Design and Analysis of An Effective Securing Consensus Scheme for Decentralized Blockchain System

Jing Wang[1,(✉)], Lingfu Wang[1], Wei-Chang Yeh[2], and Jinhai Wang[3]

1. School of Computer Science and Information Security, Guilin University of Electronic Technology, 541004, China
2. Department of Industrial Engineering and Engineering Management, College of Engineering, National Tsing Hua University, 30071, Taiwan
3. College of Electronic and Information Engineering, Foshan University, 582000, China

{wjing@guer.edu.cn}

Abstract. Blockchain, as a decentralized network system, has been attracting increasing attention in recent years. In a blockchain system, there must be a consensus mechanism to ensure the distributed consensus among all parties. Such consensus mechanism may also be applied to guarantee fairness, correctness, and sustainability of such decentralized systems. In this paper, we propose a novel consensus mechanism, named Proof-of-Credibility (PoC), which is an improved version of Proof-of-Work (PoW) . Compared with existing consensus mechanisms, PoC provides strong resistance to resource centralization and other malicious attacks. First, we present the Serial Mining Puzzle (SMP) to resist collusive mining. SMP guarantees that participants only get negligible advantage by parallel solving. Second, PoC considers the influence of participant credibility, which is reflected by the mining behaviour of a participant. Thus, credible participants get higher probability of winning the mining competition than incredible ones. Finally, the performance of PoC is analyzed in terms of common prefix, chain quality and power cost. Our analysis indicates that PoC is security and incentive compatible with suitable security parameter settings.

Keywords: Decentralized System; Blockchain; Consensus Mechanism; Proof-of-Credibility;

1 Introduction

Distributed and decentralized network systems are gaining popularity nowadays. More and more businesses and individuals have started to access application services from the Internet, which can provide a distributed and decentralized platform for deploying and hosting application of all kinds. Compared with the traditional application platforms, it offers a number of key advantages, including scalability, flexibility, and low cost. However, the security and manageability of

decentralized platforms arise as a central challenge. *Bitcoin*, as one of the most famous cryptocurrencies systems, provides an efficient way to maintain a decentralized network system. The core of *Bitcoin*-like systems, called *blockchain*, can be viewed as a decentralized public ledger [1]. The decentralized nature implies that the system can be maintained entirely by participants instead of an appointed Trusted Third Party (TTP). Thus, a consensus mechanism, such as Proof-of-Work (PoW), Proof-of-Stake (PoS) and so on, is needed for blockchain to prevent the double-spending attack [2]. Furthermore, the consensus mechanism provides a promising direction to guarantee the security and robustness of more generalized decentralized systems.

The consensus mechanism follows a fundamental assumption, named *honest majority*, where adversaries can break the mechanism with negligible probability since it is difficult to control the majority of the mining resource [3]. However, the presence of resource coalitions may violate the honest majority assumption and incurs a large lurking threat against blockchain security [4–6]. For instance, the largest mining pool, Ghash.IO, has controlled more than 50% mining capacity of the *Bitcoin* network [7]. In fact, the presence of resource colition is inevitable in the *Bitcoin*-like systems [8]. On the one hand, it is a large incentive of solo participants colluding to hedge mining risks and obtain more stable reward [9]. On the other hand, there is a built-in design limitation of the *mining puzzle*, which admits an effective coalition enforcement mechanism [8]. Thus, a significant challenge of blockchain security is to prevent malicious participants from centralizing resource to successfully implement *51% attack* [10] and *selfish strategy attack* [8].

Recently, in order to prevent resource centralization, two kinds of solutions are proposed, which involve increasing either the risk or the cost of resource coalition. For the former, Miller et al. proposed a notion named *nonoutsourceable puzzle* [8], which allows participants of a coalition to steal reward of the coalition without producing any evidence to implicate itself. Thus, it effectively creates a disincentive for participants to join the coalition, which may incur a high risk of reward lost. For increasing the cost of coalition, Duong et al. [11] and Bentov et al. [12] proposed combined mechanisms of PoW and PoS, where malicious attacks can hold advantage in mining competition by controlling the majority of both computation power and coin stake. Thus, the cost of attack is greatly increased and the security threat can be mitigated. However, while such solutions mitigate the collusion incentives, blockchain still requires a consensus mechanism that essentially resists resource coalition, which strictly maintains the fairness of the blockchain.

In this paper, we propose an improved PoW mechanism, named Proof-of-Credibility (PoC). PoC consists of two core components: the Serial Mining Puzzle (SMP) and the Mining Credibility System (MCS). First, the SMP is a novel mining puzzle that resists to resource coalition. Different from nonoutsourceable puzzles, the serial puzzle prevents not only outsource mining but also parallel mining. Thus, computation resource coalition (e.g. mining pool) presents little advantage in the mining competition. Second, for avoiding the influence of mali-

cious participants, the MCS is introduced into PoC to provide personalized mining difficulty for participants. The MCS evaluates credibility of each participant and quantifies the credibility-based mining difficulty. Ideally, the mining difficulty should monotonically decrease with the participant credibility. As a result, PoC tends to accept *next block* created by participants with a high credibility. Thus, the proposed PoC provides an efficient way to deploy secure blockchain. The contributions of this paper can be summarized as follows:
1) We propose the *Serial Mining Puzzle* (SMP) to resist parallel mining, avoid resource centralization.
2) We provide quantified *Participant Credibility*, which is evaluated by the mining events recorded in blockchain.
3) We develop the *Personalized Mining Difficulty* to promote competitive advantage of credible participants during mining.

2 Related Work

Blockchain based Decentralized Systems. Decentralized systems provide an effective means to develop large-scale applications with loosely coupled operation and management of individual systems [13]. The decentralized nature of such system also brings novel requirements and functions [14]. The most prominent example of decentralized systems is *Bitcoin*, which was built by Nakamoto in 2009 [15]. Soon after that, extending the distributed mechanism of *Bitcoin* beyond cryptocurrency has been gaining momentum [16]. Such blockchain based decentralized systems are being rapidly developed that will play a major role in the software engineering community and beyond [17, 18].

Computational puzzles. The consensus protocol of blockchain provides an efficient mean to avoid double-spending (i.e. a bitcoin is spent more than once) for *Bitcoin*-like systems [19]. Nakamoto first proposed the Nakamoto consensus using the Proof-of-Work (PoW) computational puzzle in *Bitcoin* [15]. Following the Nakamoto consensus, the blockchain may generate several temporary forks. But one of these forks will eventually surpass others and bring the eventual consensus [3, 15, 20]. Furthermore, several modified computational puzzles are proposed to solve some specific problems with the Nakamoto consensus. In order to increase the mining revenue, participants use customized hardware to improve mining efficiency. Recently, an Application Specific Integrated Circuit (ASIC) has achieved orders of magnitude better performance than common chips in terms of mining [21]. Thus, an ASIC-resistant mining puzzle is proposed to keep the competitiveness of commodity hardware in mining competition [22]. Meanwhile, a useful puzzle is provided to avoid the energy and resource waste during mining. Kroll indicated that any useful puzzle must produce a pure public good [9]. For protecting the decentralization of Bitcoin-like systems, Miller first proposed the notion of non-outsource-able puzzle to prevent participant coalition [8]. The non-outsource-able puzzle allows a participant of mining pool to steal the mining reward without producing any evidence that can potentially implicate itself.

Virtual mining proposal. Different from computational puzzles, Proof-of-Stake (PoS) is provided as a virtual mining proposal of blockchain [23]. Instead of costing external computing resources, it costs virtual resources to extend blockchain. Thus, PoS effectively avoids the waste of real resources. Recently, there are several versions of PoS proposed to acquire better performance, such as Proof-of-coin-age [23], pure Proof-of-Stake [24], Proof-of-activity [12] and so on. However, the stability and security of virtual mining systems is still an open problem, which needs to be formally addressed. King et al. believe that, in a virtual mining system, it may be more difficult for an attacker to acquire a sufficiently large amount of digital currency than to acquire a sufficiently powerful computing equipment [23]. However, Poelstra claims that external resource consuming is necessary for blockchain security [25]. The core argument is that virtual mining is susceptible to cost-less simulation attacks. These attacks cost nothing to construct an alternate view of history, in which the allocation of currency evolves differently [10].

Security and Performance Analyzation of Blockchain Consensus Protocol. A core concern of blockchain is the security and stability of its consensus protocol. The security has initially been proven (informally) in the *honest majority* model [3, 15, 20]. However, the model is unsatisfying since it does not provide sufficient guarantee of the *honest majority* assumption. Several researches deem that the mining reward of *Bitcoin*-like systems provides the incentive for participants to participate and maintain the system [15, 26]. However, an economic analysis shows that a bitcoin-like system is not fixed, rule-driven, and incentive-compatible as some advocates claim [27]. In fact, a participant (or coalition) may deviate from the incentive compatible consensus protocol by using a selfish mining strategy when it controls more than a third of total computation power [28]. Furthermore, an optimal selfish mining strategy is provided as the best response to the honest behaviour [29]. It offers a lower bound of the resource amount (less than 25% of the total resources) needed for a profitable selfish mining strategy. This result highlights the importance of preventing the formation of participant coalitions [16]. To evaluate the blockchain performance, several researches attempt to formulate fundamental metrics of the blockchain [30]. Garay et al. provided two quantifiable properties named *common prefix* and *chain quality*, which describe the *liveness* and *persistence* of blockchain, respectively. In this paper, we also present the superiority of our PoC mechanism by these fundamental metrics.

3 The Credibility Based Consensus Mechanism

3.1 System Model and Definitions

Nakamoto proposed the detailed model of PoW based blockchain [15]. As shown in Fig.1, blockchain consists of a set of *sequential* blocks, where each block is associated with a pre-block except for the *genesis block* [15]. Furthermore, each block includes two parts: block header and transaction records. Block header contains three parameters: Pre that denotes the hash of pre-block, $Nonce$ that

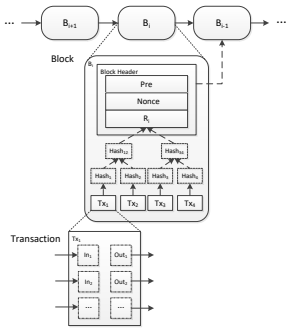

Fig. 1. PoW Blockchain

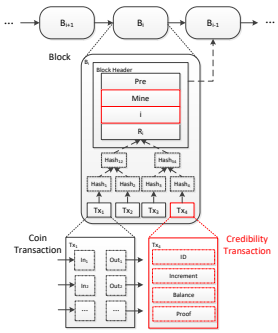

Fig. 2. PoC Blockchain

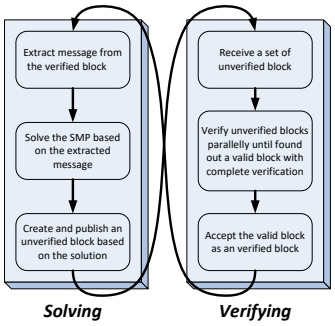

Fig. 3. Serial Mining Puzzle

denotes a PoW solution of B_i, and R_i that denotes the root of a *Merkle tree* [15] formed by transactions. Finally, each transaction record includes a set of inputs $In_1, In_2, \ldots$ (i.e. the unspent coins of the *Bitcoin* system) and a set of outputs $Out_1, Out_2, \ldots$ (i.e. the new unspent coins of the *Bitcoin* system). In such PoW-based blockchain, the block creator named *miner* persistently searches the PoW solution to generate the next block. The participant will gain monetary award when its block is confirmed by blockchain.

3.2 Proof of Credibility

PoC improves PoW by providing the capability to resist to resource centralization and collusion. There are two core functional modules of PoC: SMP and MCS. First, SMP encourages participants to mine independently because collusive mining is no longer useful. Then, MCS is proposed to quantify personalized mining difficulty based on participant credibility. It increases the success probability for credible participants during mining and provide sufficient protection against decentralization to ensure security of PoC.

Fig.2 shows the overview of PoC blockchain. There are two key differences between the PoC and PoW blockchains. First, the block header parameter $Nonce$ is replaced by mining information $Mine$ and block height i. Different from $Nonce$, $Mine$ includes two parts: *serial mining puzzle solution* and the corresponding *verification* provided by multiple participants. $Mine$ carries detailed information of mining events and reflects the credibility of block creator. Second, *credibility transaction* is introduced to quantify participant credibility. However, credibility can not be transacted. Instead, it can only be updated by specific *mining-event* with an increment. Specifically, the credibility transaction includes four parts: *ID*, *Increment*, *Balance*, and *Proof*: *ID* denotes a credibility account of a participant, *Increment* denotes the credibility increment caused by the mining-event, *Balance* denotes the updated credibility balance, and *Proof* denotes the corresponding proof of occurrence of a mining-event.

3.3 Serial Mining Puzzle

SMP is a core module of PoC, which deters resource centralization, because it provides strong guarantee against parallel mining. In the PoC blockchain, participants persistently search the SMP solution instead of the PoW solution. Different from PoW puzzle, the proposed SMP require to solve in serial and verify in parallel. Intuitively, as shown in Fig. 3, the mining process of SMP is a cycle of two phases: solving and verifying. In the solving phase, participants serially search for the solution of a SMP with the last verified block message and publishes an unverified block with the solution. In the verifying phase, participants verify the unverified block in parallel to obtain a complete verification of a valid block.

Solving Phase. Firstly, the pre-block message is extracted as

$$M = S_{sk_{i-1}}(pre_{i-1}||hash(R_{i-1}||V_{i-1})), \tag{1}$$

where S denotes a digital signature algorithm, sk_{i-1} denotes a signing key of the pre-block creator, pre_{i-1} denotes the hash value of the pre-block B_{i-1}, R_{i-1} denotes the root of the Merkel tree of accepted transactions in the pre-block, and V_{i-1} denotes the complete verification set of B_{i-1} (The explanation of complete verification is given in **Verifying Phase**). Then, the initial mining message of current block is given as $msg = S_{sk_i}(M)||I$, where I denotes the height of current block. Finally, a mining series $\{a_n\}$ is defined as follows:

$$a_j = \begin{cases} null & j = 0 \\ a_{j-1}||b_{j-1} & j > 0 \end{cases}, \qquad b_j = Bit(hash(msg||a_j)),$$

where Bit denotes a random function which inputs a equal-length string (i.e. $hash(msg||a_j)$) and outputs a bit $b_j \in \{0, 1\}$. Essentially, solving the SMP is to find the first valid a_l where $hash(msg||a_l)$ less than the specified difficulty D (see in **Algorithm 1**). It is clear that a_j can not be determined unless a_{j-1} has been determined. Thus, **Algorithm 1** must be a serial algorithm instead of a parallel algorithm. **Verifying Phase.** A weakness of SMP is the heavy computation

Algorithm 1 Serial Solving $\mathcal{S}(msg, D)$

Input: Block Message msg; Difficulty D
Output: Puzzle Solution: s

```
s ← null
tmp ← hash(msg)
while tmp ≥ D do
    b ← Bit(tmp)
    s ← s||b
    tmp ← hash(msg||s)
end while
return s
```

cost by verifying which is close to the solving cost. However, verifying can be

performed in parallel. Specifically, a block B_i is verified as a valid block if and only if it satisfies the following criteria: 1) $hash(B_i.s) < B_i.D$ where $B_i.s$ denotes the SMP solution of block B_i and $B_i.D$ denotes the mining difficulty; 2) each bit of $B_i.s$ is verified as a valid bit. Note that, the j^{th} bit b_j of $B.s$ is *valid* iff $b_j = Bit(hash(msg||a_j))$ where msg is the initial mining message and $a_j = b_0||\cdots||b_{j-1}$ is a part of $B_i.s$. The parallel multi-party verifying process is given in **Algorithm 2**. The participant continuously chooses an unverified block to verify until a complete verification set of a block is achieved.

Algorithm 2 Parallel Verifying $\mathcal{V}(\mathcal{S}_{ID})$

Input: A Unverified Block Set $\mathcal{S}_{ID}$
Output: Accepted Block $B_i \in \mathcal{S}_{ID}$, Verification set V_i

$v \leftarrow 0$
while $\mathcal{S}_{ID} \neq \emptyset \wedge v = 0$ **do**
  Choose a block $B_i \in \mathcal{S}_{ID}$
  if $hash(msg||B_i.s) < B_i.D$ **then** //$B_i.s$ denote the mining message of B_i, $B_i.D$ denotes the mining difficulty of B_i
    Extract verifying bit from B_i: $bstr_{i,ID} = Extract(B_i.s, U_{ID})$ // U_{ID} denotes the identity of verifier
    Verify the bit of $B_i.s$ indicated in $bstr_{i,ID}$
    if each verified bit is valid **then**
      Generate a successful verification and broadcast it:
      $V^+_{i,ID} = (S_{ID}(bstr_{i,ID}), B.s)$, $ORbstr_i \leftarrow bstr_{i,ID}$, $V_i \leftarrow \{V^+_{i,ID}\}$
      while $ORbstr \neq 111\ldots1$ **do**
        Receive verification of B_i broadcasted by others
        if receive a unsuccessful verification V^-_{i,ID_K} of B_i **then**
          $\mathcal{S}_{ID} \leftarrow \mathcal{S}_{ID} - \{B_i\}$
          break
        else
          if receive a successful verification V^+_{i,ID_K} of B_i **then**
            $ORbstr_i \leftarrow ORbstr_i|bstr_{i,ID_K}$, $V_i \leftarrow V_i \cup \{V^+_{i,ID_K}\}$
            if $ORbstr_i = 111\ldots1$ **then** //The complete verification set is achieved
              $v \leftarrow 1$
            end if
          end if
        end if
      end while
    end if
  else
    Generate a unsuccessful verification and broadcast: $V^-_{i,ID} = (S_{ID}(estr_{i,ID}), B_i.s)$ //$estr_{i,ID}$ indicates invalid bit
    $\mathcal{S}_{ID} \leftarrow \mathcal{S}_{ID} - \{B\}$
  end if
end while
return B_i, V_i

3.4 The Mining Credibility System

In the blockchain, each block includes not only direct *transaction records* but also indirect *credibility records*. It implies that each block indirectly records the mining events, which actually reflect the credibility of participants. Thus, MCS is developed to evaluate participant credibility and quantifies credibility-based mining difficulty.

Credibility Account. Different from a coin account, acquiring a credibility account is more strict. The credibility account can be viewed as a coin account bounded with a unique global IP address. Specifically, a credibility account can validly gain block award when an IP binding certificate of the account has been confirmed by the blockchain. In this way, a credibility account is uniquely identified by a global IP address, which can also mitigate the witch attack in the PoC based blockchain. Note that, in the MCS of PoC blockchain, the credibility accounts sacrifice the anonymity for their credibility while the coin accounts keeping their anonymity without credibility.

Credibility Quantification. First of all, an ideal MCS requires that the participant credibility accurately reflects the mining behaviour of the participant.

In MCS, each cridibility participant $C_{\mathfrak{p}}$ of each participant $\mathfrak{p}$ is initialized to be 0 and the following mining-events are specified to affect $C_{\mathfrak{p}}$:
(1) $\mathcal{E}_i$, inserting a block into the chain. The credibility increment caused by $\mathcal{E}_i$ is calculated as $\Delta_i = \alpha(1 - e^{-\lambda_i A(\mathcal{E}_i)})$, where α is a positive constant that denotes the upper bound of the increment, λ_i is a positive constant that describes the rising tendency of increment, and $A(\mathcal{E}_i)$ denotes the transaction amount confirmed while $\mathcal{E}_i$ is occurring.
(2) $\mathcal{E}_s$, contributing a verification of inserted blocks. If participant $\mathfrak{p}$ successful submits a verification to block-chain, the increment of $C_{\mathfrak{p}}$ can be calculated as $\Delta_s = \beta(1 - e^{-\lambda_s L_b(\mathcal{E}_s)})$, where $\beta > 0$ denotes the upper bound of increment Δ_s, $\lambda_s > 0$ controls the rising tendency of Δ_s and $L_b(\mathcal{E}_s)$ denotes the number of verified bits indicated in the verification.
(3) $\mathcal{E}_d$, detecting a forged block includes invalid bit. Let Δ_d be the increment of $C_{\mathfrak{p}}$ while $\mathfrak{p}$ detects a forged block that includes an invalid bit. Thus, $\Delta_d = \gamma(1 - e^{-\lambda_d L_v(\mathcal{E}_d)})$, where $\gamma > 0$ denotes the upper bound of Δ_d, λ_d denotes the rising tendency of Δ_d, and $L_v(\mathcal{E}_d)$ denotes the length of mining information of the detected block.
(4) $\mathcal{E}_c$, creating a forged block. The increment Δ_c of $C_{\mathfrak{p}}$ is produced while a block published by $\mathfrak{p}$ is verified as a forged block. Furthermore, $\Delta_c = min\{-\eta e^{\lambda_c C_p}, T\}$, where $T < 0$ denotes the upper bound of Δ_c, η and λ_c denotes two positive parameters influence increment Δ_c.
(5) $\mathcal{E}_a$, accepting a forged block. The event $\mathcal{E}_a$ represents that participant p has published a block with a forged pre-block. It implies that p accepts an incomplete or forged verification. Thus, the increment $\Delta_a = -\rho e^{-\lambda_a L_v(\mathcal{E}_a)}$, where $-\rho < 0$ denotes the lower bound of Δ_a, $-\lambda_a$ denotes the constant that controls the rising tendency of Δ_a, and $L_v(\mathcal{E}_a)$ denotes the length of the mining information of the pre-block.
(6) $\mathcal{E}_p$, publishing two blocks or verification with a close block height in different forks. It will produce a serious forking issue when a participant performs mining with different forks in parallel. Thus, such a dishonest behaviour will result in the following

credibility increment $\Delta_p = -\tau e^{\lambda_p L_{\mathfrak{p}}(\mathcal{E}_p)}$, where τ denotes a positive constant coefficient, λ_a denotes the constant that controls the rising tendency, and $L_{\mathfrak{p}}(\mathcal{E}_p)$ denotes the total length of such blocks or verification published by $\mathfrak{p}$ with close block height.

Additionally, the influence of mining-event will decay with time. Thus, it is reasonable for each credibility increment to multiply a *exponential time-decay factor* $e^{-\lambda_t T}$, where $-\lambda_t < 0$ denotes an assigned constant and T denotes the height difference between the block when the mining-event occurs and the current block.

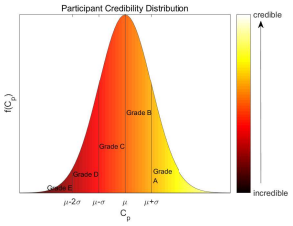

Fig. 4. $C_{\mathfrak{p}}$

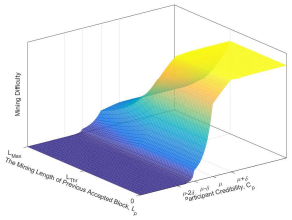

Fig. 5. $D_{\mathfrak{p}}$

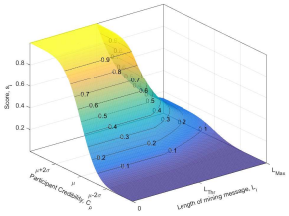

Fig. 6. s_i

Credibility Grading. Let credibility increments of a participant be a sequence of independent random variables $\Delta_1, \Delta_2, \ldots \Delta_n$ and each Δ_k. Assume that, following the Lyapunov central limit theorem [31], the distribution of $C_{\mathfrak{p}} = \sum_{k=1}^{n} \Delta_k$ tends to be normally distributed $N(\mu, \sigma^2)$, where $\mu = \sum_{k=1}^{n} \mu_k$ and $\sigma^2 = \sum_{k=1}^{n} \sigma_k^2$. Then, $C_{\mathfrak{p}}$ is graded based on its probability density function, as shown in Fig. 4: 1) Grade A, $C_{\mathfrak{p}} \in (\mu + \sigma, +\infty)$; 2) Grade B, $C_{\mathfrak{p}} \in (\mu, \mu + \sigma]$; 3) Grade C, $C_{\mathfrak{p}} \in (\mu - \sigma, \mu]$; 4) Grade D, $C_{\mathfrak{p}} \in (\mu - 2\sigma, \mu - \sigma]$; 5) Grade E, $C_{\mathfrak{p}} \in (-\infty, \mu - 2\sigma]$.

Credibility Based Mining Difficulty. Credibility grade represents the historical mining behaviour of a participant. Thus, it is reasonable to evaluate the current or future mining behaviour of a participant via its credibility grade. To encourage credible participants and penalize discredited participants, PoC introduces personalized mining difficulty for a participant based on its credibility. Let $D_{\mathfrak{p}}$ calculated by following piecewise function:

$$D_{\mathfrak{p}} = \begin{cases} 2^{\lfloor (\theta_a + \delta_a F(C_{\mathfrak{p}})) - \lambda \Delta_L \rceil}, & C_{\mathfrak{p}} \in (\mu + \sigma, +\infty) \\ 2^{\lfloor (\theta_b + \delta_b F(C_{\mathfrak{p}})) - \lambda \Delta_L \rceil}, & C_{\mathfrak{p}} \in (\mu, \mu + \sigma] \\ 2^{\lfloor (\theta_c + \delta_c F(C_{\mathfrak{p}})) - \lambda \Delta_L \rceil}, & C_{\mathfrak{p}} \in (\mu - \sigma, \mu] \\ 2^{\lfloor (\theta_d + \delta_d F(C_{\mathfrak{p}})) - \lambda \Delta_L \rceil}, & C_{\mathfrak{p}} \in (\mu - 2\sigma, \mu - \sigma] \\ 0, & C_{\mathfrak{p}} \in (-\infty, \mu - 2\sigma] \end{cases}, \tag{2}$$

where $F(C_{\mathfrak{p}})$ denotes the cumulative probability function of $C_{\mathfrak{p}}$, $\Delta_L = \max\{L_{\mathfrak{p}} - L_{Thr}, 0\}$ with $L_{\mathfrak{p}}$ denoting the length of the pre-block mining information produced by $\mathfrak{p}$, and L_{Thr} denotes a threshold. Furthermore, it is set $\theta_a = \theta_b + F(\mu + \sigma)(\delta_b - \delta_a), \theta_b = \theta_c + F(\mu)(\delta_c - \delta_b), \theta_c = \theta_d + F(\mu - \sigma)(\delta_d - \delta_c), \theta_d = -\delta_d F(\mu - 2\sigma)$ and $0 < \delta_a < \delta_b \leq 1 < \delta_c < \delta_d$. Fig.5 shows how the personalized mining difficulty increases with $C_{\mathfrak{p}}$ and $L_{\mathfrak{p}}$.

3.5 Fork Selecting Strategy

There is a strategy for a participant to select a fork to extend from multiple received forks. First, each block B_i is assigned with a score s_i calculated as $s_i = F(C_{\mathfrak{p}})e^{-\max\{0,L_i-L_{Thr}\}}$, where $F(\cdot)$ denotes the cumulative probability function of participant credibility, $C_{\mathfrak{p}}$ denotes the credibility of $\mathfrak{p}$ that generates block B_i, L_i denotes the length of mining information of B_i, and L_{Thr} denotes a specified threshold. Fig.6 shows that $s_i \in (0, 1)$ and parameters $C_{\mathfrak{p}}, L_i$ incur different influences on s_i. Then, the chain score of a fork can be defined as $s_{\mathcal{C}} = \sum_{B_i \in \mathcal{C}} s_i$. Because the largest chain score implies the best chain quality, a participant accepts the fork with the largest score instead of the longest fork.

3.6 PoC Based Blockchain Protocol

Algorithm 3 PoC Protocol Π

1: Initialize: $\mathcal{C} \leftarrow B_0$ //$\mathcal{C}$ denotes the current chain and B_0 denotes the genesis block of the chain
2: **while** True **do**
3: Upon receiving chain set $\mathcal{S}_{\mathcal{C}}$ and unrecorded information
4: Extracting unverified block set $\mathcal{S}_{ID}$ which includes all current $\mathcal{B}_i$ of the chain $\mathcal{C}_i \in \mathcal{S}_{\mathcal{C}}$
5: verifying the block $B_i \in \mathcal{S}_{ID}$ following the chain score s_i descending order
6: $(B_i, V_i) \leftarrow \mathcal{V}(\mathcal{S}_{ID})$
7: $\mathcal{C} \leftarrow \mathcal{C}_i$
8: Updating the mining difficulty D following the current chain $\mathcal{C}$
9: Calculated the initial mining message msg
10: Solving the current mining puzzle
11: $s_k \leftarrow \mathcal{S}(msg, D)$
12: Generating new block B_k of $\mathcal{C}$
13: $\mathcal{C} \leftarrow \mathcal{C}||B_k$
14: Broadcast the current chain $\mathcal{C}$
15: **end while**

The overview of PoC based blockchain protocol is presented as algorithm 3. Firstly, the chain $\mathcal{C}$ is initialized with the genesis block. Then, in each round, participants receive a set of chain $\mathcal{S}_{\mathcal{C}}$ from the whole network. The chain in $\mathcal{S}_{\mathcal{C}}$ is sorted by chain score. Thirdly, participants invoke the algorithm 2 to verify the validity of current block B_i and get the corresponding verification set V_i. Finally, the participants invoke the algorithm 1 to mine new block B_k of current chain with updated information. The new block B_k is broadcast to the network, which implies the current round of the PoC Protocol is finished.

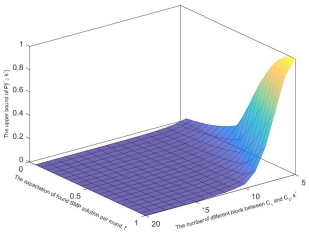

Fig. 7. Upper Bound of Probability $P[l^* \geq k^*]$

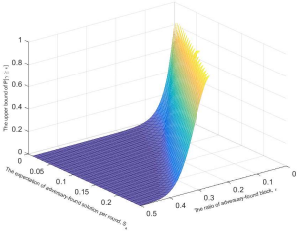

Fig. 8. Upper Bound of Probability $P[\gamma \geq \epsilon]$

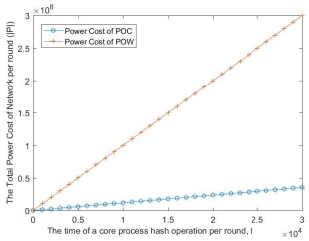

Fig. 9. Power Cost of PoC and PoW

4 Security Analysis and Performance Evaluation

Common Prefix Property. In PoC, a participant is punished with decreased credibility when it publishes blocks or verification of different forks with a similar height. Thus, it is reasonable to assume that the honest participant will only mine on one fork and the adversary only publishes blocks/verification on one fork in a temporal interval. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be the chains of two honest participants at a given round, k^* be the minimum integer such that$\mathcal{C}_1^{\lceil k^* \rceil} \preceq \mathcal{C}_2$ and $\mathcal{C}_2^{\lceil k^* \rceil} \preceq \mathcal{C}_1$[1]. Assume that all the last k^* blocks of C_1 and C_2 are produced in l rounds. The total length $2k^*$ of the sub-chains cannot be greater than the solution number X obtained by all participants in l rounds. Furthermore, let l also denote the minimum number of rounds that a participant is allowed to mine for different forks without punishment, $\mathcal{H}$ and $\mathcal{A}$ denote the sets of honest participants and adversaries, respectively. Let $X_{i,k}$ denote a Boolean random variable, where $X_{i,k} = 1$ iff there is a solution produced for $\mathcal{C}_1$ or $\mathcal{C}_2$ by participant i in the last $(l-k)^{th}$ round, the probability $P[X_{i,k} = 1]$ is calculated as follows:

$$P[X_{i,k}=1] = \begin{cases} p_i, & i \in \mathcal{H} \\ \bar{p}_i^{2(k-1)}(1-\bar{p}_i^2) + (1-\bar{p}_i^{2(k-1)})p_i, & i \in \mathcal{A}, k < l \\ \bar{p}_i^{2(l-1)}(1-\bar{p}_i^2) + (1-\bar{p}_i^{2(l-1)})p_i, & i \in \mathcal{A}, k \geq l \end{cases}, \tag{3}$$

where $\bar{p}_i = (1 - D_{\mathtt{p}_i}/N)^q$, $p_i = 1 - \bar{p}_i$ and q denotes the maximum times a participant processes the hash function in a round. Thus, the expectation of random variable $X = \sum_{i \in \mathcal{H} \cup \mathcal{A}} \sum_{k=1}^{l} X_{i,k}$ is calculated as follows:

$$\mu_1 = \sum_{i \in \mathcal{H} \cup \mathcal{A}} \sum_{k=1}^{l} E(X_{i,k}) \geq lf, \tag{4}$$

where $f = S_a + S_h$, $S_h = \sum_{i \in \mathcal{H}} p_i$ and $S_a = \sum_{i \in \mathcal{A}} p_i$ are the expected number of solutions that may be found for a chain per round by all participants, honest participants and adversary participants, respectively. The length of chains l^* cannot be greater than the number of solutions X obtained in l rounds. Thus,

[1] The equantion is defined in [3]: $\mathcal{C}_i^{\lceil k^* \rceil}$ denotes the sub-chain of $\mathcal{C}_i$ remove the latest k blocks, $* \preceq **$ denotes chain $**$ is contained in chain $*$.

$P[l^* \geq k^*] \leq P[X \geq 2k^*]$. By the Chernoff bound,

$$\begin{cases} P[l^* \geq k^*] \leq P[X \geq (1+\delta)\mu_1] \leq e^{-\frac{\delta^2\mu_1}{3}}, & 0 < \delta \leq 1 \\ P[l^* \geq k^*] \leq P[X \geq (1+\delta)\mu_1] \leq e^{-\frac{\delta\mu_1}{3}}, & 1 < \delta \end{cases}, \tag{5}$$

where $(1+\delta)\mu_1 \geq (1+\delta)lf = 2k^*$. Fig. 7 shows the upper bound of $P[l^* \geq k^*]$ for $l = 10$, $f \in (0,1)$ and $k^* \in [5,20]$. It is clear that the probability $P[l^* \geq k^*]$ drops exponentially with k^*.

Chain Quality Property. Let random variable $Y = \sum_{i\in\mathcal{A}} \sum_{k=1}^{L} X_{i,k}$ be the number of solutions of a chain $\mathcal{C}$ found by adversary participants in L rounds and the expectation of Y can be calculated as $\mu_2 = \sum_{i\in\mathcal{A}} \sum_{k=1}^{L} E(X_{i,k}) = \sum_{j\in\mathcal{A}} Lp_j = LS_a$. Furthermore, let γ be the ratio of adversary-provided block to continuous blocks of chain $\mathcal{C}$ produced in L rounds. It is clear that $P[\gamma \geq \epsilon] \leq P[Y \geq \epsilon Lf]$. Following the Chernoff bound,

$$\begin{cases} P[\gamma \geq \epsilon] \leq P[Y > (1+\delta_a)\mu_2] \leq e^{-\frac{\delta^2\mu_2}{3}}, & 0 < \delta \leq 1 \\ P[\gamma \geq \epsilon] \leq P[Y > (1+\delta_a)\mu_2] \leq e^{-\frac{\delta\mu_2}{3}}, & 1 < \delta \end{cases}, \tag{6}$$

where $(1+\delta)\mu_2 = (1+\delta)LS_a = \epsilon Lf$. Fig. 8 shows the upper bound of $P[\gamma \geq \epsilon]$ where $L = 100$, $f = 1$, $\gamma \in (0, 0.5]$ and $S_a \in (0, 0.25]$.

It is implies that, the chain quality of PoC mechanism is high enough while S_a is small enough. However, S_a must be small when the credibility of an adversary is lower than grade C. The condition is easy to obtain when participant credibility is accurately reflected by the credit rating.

Power Cost. Let n be the number of active participants, m be the average number of computing core per active participant, P be the average power cost that a core processes a hash operation, t be the times that a core runs a hash function per round and λ be the ratio of verified times to a round. Thus, the power cost of PoC mechanism of the whole network can be calculated as $\mathcal{P}_{PoC} = (1-\lambda)ntP + \lambda mntP$. Let $(1-\lambda)t \to q$ and $\lambda mt \to \alpha q$ where q denotes the maximum length of mining information and αq denotes the maximum length of verification. Thus, $\lambda \to \alpha/(m+\alpha)$, it can be simplified as $\lambda \to \alpha/m$ while $m >> \alpha$. Similarly, the power cost of PoW can be calculated as $\mathcal{P}_{PoW} = (t-1)mnP + nP$. Let $\alpha = 0.2$, $m = 10$ and $n = 1000$, Fig.9 shows the power cost of PoC and PoW increase with parameter t. However, PoC cost more power than PoW during verifying, the total power cost is much lower during mining.

5 Conclusions

In this paper, we propose a novel consensus mechanism named PoC. Compared with traditional consensus mechanisms, PoC provides strong resistance to resource centralization and malicious participant attacks. First, resource coalition gets negligible advantage in mining competition, because SMP is introduced in PoC. Second, in PoC, each participant is provided with personalized mining difficulty which depends on the participant credibility. Furthermore, the credibility

of each participant is quantified by its mining behaviour, which guarantees that the more credible participants get the higher successful mining probability. Finally, the performance of PoC is thoroughly analyzed in terms of common prefix, chain quality and power cost. The analysis justifies that PoC is security and incentive compatible when suitable parameters are set, It also can provide strong security and robustness for blockchain based system.

Acknowledgment

This work was supported by the National Science Foundation of China (No. 61802083, 61862011), the Natural Science Foundation of Guangxi (2018GXNSFBA281164, 2018GXNSFAA138116).

References

1. David S Evans. Economic aspects of bitcoin and other decentralized public-ledger currency platforms. *University of Chicago Coase-Sandor Institute for Law & Economics Research Paper*, (685), 2014.
2. Jae Kwon. Tendermint: Consensus without mining. *URL: http://tendermint.com/docs/tendermint {_} v04.pdf*, 2014.
3. Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.
4. Aron Laszka, Benjamin Johnson, and Jens Grossklags. When bitcoin mining pools run dry. In *International Conference on Financial Cryptography and Data Security*, pages 63–77. Springer, 2015.
5. Arthur Gervais, Ghassan Karame, Srdjan Capkun, and Vedran Capkun. Is bitcoin a decentralized currency? *IEEE Security & Privacy*, 12(3):54–60, 2014.
6. Danny Bradbury. The problem with bitcoin. *Computer Fraud & Security*, 2013(11):5–8, 2013.
7. Jon Matonis. The bitcoin mining arms race: Ghash. io and the 51% issue, 2014.
8. Andrew Miller, Ahmed Kosba, Jonathan Katz, and Elaine Shi. Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 680–691. ACM, 2015.
9. Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining , or bitcoin in the presence of adversaries. *Proceedings of WEIS*, 2013.
10. Nicolas Houy. It will cost you nothing to "kill" a proof-of-stake crypto-currency. *URL: http://papers.ssrn.com/sol3/papers.cfm*, 2014.
11. Tuyet Duong, Lei Fan, and Hong-Sheng Zhou. 2-hop blockchain: combining proof-of-work and proof-of-stake securely. *URL: https://eprint. iacr. org/2016/716*, 2016.
12. Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of activity: extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Performance Evaluation Review*, 42(3):34–37, 2014.
13. Michael Vierhauser, Rick Rabiser, and Paul Grünbacher. A case study on testing, commissioning, and operation of very-large-scale software systems. In *Companion Proceedings of the 36th International Conference on Software Engineering*, pages 125–134. ACM, 2014.

14. Swan Dubois, Rachid Guerraoui, Petr Kuznetsov, Franck Petit, and Pierre Sens. The weakest failure detector for eventual consistency. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 375–384. ACM, 2015.
15. Satoshi Nakamoto. Bitcoin: a peer-to-peer electronic cash system, 2008.
16. Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 315–324. ACM, 2017.
17. Asoke K Talukder, Manish Chaitanya, David Arnold, and Kouichi Sakurai. Proof of disease: A blockchain consensus protocol for accurate medical decisions and reducing the disease burden. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 257–262. IEEE, 2018.
18. Zhongli Dong, Young Choon Lee, and Albert Y Zomaya. Proofware: Proof of useful work blockchain consensus protocol for decentralized applications. *arXiv preprint arXiv:1903.09276*, 2019.
19. Joseph Bonneau Andrew Miller Jeremy Clark, Arvind Narayanan Joshua A Kroll Edward, and W Felten. Research perspectives on bitcoin and second-generation cryptocurrencies. In *IEEE Symposium on Security and Privacy*, 2015.
20. Andrew Miller and Joseph J LaViola Jr. Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin. *URL: http://nakamotoinstitute.org/research/anonymous-byzantine-consensus*, 2014.
21. M. Bedford Taylor. Bitcoin and the age of bespoke silicon. In *International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, pages 1–10, 2013.
22. John Tromp. Cuckoo cycle: a memory-hard proof-of-work system. *IACR Cryptology ePrint Archive*, page 59, 2014.
23. Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19, 2012.
24. Pavel Vasin. Blackcoin's proof-of-stake protocol v2. *URL: http://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf*, 2014.
25. Andrew Poelstra et al. Distributed consensus from proof of stake is impossible, 2014.
26. Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to better: how to make bitcoin a better currency. In *International Conference on Financial Cryptography and Data Security*, pages 399–414. Springer, 2012.
27. Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*. Citeseer, 2013.
28. Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.
29. Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2015.
30. Juan A. Garay. Basic properties of the blockchain: (invited talk). In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 1–1. ACM, 2017.
31. Patrick Billingsley. Convergence of probability measures, second edition. 2008.