基于激励相容的权益分散共识算法

田有亮[1,2,3,4]，袁延森[1,2,3,4]，高鸿峰[2,3,4]，杨旸[5]，熊金波[6]

（1. 贵州大学公共大数据国家重点实验室，贵州 贵阳 550025；2. 贵州大学计算机科学与技术学院，贵州 贵阳 550025；
3. 贵州大学密码学与数据安全研究所，贵州 贵阳 550025；4. 贵州省密码学与区块链技术特色重点实验室，贵州 贵阳 550025；
5. 新加坡管理大学计算机与信息系统学院，新加坡 188065；6. 福建师范大学计算机与网络空间安全学院，福建 福州 350117）

摘 要：PoW 共识算法被证明是激励不相容的，存在高奖励差异下的算力中心化和极端情况下的分叉收敛速度较慢等问题。基于此，提出了一种基于激励相容的 SSPoW 共识算法。通过引入局部解来计算区块链的聚合算力，利用算力的显性量化加快分叉收敛速度，从而满足区块链的一致性。通过改进奖励方案实现激励相容，减少因高奖励差异导致的算力中心化问题。仿真结果证明，所提算法能有效削减奖励差异，并且效率高于传统 PoW 共识算法，对提高系统安全性和共识效率有积极意义。

关键词：共识算法；合作挖矿；分叉收敛；奖励方案

中图分类号：TP302

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2022232

Equity decentralized consensus algorithm based on incentive compatibility

TIAN Youliang[1,2,3,4], YUAN Yansen[1,2,3,4], GAO Hongfeng[2,3,4], YANG Yang[5], XIONG Jinbo[6]

1. State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China
2. College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
3. Institute of Cryptography & Date Security, Guizhou University, Guiyang 550025, China
4. Guizhou Province Key Laboratory of Cryptography and Block Chain Technology, Guizhou University, Guiyang 550025, China
5. School of Computing and Information Systems, Singapore Management University, Singapore 188065, Singapore
6. College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China

Abstract: The PoW consensus algorithm has been proved to be incentive incompatible, existing computing centralization under high reward differences and slow convergence of forks in extreme cases. Based on this, an incentive-compatible-based consensus algorithm SSPoW was proposed. By introducing local solutions to calculate the computing power aggregated on the block chain, the explicit quantification of computing power was used to speed up the convergence of the fork, thus satisfying the consistency of the blockchain. Incentive compatibility was achieved by improving the reward scheme, which reduced the problem of computing centralization caused by high reward differences. Simulation results prove that the proposed algorithm could effectively reduce the reward differences and is more efficient than the traditional PoW consensus algorithm, which has positive implications for improving system security and consensus efficiency.

Keywords: consensus algorithm, cooperative mining, fork convergence, rewarding scheme

收稿日期：2022−08−09；**修回日期：**2022−11−08

通信作者：高鸿峰，jfgao@gzu.edu.cn

基金项目：国家重点研发计划基金资助项目（No.2021YFB3101100）；国家自然科学基金联合基金重点资助项目（No.U1836205）；贵州省高层次创新型人才项目（黔科合平台人才[2020]6008）；贵阳市科技计划基金资助项目（筑科合[2021]1-5），筑科合[2022]2-4）；贵州省科技计划基金资助项目（黔科合平台人才[2020]5017，黔科合支撑[2022]一般 065）

Foundation Items: The National Key Research and Development Program of China (No.2021YFB3101100), Key Program of the National Natural Science Union Foundation of China (No.U1836205), Project of High-level Innovative Talents of Guizhou Province (No.[2020]6008), Science and Technology Program of Guiyang (No.[2021]1-5, No.[2022]2-4), Science and Technology Program of Guizhou Province (No.[2020]5017, No.[2022]065)

0 引言

区块链作为比特币[1]等加密货币的底层技术，自问世以来引起广泛的关注。区块链是在分布式、不可信环境中，所有节点通过特定的共识算法就区块链状态的更新达成一致的技术。共识算法为区块链提供了核心功能[2]，从根本上决定了整个区块链系统的安全性、可用性等性能[3]。

区块链系统中涉及的数据记录由全网节点进行维护和生成，从而实现了去中心化。由于任何人都可以参与创建和验证区块，因此为防止节点的恶意行为，系统需要参与节点付出一定代价。工作量证明（PoW, proof of work）也叫 PoW 共识[1]，通过引入节点间的算力竞争来保证数据一致性，使女巫攻击[4]等恶意行为的作恶成本增加。作为成功挖矿的回报，矿工将获得与他们投入的计算能力成比例的奖励。

然而，PoW 共识的安全性存在一些隐患。首先，由于挖矿过程受到计算能力和概率因素的影响，导致单个矿工的期望收入较低，奖励波动较高，因此，矿工通过加入矿池聚合计算能力来稳定收入。虽然矿池对稳定矿工收益有利，但大型矿池对系统来说是有风险的[5-6]。其次，PoW 共识已被证明是激励不相容的[5,7]，在某些情况下，矿工的最佳策略是不立即公布他们找到的区块，而是将其保留一段时间，通过偏离协议来获得不成比例的利益，即自私挖矿攻击[8-9]。最后，PoW 共识在不一致的情况下（即区块链中的分叉），系统收敛速度会减缓，浪费大量计算资源[10-11]，同时降低发动自私挖矿攻击的算力阈值[5,12]，严重危害系统安全性。

针对上述问题，本文提出一种基于搜索空间划分的工作量证明（SSPoW, proof of work based on spatial segmentation）算法。SSPoW 的核心设计是局部解，局部解视为矿工的部分工作量证明，公布该局部解的矿工获得对应的部分收益。同时定义了全局解和局部解的算力值，以此来量化最终区块的聚合算力，并通过在全局解中附加局部解来提高区块算力，以此加快分叉的收敛速度。局部解的引入和附加提高了挖矿的透明度和共识效率。本文主要贡献如下。

1) 改进挖矿算法：基于搜索空间划分引入局部解，激励矿工合作挖矿提高共识效率。

2) 重新设计奖励方案：通过削减奖励差异来提高矿工挖矿积极性，抑制算力中心化。

3) 更改主链选择协议：通过量化最终区块的聚合算力来加快分叉的收敛速度。

1 相关工作

文献[13]利用不可伪造的时间戳提出了新鲜度首选（FP, freshness preferred）策略，优先选择较新的区块，降低保留区块的竞争力从而降低自私挖矿的收益能力，然而，这种方法需要一个可信的时间戳机构来生成不可伪造的时间戳，并要求诚实的矿工记录所有最近的时间戳发布日志。因此，文献[14]利用工作量证明的泊松性质提出了一种无时间戳防御策略 Zeroblock，通过设置区块的最大接受时间来限制自私挖矿，但由于哈希率波动，区块的预期时间可能会出现高方差，使有效区块失效。文献[15]引入区块的“真实状态”概念，通过为每个交易分配一个预期的确认高度，以检测网络中的自私挖矿行为，但网络中诚实区块传播的意外时延会导致交易实际高度增大，反而会有利于自私的矿工。文献[16]引入透明度并激励参与者尽快公布区块来避免自私挖矿，但是诚实矿工的算力仍会被大量浪费。同时在奖励机制设计方面，文献[17]证明当可用费用不足时，理性矿工会避免挖矿，并会倾向于加入自私挖矿矿池。文献[18]提出一种奖励共享方案（RSS, reward sharing scheme），通过适当激励参与者，实现效率和安全性的均衡，但是存在富者愈富的现象。文献[19]基于以太坊提出权益证明检查点协议 Casper，其结果表明该协议通过激励相容确保了区块链系统的活性，同时提供了比标准工作量证明协议更好的安全保证。在分叉处理方面，文献[20]提出了一种概率验证方案 PvScheme，可以有效降低区块传播时延，从而避免区块链分叉的发生，但安全设计增加了系统的时延，随着验证度的提高，分叉概率也呈现出递增的趋势。上述方案均存在一些缺陷，本文针对现有共识算法存在的激励不相容等问题，设计一种基于激励相容的共识算法，在保证挖矿参与者的积极性的同时，提高共识效率，且可更快地进行分叉处理以及对抗自私挖矿攻击，对区块链技术的发展具有重要意义。

2 基础知识

2.1 PoW 共识算法

区块链系统中 PoW 共识算法的常见形式为

$$H(\text{param} \,\|\, \text{nonce}) < \text{Target} \tag{1}$$

其中， param 表示与区块头部信息相关的数据，nonce 表示随机数，Target 表示目标值（由网络中当前难度值决定）。由哈希函数的性质可知，想要找到符合条件的 nonce 就必须通过穷举的方法来实现。最先求得满足式(1)的 nonce 值的矿工获得记账权，其余矿工仅需将各参数代入即可验证其正确性。PoW 算法通信复杂度为 $O(n)$，节点数可扩展，参与过程不需要身份验证。但是，PoW 算法浪费算力资源，效率较低，且容易导致算力中心化。

同时，区块链网络的整体哈希率和挖矿难度决定了整个网络生成一个新区块的时间（区块间隔）。以比特币为例，为了将区块间隔稳定在 10 min 左右，以适应不断变化的总挖矿能力，比特币网络每隔 2 016 个区块（即一个难度窗口）就会根据式(2)调整目标 Target，即

$$\text{Target}_{\text{new}} = \frac{\text{Target}_{\text{old}} \sum_{i=1}^{2016} \text{Time}_i}{20160} \tag{2}$$

其中，Time_i 表示过去第 i 个块的生成时间，$i \in [1, 2016]$。如果平均区块间隔超过 10 min，目标值就会增加，即难度降低。

2.2　自私挖矿攻击

PoW 共识算法并不总是激励相容的。通过偏离协议和战略性地扣留发现的区块，拥有总计算能力比例 α 的矿工可能会占据区块链上超过 α 的区块，因此获得与其算力份额不成比例的高回报[21-22]。更具体地说，只要区块链处于比当前主链分支多一个或多个区块的有利位置，攻击者就试图通过对发现的区块保密来创建一条私有链。只有当主链几乎赶上时，他才会公布他的私有链，从而使公有链和诚实的矿工所做的一切努力失效，这种攻击被称为自私挖矿。当一个自私矿工的计算能力超过一定的阈值（约 33%）[5]时，会更有效率。因此，如果算力足够分散，自私挖矿攻击对比诚实挖矿将无效率优势。

2.3　矿池和算力中心化

由于 PoW 共识的挖矿收益取决于算力，因此矿工倾向于集中计算资源以获取更高收益。挖矿是零和博弈，只有竞争成功得到记账权的幸运矿工才会得到奖励，而其他为创造区块贡献计算资源但未得到记账权的矿工则完全无收益。因此矿工会激烈地竞争区块记账权，导致计算能力有限的矿工获得的回报出现极高的差异，单独的矿工可能需要等待极长时间才能得到回报。基于上述情况，矿工聚合计算资源形成矿池以提高获得记账权的概率，并根据矿工的算力贡献来分享奖励。以比特币为例，截至 2022 年 4 月，全球最大的 3 个矿池占据了整个网络 50%以上的算力。矿池不仅破坏了区块链的去中心化特性，还引发了各种池内或跨池安全问题[23-24]。

2.4　最长链原则和分叉

最长链原则指网络中最长的链被定义为正确链（即主链），要求矿工在主链上挖矿。矿工在接收一个新区块时，必须停止当前挖矿过程，验证新区块是否有效，否则无法保证自己始终在主链上工作。借助最长链原则，保证每个新区块均被诚实矿工承认。然而，由于网络时延等问题，偶尔会出现分叉现象，即在某个区块高度，多名矿工同时挖出区块，导致矿工对最新区块链状态做出不同判断。虽然基于区块确认机制，上述冲突最终会被解决（仅剩一个分支被承认，其他竞争失败的分支链被废弃），但是由于区块具有相同的算力权重，难以进行量化比较，导致分叉收敛过程较缓慢。因此从系统一致性角度考虑，区块算力的量化具有重要意义。

3　SSPoW 算法设计

3.1　设计概述

基于 PoW 的挖矿并不是透明的，在产生区块前很难快速评估不同矿工的算力。以比特币为例，由于生成区块的平均时间是 10 min，且每个区块的隐含计算能力接近相等。因此，出现分叉问题时，比特币采用最长链原则来解决，而该过程相对耗时，且会造成自私挖矿等问题。理想情况下，恶意隐藏区块应该受到惩罚，然而在实际情况下难以检测和证明一个区块是否被隐藏。通过鼓励矿工公布当前自身计算状态可以显著增加挖矿透明度。例如，通过给予公布计算状态的矿工一定奖励，并要求最终区块包含更多的计算结果来变相增强自己的算力。随着时间的推移，其他矿工可能会创造出具有更高算力的区块，从而给刻意隐藏区块的恶意矿工带来较大的风险，这将激励矿工立即公布他们的最终区块。此外，基于零和博弈的奖励方案会导致矿工间进行激励竞争，所引发的矿池问题会影响

算力分布，进而违背区块链的去中心化特性。为避免此问题，本文重新设计了奖励方案。通过引入局部解机制，鼓励矿工间相互合作从而达到激励相容，同时消除中心化风险。

SSPoW 系统模型如图 1 所示，整个区块链网络共有 n 个矿工 $\{P_1,P_2,P_3,\cdots,P_n\}$，基于矿工数将整个随机数搜索空间划分为 N 份 $\{S_1,S_2,S_3,\cdots,S_N\}$，每个空间大小为 k。每位矿工选择一定数量的搜索空间进行随机数 Nonce 的哈希计算，相当于整个系统中的矿工合作进行挖矿。为保证挖矿透明性，矿工间交换各自的搜索结果，从而快速排除无用搜索空间，避免重复计算。

本文方案中，矿工对搜索空间内的随机数 Nonce 进行计算验证，并将未找到正确 Nonce 值的搜索空间作为局部解公布。该局部解为矿工已付出算力的工作量证明，同时避免了其余矿工的重复无效计算。找到正确 Nonce 值的矿工可将其他矿工已发布的局部解包含至全局解（最终区块）中，从而更好地反映当前区块的聚合算力。当发生分叉时，包含更多算力的区块所在分支将被规定为主链。

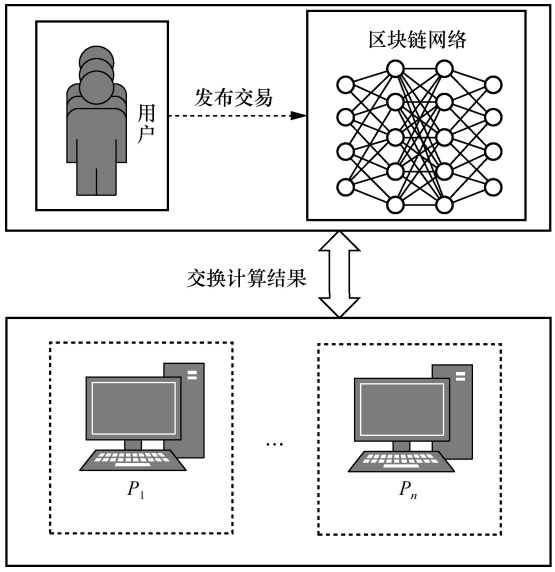


图 1 SSPoW 系统模型

此外，本文基于局部解重新设计奖励方案，旨在杜绝零和博弈。全局解若包含局部解，则公布该局部解的矿工将获取一定比例的奖励，此奖励的大小取决于所有局部解的算力贡献程度。局部解的奖励并不会影响出块矿工的奖励，这将有利于矿工公布区块时包含更多的局部解，以此来增强自身区块的聚合算力。

3.2 详细设计

3.2.1 区块结构

在本文方案中，区块由交易列表、局部解列表和全局解头部组成，其中全局解头部负责验证交易和局部解列表。区块头部字段组成如表 1 所示。本文根据 Coinbase 字段对应的地址奖励局部解的公布者（详见第 3.2.4 节）。因为所有奖励都是由头部信息决定的，所以交易列表中不再需要 Coinbase 交易。为验证局部解的正确性，本文为局部解引入 Addresses 字段代表局部解对应的搜索空间起始地址。由于局部解并不包含交易信息，因此局部解删除了 MerkleRoot 字段。局部解作为区块的组成部分，需在矿工间进行交换，本文利用哈希函数的抗碰撞性来确保所有相关局部解的完整性，如式(3)所示。

$$\text{OP\_RETURN}\ H(\text{header}_0 \parallel \text{header}_1 \| \cdots | \text{header}_n) \quad (3)$$

表 1 区块头部字段

字段名	含义
Version	当前区块版本号，表示本区块遵守的验证规则
PreHash	前一个区块头的哈希值
Target	当前难度目标值，定义了寻找新区块的难度
Nonce	一个随机数，用于生成哈希值
Timestamp	一个 Unix 时间戳
MerkleRoot	Merkle 树的根，汇总了该区块的所有交易
Coinbase	获得奖励的矿工的地址

利用 OP_RETURN 操作来存储聚合局部解头部的哈希值，该哈希值作为数据项存储于区块中，其中，header_i 是一个指向前一个全局解头部的局部解头部。与全局解区块头相比，局部解区块头并不用于验证交易，包含局部解的主要目的是反映集中在区块链某个分支上的聚合算力。SSPoW 区块链的片段如图 2 所示。每个区块包含一个单一的全局解头部、交易列表和聚合局部解的哈希值。在收到一个新区块时，矿工通过检查以下内容来验证该区块（见算法 4 中的 validateBlock()）。

1) 全局解头部哈希值满足要求，并指向之前的全局解头部。

2) 头部字段值正确，即版本、目标和时间戳等都设置正确。

3) 所有包含的交易均正确，即 MerkleRoot 验

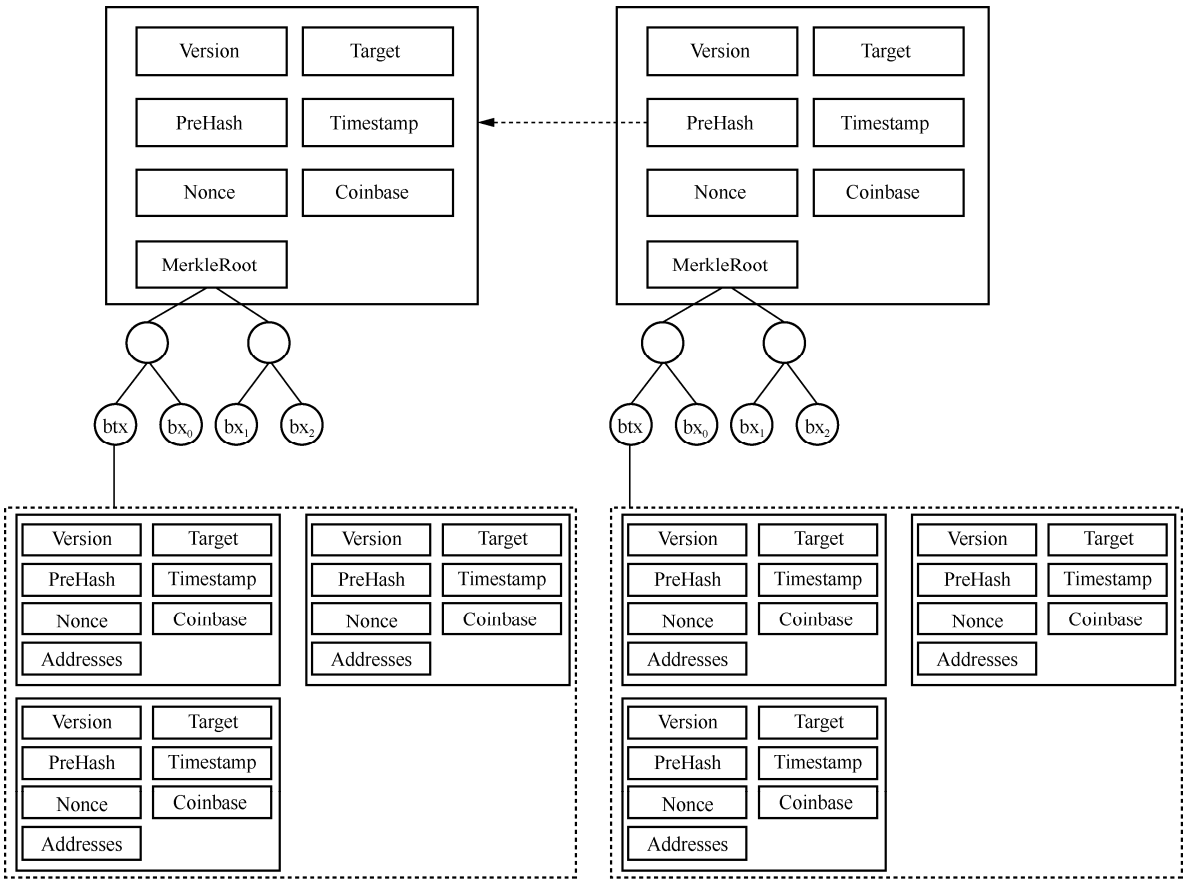


图 2　区块链结构

证正确，且聚合局部解的哈希值验证正确。

4) 所有包含的局部解都是正确的：a) 局部解 nonce 值计算出的头部哈希值符合要求；b) 局部解的 PreHash 字段指向前一个全局解头部；c) 局部解的版本、目标和时间戳均正确。如果验证成功，该区块被追加上链。

3.2.2　*挖矿算法*

与 PoW 共识算法相同，在 SSPoW 中，矿工将交易收集到区块中进行验证。竞争新区块的记账权时，每个矿工试图通过自身算力对同一哈希难题进行穷举求解，算法 1 中的 mineBlock() 函数对 SSPoW 中的挖矿过程进行了简化描述。矿工通过修改区块 Nonce 字段不断改变区块头部 header，当一个头部的哈希值 $h = H(\text{header})$ 小于目标值 Target，即 $h < \text{Target}$ 时，相应的区块会被公布到网络上，其他矿工验证通过后，将更新本地区块链状态，同时启动下一轮挖矿。

算法 1　挖矿算法

```
/*挖矿主函数*/
function mineBlock() {
    result=0;
    minhdr = Target_max;
    for nonce ∈ {0, 1, 2, ⋯} do{
        /*变更nonce计算头部哈希值*/
        hdr = headerHash(nonce);
        if hdr < minhdr then
            minnonce = nonce;
        *若头部哈希值符合要求则打包区块并
公布到网络中*/
        if hdr < Target then{
            B = createBlock(hdr, localHdrsList, Txs);
            broadcast(B);
            result = 1;
            return;
        }
    }
    /*若头部值不符合要求则打包并公布局部解*/
    if result = 0 then{
        LocalBlock = mineLocalBlock
(minnonce,address,size);
```

```
broadcast(LocalBlock)
    }
}
```

当矿工在搜索空间 S_i 上进行挖矿后并未发现正确的 nonce 时，则可以公布关于该搜索空间的局部解，如算法 2 所示。局部解仅包含头部，不包含区块体（交易列表），局部解头部需要提供该搜索空间上的最小头部哈希值 $\min(h_i)$（$i \in [a, a+k]$，a 为该搜索空间起始地址）以及 $\min(h_i)$ 对应的 Nonce 。当矿工公布一个局部解时，除了将其添加到本地局部解列表中，还将在区块链网络中广播该局部解。每个矿工对收到的局部解进行验证（见算法 3 中 onRecvLocalBlock() 函数），并检查该局部解是否指向当前最后一个区块的头部，及其他字段是否正确。另外，本文方案不限制附加的局部解数量，但同一矿工在同一搜索空间的局部解仅会被附加一次。

算法 2 局部解生成算法

```
/*创建局部解函数*/
function mineLocalBlock(nonce,address,size){
    localhdr = headerHash(nonce);
    /*根据nonce值，局部解头部哈希值，
    搜索空间字段来标识局部解*/
    localBlock= createLocalBlock
(nonce,localhdr,address,size)
    return localBlock
}
```

算法 3 局部解验证算法

```
/*验证局部解函数*/
function onRecvLocalBlock(localBlock) {
    hdr = headerHash (localBlock.nonce);
    /*仅接收头部字段正确且头部指向
前一区块的局部解*/
    assert(hdr == localBlock. localhdr > Target );
    assert(validHeader(localBlock));
    assert
(localBlock.PreHash == H(lastBlock.hdr));
    localHdrsList.add(localBlock);
}
```

算法 4 全局解验证算法

```
/*验证全局解函数*/
function validateBlock(B){
    /*仅接收头部字段正确且头部指向前一区块
的全局解*
    assert(B.hdr < Target and validHeader(B));
    assert(B. PreHash == H(lastBlock.hdr)) ;
    assert(validTransactions(B));
    /*验证全局解包含的局部解列表*/
    for hdr ∈ B. localHdrsList do{
        assert(hdr==localBlock.localhdr>Target );
        assert(validHeader(localBlock));
        assert
(localBlock.PreHash == H(lastBlock.hdr)) ;
    }
}
```

3.2.3 分叉处理

通过在区块中引入局部解，区块能更精确地反映聚合挖矿能力。每个区块包含多个局部解，这些局部解对区块的聚合算力有贡献。在分叉的情况下，本文方案依赖于最强链原则，即拥有最高聚合算力的分支链被选定为主链。每条分支链的聚合算力根据算法 5 中的 chainPoW() 函数计算。每条链上的每个区块都会被解析，每个区块的聚合算力分为两部分。

1) 全局解头部的算力。计算式为 $\frac{\text{Target}_{\max}}{\text{Target}}$，其中，$\text{Target}_{\max}$ 是最大的目标值。

2) 所有相关局部解头部的聚合算力。每个局部解头部的算力受搜索空间影响，计算式为 $\frac{\gamma \text{Target}_{\max}}{N\text{Target}}$，其中，$N$ 为划分的搜索空间数量，γ 为相对影响因子，表示局部解的权重。

该分支链的聚合算力被表示为其所有区块的算力之和。通过比较各链的聚合算力，将具有最大聚合算力的链确定为主链。假设一个区块 C_0 包含 x 个局部解，同时存在包含 y 个局部解的区块 C_1，那么如果 $x > y$，该矿工将选择 C_0 而不是 C_1。如图 3 所示，在区块链分叉时，拥有 C_0 和 C_1 区块的矿工将立即决定跟随 C_0 区块，因为 C_0 比 C_1 区块聚合了更多的算力。上述规则激励矿工尽快公布他们的计算结果，避免随着时间的推移，竞争区块算力更高导致自身区块被废弃。

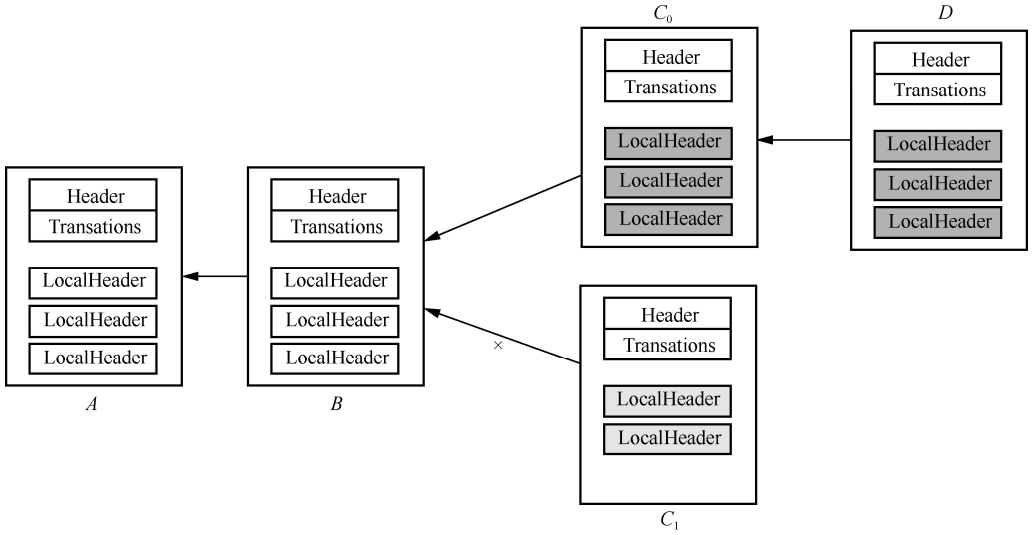


图 3 分叉示例

算法 5　聚合算力计算算法

```
/*聚合算力计算函数*/
function chainPoW(chain){
    sum= 0;
    for B ∈ chain do
        /*计算每个解的算力*/
        T = Target_max / B.Target;
        sum = sum + T;
        for hdr ∈ B. localHdrsList do{
            sum = sum + γT / N;
        }
    return sum;
}
```

3.2.4　区块奖励

SSPoW 共识算法的奖励分配由算法 6 中的 rewardBlock() 函数实现。在 PoW 共识中，只有成功出块的矿工才能获得全额奖励，而在本文方案中，所有为区块的生成做出贡献的矿工（发布局部解的矿工也被包括在内）都将获得与其所提供的算力相称的奖励。一个局部解的公布者获得全额出块奖励 R 的一部分，即 $\frac{\gamma R}{N}$，作为其相应的局部解对特定分支总算力的贡献奖励，其中 γ 为相对影响因子，其大小影响局部解的奖励权重。本文方案不限制局部解的奖励数量，矿工针对同一区块可获得其发布的多个不同局部解的奖励，因此奖励方案不是零和博弈，出块矿工无法通过舍弃其他矿工的局部解来增加自己的奖励；相反，若舍弃其他局部解，将导致自身区块的聚合算力减少，从而影响自身收益。

算法 6　奖励分配算法

```
/*奖励分配函数*/
function rewardBlock(B){
    /*出块矿工获得全额出块奖励和交易手续费*/
    reward(B. Coinbase, R + B.TxFees);
    /*给予局部解矿工一定比例的奖励*/
    w = γR / N;
    for localBlock ∈ B. localHdrsList do{
        reward(localBlock.Coinbase, w);
    }
}
```

基于局部解的机制使诚实矿工收入趋于稳定，降低了因竞争记账权的不确定性所带来的收益波动，使系统算力更加分散。基于上述情况，同一区块中所有不同局部解均被奖励，且局部解求解速度与矿工自身算力成正比。为了保证出块矿工的积极性，将区块交易费（B.TxFees）归于全局解头部的公布者（他也获得了全部的奖励 R）。在本文奖励方案中，新铸币的数量恒定为 R，且货币的总供应量没有上界。

4 安全性分析

区块链分叉、算力中心化以及自私挖矿对区块链系统的安全性造成了极大的危害，本节根据算法特性对上述问题的解决过程进行详细阐述与解释。

4.1 区块链分叉

采用 SSPoW 共识算法的挖矿时间相比于 PoW 共识更短，导致分叉概率上升。因此本文引入了聚合算力的概念，当出现 2 个或多个分叉时，矿工优先选择在聚合算力最高的分支链上进行挖矿。PoW 共识则选择最长链，如图 3 所示，区块链在 C 处出现分叉，若采用 PoW 共识，新区块 D 则会在 C_0 和 C_1 之中任选一个作为父区块。由于没有具体的量化指标，区块链最多需要 6 个区块的时间进行收敛。如图 4 所示，区块 C_1 经过后续 6 个区块的确认保证其不可更改，因此 fork_2 被选定为主链，而 fork_1 上的区块则会成为孤块，这将导致巨大的资源浪费。如果采用本文所提的 SSPoW 算法，当出现分叉时，由于 C_0 的聚合算力大于 C_1，因此后续的区块 D 会立即选择 C_0 作为父区块，fork_1 会被选定为主链，分叉解决仅需一个区块的生成时间，避免重复无意义挖矿过程所造成的资源浪费。

4.2 算力中心化

矿池导致的算力中心化问题在很大程度上可归因于单独挖矿的高奖励差异。因此，将单个矿工的奖励差异保持在较低水平是本文的核心设计目标。设拥有全网 α 算力的独立矿工 P，记 M_1 为采用 PoW 共识进行挖矿后每个区块奖励值的随机变量，M_2 为采用 SSPoW 共识进行挖矿后每个区块奖励值的随机变量。定义随机变量 $X \sim B(1,\alpha)$ 为

$$X=\begin{cases}1\text{，挖矿成功}\\0\text{，挖矿失败}\end{cases} \tag{4}$$

可得 $\mathrm{E}(X)=\alpha$ 和 $D(X)=\alpha(1-\alpha)$，其中，E 和 D 分别表示随机变量 X 的期望和方差。在 PoW 共识中，出块矿工获得固定的控矿奖励 R 以及交易手续费，用随机变量 T 表示交易手续费，定义 $T \sim U(a,b)$，可得

$$M_1 = X(R+T) \tag{5}$$

对式(5)求方差可得

$$D(M_1)=R^2D(X)+D(XT)+2\operatorname{cov}(XR,XT) \tag{6}$$

由于 XR 和 XT 不独立，可得

$$\begin{aligned}\operatorname{cov}(XR,XT)&=R(\mathrm{E}(X^2T)-\mathrm{E}(X)\mathrm{E}(XT))=\\&R\mathrm{E}(T)(\mathrm{E}(X^2)-\mathrm{E}^2(X)=\\&R\mathrm{E}(T)D(X)\end{aligned} \tag{7}$$

将式(7)代入式(6)可得

$$\begin{aligned}D(M_1)&=R^2D(X)+D(XT)+2R\mathrm{E}(T)D(X)=\\&RD(X)(R+2\mathrm{E}(T))+D(XT)\end{aligned} \tag{8}$$

设 Y 为事件 X 的示性函数，定义为

$$Y_X=\begin{cases}1\text{，事件}X\text{发生}\\0\text{，事件}X\text{不发生}\end{cases} \tag{9}$$

代入 $XT=Y_{(X=1)}T$，可得

$$D(XT)=D(X)D(T)+D(X)\mathrm{E}^2(T)+D(T)\mathrm{E}^2(X) \tag{10}$$

将式(10)代入式(8)得

$$\begin{aligned}D(M_1)&=RD(X)(R+2\mathrm{E}(T))+D(X)D(T)+\\&D(X)\mathrm{E}^2(T)+D(T)\mathrm{E}^2(X)\end{aligned} \tag{11}$$

因为 T 服从均匀分布，所以 $\mathrm{E}(T)=\dfrac{a+b}{2}$，$D(T)=\dfrac{(b-a)^2}{12}$，当交易手续费用 T 与挖矿奖励 R 相比较小时，式(11)将简化为

$$D(M_1)=R^2D(X)=R^2\alpha(1-\alpha) \tag{12}$$

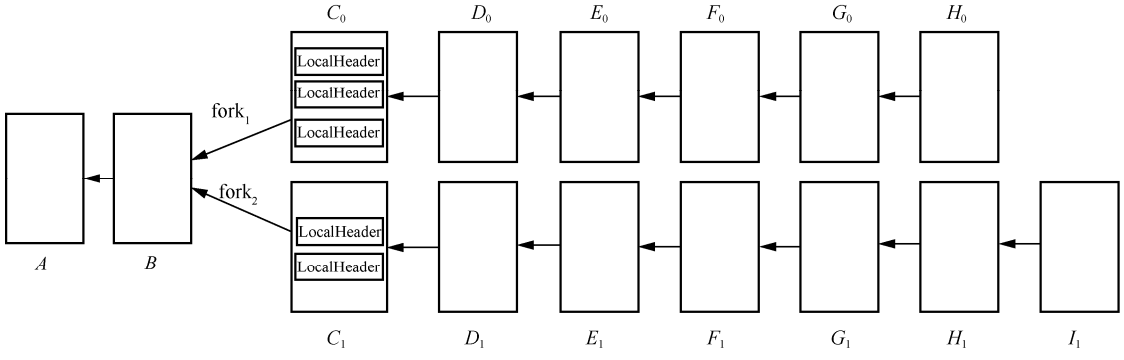


图 4 分叉展示

在 SSPoW 中，假设区块中包含了 L 个局部解，随机变量 $L \sim P(\lambda)$，随机变量 $X_i \sim B(1,\alpha)$。为便于与 PoW 共识进行比较，本文引入缩放常数 c ($0<c<1$)，以确保区块总奖励保持不变。可得

$$M_2 = X(cR+T)+\frac{(1-c)R}{L}\sum_{i=1}^{L}X_i \tag{13}$$

因此可得

$$\mathrm{E}(M_1) = R\mathrm{E}(X)+\mathrm{E}(X)\mathrm{E}(T) = R\alpha+\frac{\alpha}{2}(a+b) \tag{14}$$

$$\mathrm{E}(M_2) = \mathrm{E}(X(cR+T))+\mathrm{E}\left(\frac{(1-c)R}{L}\sum_{i=1}^{L}X_i\right) \tag{15}$$

而

$$\mathrm{E}\left(\frac{(1-c)R}{L}\sum_{i=1}^{L}X_i\right) = (1-c)R\mathrm{E}\left(\frac{1}{L}\sum_{i=1}^{L}X_i\right) = (1-c)R\alpha \tag{16}$$

将式(16)代入式(15)可得

$$\mathrm{E}(M_2) = cR\alpha+\frac{\alpha(a+b)}{2}+(1-c)R\alpha = R\alpha+\frac{\alpha(a+b)}{2} \tag{17}$$

显然可得

$$\mathrm{E}(M_1) = \mathrm{E}(M_2) \tag{18}$$

对式(13)求方差可得

$$D(M_2) = D(X(cR+T))+D\left(\frac{(1-c)R}{L}\sum_{i=1}^{L}X_i\right) \tag{19}$$

当交易手续费用 T 与挖矿奖励 R 相比较小时，式(19)将简化为

$$D(M_2) = (cR)^2 D(X) = c^2R^2\alpha(1-\alpha) \tag{20}$$

显然，$D(M_2)<D(M_1)$，SSPoW 算法降低了奖励的差异，因此，在采用本文算法的区块链系统中，矿工加入大型矿池的动机会降低，对系统的算力去中心化有积极影响。

4.3　虚假局部解

与 PoW 共识不同的是，SSPoW 算法中系统奖励公布局部解的矿工，因此存在着恶意节点公布虚假局部解的行为。本节将通过分析针对局部解的攻击来论述该算法的安全性。

假设在搜索空间 S_i（$i\in[1,N]$）中，恶意矿工 P_d 公布虚假 nonce_f，其对应的局部解哈希值 h_f，该局部解表示为 $\langle S_i,\mathrm{nonce}_f,h_f\rangle_{\sigma_d}$，$\sigma_d$ 为矿工 P_d 的签名。设 S_i 上的正确局部解哈希值为 h_t，则满足 $\mathrm{Target}<h_t<h_f$。当矿工 P_i 收到局部解时，将其存入本地局部解列表中。若收到来自同一搜索空间 S_i，但是 nonce 值和哈希值不同的局部解 $\langle S_i,\mathrm{nonce}_x,h_x\rangle_{\sigma_h}$（$\mathrm{nonce}_x\neq\mathrm{nonce}_f$，$h_x\neq h_f$，$\sigma_h$ 为发送该局部解矿工 P_h 的签名），则会触发检举机制。矿工 P_i 验证哈希值较小的局部解的正确性，若其字段正确，则哈希值较大的局部解被视为恶意结果。例如，局部解 $\langle S_i,\mathrm{nonce}_x,h_x\rangle_{\sigma_h}$ 和局部解 $\langle S_i,\mathrm{nonce}_f,h_f\rangle_{\sigma_d}$，若 $h_x<h_f$，且 $\langle S_i,\mathrm{nonce}_x,h_x\rangle_{\sigma_h}$ 字段验证正确，则另一局部解 $\langle S_i,\mathrm{nonce}_f,h_f\rangle_{\sigma_d}$ 被视为恶意结果，此时矿工 P_i 将发布相关证据 $\left\{P_d,\langle S_i,\mathrm{nonce}_x,h_x\rangle_{\sigma_h},\langle S_i,\mathrm{nonce}_f,h_f\rangle_{\sigma_d}\right\}_{\sigma_i}$ 来检举恶意矿工 P_d，σ_i 为矿工 P_i 的签名。其他矿工收到证据后，验证其正确性，若证据正确，则继续广播该证据，并删除本地局部解列表中 P_d 公布的所有局部解，同时拒绝接收所有来自 P_d 的消息。因此恶意矿工若发布虚假结果，则面临着自身计算结果被拒收的风险，将不会获得任何收益。这种检举机制可以遏制恶意矿工的行为。

4.4　自私挖矿

SSPoW 的另一特点是抑制了自私挖矿攻击。在自私挖矿中，当恶意矿工优先于诚实矿工挖出区块时，恶意矿工不立即将挖掘到的区块公布到区块链网络中，而是选择隐瞒。当诚实矿工在原有链上挖到新的区块时，恶意矿工突然释放之前所保留的区块，增加链的长度，使区块链网络出现分叉，根据最长链原则，诚实矿工挖出的区块无效，从而浪费了大量的算力资源。与此同时，恶意矿工创造的分支链成为最长链，诚实矿工最终也会趋向于在自私链上进行挖矿，严重破坏了区块链系统的安全性。

通过分析得知，自私挖矿是针对区块链最长链原则的攻击，而在 SSPoW 中引入了基于局部解的新挖矿策略，主链选择的并非最长链，而是基于聚合算力的最强链。由于更改了链选择协议，同时通过奖励方案激励矿工尽早公布计算结果，提高了挖矿过程的透明度。因此如果恶意矿工不公布任何新发现的局部解（“隐蔽”挖矿），推迟发布新区块或拒绝包含其他矿工的局部解（“恶意”挖矿），那么在前一种情况下，恶意矿工将无法获得局部解的收益；而在后一种情况下，若诚实矿工公布的区

块聚合算力较高，恶意节点将面临着自身区块被废弃的风险。综上所述，这些策略都会给恶意矿工带来利益损失，极大地抑制自私挖矿。

5 仿真结果

本文仿真实验主机配置如下。CPU 为 Intel Core i5-9500，内存为 8 GB，操作系统为 Window 10 企业版。实验选用 Python3.8 为主要编程语言，并使用其 Matplotlib3.5.2 模块实现数据的可视化。

5.1 性能指标

因为 SSPoW 算法是一个划分搜索空间的合作挖矿，所以除了核心的挖矿模块和通信模块，还需要有划分搜索空间模块。为了简化实验，仿真实验指定了各个节点的搜索空间。实验主要比较 SSPoW 与 PoW 的算法效率和挖矿奖励差异。算法效率通过对比吞吐量和时延进行衡量，奖励差异则通过对比挖矿奖励的离散系数进行衡量。离散系数 c 是度量数据离散程度的相对统计量，用于比较不同样本数据的离散程度，定义为标准差 σ 与平均值 μ 之比，表示为

$$c=\frac{\sigma}{\mu} \tag{21}$$

在本文算法中，挖矿奖励的离散系数越大，说明挖矿奖励的差异程度也越大。吞吐量指单位时间内区块链网络中的交易从产生到被打包并写入区块链中的交易总数，本文使用每秒交易数（TPS, transaction per second）来表示吞吐量，表示为

$$\text{TPS}=\frac{\text{NumofTxs}}{\text{Time}} \tag{22}$$

其中，Time 为交易产生到区块被确认的时间间隔，NumofTxs 为在时间间隔 Time 内被确认区块中包含的交易总数。算法的吞吐量越高，则表明算法的性能效率越高。

5.2 吞吐量对比

在区块链系统中还需考虑算法的使用规模，使用规模指算法在系统中运用后所能承载的网络节点数量。本文分别测试了 PoW 和 SSPoW 这 2 种共识算法在不同节点数量下的吞吐量。为避免误差，实验结果取算法运行 10 次的平均值，如图 5 所示。通过调整系统的节点数量来测试不同的使用规模对算法吞吐量的影响。由图 5 可知，在同一测试平台下，SSPoW 算法的吞吐量接近 400 tps，而 PoW 共识的吞吐量只有 100 tps。但是随着节点数量的增多，吞吐量大小存在着较大的波动。这是因为区块中额外引入了局部解，导致 SSPoW 区块大小相比 PoW 算法有所提高，节点之间的通信开销也会增大，所以随着节点数量的增多，受网络因素的影响也会增大。

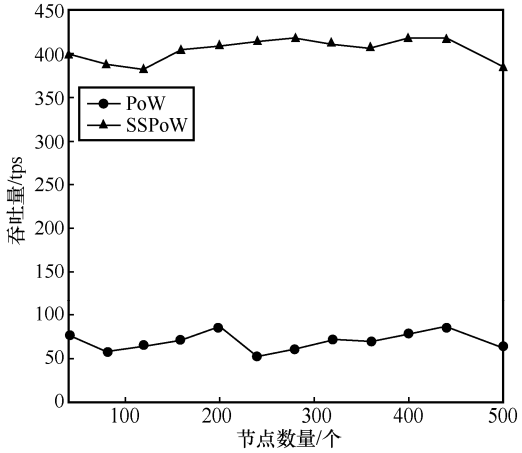


图 5 算法吞吐量对比

5.3 奖励差异对比

由于本文算法为局部解提供一定比例的奖励，为模拟不同区块链网络下算法的奖励分配情况，本节通过在不同节点规模下进行挖矿，测试节点奖励的离散系数并以此来评估奖励差异情况。在同一测试平台下，每个节点规模挖矿 100 次，设置搜索空间数 $N=256$，相对影响因子 $\gamma=1$，表示每个搜索空间对应的局部解奖励之和与全局解奖励相等。每组实验结束后，统计每个节点获得的奖励，并由式(21)计算出该组奖励的离散系数。由图 6 可知，随着节点规模的增大，PoW 算法的奖励离散系数急剧增大，说明随着网络规模的增大，节点的奖励将会出现较大的差异，因此矿工加入矿池的意愿也会更强烈。而 SSPoW 算法的离散系数上升则较缓慢，说明本文算法削减了节点的奖励差异，在一定程度上降低了算力中心化风险。

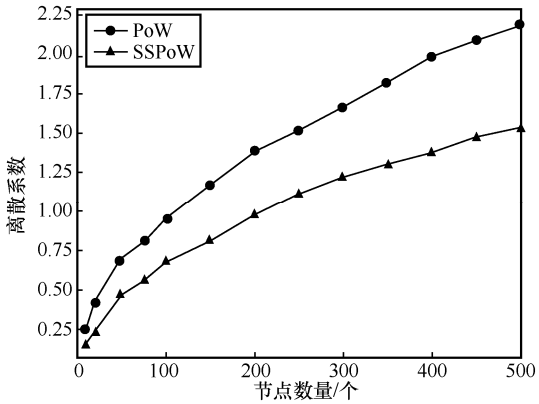


图 6 算法奖励差异对比

5.4 时延对比

SSPoW 算法通过引入局部解提高了共识效率，与此同时，额外字段的引入也会增加通信开销，因此本节通过测试算法的时延来评估通信开销。本节实验对比选用 8 台主机，并在同一测试平台上分别运行 PoW 和 SSPoW 算法来测试挖矿的时延，评估指标为 50 次挖矿的时间开销。由图 7 可知，在同一实验环境下，SSPoW 算法的时延明显低于 PoW 算法。同时，本节实验也计算了 2 种算法挖矿 50 次的平均时延，SSPoW 算法的平均时延显著低于 PoW 算法，说明本文算法在提高共识速度的同时并未带来过大的通信开销。

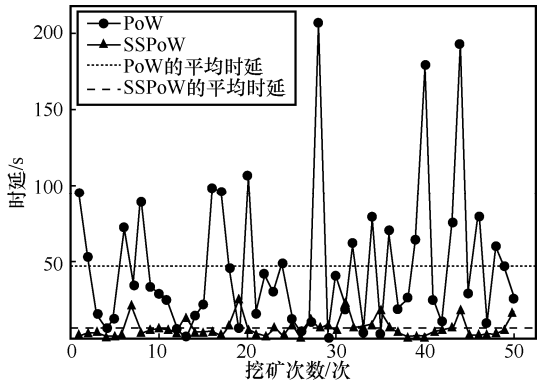


图 7　算法时延对比

6 结束语

对于 PoW 共识而言，高奖励差异带来的中心化问题以及自私挖矿等问题是危害区块链系统安全的关键因素，因此，本文设计并提出了一种新的工作量证明算法——基于 PoW 共识的 SSPoW 算法。本文算法引入局部解，并基于局部解重新设计奖励方案，减少了矿工加入矿池的意愿，从而降低了因算力中心化导致的风险，对区块链的安全属性有积极贡献。最后，理论分析与实验结果表明，相比于传统 PoW 共识算法，本文算法的共识效率和安全性均有提高。未来，笔者希望通过使用新颖的框架和工具来扩展本文工作。

参考文献：

[1] 沈鑫,裴庆祺,刘雪峰. 区块链技术综述[J]. 网络与信息安全学报, 2016, 2(11): 11-20.
SHEN X, PEI Q Q, LIU X F. Survey of block chain[J]. Chinese Journal of Network and Information Security, 2016, 2(11): 11-20.

[2] XIAO Y, ZHANG N, LOU W J, et al. A survey of distributed consensus protocols for blockchain networks[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 1432-1465.

[3] 刘懿中, 刘建伟, 张宗洋, 等. 区块链共识机制研究综述[J]. 密码学报, 2019, 6(4): 395-432.
LIU Y Z, LIU J W, ZHANG Z Y, et al. Overview on blockchain consensus mechanisms[J]. Journal of Cryptologic Research, 2019, 6(4): 395-432.

[4] DOUCEUR J R. The sybil attack[C]//Peer-to-Peer Systems. Berlin: Springer, 2002: 251-260.

[5] EYAL I, SIRER E G. Majority is not enough: bitcoin mining is vulnerable[C]//Financial Cryptography and Data Security. Berlin: Springer, 2014: 436-454.

[6] LEONARDOS N, LEONARDOS S, PILIOURAS G. Oceanic games: centralization risks and incentives in blockchain mining[C]//Mathematical Research for Blockchain Economy. Berlin: Springer, 2020: 183-199.

[7] EYAL I. The miner's dilemma[C]//Proceedings of 2015 IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2015: 89-103.

[8] LI T, WANG Z J, YANG G Y, et al. Semi-selfish mining based on hidden Markov decision process[J]. International Journal of Intelligent Systems, 2021, 36(7): 3596-3612.

[9] NEGY K A, RIZUN P R, SIRER E G. Selfish mining re-examined[C]//International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2020: 61-78.

[10] BIAIS B, BISIÈRE C, BOUVARD M, et al. The blockchain folk theorem[J]. The Review of Financial Studies, 2019, 32(5): 1662-1715.

[11] NEUDECKER T, HARTENSTEIN H. Short paper: an empirical analysis of blockchain Forks in bitcoin[C]//Financial Cryptography and Data Security. Berlin: Springer, 2019: 84-92.

[12] LIAO K, KATZ J. Incentivizing blockchain Forks via whale transactions[C]//Financial Cryptography and Data Security. Berlin: Springer, 2017: 264-279.

[13] HEILMAN E. One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (poster abstract)[C]//Financial Cryptography and Data Security. Berlin: Springer, 2014: 161-162.

[14] SOLAT S, POTOP-BUTUCARU M. Brief announcement: ZeroBlock: timestamp-free prevention of block-withholding attack in bitcoin[C]//Stabilization, Safety, and Security of Distributed Systems. Berlin: Springer, 2017: 356-360.

[15] SAAD M, NJILLA L, KAMHOUA C, et al. Countering selfish mining in blockchains[C]//Proceedings of 2019 International Conference on Computing, Networking and Communications (ICNC). Piscataway: IEEE Press, 2019: 360-364.

[16] SZALACHOWSKI P, REIJSBERGEN D, HOMOLIAK I, et al. StrongChain: transparent and collaborative proof-of-work consensus[C]//Proceedings of the 28th USENIX Security Symposium. Berkeley: USENIX Association, 2019: 819-836.

[17] TSABARY I, EYAL I. The gap game[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2018: 713-728.

[18] BRÜNJES L, KIAYIAS A, KOUTSOUPIAS E, et al. Reward sharing schemes for stake pools[C]//Proceedings of 2020 IEEE European

Symposium on Security and Privacy (EuroS&P). Piscataway: IEEE Press, 2020: 256-275.

[19] BUTERIN V, REIJSBERGEN D, LEONARDOS S, et al. Incentives in Ethereum's hybrid Casper protocol[J]. International Journal of Network Management, 2020, 30(5): e2098.

[20] LIU B, QIN Y, CHU X W. Reducing Forks in the blockchain via probabilistic verification[C]//Proceedings of 2019 IEEE 35th International Conference on Data Engineering Workshops. Piscataway: IEEE Press, 2019: 13-18.

[21] SAPIRSHTEIN A, SOMPOLINSKY Y, ZOHAR A. Optimal selfish mining strategies in bitcoin[C]//Financial Cryptography and Data Security. Berlin: Springer, 2017: 515-532.

[22] MIRKIN M, JI Y, PANG J, et al. BDoS: blockchain denial-of-service[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2020: 601-619.

[23] HU Q, WANG S L, CHENG X Z. A game theoretic analysis on block withholding attacks using the zero-determinant strategy[C]//Proceedings of 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS). Piscataway: IEEE Press, 2019: 1-10.

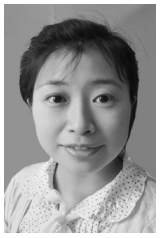
[24] SHI H W, WANG S L, HU Q, et al. Fee-free pooled mining for countering pool-hopping attack in blockchain[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(4): 1580-1590.

[作者简介]

田有亮（1982– ），男，贵州盘县人，博士，贵州大学教授、博士生导师，主要研究方向为算法博弈论、密码学与安全协议、大数据安全与隐私保护等。

袁延森（1998– ），男，河南南阳人，贵州大学硕士生，主要研究方向为区块链技术、共识算法等。

高鸿峰（1975– ），男，贵州遵义人，贵州大学副教授、硕士生导师，主要研究方向为网络与信息安全。



杨旸（1984– ），女，湖北随州人，新加坡管理大学在站博士后，福州大学教授、博士生导师，主要研究方向为区块链、密文搜索、大数据安全等。

熊金波（1981– ），男，湖南益阳人，博士，福建师范大学教授、博士生导师，主要研究方向为大数据安全与隐私保护、区块链技术、安全深度学习。