IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 14, 2020, accepted June 29, 2020, date of publication July 8, 2020, date of current version July 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007938

GSCS: General Secure Consensus Scheme for Decentralized Blockchain Systems

JING WANG[1], YONG DING[1], NEAL NAIXUE XIONG[2], WEI-CHANG YEH[3], (Senior Member, IEEE), AND JINHAI WANG[4]

[1]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[2]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA
[3]Department of Industrial Engineering and Engineering Management, College of Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan
[4]College of Electronic and Information Engineering, Foshan University, Foshan 528011, China

Corresponding authors: Jing Wang (wjing@guet.edu.cn) and Neal Naixue Xiong (xiongnaixue@gmail.com)

This study was supported by the National Science Foundation of China (grant Nos. 61802083, 61862011, 61772150 and 61862012), the Natural Science Foundation of Guangxi (grant Nos. 2018GXNSFBA281164, 2018GXNSFAA138116, 2018GXNSFDA281054 and 2018GXNSFAA281232) and the National Cryptography Development Foundation of China (grant No. MMJJ20170217).

ABSTRACT Blockchain, a type of a decentralized network system that allows mutually distrustful parties to transact securely without involving third parties, has recently been attracting increasing attention. Hence, there must be a consensus mechanism to ensure a distributed consensus among all participants. Such a consensus mechanism may also be used to guarantee fairness, correctness and security of such decentralized systems. Thus, in this paper we propose a novel consensus mechanism named GSCS that is an improved version of PoW. Compared with existing consensus mechanisms (such as PoW, PoS and so on), GSCS provides strong resistance to resource centralization, the quantum attack and other malicious attacks. In this work, we first present the serial mining puzzle to resist collusive mining and the quantum attack. It guarantees that participants can only obtain a negligible advantage by solving the relevant problem in parallel. Second, GSCS considers the influence of participant credibility. The credibility is reflected by the mining behavior of each participant and directly influence to the mining difficulty of participant. Thus, credible participants enjoy a higher probability of winning the mining competition than do participants who are not credible. Finally, performance of GSCS is analyzed in terms of the common prefix, chain quality, chain growth, and power cost. The results indicate that GSCS is security- and incentive-compatible with suitable security parameter settings. In brief, GSCS has the potential to ensure a more secure and robust environment for decentralized blockchain systems.

INDEX TERMS Decentralized system, blockchain, consensus mechanism, GSCS.

I. INTRODUCTION

Distributed and decentralized network systems are gaining popularity. A growing number of businesses and individuals have adopted such systems to access application services available on the Internet. In other words, they can deploy and host various kinds of applications on a distributed and decentralized network platform. Compared with centralization application platforms, a decentralized platform offers numerous advantages of scalability, flexibility, and low cost. However, security and manageability arise as central challenges at the same time. Fortunately, blockchain proposed a perfect solution for such security and manageability problem. *Bitcoin*, a well-known cryptocurrency system, provides an efficient way to maintain a decentralized network system [1]. The core of *Bitcoin*-like systems named *blockchain* can be viewed as a decentralized public ledger [2]. Because of the decentralized nature of systems using it, they must be maintained entirely by all participants instead of an appointed trusted third party (TTP). Thus, in blockchain systems, a consensus mechanism, such as a proof-of-work (PoW), a proof-of-stake (PoS), etc., is required to prevent attacks such as double-spending (i.e., the scenario of an attacker spending a Bitcoin more than once) [3]. Furthermore, the consensus mechanism

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

provides a promising direction for guaranteeing the fairness, correctness and security of more generalized decentralized systems.

However, the emergence of a resource coalition is inevitable in such Bitcoin -like systems [4]. It will Seriously threat to the blockchain security. Because it can break the *honest majority* and trigger *51% attack*. Currently, the consensus mechanism follows a fundamental assumption named *honest majority*. It assumes that adversaries can break the mechanism with a negligible probability since it is difficult to control the majority (greater than than 51%) of mining resources [5]. However, the presence of resource coalitions may violate the honest majority assumption and incurs a large lurking threat to the security of the blockchain [6]–[8]. For instance, the largest mining pool called Ghash.IO had controlled more than 30% of mining capacity of *Bitcoin* at once [9]. In fact, the phenomenon of a resource coalition will become more and more serious in future [4]. On the one hand, solo participants have a strong incentive to collude to hedge mining risks and obtain more stable rewards [10]. On the other hand, there is a built-in design limitation of the *mining puzzle* of *Bitcoin*, which admits an effective coalition enforcement mechanism [4]. It implies that the PoW mining puzzle connive the miner coalition(e.g. mining pool). That will make the appearance of resources centralization more and more serious. Additionally, the ability of the consensus mechanism to resist the quantum attack is indispensable because a quantum computer can provide greater parallel computing power than can a recent traditional computer [11], [12]. Thus, a significant challenge of blockchain security is preventing malicious miners from implementing the *51% attack* and the *selfish strategy attack* by centralizing resources or using a quantum computer [4], [11], [13].

To prevent resource centralization, two kinds of solutions have been proposed recently: increasing the resource coalition risk and increasing the resource coalition cost. However, each of them gets weaknesses. For the former, Miller *et al.* proposed a notion named *nonoutsourceable puzzle* [4]. It allows participants in a coalition to steal the reward of the coalition without any evidence and negative impact. Thus, it effectively creates a disincentive for participants to join the coalition because doing so would incur a high risk of reward loss. For the latter, Duong *et al.* [14] and Bentov *et al.* [15] proposed the combined consensus mechanisms of PoW and PoS. In these mechanisms, attacks can only yield advantage in a mining competition if attackers control the majority of both computational power and coin stake. Thus, the attack cost is greatly increased, and the security threat is mitigated. Additionally, Aggarwal *et al.* proposed a quantum-resistant PoW puzzle that could mitigate the problem-solving advantage of quantum computers [11]. However, the puzzle still could not achieve a complete elimination of the quantum advantage. In brief, the approaches of [4], [14], [15] impede resource centralization by providing various coalition disincentives, and the solution of [11] only provides a limited quantum attack resistance. However, the above solutions can mitigate the incentive to collude and the power of the quantum attack, they are still hard to completely eliminate the threat of resource centralization. The blockchain still requires a consensus mechanism that essentially resists a resource coalition and the quantum attack. Such a consensus mechanism would strictly maintain the fairness of the blockchain system.

In this paper, we propose an improved PoW mechanism named the general secure consensus scheme (GSCS). It is an extended version of our previous work [16]. Our GSCS consists of two core components: the serial mining puzzle (SMP) and the mining credibility system (MCS). The first component – SMP – is a novel mining puzzle that naturally resists a resource coalition. In contrast to the nonoutsourceable puzzle, SMP prevents not only outsourced mining but also parallel mining. It implies that the mining power heavily relies on the single CPU's computational power instead of the aggregate computational power of multiple CPUs. Thus, a computational resource coalition (e.g., a mining pool) possesses a negligible advantage in the SMP mining competition. Second, MCS is introduced into GSCS to avoid the influence of malicious participants. In fact, the entire sequence of participants' mining actions is indirectly recorded in the blockchain, which reflects each participant's credibility. MCS evaluates the latter and quantifies the credibility-based mining difficulty. Thus, in GSCS each participant is assigned a personalized mining difficulty dependent on that participant's credibility. Ideally, the mining difficulty should monotonically decrease with the participant's credibility. As a result, GSCS tends to accept the *next block* created by the participants with a high credibility. That creates a disincentive for participants to break the protocol of consensus mechanism. Furthermore, the proposed GSCS provides an efficient way to deploy a secure blockchain by using SMP and MCS. In summary, the contributions of this paper are as follows:

1) We propose the *serial mining puzzle* to resist parallel mining. It can effectively avoid resource centralization and resist the quantum attack.
2) We provide a quantitative method for *participant credibility*, which is directly evaluated based on the mining events recorded in blockchain. It allows the decentralized blockchain system to efficiently detect participants who are not credible.
3) We develop a *personalized mining difficulty* to allow credible participants to have a competitive advantage during mining. It significantly increases the difficulty of controlling a GSCS-based decentralized system with an illegal mining policy.

The rest of this paper is organized as follows. Section II presents the related studies. Section III introduces important preliminaries of the proposed scheme. Section IV describes the proposed GSCS in detail. Section V presents a detailed performance analysis and evaluation of GSCS. Section VI presents the conclusions and discusses directions of our future research.

II. RELATED WORK

A. DECENTRALIZED SYSTEMS AND BLOCKCHAIN SYSTEMS

Decentralized systems provide an effective way to develop large-scale applications with loosely coupled operations in a networked environment [17]–[20]. Significantly, the decentralized nature of such systems also introduces numerous novel requirements and functions [21]–[24]. One of the best-known examples of decentralized systems is a decentralized cryptocurrency system named *Bitcoin*. It was proposed by Nakamoto in 2009 [1]. Soon after that, the growth of decentralized systems beyond cryptocurrency gained momentum [25]. As a successful example, a blockchain-oriented application called a smart contract has attracted particularly wide attention [26]. For instance, Jun *et al.* built a decentralized service contract management scheme based on blockchain [27], and Christidis *et al.* combined smart contracts and the Internet of Things (IoT) to implement a decentralized blockchain-IoT system [28]. Additionally, Chao Lin *et al.* proposed a blockchain-based secure mutual authentication system called Bsein [29], Asoke K Talukder *et al.* proposed an accurate medical decision-making system based on blockchain named "proof of disease" [30], Zhongli Dong *et al.* proposed a blockchain consensus protocol for decentralized applications named Proofware [31], etc. Thus, blockchain-based decentralized systems were going to be a research area of interest to a number of researchers in the future.

B. CONSENSUS MECHANISM OF BLOCKCHAIN

1) COMPUTATIONAL POWER-BASED CONSENSUS MECHANISM

The consensus mechanism of blockchain provides an efficient approach to avoiding double-spending for *Bitcoin*-like systems [32]. In *Bitcoin*, Nakamoto first proposed the Nakamoto consensus using the proof-of-work computational puzzle [1]. It is a novel consensus mechanism dependent on each participant's computational power without third-party involvement. Following the Nakamoto consensus, the blockchain may generate several temporary forks, but one of those forks will eventually surpass others and bring the eventual consensus to the entire network [1], [5], [33]. Afterwards, several modified computational puzzles have been proposed to solve some specific problems of the Nakamoto consensus. To increase the mining reward, some miners may tend to use customized hardware, e.g., mining rigs, to improve the mining efficiency. Application-specific integrated circuits (ASIC) have recently achieved orders of magnitude better efficiency than that of common chips during mining [34]. Thus, considering fairness, an ASIC-resistant mining puzzle has been proposed to maintain the competitiveness of commodity hardware in a mining competition [35]. Similarly, a quantum-resistant PoW has been developed to reduce the mining advantage of a quantum computer [11]. Meanwhile, a useful puzzle has been introduced to avoid consuming the energy and resources solely for mining. Kroll indicated that any useful puzzle must produce a valuable good [10]. As an example, Primecoin introduced the first useful puzzle [36]. The latter requires miners to find sequences of large prime numbers that can be provided as parameters for many cryptographic protocols. To protect the decentralization of Bitcoin-like systems, Miller first proposed the notion named nonoutsourceable puzzle to prevent a miner coalition (i.e., a mining pool) [4]. The nonoutsourceable puzzle allows a participant of a mining pool to steal the mining reward without any negative effect for that participant. Focus on fairness and throughput, Bitcoin-NG [37], Conflux [38], fruitchains [25] et. al are proposed as a kinds of hierarchical blockchain protocol based on primary PoW puzzle. Furthermore, in Monoxide, the notions of *Chu-ko-nu Mining* and *Eventual Atomicity* are proposed to improve efficiency, security and decentration of blockchain system [39].

TABLE 1. Consensus mechanisms comparison.

Consensus Mechanisms	Advantage Parallel Mining	Personalized Mining Difficulty
PoW	Stronge	No
PoS	Stronge	Yes
nonoutsourceable puzzle	limited	No
GSCS	limited	Yes

2) VIRTUAL RESOURCE-BASED CONSENSUS MECHANISM

In contrast to a computational power-based consensus mechanism, the proof-of-stake (PoS) provides a virtual resource-based consensus mechanism [40]. Several versions of PoS have been proposed recently, such as proof-of-coin-age [40], pure proof-of-stake [41], proof-of-deposit [3], proof-of-burn [42], proof-of-activity [15], etc. A representative example is a novel PoS protocol proposed by Sarah Azouvi *et al.*, named Fantomette [43], for which formal game-theoretic proofs of security have been provided. Instead of costing actual computational resources, PoS costs virtual resources to maintain blockchain. It effectively avoids the waste of such computational resources. However, some researchers insist that the stability and security of PoS consensus mechanisms remains an open problem that needs to be formally addressed. Poelstra claims that external resource consumption is necessary for blockchain security [44]. However, King *et al.* believe that in PoS blockchains it may be more difficult for an attacker to acquire a sufficiently large amount of the virtual resource than to acquire sufficiently powerful computing equipment [40]. The core argument is that a resource-based consensus mechanism is susceptible to costless simulation attacks. It implies that such attacks allow construction of an alternate view of history at no cost, and lead to a different currency allocation of blockchain-based cryptocurrency systems [13].

3) COMPARISON OF CONSENSUS MECHANISMS

Table 1 presents a comparison of various consensus mechanisms. PoW, PoS and nonoutsourceable puzzle are chosen as representatives to compare with proposed GSCS. The table shows the resistibility of resource centralization for consensus mechanisms in two aspects: parallel mining advantage and personalized mining difficulty. Firstly, in PoW and PoS, miners can get significant competitive advantage by parallel mining. Thus, resource centralization is potential encouraged in PoW or PoS based blockchain system. Inversely, nonoutsourceable puzzle and GSCS are all provided resistibility to parallel mining. Thus, the threat of resource centralization must be mitigated. Second, the personalized mining difficulty is an efficient way to implement perennial rewards and punishments. It can encourage miners following the protocol of consensus mechanisms. However, in PoS, the personalized mining difficulty directly relies on the e-currency of miner. That would incur some risk [44]. Different from PoS, GSCS provided a mining credibility system to evaluate the mining behavior of miner. The credibility based mining difficulty can efficiently encourage honest miners.

C. PERFORMANCE EVALUATION OF BLOCKCHAIN

A core concern of a blockchain system is the security and stability of its consensus mechanism. Security of a blockchain has initially been informally proven in the *honest majority* model [1], [5], [33]. However, the model is insufficient because a sufficient guarantee is not provided for the *honest majority* assumption. Several researches deem that the mining reward of *Bitcoin*-like systems provides the essential incentive for participants to take part in the system [1], [45]. However, an economic analysis indicates that *Bitcoin*-like systems are not fixed, rule-driven, and incentive-compatible as some advocates claim [10]. In fact, a participant (or a coalition) can cause a blockchain system to deviate from the incentive-compatible state by using a selfish mining strategy with a third of the total computational power [46]. Furthermore, an optimal selfish mining strategy has been provided as the best response to the honest mining strategy [47]. It presents a lower bound of the resource amount (less than 25%) needed for a profitable selfish mining strategy. This result highlights the importance of preventing miner coalitions [25]. To evaluate blockchain performance, some researches have attempted to formulate the fundamental metrics of blockchain. First, Garay *et al.* provided two quantifiable properties named the *common prefix* and *chain quality* that are similar to concepts named *liveness* and *persistence* in the Byzantium protocol, respectively [48]. Next, Kiayias *et al.* proposed a novel concept named *chain growth* that evaluates the efficiency of block generating in a blockchain [49]. Additionally, Garay *et al.* focused on the *trusted setup* properties of blockchain. The cited study presents a *bootstrapped* blockchain to guarantee an unpredictable *genesis block* [50]. Pass *et al.* focused on the *fairness* of blockchain and proposed FruitChains to avoid *selfish mining*. Similarly, in this paper we demonstrate the superiority of our GSCS mechanism according to these fundamental metrics. The analysis and evaluation show that GSCS offers strong security and low cost to blockchain systems.

III. PRELIMINARIES

A blockchain protocol provides an algorithm for a set of nodes (also called miners or participants) to interact with each other. The execution of blockchain protocol Π is directed by environment $Z(1^\kappa, n, \rho)$ and entails maintaining *chain* $\mathcal{C}$, where $\mathcal{C}$ denotes an ordered sequence of records, κ denotes the security parameter which used to describe mining difficulty, n denotes the number of participants, and ρ denotes the fraction of honest participants who faithfully follow the prescription of protocol Π. Furthermore, Z is responsible for providing inputs to honest nodes and receiving outputs from them. The inputs of protocol Π may be provided by external applications(i.e. nodes of blockchain network), and the outputs of Π can be received by such applications. Thus, all external applications and protocols running in the blockchain system are viewed as parts of Z. Rafael *et al.* proposed a framework for a blockchain protocol that is simplified as follows [51]:

1) Π proceeds over *rounds* that model time steps. In each round r, each honest participant receives message m from Z and attempts to add it into $\mathcal{C}$ as the latest record. A participant may broadcast the current state of $\mathcal{C}$ to all other participants while m has been successfully recorded in $\mathcal{C}$.
2) An adversary A exists in Z, and controls dishonest participants. In any round, Z can *corrupt* an honest participant (i.e., a participant is controlled by A) or *"uncorrupt"* a dishonest participant (i.e., a participant no longer controlled by A). However, the total number of dishonest participants is a constant $(1-\rho)n$.

A. OVERVIEW OF NAKAMOTO'S BLOCKCHAIN PROTOCOL

Nakamoto proposed a detailed model of a PoW blockchain [1]. As shown in fig. 1, Nakamoto's blockchain consists of a set of *sequential* blocks. It implies that each block is associated with a preceding block (a "pre-block") except for the *genesis block*.[1] Furthermore, each block of blockchain includes two parts: a block header and a set of transaction records. The first part – a block header – contains three parameters: *Pre* denotes the hash of the pre-block, *Nonce* denotes a PoW solution of B_i, and R_i denotes the root of a *Merkle tree*[2] formed by transactions. Intuitively, Blockchain is a set of sequential blocks, each block includes a hash value of pre-block. The parameter *Pre* of B_i is used to determine its unique pre-block B_{i-1}. However, B_i cannot determine its pro-block B_{i+1}, because B_{i+1} cannot be precomputed at the i^{th} round. Second, each transaction 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

[1] The first block of the whole blockchain.

[2] A specific binary tree, in which the value of each leaf node is the hash value of a record, and the value of each non-leaf node is the hash value of its children.

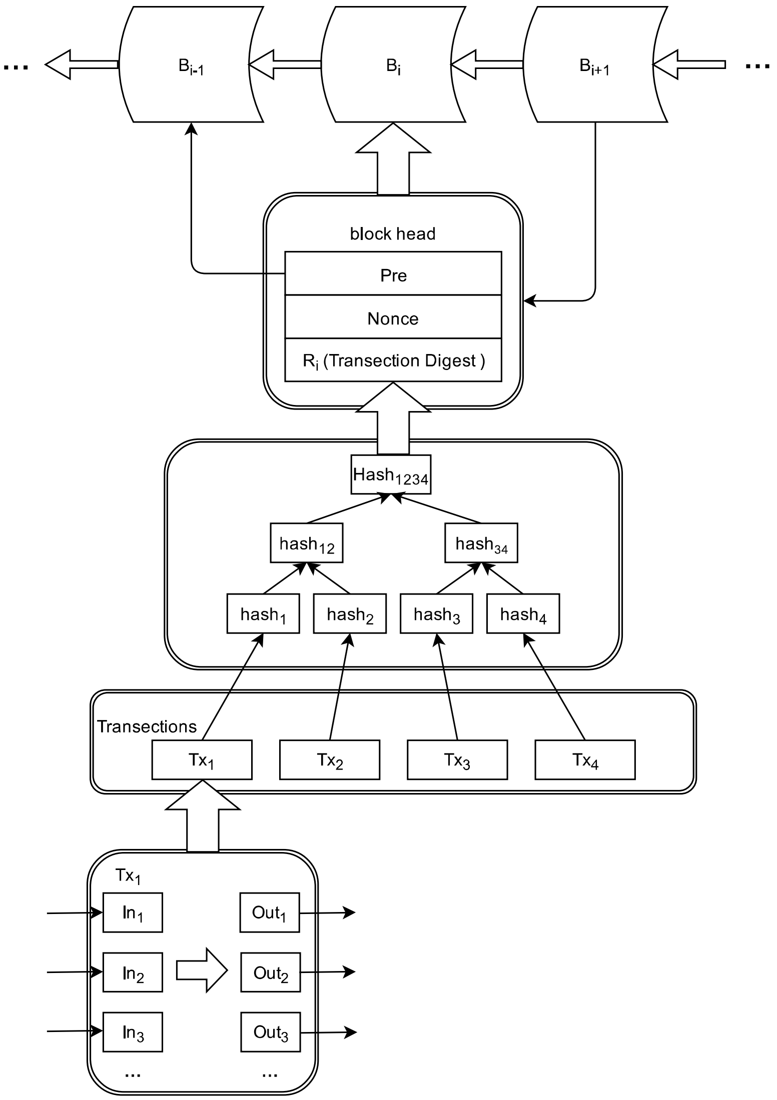


FIGURE 1. **PoW-based blockchain.**

Bitcoin system). Note that for each transaction, the total value of its Inputs must be larger than the total value of its outputs.

In such a PoW-based blockchain system, a block generator is called a *miner*. A miner will persistently search for PoW solutions to generate the next block and will gain a monetary award when its block is confirmed by the blockchain. However, in theory it can never be completely confirmed whether a block has been permanently included in the blockchain. It has only been theoretically proven that the unconfirmed probability of a block decreases exponentially with the number of blocks following it. In practice, 6 following blocks essentially indicate that a block has been accepted as *permanently confirmed*.

B. CONSENSUS PUZZLE

The consensus puzzle of the blockchain has been proposed to guarantee consistency among participants. PoW puzzles and PoS puzzles are widely used by current blockchain systems. In such puzzles, the puzzle-solving efficiency depends on the computing power and a kind of virtual resource possessed by participants. In contrast to such consensus puzzles, the notion of a *time-lock puzzle* has been proposed by Mohammad *et al.* [52] and Ronald *et al.* [53]. Initially, the *time-lock puzzle* was developed for sending messages *to the future*. Afterwards, based on this puzzle's notion, proof of sequential work (PoSW) was proposed by Bram *et al.* [54] and Mohammad *et al.* [55]. The original motivation of PoSW included non-interactive time-stamping and universally verifiable CPU benchmarks. A recent novel application of PoSW is in blockchain protocol design, where it can be used as a more ecological and economic substitute of PoW. More specifically, the definition of a *time-lock puzzle* proposed by Mahmoody *et al.* [55] is as follows:

Definition 1: A time-lock puzzle is a game involving three parties: puzzle generator $\mathcal{G}$, puzzle solver $\mathcal{S}$ and solution verifier $\mathcal{V}$. The parties receive the common input 1^n for security parameter n and $N = poly(n)$ as the *complexity of the puzzle* and act as follows:

1) $\mathcal{G}$ generates puzzle $\mathcal{P}$;
2) $\mathcal{S}$ receives $\mathcal{P}$ and outputs some solution s in time N, and
3) $\mathcal{V}$ receives $\mathcal{P}, s$ and decides whether to accept solution s.

The time-lock puzzle also requires following properties:

1) **Completeness:** In an honest execution, $\mathcal{V}$ accepts a valid solution s with probability $1 - \epsilon(n)$.
2) η**-Soundness:** every nonuniform adversary $\mathcal{A}$ that runs in parallel time $T_{\mathcal{A}}$ that is slightly smaller than the time of the honest solver $T_{\mathcal{S}}$ (i.e., $T_{\mathcal{A}} < \eta \cdot T_{\mathcal{S}}$, where $0 < \eta < 1$) will fail to convince $\mathcal{V}$ with more than a negligible probability. It implies that the probability of output of $\mathcal{A}$ being accepted by $\mathcal{V}$ is negligible.

C. SECURITY OF BLOCKCHAIN

In order to evaluate the security performance of blockchain, there are three quantitative indexes of blockchain security have been introduced: chain growth, chain growth and common prefix.

1) CHAIN GROWTH

The notion of chain growth is used to describe how chain $\mathcal{C}$ grows proportionally to the number of rounds of protocol Π. Assume that **view** denotes a randomly sampled execution trace of Π, and $|\mathbf{view}|$ denotes the number of rounds in the execution trace **view**. Furthermore, let

$$\underline{\mathcal{C}}_{r,t} = \min_{i,j}\{|\mathcal{C}_i^{r+t}| - |\mathcal{C}_j^r|\}, \tag{1}$$

$$\overline{\mathcal{C}}_{r,t} = \max_{i,j}\{|\mathcal{C}_i^{r+t}| - |\mathcal{C}_j^r|\}, \tag{2}$$

where i, j are the honest participants in rounds $r + t, r$, respectively, C_i^r is the local chain of i in round r, C_j^{r+t} is the local chain of j in round $r + t$, and $|\mathcal{C}|$ denotes the length of $\mathcal{C}$. Let $growth^{t_0,t_1}(\mathbf{view}, \delta, T) = 1$ iff

1) $\forall r, r', r+\delta \leq r' \leq |\mathbf{view}|$ and for arbitrary participants i, j such that i is honest in r and j is honest in r', it holds that $|\mathcal{C}_j^{r'}| \geq |\mathcal{C}_i^r|$.

2) $\forall r, r \leq |\mathbf{view}| - t_0$, it holds that $\underline{\mathcal{C}}_{r,t_0} \geq T$.

3) $\forall r, r \leq |\mathbf{view}| - t_1$, it holds that $\overline{\mathcal{C}}_{r,t_1} \leq T$.

Thus, a formal definition of chain growth is given as follows.

Definition 2 (Chain Growth): A blockchain protocol Π satisfies chain growth of (T_0, g_0, g_1) in environment Z if there exists a negligible function $\epsilon(\kappa)$ such that for every $T \geq T_0, t_0 \geq T/g_0, t_1 \leq T/g_1$ it holds that

$$Pr[growth^{t_0,t_1}(\mathbf{view}, \delta, T) = 1] \geq 1 - \epsilon(\kappa). \tag{3}$$

It is important that chain growth describes the growth efficiency of blockchain. It determines the delay of block confirming and the difficulty of block tampering. The larger of $Pr[growth^{t_0,t_1}(\mathbf{view}, \delta, T) = 1]$, the better stability for blockchain.

2) CHAIN QUALITY

The notion of chain quality describes the phenomenon that the number of records contributed by the adversary is proportional to its relative power. Let $quality^T(\mathbf{view}, \mu) = 1$ iff

1) $\forall i, r$, participant i is honest in round r;

2) for an arbitrary consecutive sequence of T blocks $\mathcal{C}_i^r[t, t + T]$ of $\mathcal{C}_i^r$, the fraction of blocks $B \in \mathcal{C}_i^r[t, t + T]$ mined by honest miners is at least μ.

Thus, a formal definition of chain quality is given as follows.

Definition 3 (Chain Quality): A blockchain protocol Π has chain quality of (κ, T, μ) in environment Z if there exists a negligible function $\epsilon(\kappa)$ such that for every $T_0 \geq T$ it holds that

$$Pr[quality^{T_0}(\mathbf{view}, \mu) = 1] \geq 1 - \epsilon(\kappa). \tag{4}$$

Significantly, chain quality describes the advantage of dishonest nodes(adversary) in the mining competition. The larger $Pr[quality^{T_0}(\mathbf{view}, \mu) = 1]$, the less threat of adversary.

3) COMMON PREFIX

A common prefix, also called consistency, describes the consistency of the local chain state of all participants. Let $consistency^T(\mathbf{view}) = 1$ iff for all rounds $r \leq r'$ and participants i, j honest in rounds r and r', respectively, it holds that the prefixes of $\mathcal{C}_i^r$ and $\mathcal{C}_j^{r'}$ consisting of the first $l = |\mathcal{C}_i^r| - T$ blocks are identical. Thus, a formal definition of a common prefix is given as follows.

Definition 4 (Common Prefix): A blockchain protocol Π has a common prefix of (κ, T) in environment Z if there exists a negligible function $\epsilon(\kappa)$ such that for every $T_0 \geq T$,

$$Pr[consistency^{T_0}(\mathbf{view}) = 1] \geq 1 - \epsilon(\kappa). \tag{5}$$

Thus, Common prefix describes the possibility of chain forking. It determines the ability to resist double spending attack. The larger $Pr[consistency^{T_0}(\mathbf{view}) = 1]$, the less success probability of double spending attack.

IV. PROPOSED GENERAL SECURE CONSENSUS SCHEME

GSCS is an improved version of PoW that provides the ability to resist resource centralization and collusion. More specifically, there are two core functional modules of GSCS: a serial mining puzzle(SMP) and a mining credibility system(SMP). The first module–SMP–encourages participants to mine independently. The reason is that parallel mining is no longer useful for solving SMP. The second module–MCS–is proposed to quantify each participant's mining credibility. Furthermore, each participant can be assigned a personalized mining difficulty based on that participant's credibility. Ideally, MCS can increase the success probability for credible participants during mining. Thus, the two functional modules provide sufficient protection against decentralization to ensure security of GSCS.

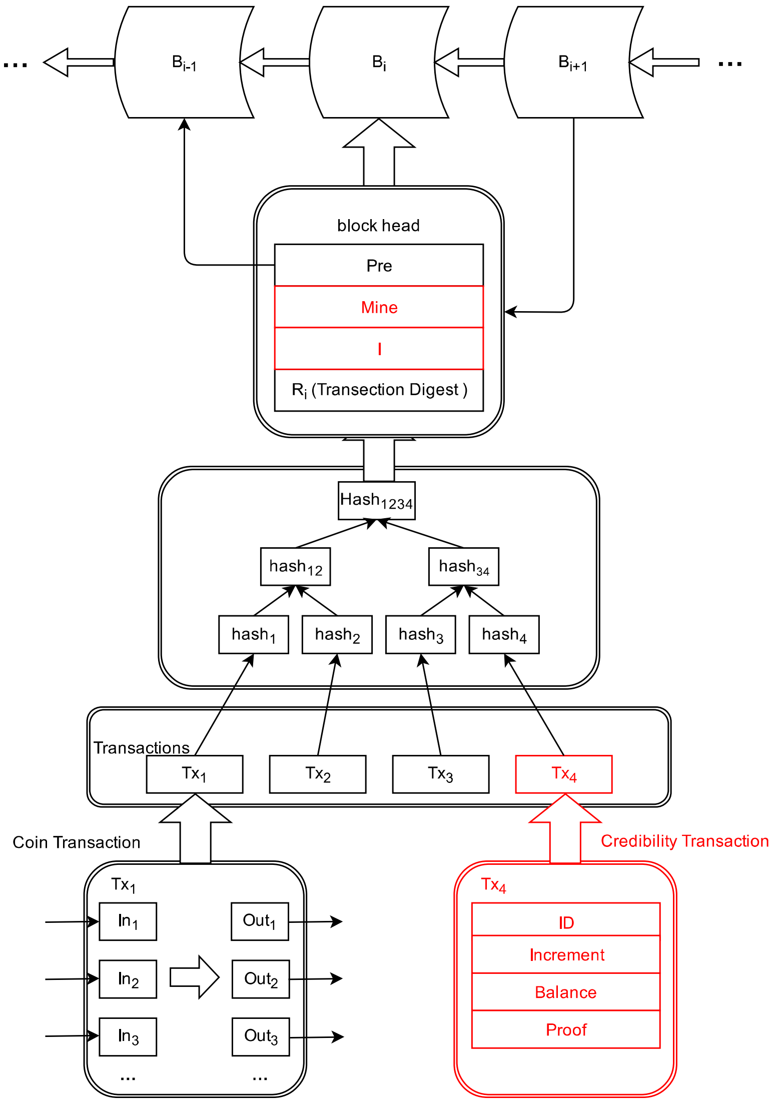


FIGURE 2. Design of a general secure consensus scheme-based blockchain.

Fig.2 presents a visual overview of a GSCS blockchain. It is clear that there are two prominent differences between the GSCS and PoW blockchains. First, the block header parameter *Nonce* is replaced by mining information *Mine* and block height *i* in GSCS. In contrast to *Nonce*, *Mine* includes two parts: a serial mining puzzle *solution* and the corresponding *verification* provided by a set of participants. It implies that *Mine* contains detailed information of mining events and reflects the credibility of the block generator. Second, a *credibility transaction* is introduced into the blockchain system to quantify participants' credibility. However, the value of credibility cannot be transacted, and can only be updated by a specific *mining-event*. Furthermore, a credibility transaction includes four parameters: *ID*, *Increment*, *Balance* and *Proof*. In particular, *ID* denotes a credibility account in GSCS, *Increment* denotes the credibility increment caused by a mining-event, *Balance* denotes the updated credibility balance of the account, and *Proof* denotes the corresponding proof of a mining event's occurrence. Additionally, for readability we list and explain the primary symbols used in this paper in table 2.

A. SERIAL MINING PUZZLE

SMP is one of the core modules in GSCS. It can efficiently deter resource centralization and the quantum attack because it provides a strong guarantee against parallel mining. In the GSCS blockchain, participants always solve SMP instead of a PoW puzzle. Similarly to the time-lock puzzle, SMP cannot be efficiently solved by any parallel algorithm. Thus, a resource coalition and a quantum computer can offer negligible advantages in a mining competition. Similarly, following the concept of a time-lock puzzle, we propose a formal SMP definition as follows.

Definition 5 (Serial Mining Puzzle): SMP is a protocol that has the following two phases. In the solving phase, a participant receives input information I and puzzle difficulty D.

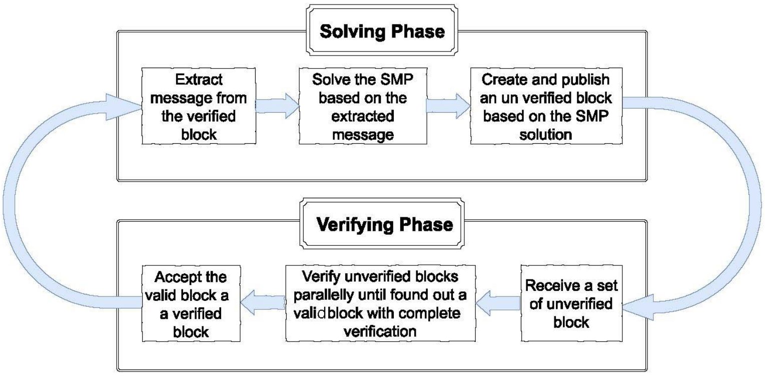

Solving Phase
Extract message from the verified block
Solve the SMP based on the extracted message
Create and publish an un verified block based on the SMP solution
Verifying Phase
Accept the valid block a a verified block
Verify unverified blocks parallelly until found out a valid block with complete verification
Receive a set of unverified block

FIGURE 3. Serial mining puzzle.

TABLE 2. Primary symbols used in the paper.

Symbol	Description
B_i	The i^{th} block of the blockchain.
Pre_i	The hash value of the i^{th} block B_i.
R_i	The root of the transaction's Merkle tree recorded by B_i.
V_i	The complete verification set of block B_i.
$V_{i,ID}^{+}$	A positive verification of B_i generated by participant $\mathfrak{p}_{ID}$.
$V_{i,ID}^{-}$	A negative verification of B_i generated by participant $\mathfrak{p}_{ID}$.
$bstr_{i,ID}$	A string that indicates the verified bit of $V_{i,ID}^{+}$.
$estr_{i,ID}$	A string that indicates the error bit of B_i indicated by $V_{i,ID}^{-}$.
$ORbstr_i$	Bitwise OR of a string set $\{bstr_{i,ID} \mid V_{i,ID}^{+} \in V_i\}$.
msg	The initial mining message of a block.
$B_i.s$	The SMP solution of B_i.
$B_i.D$	The mining difficulty of B_i.
$C_{\mathfrak{p}}$	The credibility value of participant $\mathfrak{p}$.
$D_{\mathfrak{p}}$	The mining difficulty for participant $\mathfrak{p}$.
$\mathcal{E}_*$	A mining event categorized as belonging to category $*$.
Δ_*	The credibility increment caused by a $*$-event.
s_i	The block score of B_i.
$\mathcal{C}_l$	The l^{th} branch of the blockchain.
$s_{\mathcal{C}_l}$	The branch score of $\mathcal{C}_l$.
$* \preceq **$	Chain $**$ is contained in chain $*$.

Next, the participant calculates a solution s of SMP $\mathcal{P}_{ID}(I, D)$, where ID denotes the identity of the participant. In the verification phase, all participants receive solution s and verify in parallel the validity of s by processing $\mathcal{V}_{ID}(\mathcal{P}(I, D), s)$. SMP must have the following properties:

1) **Completeness.** The honest participants can solve puzzle $\mathcal{P}$ in expected runtime t, and the solution obtained honestly can be accepted by other participants with overwhelming probability $1 - \epsilon(n)$.
2) **Time Soundness.** The time-soundness of SMP is parameterized by $\eta_1 \leq 1$ and $\eta_2 \leq 1$. Additionally, SMP is η_1-solving-sound if for every participant it holds that $T_P(\mathcal{P}) \geq \eta_1\ T_S(\mathcal{P})$, where $T_P()$ denotes the parallel execution time of an algorithm, and $T_S()$ denotes the serial execution time of the algorithm. SMP is η_2-verifying-sound if for every participant it holds that $T_P(\mathcal{V}) \leq \eta_2\ T_S(\mathcal{P})$.
3) **Unpredictability and Irreversibility.** Each solution s of SMP $\mathcal{P}$ takes current information I as input, which makes solution s unpredictable and the recorded information I irreversible.
4) **Participant Authentication.** Each solution s includes the identity information ID of a participant that authenticates the participant.

In contrast to a PoW puzzle, SMP is required to be solved serially and be verified in parallel. Intuitively, the SMP mining process is a cycle of two phases: solving and verifying, as shown in fig.3. In the solving phase, participants all serially solve the current SMP and publish an unverified block with the solution. In the verification phase, all participants verify the unverified block in parallel to obtain a complete verification of the received valid block.

1) SOLVING PHASE

The proposed SMP is designed based on the concept of a hash chain. First, the pre-block message is extracted as the initial mining message:

$$M = S_{sk_{i-1}}(pre_{i-1}||R_{i-1}||V_{i-1}), \tag{6}$$

where *hash* denotes a hash function, S denotes a digital signature algorithm, sk_{i-1} denotes a signing key of the pre-block generator, pre_{i-1} denotes the block head hash of pre-block B_{i-1}, R_{i-1} denotes the root of the Merkle tree of recorded transactions in the pre-block, and V_{i-1} denotes the complete verification set of B_{i-1}.[3] Afterwards, the initial mining message of the current block generator is calculated as follows:

$$msg = S_{sk_i}(M)||I, \tag{7}$$

where I denotes the height of the current block. Finally, a mining series $\{a_n\}$ is calculated as follows:

$$a_j = \begin{cases} null & j = 0 \\ a_{j-1}||b_{j-1} & j > 0 \end{cases} \tag{8}$$

$$b_j = Bit(hash(msg||a_j)), \tag{9}$$

where *hash* also denotes the hash function, and *Bit* denotes a random function that takes as input an equal-length string $hash(msg||a_j)$ and outputs bit $b_j \in \{0, 1\}$. Essentially, the solution of such SMP is the first valid a_l such that $hash(msg||a_l) \leq D$, where D denotes the given mining difficulty. The detailed solution algorithm is given in **Algorithm 1**.

[3] A detailed explanation of the complete verification set will be presented in section IV-A2.

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

```
Input: Block Message $msg$; Difficulty $D$
Output: Puzzle Solution: $s$
$s \leftarrow null$
$tmp \leftarrow hash(msg)$
while $tmp \geq D$ do
    $b \leftarrow Bit(tmp)$
    $s \leftarrow s||b$ //i.e. calculate the series $\{a_n\}$
    $tmp \leftarrow hash(msg||s)$
end while
return $s$
```

It is clear that a_j cannot be calculated unless a_{j-1} has been calculated. Thus, **Algorithm 1** is a serial algorithm that cannot be run in parallel. It implies that $T_P(\mathcal{S}) \geq \eta_1 \ T_S(\mathcal{S})$, where $\eta_1 \rightarrow 1$. As a result, SMP provides an effective way to mitigate resource centralization and miner collusion, which become increasingly more significant in existing blockchain systems.

2) VERIFICATION PHASE

In SMP, 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, and

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 $B_i.s$. Intuitively, a weakness of SMP is the significant computational cost of verification. This makes it very different from a traditional PoW puzzle. However, SMP verification can be performed in parallel to improve efficiency. The respective parallel multiparty verification process is given by **Algorithm 2**. In the verification phase, each participant continuously chooses an unverified block from the received block set to verify until a complete verification set of a block has been obtained. Significantly, the verification algorithm can be executed in parallel by multiple participants. Thus, $T_P(\mathcal{V}) \leq \eta_2 \ T_S(\mathcal{P})$, where $\eta_2 \rightarrow 1/n$, and n denotes the expected verification number of a complete verification set. It implies that SMP is time-sound in the verification phase.

For instance, fig. 4 shows in detail the verification process of block B_i that entails the following three phases:

1) The participant extracts a verifying-bit string $bstr_{i,ID}$ based on mining message $B_i.s$ of the unverified block B_i and the identity ID of the participant.

2) The participant verifies each bit b_j of $B_i.s$ in parallel, where j is such that the j^{th} bit of $bstr_{i,ID}$ is set to 1. As shown in fig. 4(b), the verifying participant signs and publishes a positive verification V^{+}_{i,ID_K} iff each specified bit b_j is valid. Otherwise, it signs and publishes an negative verification V^{-}_{i,ID_K} to indicate an invalid bit of $B_i.s$.

3) As shown in fig. 4(c), after publishing verification V^{+}_{i,ID_K}, the participant with identity ID_k persistently receives verifications of B_i produced by others until reaching an all-ones string $ORbstr_i$ that corresponds to a complete verification set V_i. The complete verification set implies that each bit of $B_i.s$ has been verified as valid.

B. MINING CREDIBILITY SYSTEM

In a blockchain, each block includes not only direct *transaction records* but also indirect *credibility records* of participants. This implies that each block indirectly records the mining events that in fact reflect the credibility of participants, especially miners. Thus, MCS is developed to evaluate participant credibility and provide a credibility-based mining difficulty for miners.

1) CREDIBILITY ACCOUNT

In GSCS, a credibility account is introduced to reflect miner credibility. The process of acquiring a credibility account is stricter than that of a coin account (Note that, the coin account of GSCS just like the account of *Bitcoin*, use to traffic electronic currency). Intuitively, a credibility account can be viewed as a coin account bound to a globally unique IP address. Specifically, a credibility account can legally acquire a mining award iff an IP binding certificate of the account has been confirmed by the blockchain. Because a credibility account is uniquely identified by a global IP address, GSCS can also mitigate the witch attack during mining. Because the credibility accounts are allowed to take part in mining competition and maintain the data of blockchain, they should be restricted more strictly. However, in MCS the credibility accounts sacrifice anonymity for credibility, while other coin accounts retain their anonymity. Additionally, in the register step, the credibility account only need to provide a proof to blockchain using a zero-knowledge proof protocol, which proofs that the account has been bound to a *valid* IP address. Thus, the bounded IP address is not published to whole network, it can partly keep anonymity and resist DDoS attack.

2) CREDIBILITY QUANTIFICATION

First, an ideal MCS requires that miner credibility accurately reflect the mining behavior of the participant. Thus, a quantitative representation of miner credibility $C_\mathfrak{p}$ must satisfy the following principles:

1) Range of $C_\mathfrak{p}$: $C_\mathfrak{p} \in [-\infty, +\infty]$. A positive (respectively, negative) value of $C_\mathfrak{p}$ indicates that miner $\mathfrak{p}$ is potentially honest (respectively, dishonest). Additionally, 0 denotes the critical value of $C_\mathfrak{p}$.

2) Initial value: $C_\mathfrak{p}$ of each participant $\mathfrak{p}$ is initialized to 0 because there is little usable information about credibility in the very beginning.

3) Credibility updating: $C_\mathfrak{p}$ can be updated by the specified mining events. This implies that each mining event is associated with a credibility increment.

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

```
Input: An Unverified Block Set $\mathcal{S}_{ID}$
Output: Accepted Block: $B_i \in \mathcal{S}_{ID}$, Verification set of $B_i$: $V_i$
$v \leftarrow 0$
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 //$B_i.s$ denotes the mining message of $B_i$; $B_i.D$ denotes the mining difficulty of $B_i$
        Extract the verifying bit from $B_i$:
        $bstr_{i,ID} \leftarrow Extract(B_i.s, U_{ID})$ // $U_{ID}$ denotes the identity of the 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 $B_i$ broadcast by others
                if receive an 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 has been obtained
                            $v \leftarrow 1$
                        end if
                    end if
                end if
            end while
        end if
    else
        Generate an unsuccessful verification and broadcast
        $V^-_{i,ID} = (S_{ID}(estr_{i,ID}), B_i.s)$ // $estr_{i,ID}$ indicates an invalid bit
        $\mathcal{S}_{ID} \leftarrow \mathcal{S}_{ID} - \{B\}$
    end if
end while
return $B_i$, $V_i$
```

4) Credibility transaction: $C_{\mathfrak{p}}$ cannot be transacted at all. Only mining events are permanently recorded in the blockchains, and can be viewed as *credibility transactions*.

Furthermore, six categories of mining events are associated with increments of participant credibility $C_{\mathfrak{p}}$.

(1) $\mathcal{E}_i$, inserting a new block into the chain. The corresponding credibility increment of $\mathcal{E}_i$ is calculated as follows:

$$\Delta_i = \alpha(1 - e^{-\lambda_i A(\mathcal{E}_i)}), \tag{10}$$

where α is a positive constant that represents the upper bound of the increment, λ_i is also a positive constant that represents the rising tendency of the increment, and $A(\mathcal{E}_i)$ denotes the transaction amount confirmed with $\mathcal{E}_i$. The reasoning is that a larger $A(\mathcal{E}_i)$ corresponds to a greater contribution of $\mathcal{E}_i$.

(2) $\mathcal{E}_s$, contributing a verification of an unverified block. As miner $\mathfrak{p}$ successfully submits a verification to the blockchain, the increment of $C_{\mathfrak{p}}$ is calculated as

$$\Delta_s = \beta(1 - e^{-\lambda_s L_b(\mathcal{E}_s)}), \tag{11}$$

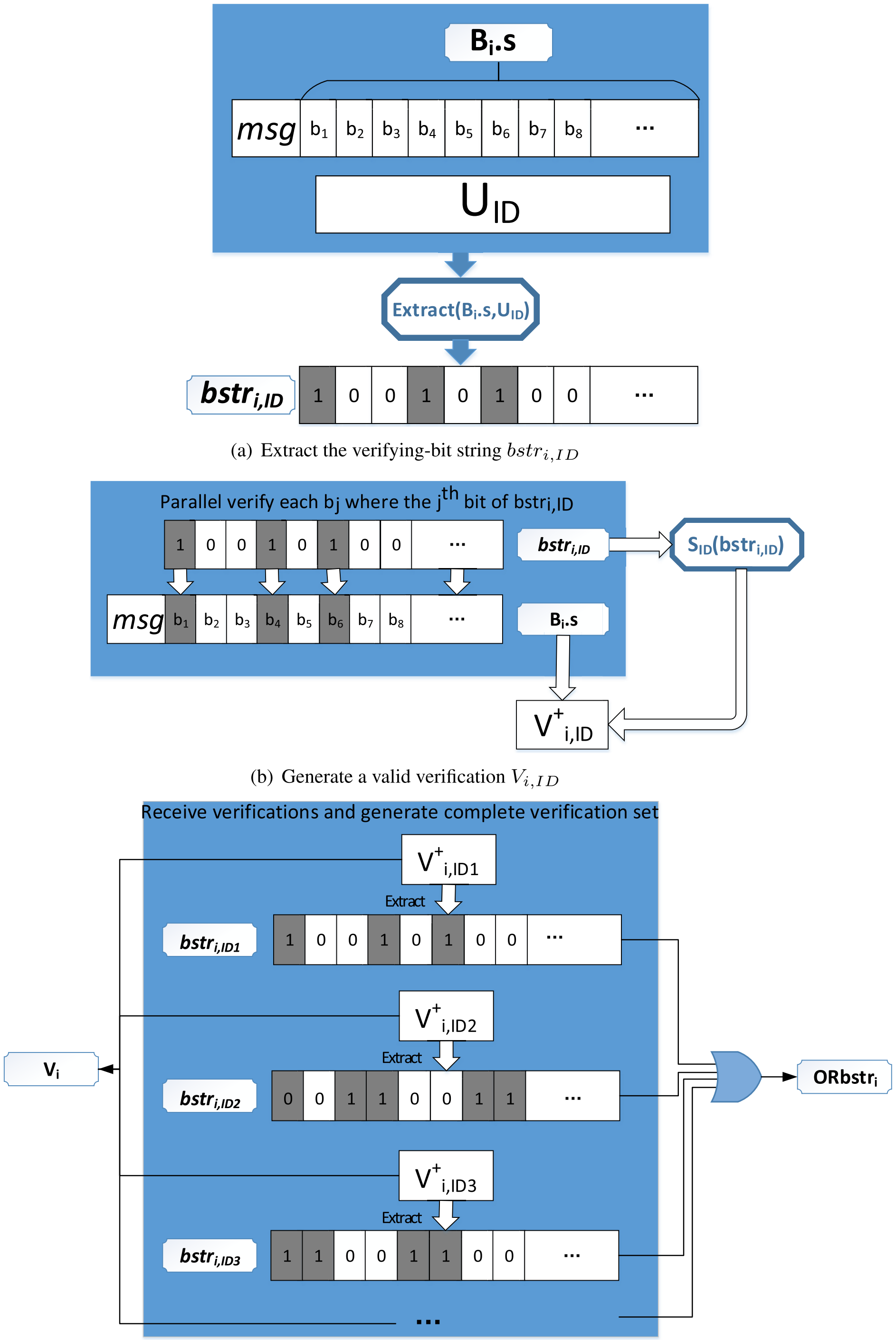
where $\beta > 0$ is the upper bound of increment Δ_s, $\lambda_s > 0$ controls the rising tendency of Δ_s, and $L_b(\mathcal{E}_s)$ is the verified bit number indicated in the verification. This implies that the contribution of $\mathcal{E}_s$ is accurately reflected by variable $L_b(\mathcal{E}_s)$.

(3) $\mathcal{E}_d$, detecting an invalid block with an invalid bit. Let Δ_d denote the increment of $C_{\mathfrak{p}}$, while $\mathfrak{p}$ detects an invalid block. Thus,

$$\Delta_d = \gamma(1 - e^{-\lambda_d L_v(\mathcal{E}_d)}), \tag{12}$$

where $\gamma > 0$ is the upper bound of Δ_d, λ_d is the rising tendency of Δ_d, and $L_v(\mathcal{E}_d)$ is the length of detected mining data.

(4) $\mathcal{E}_c$, creating an invalid block. In this case, the increment Δ_c of $C_{\mathfrak{p}}$ is produced iff the invalid block has been published



(a) Extract the verifying-bit string $bstr_{i,ID}$

(b) Generate a valid verification $V_{i,ID}$

(c) Generate a complete verification set V_i

FIGURE 4. Verification of a serial mining puzzle.

and verified as a forged block. Furthermore, the increment is calculated as follows:

$$\Delta_c = min\{-\eta e^{\lambda_c C_p}, T\}, \tag{13}$$

where $T < 0$ denotes the upper bound of Δ_c, and η and λ_c denote two positive parameters that affect increment Δ_c. Significantly, equation 13 implies that the punishment incurred by more credible participants is more significant.

(5) $\mathcal{E}_a$, accepting an invalid block. Event $\mathcal{E}_a$ represents that miner p has published a block with an invalid pre-block. This implies that p has accepted an incomplete or forged verification. Thus, the increment is computed as follows:

$$\Delta_a = -\rho e^{-\lambda_a L_v(\mathcal{E}_a)}, \tag{14}$$

where $\rho < 0$ is used to represent the lower bound of Δ_a, $\lambda_a > 0$ is a constant that controls the rising tendency of Δ_a, and $L_v(\mathcal{E}_a)$ denotes the length of mining data of the pre-block.

(6) $\mathcal{E}_p$, publishing blocks or verifications with similar block heights in different forks. The reason is that $\mathcal{E}_p$ will produce a significant forking issue if a miner engages in mining with different forks in parallel. Thus, such dishonest behavior would result in the following credibility increment:

$$\Delta_p = -\tau e^{\lambda_p L_{\mathfrak{p}}(\mathcal{E}_p)}, \tag{15}$$

where τ is a positive constant coefficient, λ_a is used to control the rising tendency, and $L_{\mathfrak{p}}(\mathcal{E}_p)$ denotes the total length of such blocks or verifications of different forks.

Additionally, in MCS the influence of a mining event must decay over time. Thus, it is reasonable for each credibility increment to involve a multiplication by an *exponential time decay factor* $e^{-\lambda_t T}$, where $\lambda_t > 0$ is an assigned constant, and T denotes the height difference between the mining event's record block and the current block. Thus, the final expression of $C_{\mathfrak{p}}$ is as follows:

$$\begin{aligned} C_{\mathfrak{p}} = &\sum_{\mathcal{E}_i \in S_i} e^{-\lambda_t T_i}\Delta_s + \sum_{\mathcal{E}_s \in S_s} e^{-\lambda_t T_s}\Delta_s + \sum_{\mathcal{E}_d \in S_d} e^{-\lambda_t T_d}\Delta_d \\ &+ \sum_{\mathcal{E}_c \in S_c} e^{-\lambda_t T_c}\Delta_c + \sum_{\mathcal{E}_a \in S_a} e^{-\lambda_t T_a}\Delta_a + \sum_{\mathcal{E}_p \in S_p} e^{-\lambda_t T_p}\Delta_p, \end{aligned} \tag{16}$$

For convenience, equation 16 can be simplified as follows:

$$C_{\mathfrak{p}} = e^{-\lambda_t \Delta T}\bar{C}_{\mathfrak{p}} + \Delta_*, \tag{17}$$

where ΔT denotes the height difference between the current block and the previous block that records the last update of $C_{\mathfrak{p}}$, $\bar{C}_{\mathfrak{p}}$ denotes the last balance of $C_{\mathfrak{p}}$, and Δ_* denotes the current increment of $C_{\mathfrak{p}}$.

3) CREDIBILITY GRADING

Let credibility increments of a miner be a sequence of independent random variables $\Delta_1, \Delta_2, \ldots, \Delta_n$ such that each increment Δ_k has mean μ_k and variance σ_k^2. It is reasonable to assume that $\exists\delta > 0$ such that the Lyapunov's condition holds:

$$\lim_{n\to\infty} \frac{1}{B_n^{2+\delta}} \sum_{k=1}^{n} E|\Delta_k - \mu_k|^{2+\delta} \to 0, \tag{18}$$

where $B_n^2 = \sum_{k=1}^n \sigma_k^2$. According to Lyapunov's central limit theorem,[4] the distribution of $C_{\mathfrak{p}} = \sum_{k=1}^n \Delta_k$ tends to be normal, $N(\mu, \sigma^2)$, where $\mu = \sum_{k=1}^n \mu_k$ and $\sigma^2 = \sum_{k=1}^n \sigma_k^2$. Thus, for $x \in [-\infty, +\infty]$, we obtain

$$F(x) = \lim_{n\to\infty} P\{C_{\mathfrak{p}} \le x\} = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t-\mu)^2}{2\sigma^2}} dt, \tag{19}$$

Then, we assume that $A(\mathcal{E}_i) \sim P(\lambda_i), L_b(\mathcal{E}_s) \sim P(\lambda_s), L_v(\mathcal{E}_d) \sim P(\lambda_i), L_v(\mathcal{E}_a) \sim P(\lambda_a), L_p(\mathcal{E}_p) \sim P(\lambda_p)$, $T \sim P(\lambda_t)$. Considering the probability distribution and influence of each kind of mining events, the constant parameters must satisfy following inequality:

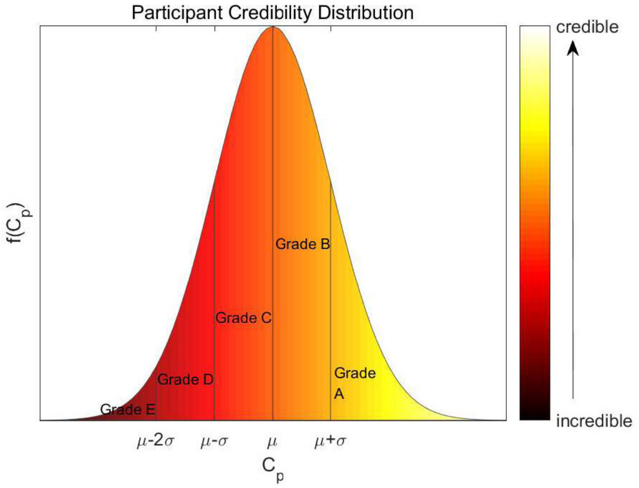


FIGURE 5. Grading of credibility $C_{\mathfrak{p}}$.

$$\lambda_t > \lambda_s > \lambda_i > \lambda_d > \lambda_p > \lambda_c > \lambda_a, \tag{20}$$

$$\eta > \tau > \rho > \alpha > \gamma > \beta. \tag{21}$$

Furthermore, we suppose that

$$\mu = E(e^{-\lambda_t T})E(\Delta_i + \Delta_s + \Delta_d + \Delta_c + \Delta_a + \Delta_p), \tag{22}$$

$$\delta = E(e^{-2\lambda_t T})E((\Delta_i + \Delta_s + \Delta_d + \Delta_c + \Delta_a + \Delta_p)^2) - \mu^2. \tag{23}$$

Additionally, assume that credibility $C_{\mathfrak{p}}$ of a miner tends to be normally distributed as $N(\mu, \sigma^2)$. Then, as shown in fig. 5, $C_{\mathfrak{p}}$ is graded based on its probability density function:

1) Grade A: $C_{\mathfrak{p}} \in (\mu + \sigma, +\infty)$. In this case, $\mathfrak{p}$ is regarded as a completely credible participant.

2) Grade B: $C_{\mathfrak{p}} \in (\mu, \mu + \sigma]$. In this case, the credibility grade represents that $\mathfrak{p}$ is a mostly credible participant.

3) Grade C: $C_{\mathfrak{p}} \in (\mu - \sigma, \mu]$. In this case, grade C shows that participant $\mathfrak{p}$ is not credible.

4) Grade D: $C_{\mathfrak{p}} \in (\mu - 2\sigma, \mu - \sigma]$. The credibility of $\mathfrak{p}$ is considered to be poor in this case.

5) Grade E: $C_{\mathfrak{p}} \in (-\infty, \mu - 2\sigma]$. In this case, $\mathfrak{p}$ is regarded as entirely discredited.

4) CREDIBILITY-BASED MINING DIFFICULTY

Credibility grades indicate various categories of participant mining behavior. Thus, it is reasonable to evaluate the current or future mining behavior of a miner by the respective credibility grade. To encourage credible miners and penalize discredited miners, GSCS introduces a personalized mining difficulty based on credibility of miners.

In our SMP, the solution is the first element a_l of series $\{a_n\}$ that satisfies $hash(a_l) < D$. Function $hash(\cdot)$ takes a message of arbitrary length as input and calculates a fixed-length value $h \in \{0, 1\}^n$ as output. Probability $Pr[hash(a_l) < D]$ is approximately equal to $D/2^n$, while

[4] A detailed description of Lyapunov's central limit theorem is given in [56].

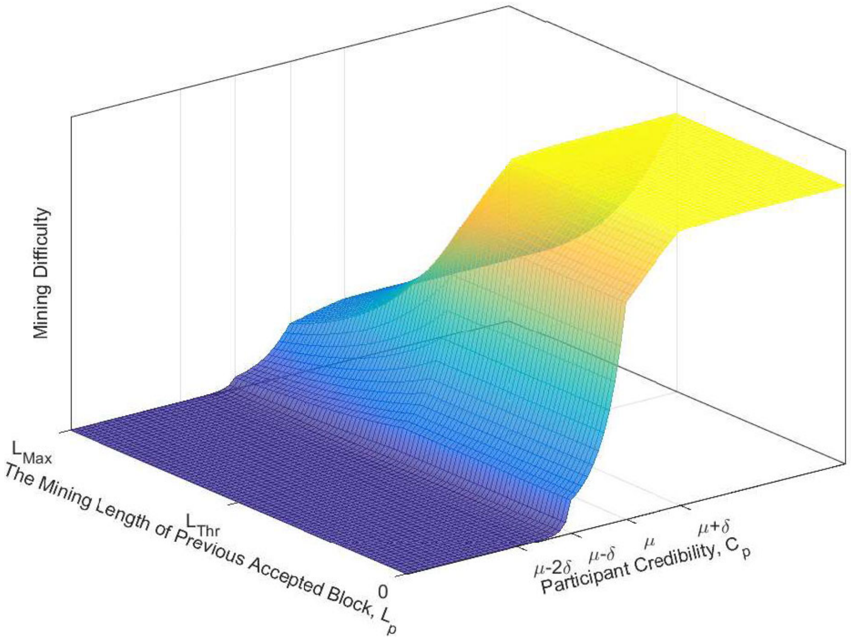


FIGURE 6. Personalized mining-difficulty $D_\mathfrak{p}$ based on participant credibility $C_\mathfrak{p}$.

function $hash(\cdot)$ has the ideal *one-way property*. Thus, a smaller D implies a greater mining difficulty. Furthermore, the ideal personalized mining difficulty $D_\mathfrak{p}$ of participant $\mathfrak{p}$ must have the following properties:

1) The value of $D_\mathfrak{p}$ depends significantly on the distribution and grade of $C_\mathfrak{p}$. Furthermore, $\mathfrak{p}$ can no longer mine iff $C_\mathfrak{p}$ is lower than a threshold.

2) The serial computational power of a participant must be considered as an important factor affecting $D_\mathfrak{p}$. Significantly, the factor can be described as the mining message length of the last block produced by $\mathfrak{p}$.

3) Function $D_\mathfrak{p}$ is continuous in interval $C_\mathfrak{p} \in (-\infty, +\infty)$.

Thus, let $D_\mathfrak{p}$ be calculated by the following piecewise function:

$$D_\mathfrak{p} = \begin{cases} 2^{\lfloor(\theta_a+\delta_a F(C_\mathfrak{p}))-\lambda\Delta_L\rceil}, & C_\mathfrak{p} \in (\mu+\sigma, +\infty) \\ 2^{\lfloor(\theta_b+\delta_b F(C_\mathfrak{p}))-\lambda\Delta_L\rceil}, & C_\mathfrak{p} \in (\mu, \mu+\sigma] \\ 2^{\lfloor(\theta_c+\delta_c F(C_\mathfrak{p}))-\lambda\Delta_L\rceil}, & C_\mathfrak{p} \in (\mu-\sigma, \mu] \\ 2^{\lfloor(\theta_d+\delta_d F(C_\mathfrak{p}))-\lambda\Delta_L\rceil}, & C_\mathfrak{p} \in (\mu-2\sigma, \mu-\sigma] \\ 0, & C_\mathfrak{p} \in (-\infty, \mu+2\sigma], \end{cases} \tag{24}$$

where $F(C_\mathfrak{p})$ denotes the cumulative probability function of $C_\mathfrak{p}$, $\Delta_L = \max\{L_\mathfrak{p} - L_{Thr}, 0\}$, $L_\mathfrak{p}$ denotes the length of the pre-block mining data produced by $\mathfrak{p}$, and L_{Thr} denotes a threshold. Furthermore, $\theta_a, \theta_b, \theta_c, \theta_d, \delta_a, \delta_b, \delta_c, \delta_d$ denote the constants that satisfy the following equations:

$$\theta_a = \theta_b + F(\mu+\sigma)(\delta_b - \delta_a), \tag{25}$$

$$\theta_b = \theta_c + F(\mu)(\delta_c - \delta_b), \tag{26}$$

$$\theta_c = \theta_d + F(\mu-\sigma)(\delta_d - \delta_c), \tag{27}$$

$$\theta_d = -\delta_d F(\mu - 2\sigma), \tag{28}$$

$$0 < \delta_a < \delta_b \leq 1 < \delta_c < \delta_d. \tag{29}$$

Intuitively, fig.6 shows how personalized mining difficulty $D_\mathfrak{p}$ increases with $C_\mathfrak{p}$ and $L_\mathfrak{p}$. First, in different credibility grade intervals $D_\mathfrak{p}$ exhibits different rising trends with respect to $C_\mathfrak{p}$:

1) For $C_\mathfrak{p} \in (\mu+\delta, +\infty)$, we can reach the upper bound of $D_\mathfrak{p}$:

$$\lim_{C_\mathfrak{p}\to\infty} D_\mathfrak{p} = 2^{\theta_a+\delta_a-max\{L_\mathfrak{p},0\}}. \tag{30}$$

2) To rapidly distinguish miners who are credible and those who are not, constants δ_b, δ_c should be large enough if $C_\mathfrak{p} \in (\mu-\sigma, \mu+\sigma]$.

3) The probability of solving SMP is negligible if $C_\mathfrak{p} \in (\mu-2\sigma, \mu-\sigma]$ because $D_\mathfrak{p}$ is sufficiently small for miner $\mathfrak{p}$ in this case.

4) It is impossible for miner $\mathfrak{p}$ to solve SMP if $C_\mathfrak{p} \in (-\infty, \mu-2\sigma]$; because of that, $D_\mathfrak{p} = 0$ in this case.

Second, $D_\mathfrak{p}$ decreases exponentially while $L_\mathfrak{p}$ is greater than threshold L_{Thr}. In fact, $L_\mathfrak{p}$ can partly reflect the serial computational power of $\mathfrak{p}$. Thus, $D_\mathfrak{p}$ should be reduced while participant $\mathfrak{p}$ is provided with a significant advantage of serial computational power. In brief, the mining difficulty given by equation (24) completely satisfies the given two conditions.

C. BRANCH SELECTION STRATEGY

Miners can follow a novel strategy to select a branch from multiple received branches to extend in GSCS. First, each block B_i is assigned a credibility-based score s_i calculated as

$$s_i = F(C_\mathfrak{p})e^{-\max\{0, L_i - L_{Thr}\}}, \tag{31}$$

where $F(\cdot)$ denotes the cumulative probability function of participant credibility, $C_\mathfrak{p}$ denotes the credibility of the participant who generated block B_i, L_i denotes the length of mining

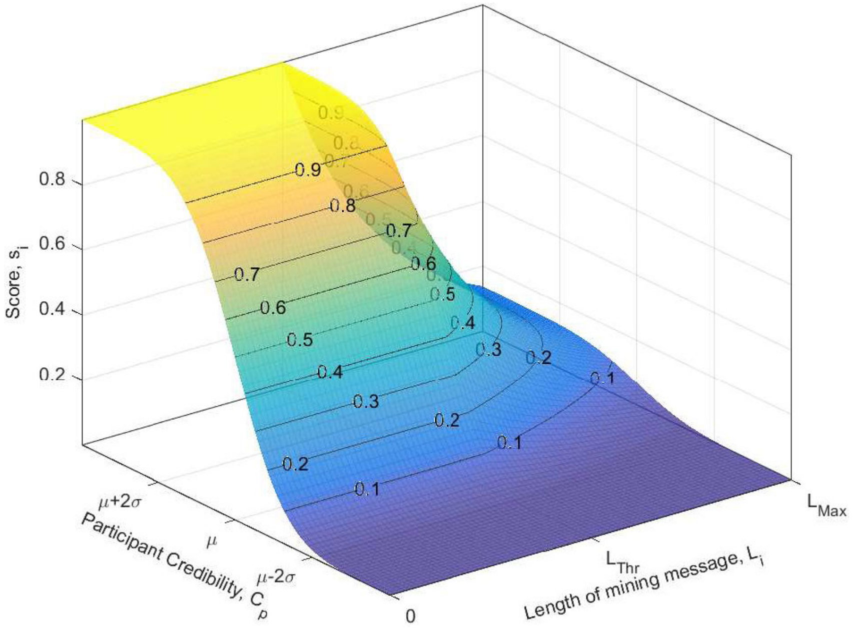


FIGURE 7. **Block score.**

data of B_i, and L_{Thr} denotes a specified threshold. As shown in fig. 7, parameters C_{p}, L_i have different influences on s_i (note that the range of a block score is (0, 1)). Then, the chain score of a fork can be defined as follows:

$$s_{\mathcal{C}} = \sum_{B_i \in \mathcal{C}} s_i. \tag{32}$$

Because the largest chain score implies the best chain quality, a participant accepts the fork with the largest score instead of the longest fork in GSCS. This is different from the strategy of a PoW blockchain.

Without a loss of generality, $s_{\mathcal{C}}$ can be viewed as a weighted sum of chain lengths. Each weight depends on miner credibility and *serial computing power*. Following such fork selection strategy, a miner tends to accept the longest chain while generators of each branch gain similar credibility values. However, a fork generated by credible miners always attains a high score while all branches reach similar lengths. In brief, the proposed strategy considers not only chain length but also miner credibility.

D. GSCS-BASED BLOCKCHAIN PROTOCOL

An overview of GSCS protocol is presented in algorithm 3. First, chain $\mathcal{C}$ is initialized with a genesis block. Second, in each round miners receive a set of chain branches $\mathcal{S}_{\mathcal{C}}$ from the entire network. Furthermore, branches of $\mathcal{S}_{\mathcal{C}}$ are all sorted by branch score. Third, miners invoke algorithm 2 to verify the validity of current block B_i and obtain the corresponding verification set V_i. Finally, miners invoke algorithm 1 to mine new block B_{i+1} of the current branch with the updated data. The new block B_{i+1} is broadcast to the network, which implies that the current round of the GSCS protocol has been completed, and the next round of GSCS will begin.

V. PERFORMANCE ANALYSIS AND EVALUATION

A. COMMON PREFIX PROPERTY

GSCS essentially rejects miners that mine with a different branch in a temporal interval. In contrast, in a PoW blockchain a miner is allowed to mine on different branches at the same time. Additionally, in some cases a miner may mine on multiple branches to mitigate the risk of mining award competition. However, in GSCS a miner is punished with reduced credibility if the miner publishes blocks or verifications of different branches with similar heights. Thus, it is reasonable to assume that honest miners will only mine on one fork, and the adversary only publishes blocks/verifications on one branch in a temporal interval. Let $\mathcal{C}_i^r$ and $\mathcal{C}_i^r$ be the chains of two honest miners i, j in a given round r, and k^* be the minimum integer such that $\mathcal{C}_i^r[0, r-k^*] \preceq \mathcal{C}_j^r$ and $\mathcal{C}_j^r[0, r-k^*] \preceq \mathcal{C}_i^r$. Assume that all the last k^* blocks of $\mathcal{C}_i^r$ and $\mathcal{C}_j^r$ are generated in l rounds. The block number of such branches $2k^*$ cannot be larger than the solution number X obtained by all miners in l rounds. Furthermore, let l denote the minimum number of rounds that a participant is allowed to mine for different branches without punishment; $\mathcal{H}$ and $\mathcal{A}$ denote the sets of honest miners and adversaries, respectively. Let $X_{i,k}$ denote a Boolean random variable such that $X_{i,k} = 1$ iff there is a solution produced for $\mathcal{C}_1$ or $\mathcal{C}_2$ by miner i in the last $(l-k)^{th}$ round. Thus, probability $P[X_{i,k} = 1]$ is calculated as follows:

$$P[X_{i,k}=1] = \begin{cases} p_i, \quad i \in \mathcal{H} \\ \bar{p}_i^{2(k-1)}(1-\bar{p}_i^2) + (1-\bar{p}_i^{2(k-1)})p_i, \\ \qquad i \in \mathcal{A}, k < l \\ \bar{p}_i^{2(l-1)}(1-\bar{p}_i^2) + (1-\bar{p}_i^{2(l-1)})p_i, \\ \qquad i \in \mathcal{A}, k \geq l, \end{cases} \tag{33}$$

Algorithm 3 GSCS Protocol Π

```
Initialize: $\mathcal{C} \leftarrow B_0$ // $\mathcal{C}$ denotes the current chain, and $B_0$ denotes the genesis block of $\mathcal{C}$
while True do
    Upon receiving chain branch set $\mathcal{S}_{\mathcal{C}}$ and unrecorded information,
    Extract unverified block set $\mathcal{S}_{ID}$ that includes the last block $\mathcal{B}_i$ of each branch $\mathcal{C}_i \in \mathcal{S}_{\mathcal{C}}$
    Choose and verify block $B_i \in \mathcal{S}_{ID}$, following branch scores $s_{\mathcal{C}_i}$ in the descending order:
    $(B_i, V_i) \leftarrow \mathcal{V}(\mathcal{S}_{ID})$
    $\mathcal{C} \leftarrow \mathcal{C}_i$
    Update mining difficulty $D$, following the current chain $\mathcal{C}$
    Calculate the initial mining message $msg$
    Solve the current mining puzzle
    $s_{i+1} \leftarrow \mathcal{S}(msg, D)$
    Generate a new block $B_{i+1}$ of $\mathcal{C}$
    $\mathcal{C} \leftarrow \mathcal{C}||B_{i+1}$
    Broadcast the current chain $\mathcal{C}$
end while
```

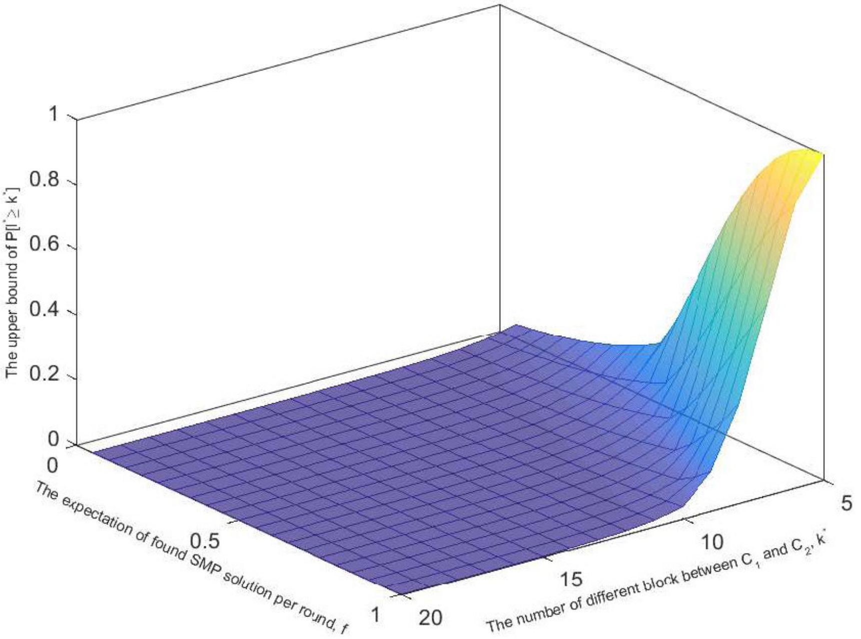


FIGURE 8. Upper bound of probability $P[l^* \geq k^*]$.

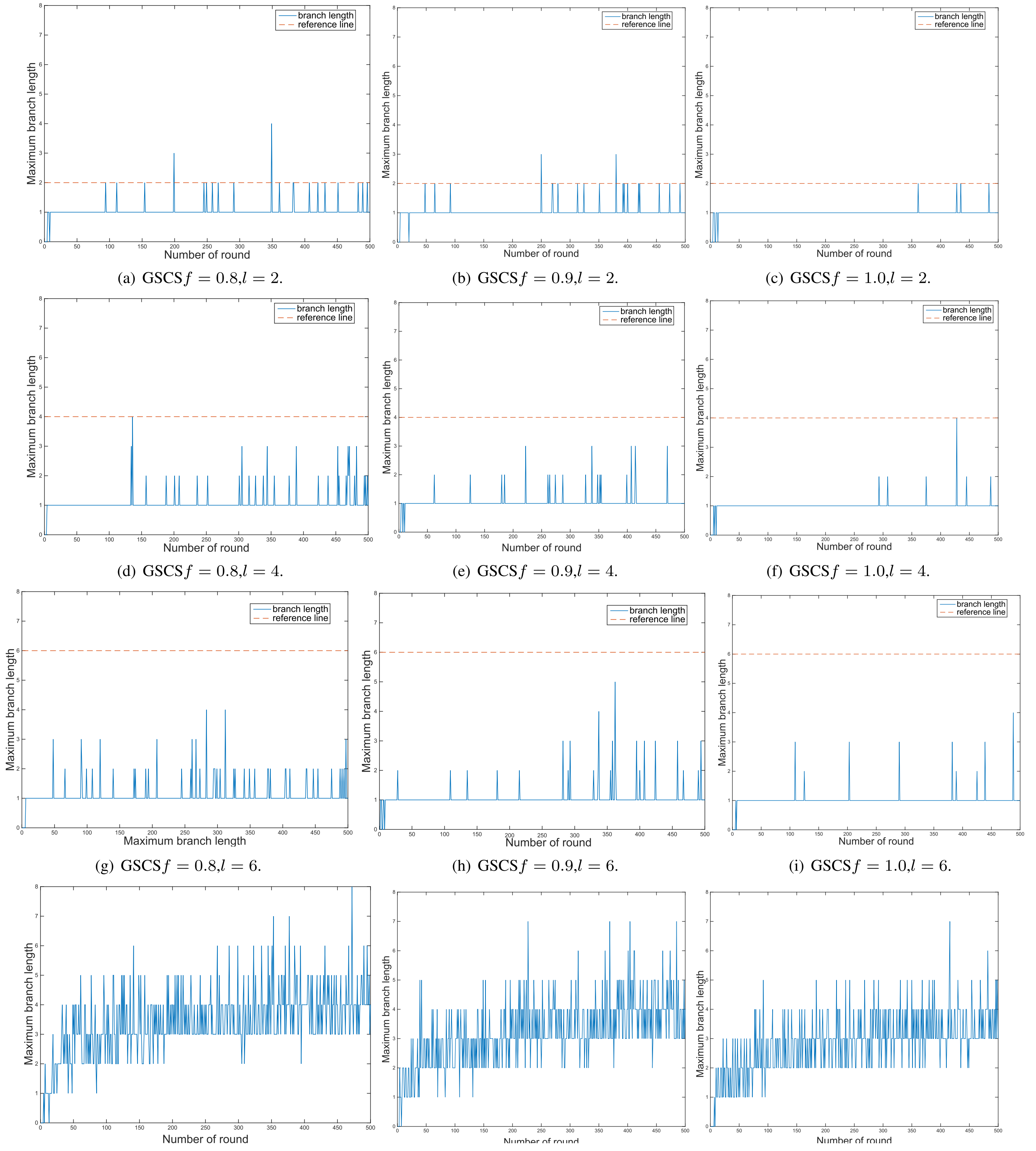
where $\bar{p}_i = (1 - D_{\mathfrak{p}_i}/N)^q$, $p_i = 1 - \bar{p}_i$, and q denotes the number of times a miner can run the hash function in a round. Thus, the expectation of random variable $X = \sum_{i \in \mathcal{H} \cup \mathcal{A}} \sum_{k=1}^{l} X_{i,k}$ can be calculated as follows:

$$\begin{aligned}
\mu_1 &= \sum_{i \in \mathcal{H} \cup \mathcal{A}} \sum_{k=1}^{l} E(X_{i,k}) \\
&= \sum_{i \in \mathcal{H}} \sum_{k=1}^{l} p_i + \sum_{i \in \mathcal{A}} \sum_{k=1}^{l} E(X_{i,k}) \\
&\geq \sum_{i \in \mathcal{H}} \sum_{k=1}^{l} p_i + \sum_{i \in \mathcal{A}} \sum_{k=1}^{l} p_i^{2(k-1)}(1 - \bar{p}_i^2) + (1 - \bar{p}_i^{2(k-1)}) p_i
\end{aligned}$$

$$\begin{aligned}
&= \sum_{i \in \mathcal{H}} \sum_{k=1}^{l} p_i + \sum_{i \in \mathcal{A}} \sum_{k=1}^{l} p_i + \sum_{i \in \mathcal{A}} \sum_{k=1}^{l} p_i \bar{p}_i^{2k-1} \\
&= lS_h + lS_a + \sum_{j \in \mathcal{A}} \bar{p}_i \frac{1 - \bar{p}_i^{2l}}{1 + \bar{p}_i} \\
&\geq lS_h + lS_a \\
&\geq lf, \qquad (34)
\end{aligned}$$

where $S_h = \sum_{i \in \mathcal{H}} p_i$, $S_a = \sum_{i \in \mathcal{A}} p_i$ and $f = S_a + S_h$ are the expected numbers of solutions found by honest miners, adversarial participants and all miners, respectively, in a round. The total growing branch length l^* during the last l rounds cannot be greater than random variable X. Thus,

$$P[l^* \geq k^*] \leq P[X \geq 2k^*]. \qquad (35)$$



(a) GSCS $f = 0.8, l = 2$.

(b) GSCS $f = 0.9, l = 2$.

(c) GSCS $f = 1.0, l = 2$.

(d) GSCS $f = 0.8, l = 4$.

(e) GSCS $f = 0.9, l = 4$.

(f) GSCS $f = 1.0, l = 4$.

(g) GSCS $f = 0.8, l = 6$.

(h) GSCS $f = 0.9, l = 6$.

(i) GSCS $f = 1.0, l = 6$.

(j) PoW $f = 0.8$.

(k) PoW $f = 0.9$.

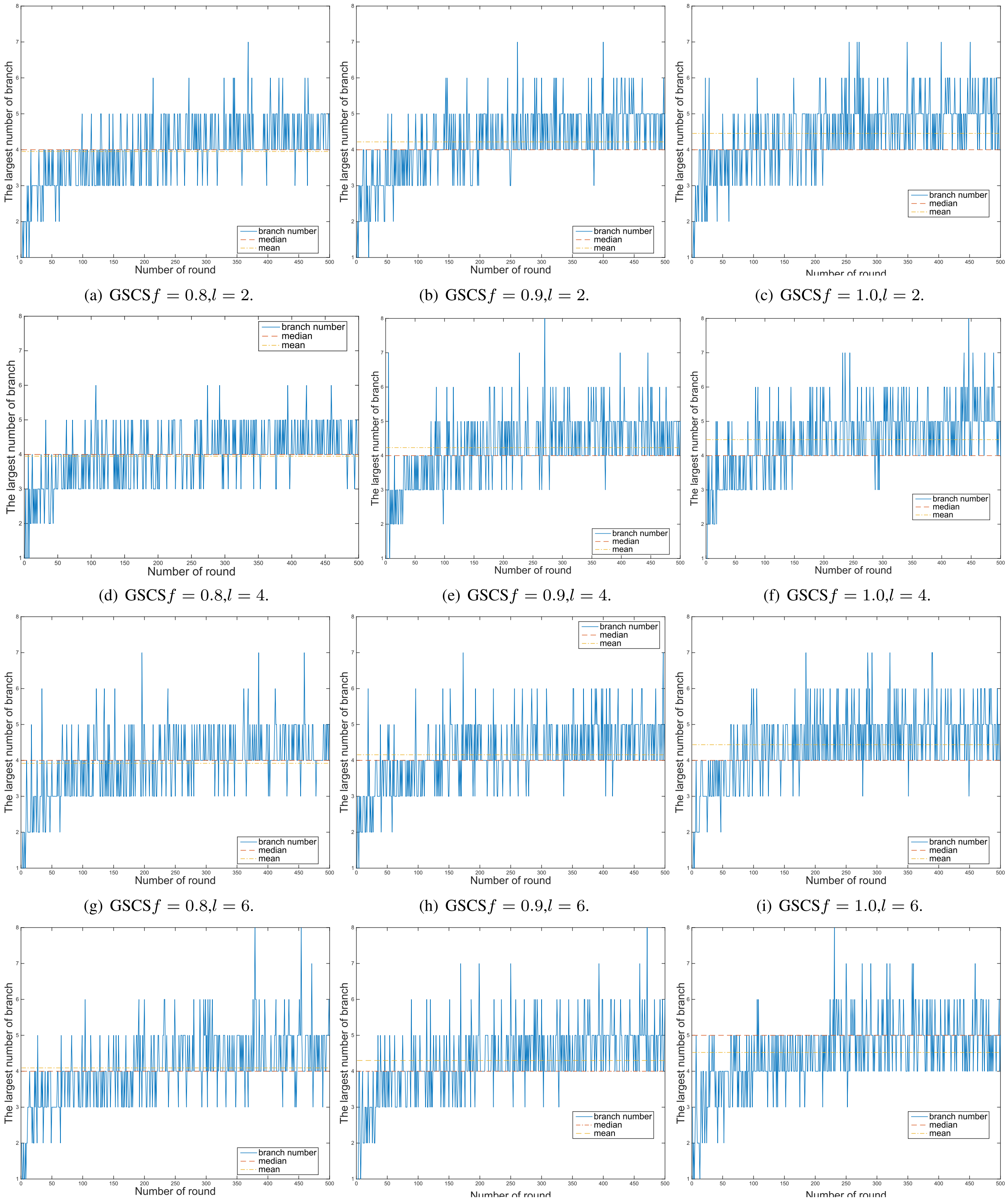
(l) PoW $f = 1.0$.

FIGURE 9. **Growth trend of the branches length at different *f* and *l*.**

By Chernoff's bound,

$$\begin{cases} P[l^* \geq k^*] \leq P[X \geq (1+\delta)\mu_1] \leq e^{-\frac{\delta^2 \mu_1}{3}}, & 0 < \delta \leq 1 \\ P[l^* \geq k^*] \leq P[X \geq (1+\delta)\mu_1] \leq e^{-\frac{\delta \mu_1}{3}}, & 1 < \delta, \end{cases} \tag{36}$$

where $(1+\delta)\mu_1 \geq (1+\delta)lf = 2k^*$. Fig. 8 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 probability $P[l^* \geq k^*]$ declines exponentially with k^* and f. Thus, parameter l can be set based on a negligible constant ε_{cp} and parameter $\delta \in (0, +\infty)$



(a) GSCS $f = 0.8, l = 2$.

(b) GSCS $f = 0.9, l = 2$.

(c) GSCS $f = 1.0, l = 2$.

(d) GSCS $f = 0.8, l = 4$.

(e) GSCS $f = 0.9, l = 4$.

(f) GSCS $f = 1.0, l = 4$.

(g) GSCS $f = 0.8, l = 6$.

(h) GSCS $f = 0.9, l = 6$.

(i) GSCS $f = 1.0, l = 6$.

(j) PoW $f = 0.8$.

(k) PoW $f = 0.9$.

(l) PoW $f = 1.0$.

FIGURE 10. Growth trend of the of branches number at different *f* and *l*.

as follows:

$$l = \begin{cases} \dfrac{3\ln(\varepsilon_{cp}^{-1})}{\delta^2 f}, & \delta \in (0, 1] \\ \dfrac{3\ln(\varepsilon_{cp}^{-1})}{\delta f}, & \delta \in (0, +\infty). \end{cases} \tag{37}$$

For a more detailed analysis of a given GSCS, we simulate the GSCS and PoW, evaluate their performance at various parameter values in different parameters f and l. In the simulation, we set the average mining difficulty of miners to be 0.05 and the adversary ratio $\rho = 0.15$. Furthermore,

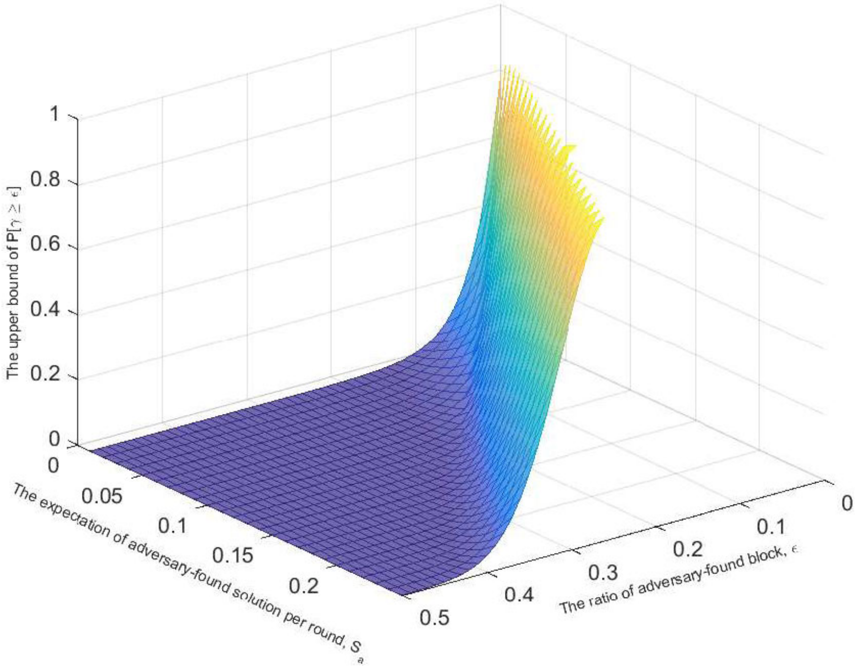
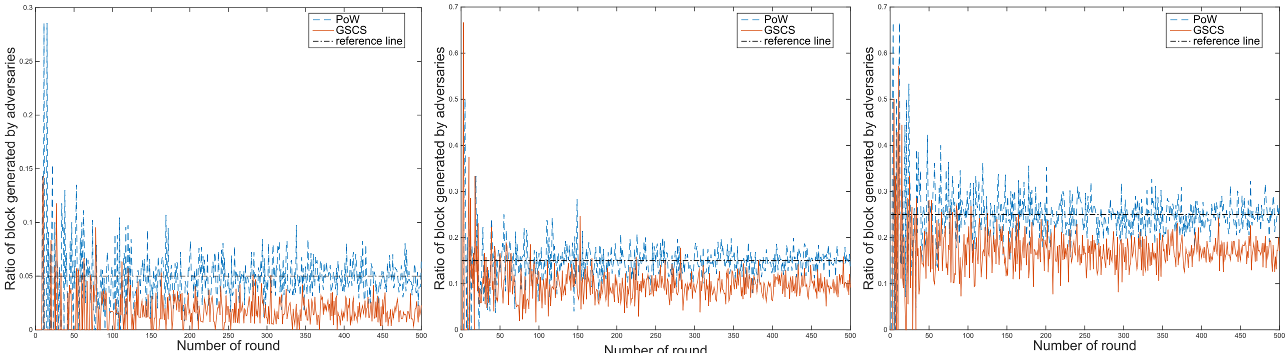


FIGURE 11. Upper bound of probability $P[\gamma \geq \epsilon]$.



(a) $S_h = 0.95, S_a = 0.05$.

(b) $S_h = 0.85, S_a = 0.15$.

(c) $S_h = 0.75, S_a = 0.25$.

FIGURE 12. Chain quality property of GSCS at different *h* and *a*.

the average verification number of a complete verification set is 5. It implies that a branch can growth iff there are more than 5 miners verify its last block. The simulation results are shown in fig.9. The figures show the growth trend of the longest branch length L_b as the round number is increasing, $f = 0.8, 0.9, 1.0$ and $l = 2, 4, 6$. It is clear that L_b is particularly magnified as l and f increases. However, the influence of f is smaller than that of l. Significantly, in each case, L_b is hard to exceed the value of l. Meanwhile, overall, the branch length of GSCS is much less than PoW. It is caused by the improved branch selection strategy. There is a negligible probability of two branches get equal score. Thus, the honest can be split into different branches with negligible probability.

Another quantity that can be used to evaluate the common prefix property is N_b, which denotes the largest valid branch number per round. Fig.10 shows that N_b grows as rounds continue for $f = 0.8, 0.9, 1.0$ and $l = 2, 4, 6$. It is clear that N_b is heavily dependent on f, but the influence of l is limited. Intuitional, the N_b of GSCS always less than PoW. Because a branch cannot grow without a complete verification set.

In brief, fig9 and fig.10 shows that L_b and N_b of GSCS and PoW are significantly dependent on l and f, respectively. However, in each case, GSCS get smaller L_b and N_b than PoW. It implies GSCS get stronger common prefix property.

B. CHAIN QUALITY PROPERTY

Let random variable $Y = \sum_{i \in \mathcal{A}} \sum_{k=1}^{L} X_{i,k}$ denote the number of solutions of chain $\mathcal{C}$ found by an adversary in L rounds; the expectation of Y can be calculated as follows:

$$\begin{aligned} \mu_2 &= \sum_{i \in \mathcal{A}} \sum_{k=1}^{L} E(X_{i,k}) \\ &= \sum_{j \in \mathcal{A}} L p_j \\ &= L S_a. \end{aligned} \tag{38}$$

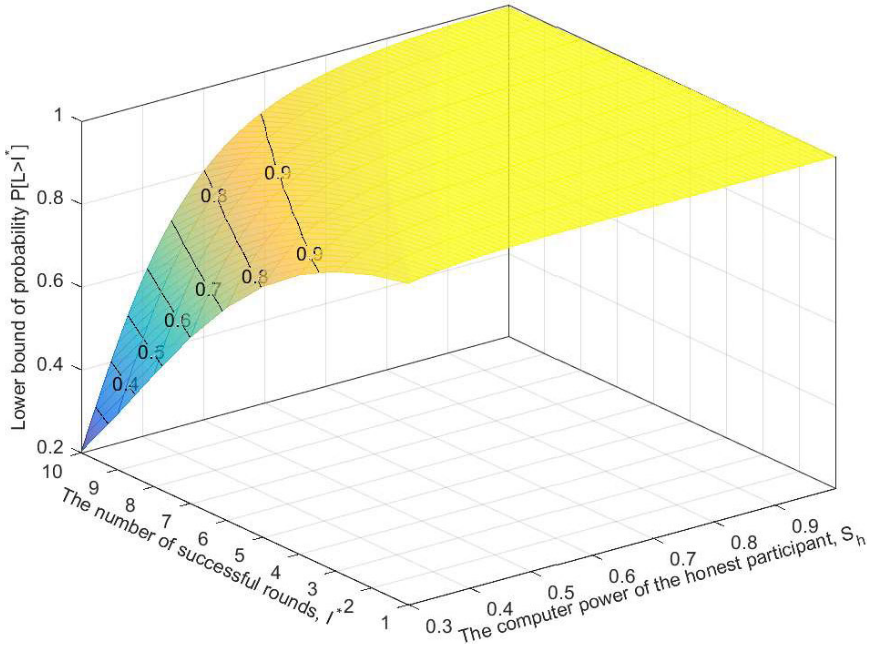


FIGURE 13. **Upper bound of probability $P[\mathcal{L}_l \ge l^*]$.**

Furthermore, let γ be the ratio of the number of adversary-provided blocks and the length of a continuous part of chain $\mathcal{C}$ produced in L rounds. It is clear that $P[\gamma \ge \epsilon] \le P[Y \ge \epsilon Lf]$. Similarly, using the Chernoff's bound, we can obtain:

$$\begin{cases} P[\gamma \ge \epsilon] \le P[Y > (1+\delta_a)\mu_2] \le e^{-\frac{\delta^2\mu_2}{3}}, & 0 < \delta \le 1 \\ P[\gamma \ge \epsilon] \le P[Y > (1+\delta_a)\mu_2] \le e^{-\frac{\delta\mu_2}{3}}, & 1 < \delta, \end{cases} \tag{39}$$

where $(1+\delta)\mu_2 = (1+\delta)LS_a = \epsilon Lf$. Fig. 11 shows the upper bound of $P[\gamma \ge \epsilon]$, where $L = 100, f = 1, \gamma \in (0, 0.5]$ and $S_a \in (0, 0.25]$. It is important that probability $P[\gamma \ge \epsilon]$ be less than the negligible constant ε_{cq} while S_a is less than the following threshold:

$$\bar{S}_a = \begin{cases} \sqrt{(\epsilon f)^2 - \frac{3\epsilon f \ln(\varepsilon_{cq}^{-1})}{2L}}, & \delta \in (0, 1] \\ \epsilon f - \frac{3\ln(\varepsilon_{cq}^{-1})}{L}, & \delta \in (1, +\infty), \end{cases} \tag{40}$$

where $\delta \in (0, +\infty)$, $\epsilon \in (0, 1)$ and $L \in \mathbb{Z}$ are three constants. Thus, the chain quality of GSCS is sufficiently high while S_a is sufficiently small. However, S_a must be small while the credibility of each adversary is lower than grade C. This condition can be satisfied while miner credibility is accurately reflected by the credibility rating.

The chain quality property of GSCS is also measured in various simulation experiments using different $R' = S_h/(S_h + S_a)$ (i.e., the computing power share of honest participants). In the simulation, there are 200 miners, the average mining difficulty is 0.05, $S_h = 0.75, 0.85, 0.95$ and $S_a = 0.25, 0.15, 0.05$. The experimental result are shown in fig.12. In the figure, the x-axis and y-axis represent the round number and the adversary-produced permanent block ratio R, respectively. Significantly, R of PoW always converge to the constant R. Meanwhile, that R of GSCS is slightly less than R'. The reason is that in each case, S_h is much larger than S_a and the honest miner always tend to chose the branch with high score. This implies that the chain quality property of GSCS is more superior while S_h is overwhelmingly greater than S_a.

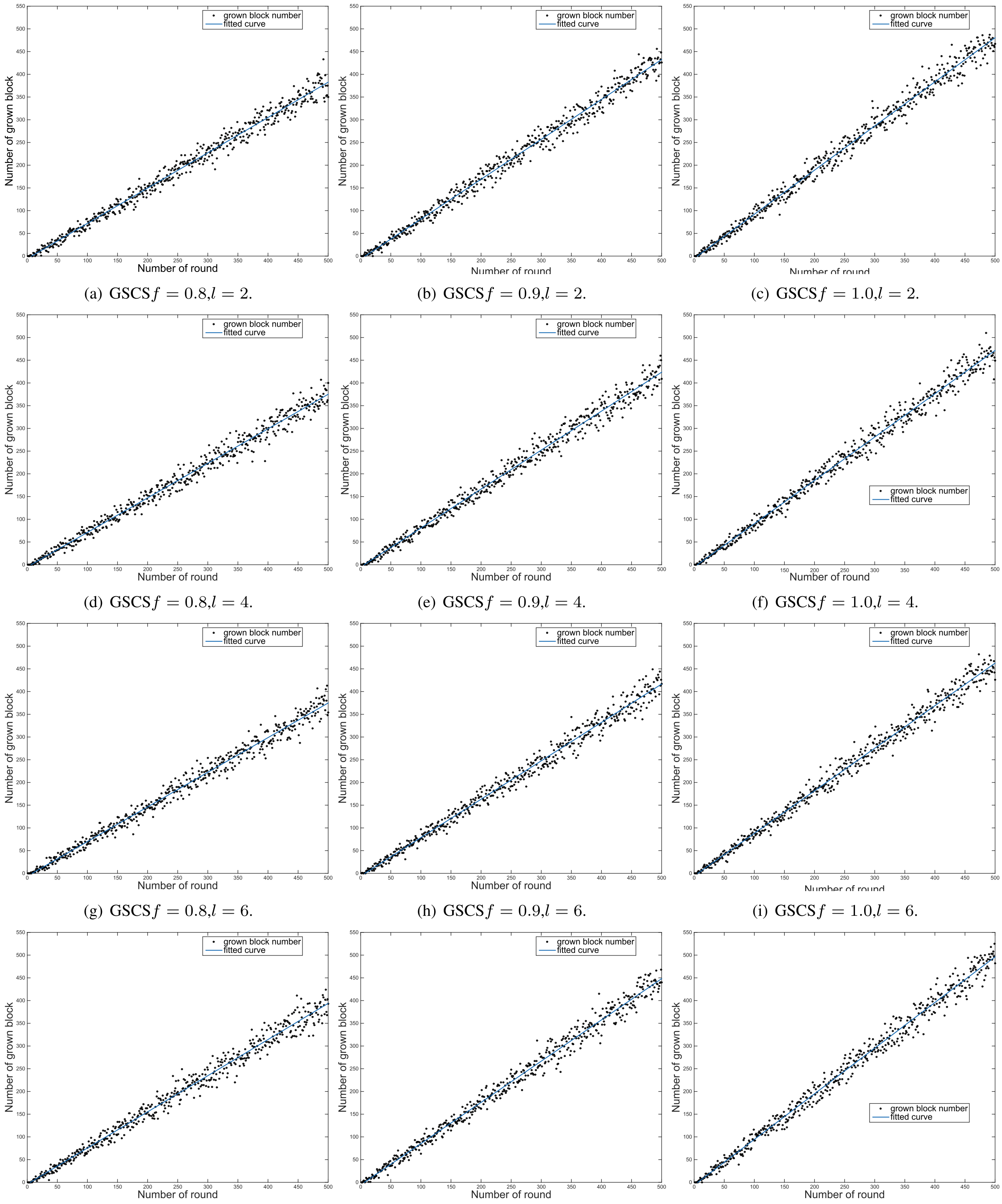
C. CHAIN GROWTH PROPERTY

The notion of chain growth has been informally discussed by Garay [5] and introduced as a formal property of a blockchain by Kiayias [49].

Let $\mathcal{L}_l = \max_i |\mathcal{C}_i^r| - \min_j |\mathcal{C}_j^{r+l}|$, where $\max_i |\mathcal{C}_i^r|$ denotes the length of the longest branch accepted by an honest miner in the r^{th} round, and $\min_j |\mathcal{C}_j^{r+l}|$ denotes the length of the shortest branch accepted by an honest miner in the $r + l^{th}$ round. It is proven that $\mathcal{L}_l \ge \zeta l$ with an overwhelming probability $1 - e^{\Omega(\delta^2 l)}$ for $\delta \in (0, 1)$, where ζ denotes the probability of occurrence of a successful round.[5] Assume that $Z_k \in \{0, 1\}$ and $Z_k = 1$ iff the k^{th} round is a successful round. Using the geometric inequality, we obtain:

$$\begin{aligned} P[Z_k = 1] &= 1 - \prod_{i\in\mathcal{H}} \bar{p}_i \ge 1 - (\frac{\sum_{i\in\mathcal{H}} \bar{p}_i}{N})^N \\ &\ge 1 - (\frac{\sum_{i\in\mathcal{H}} \bar{p}_i}{N})^N \\ &= 1 - (1 - \frac{\sum_{i\in\mathcal{H}} p_i}{N})^N \\ &= 1 - (1 - \frac{S_h}{N})^N, \end{aligned} \tag{41}$$

[5] A detailed definition of a successful round is given in [5]: there is only one block generated by an honest miner in the round.



(a) GSCS$f = 0.8$,$l = 2$.

(b) GSCS$f = 0.9$,$l = 2$.

(c) GSCS$f = 1.0$,$l = 2$.

(d) GSCS$f = 0.8$,$l = 4$.

(e) GSCS$f = 0.9$,$l = 4$.

(f) GSCS$f = 1.0$,$l = 4$.

(g) GSCS$f = 0.8$,$l = 6$.

(h) GSCS$f = 0.9$,$l = 6$.

(i) GSCS$f = 1.0$,$l = 6$.

(j) PoW $f = 0.8$.

(k) PoW $f = 0.9$.

(l) PoW $f = 1.0$.

FIGURE 14. Growth trend of blockchain at different *f* and *l*.

where $N = |\mathcal{H}|$ denotes the number of elements in $\mathcal{H}$. Thus, we obtain

$$\mu_3 = \sum_{k=s}^{s+l} E(Z_k) \geq l - l(1 - \frac{S_h}{N})^N, \tag{42}$$

where $\mu_3 = \zeta l$ denotes the expectation of $Z_l = \zeta l = \sum_{k=s}^{s+l} Z_k$, and $l - l(1 - S_h/N)^N$ denotes the upper bound of a successful round's occurrence in l rounds. Significantly,

$$\begin{aligned} P[\mathcal{L}_l \geq l^*] &\geq P[Z_l \geq l^*] \\ &= 1 - P[Z_l \leq l^*] \\ &= 1 - P[Z_l \leq (1-\delta)\mu_3] \\ &\geq 1 - e^{-\frac{\delta^2 l(1-(1-S_h/N)^N)}{3}}, \end{aligned} \tag{43}$$

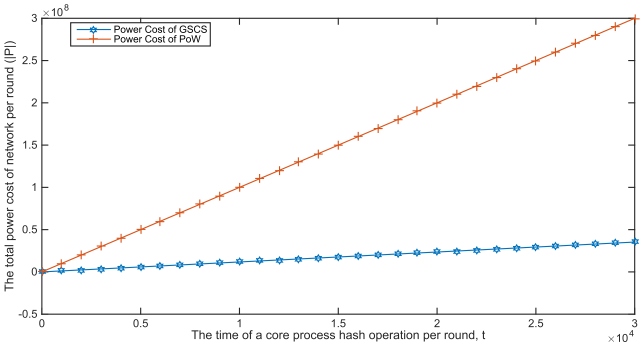


FIGURE 15. **Power cost of GSCS and PoW.**

where $l^* = (1-\delta)\mu_3 \geq (1-\delta)l(1-(1-S_h/N)^N)$. Fig.13 shows the lower bound of $P[\mathcal{L}_l \geq l^*]$ for $l^* \in (1, 10)$, $S_h \in (0.3, 1)$, $N = 100$ and $l = 50$. Assuming that $\varepsilon_{cg} = 1 - \bar{\varepsilon}_{cg}$ is the lower bound of $P[\mathcal{L}_l \geq l^*]$, and $l, N \in \mathbb{Z}^+, \delta \in (0, +\infty)$ are three constants, we obtain the following security lower bound of S_h:

$$\underline{S}_h = \begin{cases} N - N\sqrt[N]{1 - \dfrac{3\ln(\bar{\varepsilon}_{cg}^{-1})}{\delta^2 l}}, & \delta \in (0, 1] \\ N - N\sqrt[N]{1 - \dfrac{3\ln(\bar{\varepsilon}_{cg}^{-1})}{\delta l}}, & \delta \in (1, +\infty]. \end{cases} \tag{44}$$

Similarly, sufficient simulation experiments are performed to assess the chain growth property of GSCS and PoW. In the simulations, there are 200 miners, the average mining difficulty is 0.05 and $f = 0.8, 0.9, 1.0$ and $l = 2, 4, 6$. The simulation results have been shown in fig.14. In the figure, the x-axis and y-axis represent the round number and the grown block number L_c of GSCS and PoW, respectively. It is clear that, for both GSCS and PoW, L_c increases linearly with the round number. However, the influence of l on L_c is limited in GSCS. Additionally, the throughput and confirm latency of blockchain are all related to the chain growth property. The simulation results present that the chain growth property of GSCS is approximate to PoW. It implies that the throughput and confirm latency are approximate to PoW while the round period are equal. For instance, if the period of a round is set to be *10 minutes*, the throughput and latency of our GSCS is approximate to Bitcoin system.(Note that, the period of a round can be adjusted by the expectation of mining difficulty $\bar{D}$)

D. POWER COST

Because SMP must be solved serially, the total computational power cost incurred by miners is finite during mining. Let n be the number of active miners, m be the average number of computing cores per miner, P be the average power cost of a core running a hash function, t be the times of a core runs the hash function per round, and λ be the share of verified time in a round. Thus, the GSCS power cost of the entire network per round can be calculated as

$$\mathcal{P}_{GSCS} = (1-\lambda)ntP + \lambda mntP. \tag{45}$$

Let $(1-\lambda)t \rightarrow q$ and $\lambda mt \rightarrow \alpha q$. Approximately, q can represent the number of times SMP is solved per round, and αq can represent the number of times SMP is verified per round. Thus, $\lambda \rightarrow \alpha/(m+\alpha)$, which can be simplified as $\lambda \rightarrow \alpha/m$ if $m \gg \alpha$. Similarly, the power cost of the PoW mechanism can be calculated as

$$\mathcal{P}_{PoW} = (t-1)mnP + nP. \tag{46}$$

Assume that the round duration of GSCS is equal to that of PoW. Following this assumption, fig.15 shows how the power costs of GSCS and PoW increase with parameter t if $\alpha = 0.2$, $m = 10$ and $n = 1000$. Although GSCS incurs a greater power cost than does PoW during verification, the total power cost of GSCS is much lower than that of PoW during mining. Thus, GSCS requires much less computational power than does PoW if they have the same round duration.

VI. CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

In this paper, we propose a novel consensus mechanism named GSCS for blockchain. Compared with traditional consensus mechanisms (such as PoW and PoS), GSCS provides strong resistance to resource centralization and the quantum attack. First, a resource coalition or a quantum computer possess negligible advantages in mining competition since SMP has been introduced in GSCS. Second, in GSCS each miner is provided with a personalized mining difficulty level that depends on the respective mining credibility. The credibility of each participant is quantified by the respective mining behavior that guarantees that a more credible miner will be assigned a higher successful mining probability. Finally, the performance of GSCS is thoroughly analyzed in terms of the common prefix, chain quality, chain growth and power cost. The analysis shows that GSCS is security- and incentive-compatible if suitable parameters are set. It also demonstrates the potential that GSCS can provide strong security and robustness for the developing blockchain system.

In future research, we will further focus on the incentive compatibility of the mining competition in blockchain systems. Game theory and random process models will be introduced to analyze a miner's behavior in more detail. This will provide a formalized method for evaluating blockchain performance.

REFERENCES

[1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. Manubot, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] D. S. Evans, "Economic aspects of bitcoin and other decentralized public-ledger currency platforms," Coase-Sandor Inst. Law Econ., Univ. Chicago, Chicago, IL, USA, Res. Paper 685, 2014.

[3] J. Kwon. (2014). *Tendermint: Consensus Without Mining*. [Online]. Available: http://tendermint.com/docs/tendermint_v04.pdf

[4] A. Miller, A. Kosba, J. Katz, and E. Shi, "Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 680–691.

[5] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2015, pp. 281–310.

[6] A. Laszka, B. Johnson, and J. Grossklags, "When bitcoin mining pools run dry," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2015, pp. 63–77.

[7] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun, "Is bitcoin a decentralized currency?" *IEEE Security Privacy*, vol. 12, no. 3, pp. 54–60, May 2014.

[8] D. Bradbury, "The problem with bitcoin," *Comput. Fraud Secur.*, vol. 2013, no. 11, pp. 5–8, Nov. 2013.

[9] J. Matonis, "The bitcoin mining arms race: Ghash.io and the 51% issue," New York, NY, USA: CoinDesk, Tech. Rep., 2014. [Online]. Available: https://www.coindesk.com/bitcoin-mining-detente-ghash-io-51-issue

[10] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," in *Proc. WEIS*, 2013, p. 11.

[11] D. Aggarwal, G. K. Brennen, T. Lee, M. Santha, and M. Tomamichel, "Quantum attacks on bitcoin, and how to protect against them," 2017, *arXiv:1710.10377*. [Online]. Available: https://arxiv.org/abs/1710.10377

[12] E. O. Kiktenko, N. O. Pozhar, M. N. Anufriev, A. S. Trushechkin, R. R. Yunusov, Y. V. Kurochkin, A. I. Lvovsky, and A. K. Fedorov, "Quantum-secured blockchain," *Quantum Sci. Technol.*, vol. 3, no. 3, 2017, Art. no. 035004.

[13] N. Houy. (2014). *It Will Cost You Nothing to 'Kill' a Proof-of-Stake Crypto-Currency*. [Online]. Available: http://papers.ssrn.com/sol3/papers.cfm

[14] T. Duong, L. Fan, and H.-S. Zhou. (2016). *2-Hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely*. [Online]. Available: https://eprint.iacr.org/2016/716

[15] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, 2014.

[16] J. Wang, L. Wang, W.-C. Yeh, and J. Wang, "Design and analysis of an effective securing consensus scheme for decentralized blockchain system," in *Proc. Int. Conf. Blockchain Trustworthy Syst.* Berlin, Germany: Springer, 2019, pp. 212–225.

[17] M. Vierhauser, R. Rabiser, and P. Grünbacher, "A case study on testing, commissioning, and operation of very-large-scale software systems," in *Proc. Companion 36th Int. Conf. Softw. Eng.*, 2014, pp. 125–134.

[18] Z. Xia, L. Jiang, X. Ma, W. Yang, P. Ji, and N. N. Xiong, "A privacy-preserving outsourcing scheme for image local binary pattern in secure industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 629–638, Jan. 2020.

[19] C. Wu, L. Li, C. Peng, Y. Wu, N. Xiong, and C. Lee, "Design and analysis of an effective graphics collaborative editing system," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, p. 50, Dec. 2019.

[20] W. Zhang, J. Chang, F. Xiao, Y. Hu, and N. N. Xiong, "Design and analysis of a persistent, efficient, and self-contained WSN data collection system," *IEEE Access*, vol. 7, pp. 1068–1083, 2019.

[21] S. Dubois, R. Guerraoui, P. Kuznetsov, F. Petit, and P. Sens, "The weakest failure detector for eventual consistency," in *Proc. ACM Symp. Princ. Distrib. Comput.*, 2015, pp. 375–384.

[22] W. Golab, X. Li, A. López-Ortiz, and N. Nishimura, "Computing weak consistency in polynomial time," in *Proc. ACM Symp. Princ. Distrib. Comput.*, 2015, pp. 395–404.

[23] B. Wang, W. Kong, H. Guan, and N. N. Xiong, "Air quality forecasting based on gated recurrent long short term memory model in Internet of Things," *IEEE Access*, vol. 7, pp. 69524–69534, 2019.

[24] X. Zhang, C. Lyu, Z. Shi, D. Li, N. N. Xiong, and C.-H. Chi, "Reliable multiservice delivery in fog-enabled VANETs: Integrated misbehavior detection and tolerance," *IEEE Access*, vol. 7, pp. 95762–95778, 2019.

[25] R. Pass and E. Shi, "FruitChains: A fair blockchain," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2017, pp. 315–324.

[26] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, "Adding concurrency to smart contracts," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2017, pp. 303–312.

[27] J. Zou, Y. Wang, and M. A. Orgun, "A dispute arbitration protocol based on a peer-to-peer service contract management scheme," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2016, pp. 41–48.

[28] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[29] C. Lin, D. He, X. Huang, K.-K.-R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *J. Netw. Comput. Appl.*, vol. 116, pp. 42–52, Aug. 2018.

[30] A. K. Talukder, M. Chaitanya, D. Arnold, and K. Sakurai, "Proof of disease: A blockchain consensus protocol for accurate medical decisions and reducing the disease burden," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 257–262.

[31] Z. Dong, Y. Choon Lee, and A. Y. Zomaya, "Proofware: Proof of useful work blockchain consensus protocol for decentralized applications," 2019, *arXiv:1903.09276*. [Online]. Available: http://arxiv.org/abs/1903.09276

[32] J. Clark, A. Edward, and W. Felten, "Research perspectives on bitcoin and second-generation cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy*, 2015, pp. 104–121.

[33] A. Miller and J. J. LaViola, Jr. (2014). *Anonymous Byzantine Consensus From Moderately-Hard Puzzles: A Model for Bitcoin*. [Online]. Available: http://nakamotoinstitute.org/research/anonymous-byzantine-consensus

[34] M. B. Taylor, "Bitcoin and the age of bespoke silicon," in *Proc. Int. Conf. Compil., Archit. Synth. Embedded Syst. (CASES)*, Sep. 2013, pp. 1–10.

[35] J. Tromp, "Cuckoo cycle: A memory-hard proof-of-work system," in *Proc. IACR Cryptol. ePrint Arch.*, 2014, p. 59.

[36] S. King. (2013). *Primecoin: Cryptocurrency With Prime Number Proof-of-Work*. [Online]. Available: https://www.primecoin.org/static/primecoin-paper.pdf

[37] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement.*, 2015, pp. 45–59.

[38] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling nakamoto consensus to thousands of transactions per second," 2018, *arXiv:1805.03870*. [Online]. Available: https://arxiv.org/abs/1805.03870

[39] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement.*, May 2019, pp. 95–112.

[40] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," Chainwhy, Tech. Rep., Aug. 2012. [Online]. Available: https://www.chainwhy.com/upload/default/20180619/126a057fef926dc286accb-372da46955.pdf

[41] P. Vasin. (2014). *Blackcoin's Proof-of-Stake Protocol V2*. [Online]. Available: http://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf

[42] I. Stewart. (2012). *Proof Of Burn. Bitcoin. It*. [Online]. Available: https://eprint.iacr.org/2019/1096.pdf

[43] S. Azouvi, P. McCorry, and S. Meiklejohn, "Betting on blockchain consensus with fantomette," 2018, *arXiv:1805.06786*. [Online]. Available: http://arxiv.org/abs/1805.06786

[44] A. Poelstra, "Distributed consensus from proof of stake is impossible," Wpsoftware, Tech. Rep., 2014. [Online]. Available: https://download.wpsoftware.net/bitcoin/old-pos.pdf

[45] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better: How to make bitcoin a better currency," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2012, pp. 399–414.

[46] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2014, pp. 436–454.

[47] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2015, pp. 515–532.

[48] A. Juan Garay, "Basic properties of the blockchain: (Invited talk)," in *Proc. ACM Workshop Blockchain, Cryptocurrencies Contracts*, 2017, p. 1.

[49] A. Kiayias and G. Panagiotakos, "Speed-security tradeoffs in blockchain protocols," in *Proc. IACR Cryptol. ePrint Arch.*, 2015, p. 1019.

[50] J. A. Garay, A. Kiayias, N. Leonardos, and G. Panagiotakos, "Bootstrapping the blockchain-directly," in *Proc. IACR Cryptol. ePrint Arch.*, 2016, p. 991.

[51] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Advances in Cryptology—EUROCRYPT*, J.-S, Coron and J. B. Nielsen, Eds. Cham, Switzerland: Springer, 2017, pp. 643–673.

[52] M. Mahmoody, T. Moran, and S. Vadhan, "Time-lock puzzles in the random oracle model," in *Proc. Conf. Adv. Cryptol.*, 2011, pp. 39–50.

[53] L. R. Rivest, A. Shamir, and A. D. Wagner, *Time-Lock Puzzles and Timed-Release Crypto*. Cambridge, MI, USA: Massachusetts Inst. Technol., 1996.

[54] B. Cohen and K. Pietrzak, "Simple proofs of sequential work," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2018, pp. 451–467.

[55] M. Mahmoody, T. Moran, and S. Vadhan, "Publicly verifiable proofs of sequential work," in *Proc. 4th Conf. Innov. Theor. Comput. Sci. (ITCS)*, New York, NY, USA, 2013, pp. 373–388.

[56] P. Billingsley, *Convergence of Probability Measures*, 2nd ed. New York, NY, USA: ACM, 2008.

JING WANG received the Ph.D. degree in information security from Wuhan University. She is an Associate Professor with the Guilin University of Electronic Technology. Her research interests include cryptography, network security, and the blockchain.

YONG DING received the Ph.D. degree from Xidian University. He is a Professor with the Guilin University of Electronic Technology. His research interests include cryptography and the blockchain.

NEAL NAIXUE XIONG received the dual Ph.D. degrees in software engineering from Wuhan University and in dependable networks from the Japan Advanced Institute of Science and Technology. He is currently an Associate Professor with the Department of Mathematics and Computer Science, Northeastern State University. Before joining the Northeastern State University, he worked for many years at the Wentworth Technology Institution, Georgia State University. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.

WEI-CHANG YEH (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the Department of Industrial Engineering, University of Texas at Arlington. He is currently a Distinguished Professor with the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan. His research interests include network reliability theory, graph theory, the deadlock problem, linear programming, and scheduling. He is a member of INFORMS.

JINHAI WANG is currently pursuing the Ph.D. degree with Wuhan University. He is an Associate Professor with Foshan University. His research interests include cloud computing and high-performance computing.