta}(C)$ only reflects the static property of a contract from the perspective of software engineering; it cannot reflect the practicality, so it is necessary to introduce the dynamic evaluation $\text{Dyn}(C)$.

The dynamic evaluation $\text{Dyn}(C)$ reflects the popularity of smart contract C, which is indicated by the following:

$$\text{Dyn}(C) = \frac{\sum_{k=\ddot{k}-\Delta k}^{\ddot{k}} \text{CallCount}_{C[k]}}{\Delta k}. \tag{7}$$

In function (8), $\ddot{k}$ represents the current period of block, and $\text{CallCount}_{C[k]}$ represents the number of times contract C

was called during the period of $B_{[k]}$. Dyn(C) reflects how popular contract C was in the last Δk blocks.

Thanks to the Turing-complete programming language, smart contracts can perform almost any known computation, which makes them be applied to different work areas. We believe that the work area will be an important property of smart contracts in the future.

Work area a is a static property of a smart contract; it is determined by the application direction of the contract. The activity of node i in work area a is calculated by the following:

$$H_i^a = \frac{\sum_{k=\ddot{k}-\Delta k}^{\ddot{k}} \left(\left| T_i^a \right|_{[k]} + \gamma_{i[k]}^a \right)}{\Delta k}. \tag{8}$$

In function (9), node i's transaction volume and consumption of computing power during the last Δk blocks show its recent activity. γ represents the computing power consumption for maintaining the blockchain, which includes contract validation and transaction creation and verification.

As we know, a node will be rewarded with a bonus once it successfully "dig out" a new block. In PoW and PoS, the production of new blocks has certain degrees of randomness, which makes the mechanism fairer. In PoE, randomness is preserved, so the expected profits of a node in a PoE-based blockchain should be calculated to evaluate its benefits.

The expected profit P_i of node i in a PoE-based blockchain is as follows:

$$P_i = \text{reward} \times \frac{\sum_{a=a_1}^{a_m} \Omega_i^a}{\sum_{x=1}^{n} \Omega_x}. \tag{9}$$

In function (10), a node's total engagement Ω_x in a PoE-based blockchain is the sum of its engagement in all m different work areas.

3.3. The Process of PoE. The process of consensus is divided into 2 main stages according to the different types of input and output: one is the leader election among nodes, and the other is the chain updation among transactions.

3.3.1. Leader Election. The leader election stage determines which node will become the bookkeeper.

Definition 3. The leader election stage consists of the following:

(i) A set $I = \{i_1, \cdots, i_n\}$ of network nodes

(ii) A set $\Omega = \{\Omega_{i_1}, \cdots, \Omega_{i_n}\}$ of nodes' total engagement

(iii) A unique bookkeeper b

(iv) A encryption function calculation()

(v) A range target of solutions

```
Input: I, Ω, target
Output: b
1. while True do
2.     for all i ∈ I do
3.         if calculation( ).result < target × Ω_i then
4.             b ⟵ i
5.             break
6.         else
7.                 continue
8.         end if
9.     end for
10. end while
11. return b
```

Algorithm 1: Leader election.

In a PoE-based blockchain, the process of the leader election stage is shown as Algorithm 1. In Algorithm 1, the engagement is introduced as a key attribute to gain the range of the computing target, so a node with high-level engagement is more likely to be a bookkeeper.

3.3.2. Chain Updation. The chain updation stage determines which of the validated smart contracts will be updated onto the blockchain.

Definition 4. The chain updation stage consists of the following:

(i) A set $\times \tilde{C} = \{C_1, \cdots, C_j\}$ of validated smart contracts

(ii) A set $\times \check{C} = \{C_1, \cdots, C_t\}$ of smart contracts to be updated

(iii) A set $A = \{a_1, \cdots, a_m\}$ of different work areas

(iv) A set $I = \{i_1, \cdots, i_n\}$ of network nodes

(v) n sets $\times \Omega_i^a = \{\Omega_i^{a_1}, \cdots, \Omega_i^{a_m}\}$ of n nodes' engagement in different work areas

(vi) Block capacity$V$, the maximum number of contract-addresses stored in a block

(vii) A function getWorkArea() for getting the work area of a smart contract

In a PoE-based blockchain, the validation process of smart contracts is shown as Algorithm 2 and the chain updation stage is shown as Algorithm 3.

In Algorithm 2, $\widehat{S}$ represents a threshold for the static evaluation of smart contracts. In other words, a smart contract is "qualified" when its static evaluation reaches $\widehat{S}$. In Algorithm 3, network nodes vote for the validated smart contracts, and if a node agrees to contract C, he/she will use his/her engagement in the same field to endorse the contract by adding Ω_i^a to Ω_C. The mechanism will determine which contracts are eventually recorded on the blockchain, based on their engagement Ω_C and the block capacity.

Input: $\times C = \{C1, \cdots, Cn\}, \widehat{S}$
Output: $\times \tilde{C} = \{C_1, \cdots, C_m\}$
1. $\times \tilde{C} \longleftarrow$ null
2. for $C \in \times C$ do
3. $S \longleftarrow \mathrm{Sta}(C)$ in function(3)
4. if $S \geq \widehat{S}$ then
5. add C to $\times \tilde{C}$
6. end if
7. end for
8. return $\times \tilde{C}$

Algorithm 2: Validation process of smart contracts.

Input: $\times \tilde{C}, A, I, \times \Omega_i^a, V$
Output: $\times \check{C}$
1. $\times \check{C} \longleftarrow$ null
2. $L \longleftarrow$ null
3. $v \longleftarrow 0$
4. for $C \in \times \tilde{C}$ do
5. $\Omega_C \longleftarrow 0$
6. for $a \in A$ do
7. if $a ==$ getWorkArea(C) then
8. for $i \in I$ do
9. if agree then
10. $\Omega_C \longleftarrow \Omega_C + \Omega_i^a$
11. add Ω_C to L
12. else
13. continue
14. end if
15. end for
16. end if
17. end for
18. end for
19. while $v < V$ do
20. $v \longleftarrow v + 1$
21. for $\Omega_{C_i} \in L$ do
22. if $\Omega_{C_i} == \max(L)$ then
23. add C_i to $\times \check{C}$
24. end if
25. end for
26. end while
27. return $\times \check{C}$

Algorithm 3: Chain updation of smart contracts.

4. Game Analysis

In this section, the incentive compatibility of our mechanism is discussed. In a PoE-based blockchain, in order to gain more profits, a node must keep active and concentrated and try to increase its contract quality.

Theorem 5. *Suppose there are average activity values $\overline{H_{i1}}$, $\overline{H_{i2}}$ of node i, which satisfy $\overline{H_{i1}} > \overline{H_{i2}}$, and all the other variables are constant, there is $P_{i1} > P_{i2}$.*

Proof. P_i is estimated to be a function of the average activity $\overline{H_i}$ of node i:

$$P_i = \text{reward} \times \frac{\bar{H}_i \times \left(m + \sum_{j=1}^{m} Q_{C_i}\right)}{\bar{H}_i \times \left(m + \sum_{j=1}^{m} Q_{C_i}\right) + \bar{H}_{-i} \times \left(m + \sum_{j=1}^{m} Q_{C_{-i}}\right)} A$$

$$\longleftarrow m + \sum_{j=1}^{m} Q_{C_i}, B \longleftarrow \bar{H}_{-i} \times \left(m + \sum_{j=1}^{m} Q_{C_{-i}}\right)$$

$$\Rightarrow P_i = \text{reward} \times \frac{\bar{H}_i \times A}{\bar{H}_i \times A + B}. \tag{10}$$

Calculate derivative of function (11) with respect to $\bar{H}\, i$, and the outcome is as follows:

$$P_{i,\bar{H}_i} = \frac{\text{reward} \times A \times B}{\left(\bar{H}_i \times A + B\right)^2}. \tag{11}$$

Obviously, there is $P_i, \bar{H}_i > 0$, so function (11) is *strictly increasing*. It is proved that if a node is not active enough in a PoE-based blockchain, its expected profits will strictly reduce. □

Theorem 6. *In the same period of block, suppose there are contract quality values Q_{i1}, Q_{i2} of node i, which satisfy $Q_{i1} > Q_{i2}$, and all the other variables are constant, there is $P_{i1} > P_{i2}$.*

Proof. From function (11), it is obvious that P_i can also be estimated to be a function $P_{i,A}$ of A. Calculate derivative of $P_{i,A}$ with respect to $A(A = m + Q)$, and the outcome is as follows:

$$P_{i,A} = \frac{\text{reward} \times \bar{H}_i \times B}{\left(\bar{H}_i \times A + B\right)^2}. \tag{12}$$

Similarly there is $P_{i,A} > 0$, so the function $P_{i,A}$ is *strictly increasing*. It is proved that if a node succeeds in increasing its contract quality in the next period of block, it will be likely to gain more profits. □

According to the theory of mechanism design [21], a direct mechanism (q, t) is incentive-compatible if and only if

(1) q is increasing

(2) For every $\theta \in [\underline{\theta}, \overline{\theta}]$, we have

$$t(\theta) = t(\underline{\theta}) + (\theta q(\theta) - \underline{\theta} q(\underline{\theta})) - \int_{\underline{\theta}}^{\theta} q(x)dx \tag{13}$$

It is given in function (9) that the computing power consumption γ of a node for maintaining the blockchain is positively correlated with its activity. In a PoE-based blockchain,

the computing power consumption must be considered as the "mining cost" of a node.

The utility u_i of node i is calculated by the following:

$$u_i = P_i - \Gamma_i - O(q_i)\Gamma_i = \sigma \times \gamma_i. \tag{14}$$

In function (15), Γ_i represents the average "mining cost" in the recent Δk periods of blocks. $\sigma(\sigma > 0)$ represents the unit cost. $O(q_i)$ represents the cost for increasing the contract quality; it is small enough compared to Γ_i. Function (15) can also be estimated to be a function of γ_i:

$$u(\gamma_i) = P(\gamma_i) - \Gamma(\gamma_i) - O(q_i). \tag{15}$$

Lemma 7. *When σ is within a reasonable range, there exists a threshold $\widehat{\gamma}_i$ which satisfies the following:*

$$u(\widehat{\gamma}_i) = \max\,(u(\gamma_i)). \tag{16}$$

Proof. Calculate derivative of function (16):

$$u'(\gamma_i) = P'(\gamma_i) - \Gamma'(\gamma_i). \tag{17}$$

Same as function (12), $P'(\gamma_i)$ decreases strictly and approaches to $0.\Gamma'(\gamma_i) = \sigma(\sigma > 0)$, when σ satisfies the following:

$$0 < \sigma < \frac{\text{reward} \times A}{B}. \tag{18}$$

Function (16) exhibits increasing and then decreasing. Lemma 7 stands. □

In a PoE-based blockchain, the strategy space of node i is as follows:

$$s_i = \begin{cases} \gamma_i \times q_i\, \gamma_i \in [0, \widehat{\gamma}_i), q_i \in [0, 1), \\ q_i\, \gamma_i = \widehat{\gamma}_i, q_i \in [0, 1), \\ \gamma_i\, \gamma_i \in [0, \widehat{\gamma}_i), q_i = 1, \\ \widehat{s}_i\, \gamma_i = \widehat{\gamma}_i, q_i = 1. \end{cases} \tag{19}$$

In function (20), γ_i, q_i represents increment space of the computing power consumption and the contract quality. Now function (15) can be expressed as follows:

$$u_i = \begin{cases} u(\gamma_i, q_i)\, \gamma_i \in [0, \widehat{\gamma}_i), q_i \in [0, 1), \\ u(q_i)\, \gamma_i = \widehat{\gamma}_i, q_i \in [0, 1), \\ u(\gamma_i)\, \gamma_i \in [0, \widehat{\gamma}_i), q_i = 1, \\ u(\widehat{s}_i)\, \gamma_i = \widehat{\gamma}_i, q_i = 1. \end{cases} \tag{20}$$

Theorem 8. *When s_i satisfies function (20) and σ satisfies function (17), the PoE mechanism is incentive-compatible.*

Proof. According to Theorem 5 and Theorem 6, it is obvious that P_i is increasing. According to function (14), function (15), and function (21), it is easy to infer that u_{s_i} satisfies the following:

$$u_{s_i} \geq u_{s_i'} \Leftrightarrow, \tag{21}$$

$$u_{s_i} \geq s_i \int_{\underline{s_i}}^{s_i'} \frac{\delta P_i}{\delta s_i'} dx - \left(\int_{\underline{s_i}}^{s_i'} \frac{\delta \Gamma_i}{\delta s_i'} dx + \int_{\underline{s_i}}^{s_i'} \frac{\delta O(q_i)}{\delta s_i'} dx \right) \Leftrightarrow, \tag{22}$$

$$u_{s_i} \geq s_i \vartheta\left(s_i'\right) - s_i' \vartheta\left(s_i'\right) + s_i' \vartheta\left(s_i'\right) - \tau\left(s_i'\right) \Leftrightarrow, \tag{23}$$

$$u_{s_i} \geq s_i \vartheta\left(s_i'\right) - s_i' \vartheta\left(s_i'\right) + u\left(s_i'\right) \Leftrightarrow, \tag{24}$$

$$u_{s_i} - u\left(s_i'\right) \geq \left(s_i - s_i'\right) \vartheta\left(s_i'\right) \Leftrightarrow, \tag{25}$$

$$\int_{s_i'}^{s_i} \vartheta(x) dx \geq \int_{s_i'}^{s_i} \vartheta\left(s_i'\right) dx \Leftrightarrow. \tag{26}$$

Theorem 8 stands. □

It can be seen from the establishment of Theorem 5, Theorem 6, and Theorem 8 that as a consensus mechanism, PoE is incentive-compatible. It means that under PoE, rational nodes are more likely to choose to improve the quality of their contracts and remain active to improve their profits. This explains the feasibility of PoE.

Table 2: Attributes of nodes at $B_{[0]}$ in Exp1.

Attribute	a	b	c	Remarks
Computing power	100	10	1	Constant
Coin age	10	100	1	Variable
Activity	10	1	10	Constant
Contract quality (total)	1	10	q	Variable

5. Experiment and Discussion

In this section, a method of evaluating the flexibility of a consensus mechanism is proposed. According to the generic models of PoW and PoS mentioned in Section 3.1 and Algorithm 1, we first test and compare the performances of 3 nodes under 3 different mechanisms in Exp1. Then, we test the impact of the contract quality on nodes' performances under PoE in Exp2. At last, we discuss the main application directions of PoE.

5.1. Flexibility Comparison. The 3 nodes have different characteristics and strategy preferences at the very beginning, as is shown in Table 2: node *a* performs as a full-time miner, so he has the highest computing power and always keeps active; node *b* performs as a lazy rich guy, so he has a high level of coinage and creates some smart contracts for transactions; node *c* performs as a new and hard-working developer, so he keeps active and creates a number of smart contracts for different applications.

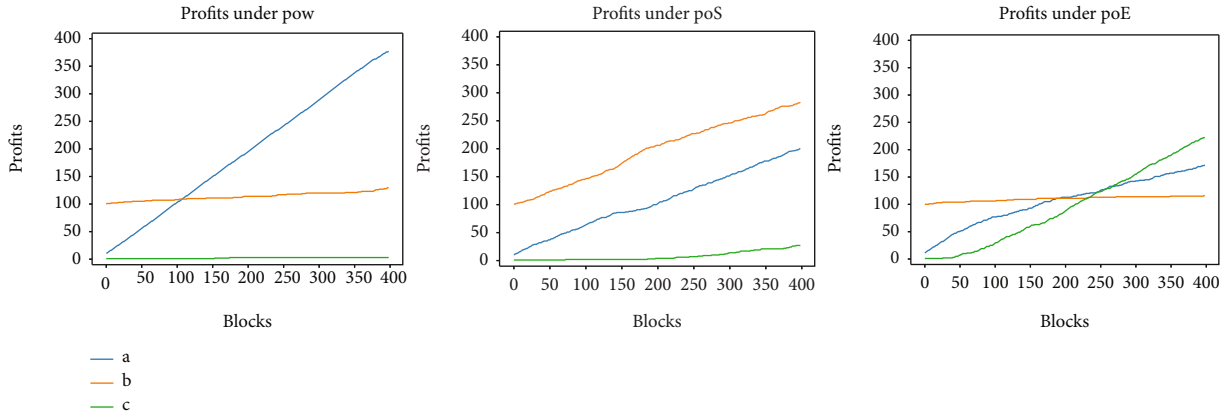

Figure 2: Profits under different consensus mechanisms in Exp1.

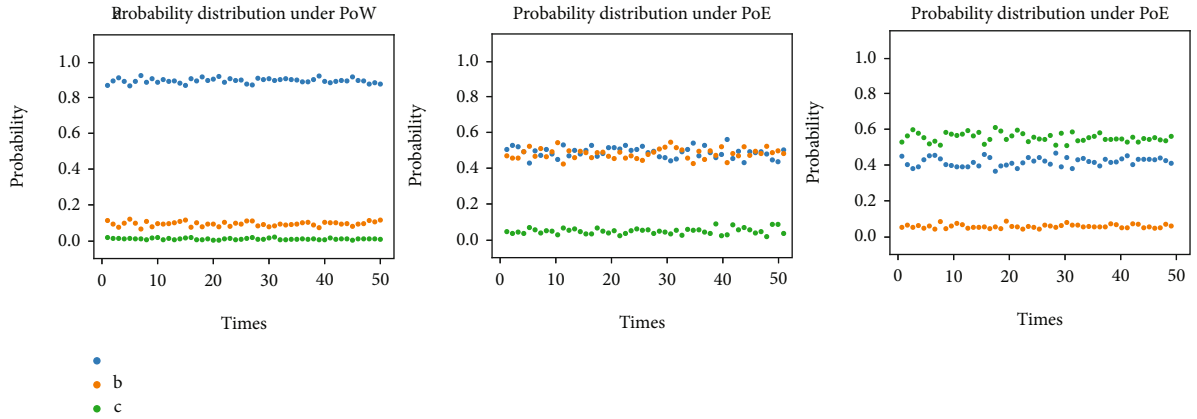

Figure 3: Probability distribution under different consensus mechanisms in Exp1.

5.1.1. Profits. The profit changes of a, b, c under PoW, PoS, and PoE during 400 periods of blocks are shown in Figure 2. It is directly shown in the line charts that compared to those under PoW or PoS, node a accumulates its profits more quickly under PoE and surpasses b and c in a short time.

5.1.2. Probability Distributions. We tested a, b, c's probability distribution of becoming the bookkeeper during 400 periods of blocks for 50 times under PoW, PoS, and PoE. The probability distributions are shown in Figure 3. Figures 2 and 3 directly indicate that node a, b, and c have their own advantages under the 3 different mechanisms.

The flexibility Φ_M of a consensus mechanism M is a property which reflects the novice-friendliness and the incentive of M.

Proposition 9. *The flexibility Φ_M of a consensus mechanism M is proportional to the average probability $\overline{\phi_{new}}$ that a new node successfully becomes the bookkeeper under M during the first k periods of blocks.*

We do not have to prove the simple proposition presented above. Note that node c is a new node in the blockchain, so we calculate the average probability $\overline{\phi_c}$ of node c under different mechanisms:

Table 3: Attributes of nodes at $B_{[0]}$ in Exp2.

Attribute	a	b	c	Remarks
Computing power	1	1	1	Constant
Coin age	1	1	1	Variable
Activity	10	10	10	Constant
Contract quality	c	c	c	Variable
Contract quality (average)	1	1	q	Variable

$$\begin{aligned}\bar{\phi}_c(\text{PoW}) &= 0.0097,\\ \bar{\phi}_c(\text{PoS}) &= 0.0452,\\ \bar{\phi}_c(\text{PoE}) &= 0.5462.\end{aligned} \tag{27}$$

Obviously, node c has excellent performance under PoE. Note that the cost of creating a smart contract (or improving the contract quality) is much less than increasing the computing power or increasing the coinage. Hence, it is not

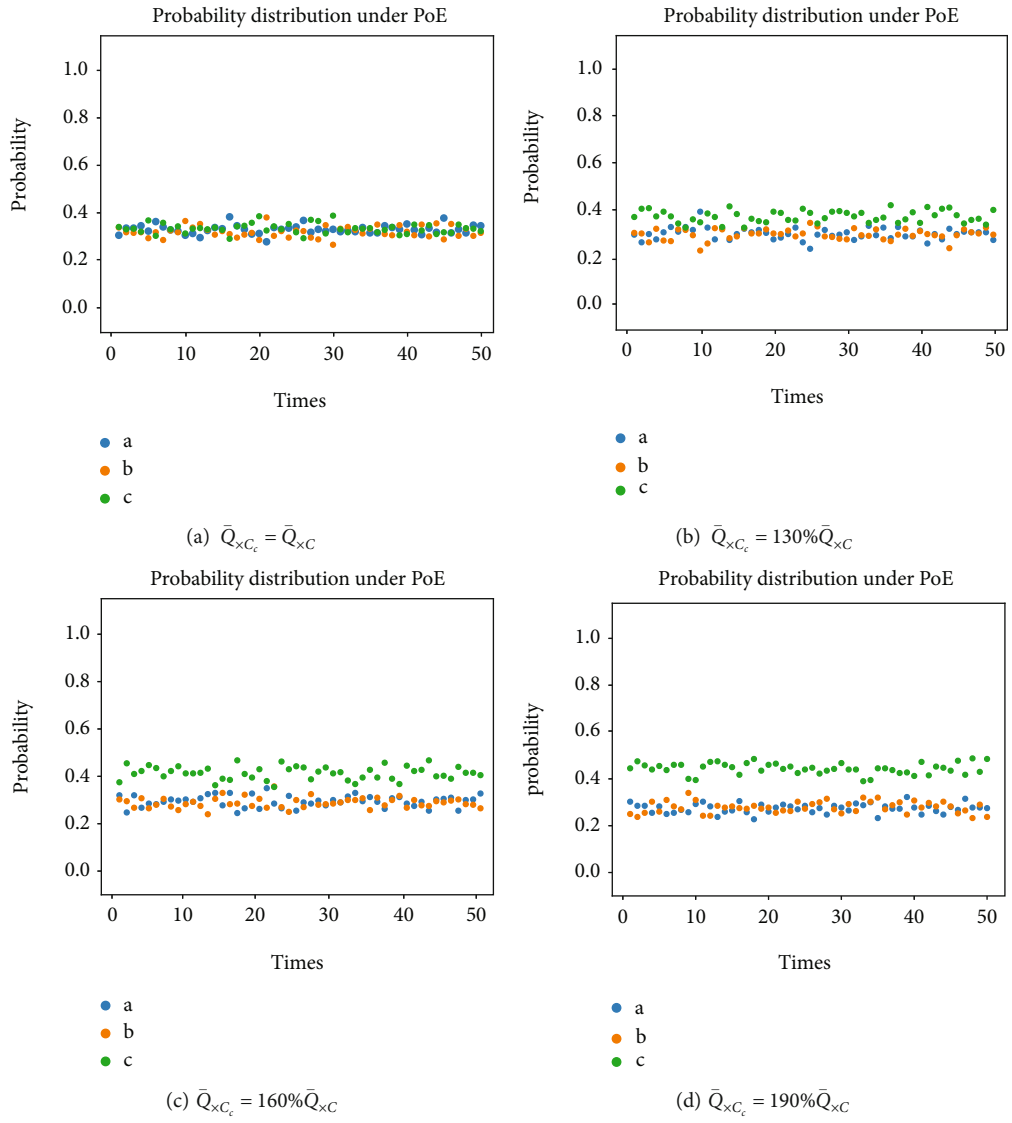

Figure 4: Probability distribution under PoE with contract quality increasing in Exp2.

difficult to summarize that PoE is more flexible than PoW and PoS.

5.2. Impact of Contract Quality. In Exp2, we assume that a, b, and c are all hard-working developers, and their characteristics are shown in Table 3.

In Exp2, the basic rules are as follows:

(i) At $B_{[0]}$, $P_a = 1, P_b = 1, P_c = 1$

(ii) $P_i = P_i + 1$ if $b \longleftarrow i$

(iii) At $B_{[0]}$, $c = 1$, and there is $c = c + 1$ when $B_{[k]} = B_{[k+1]}$

Different from Exp1, we considered the average quality of smart contracts in Exp2. We improved the average contract quality of node c by 0%, 30%, 60%, and 90%, respectively, and tested a, b, c's probability distribution of becoming the bookkeeper under PoE, and the results are shown in Figure 4.

We calculate the average probability $\overline{\phi_c}$ of node c when $\overline{Q_{\times C_c}}$ is 0%, 30%, 60%, and 90% higher than the average value $\overline{Q_{\times C}}$:

$$\begin{aligned}
\bar{\Phi}_c\left(\bar{Q}_{\times C_c} = 100\%\bar{Q}_{\times C}\right) &= 0.3384,\\
\bar{\Phi}_c\left(\bar{Q}_{\times C_c} = 130\%\bar{Q}_{\times C}\right) &= 0.3820,\\
\bar{\Phi}_c\left(\bar{Q}_{\times C_c} = 160\%\bar{Q}_{\times C}\right) &= 0.4144,\\
\bar{\Phi}_c\left(\bar{Q}_{\times C_c} = 190\%\bar{Q}_{\times C}\right) &= 0.4444.
\end{aligned} \tag{28}$$

Our experimental results directly indicate that when node c's average contract quality improves by 30%, 60%, and 90%, and its average probability to become a bookkeeper is improved by about 12.9%, 22.5%, and 31.3%.

5.3. Application Discussion. According to the characteristics of PoE, we have made some assumptions about its application directions.

5.3.1. Knowledge Payment Platforms. Higher quality, more practical knowledge deserves a higher price.

Suppose in a Q&A community, users ask and answer questions in different areas, such as medicine, finance, and law. The community will reward users with excellent answers, and the questioner also gives the answerer some appreciation for solving his/her problems. A PoE-based blockchain is suitable for building a Q&A community, for example, a senior lawyer can quickly accumulate his fame and income by answering a number of questions related to law in such a community.

5.3.2. Copyright Protection. The data should be traceable and hard to be tampered with.

Thanks to the characteristics of a PoE-based blockchain, the publication time and author information of work can be traced back and it is very difficult to be tampered with. In addition, excellent works can be endorsed by experts in the industry, so that the author's hard work can be recognized by the industry.

5.3.3. Distributed Social Networks. High-quality content creators should be recommended to be followed by users. A more active user should have a louder voice.

Suppose on a video website, users can get coins by daily logging in and watching videos, which will be given to their favorite videos as "like." Videos with more coins will be recommended to each user's home page, and the video creators will be recommended to be followed. A PoE-based blockchain can meet the needs of such distributed social networks and greatly reduce the pressure of centralized storage. In addition, it can protect users' creations from usurpation.

6. Conclusion

This study mainly investigates a novel consensus mechanism called proof of engagement. Specifically, we first present a definition of consensus mechanism and propose a generic model of proof-class consensus mechanisms, and on this basis, establish the generic and calculation models of PoE. Secondly, we present the algorithms of the consensus process. Thirdly, from the perspective of mechanism design, we analyze the incentive compatibility of our mechanism. If a node keeps active and improves its contract quality, it will always gain more profits than performing idleness. At last, we test the flexibility of our mechanism through a series of simulation experiments and discuss the main application directions of PoE. The experimental results show that a new node is more easier to increase profits if he/she maintains a high level of engagement. Our mechanism has better incentives for contract creators than PoW and PoS. This is a strong indication that to a large extent, PoE weakens the Matthew Effect. Generally speaking, PoE is a flexible, fair, and novice-friendly mechanism. This work provides a new idea for the industrial combination of blockchain; it will bring benefits to the development of smart contracts and distributed autonomous systems such as DApp, DAO, and DAS.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61872313, in part by the Key Research Projects in Education Informatization in Jiangsu Province under Grant 20180012, in part by the Yangzhou Science and Technology under Grant YZ2020174 and Grant YZ2019133, and in part by the Open Project in the State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University under Grant 1907.

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, 2008, article 21260.

[2] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2016.

[3] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[4] J. Golosova and A. Romanovs, "The advantages and disadvantages of the blockchain technology," in *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, Vilnius, Lithuania, 2018.

[5] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *International Conference on Financial Cryptography and Data Security*, Berlin, Heidelberg, 2016.

[6] A. Vries, "Renewable energy will not solve bitcoin's sustainability problem," *Joule*, vol. 3, no. 4, pp. 893–898, 2019.

[7] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: a cooperative game theoretic analysis," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, Istanbul, Turkey, 2015.

[8] L. Cong, Z. He, and J. Li, "Decentralized mining in centralized pools," *The Review of Financial Studies*, vol. 34, no. 3, pp. 1191–1235, 2021.

[9] S. King and N. Scott, *Ppcoin: peer-to-peer crypto-currency with proof-of-stake*, Self-published paper, 2012.

[10] R. Merton, "The Matthew effect in science: the reward and communication systems of science are considered," *Science*, vol. 159, no. 3810, pp. 56–63, 1968.

[11] Y. Liu, K. Wang, Y. Lin, and W. Xu, "LightChain: a lightweight blockchain system for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3571–3581, 2019.

[12] Y. Yuan, X. Ni, S. Zeng, and F. Wang, "Blockchain consensus algorithms: the state of the art and future trends," *Acta Automatica Sinica*, vol. 44, no. 11, pp. 2011–2022, 2018.

[13] F. Saleh, "Blockchain without waste: proof-of-stake," *The Review of Financial Studies*, vol. 34, no. 3, pp. 1156–1190, 2021.

[14] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On security analysis of proof-of-elapsed-time (poet)," in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, Cham, Boston, MA, USA, 2017.

[15] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: an efficient blockchain consensus protocol," in *Proceedings of the 1st Workshop on System Software for Trusted Execution*, Trento, Italy, 2016.

[16] A. Baldominos and S. Yago, "Coin. AI: a proof-of-useful-work scheme for blockchain-based distributed deep learning," *Entropy*, vol. 21, no. 8, p. 723, 2019.

[17] A. Shoker, "Sustainable blockchain through proof of exercise," in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, USA, 2017.

[18] W. Chiu and W. Meng, "EdgeTC - a PBFT blockchain-based ETC scheme for smart cities," in *Peer-to-Peer Networking and Applications*, pp. 1–13, Springer, 2021.

[19] K. Karantias, A. Kiayias, and D. Zindros, "Proof-of-burn," in *International Conference on Financial Cryptography and Data Security*, Cham, Kota, Kinabalu, Malaysia, 2020.

[20] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: extending bitcoin's proof of work via proof of stake," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, 2014.

[21] T. Börgers and J. Li, "Strategically simple mechanisms," *Econometrica*, vol. 87, no. 6, pp. 2003–2035, 2019.