软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.006892]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

带有预验证机制的区块链动态共识算法*

侯凯祥[1,2], 邱 铁[1,2], 徐天一[1,2], 周晓波[1,2], 池建成[1,2]

[1](天津大学 智能与计算学部, 天津 300350)

[2](天津市先进网络技术与应用重点实验室, 天津 300350)

通信作者: 邱铁, E-mail: qiutie@tju.edu.cn

摘 要: 委员会共识和混合共识通过选举委员会来代替全网节点完成区块验证, 可有效加快共识速度, 提高吞吐量, 但恶意攻击和收买易导致委员会发生腐败问题, 严重时将影响共识结果甚至造成系统瘫痪. 现有工作虽引入信誉机制降低委员会节点腐败的可能性, 但开销大、可信度低且无法降低腐败问题对系统的影响. 提出一种带有预验证机制的区块链动态共识算法 (DBCP), 通过预验证机制在较小开销的前提下对委员会的进行可靠的信誉评估, 及时淘汰委员会中的恶意节点. 若腐败问题已影响到共识结果, DBCP 会通过动态共识将区块验证权转移到全网节点, 并淘汰给出错误意见的委员会节点, 避免系统瘫痪. 若委员会通过迭代达到高可信状态, DBCP 会将区块验证权交由委员会, 全网节点将认同委员会的共识结果并不再对区块进行验证, 进而加快共识速度. 经实验验证, DBCP 的吞吐量较比特币提升两个数量级且与 Byzcoin 相近, 可在一个出块周期内快速应对委员会腐败问题, 安全性优于 Byzcoin.

关键词: 区块链; 混合共识; 预验证机制; 动态共识; 委员会腐败

中图法分类号: TP301

中文引用格式: 侯凯祥, 邱铁, 徐天一, 周晓波, 池建成. 带有预验证机制的区块链动态共识算法. 软件学报. http://www.jos.org.cn/1000-9825/6892.htm

英文引用格式: Hou KX, Qiu T, Xu TY, Zhou XB, Chi JC. Dynamic Blockchain Consensus with Pre-validation. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/6892.htm

Dynamic Blockchain Consensus with Pre-validation

HOU Kai-Xiang[1,2], QIU Tie[1,2], XU Tian-Yi[1,2], ZHOU Xiao-Bo[1,2], CHI Jian-Cheng[1,2]

[1](College of Intelligence and Computing, Tianjin University, Tianjin 300350, China)

[2](Tianjin Key Laboratory of Advanced Networking, Tianjin 300350, China)

Abstract: The committee consensus and hybrid consensus elect the committee to replace the whole nodes for block validation, which can effectively speed up consensus and improve throughput. However, malicious attacks and bribes can easily lead to committee corruption, affect consensus results, and even cause system paralysis. Although the existing work proposes the reputation mechanism to reduce the possibility of committee corruption, it has high overhead and poor reliability and cannot reduce the impact of corruption on the system. Therefore, this study proposes a dynamic blockchain consensus with pre-validation (DBCP). DBCP realizes reliable reputation evaluation of the committee through pre-validation with little overhead, which can eliminate malicious nodes from the committee in time. If serious corruption has undermined the consensus result, DBCP will transfer the authority of block validation to the whole nodes through dynamic consensus and eliminate the committee nodes that give wrong suggestions to avoid system paralysis. When the committee iterates to the high-credibility state, DBCP will hand over the authority of block validation to the committee, and the whole nodes will accept the consensus result from the committee without verifying the block to speed up the consensus. The experimental results show that the throughput of DBCP is two orders of magnitude higher than that of Bitcoin and similar to that of Byzcoin. In addition, DBCP can quickly

* 基金项目: 国家重点研发计划 (2019YFB1703601); 国家自然科学基金联合基金重点项目 (U2001204); 国家自然科学基金面上项目 (6227071709); 天津市杰出青年基金 (20JCJQJC00250); 之江实验室开放课题 (2021KF0AB02)

收稿时间: 2022-03-02; 修改时间: 2022-06-14; 采用时间: 2023-01-05; jos 在线出版时间: 2023-07-28

deal with committee corruption within a block cycle, demonstrating better security than Byzcoin.

Key words: blockchain; hybrid consensus; pre-validation scheme; dynamic consensus; committee corruption

以区块链为基础技术的比特币引领了一股加密货币浪潮[1], 随着以太坊将智能合约引入区块链, 区块链应用领域进一步扩展[2], 在存证溯源[3]、数据共享[4]、工业物联网[5]、隐私计算[6]等领域得到较好发展, 逐步成为数字经济时代的重要基础设施. 区块链保证了去中心化应用 (decentralized application, DApp) 安全运行, 但随着 DApp 的爆发增长, 工作量证明 (proof of work, PoW)、股权证明 (proof of stake, PoS)[7]等全网共识机制由于吞吐量 (transactions per second, TPS) 较低, 难以满足不断增长的交易需求, 成为区块链大范围落地的制约因素[8].

为提高吞吐量, 委员会共识选举少部分节点组成委员会来代替全网节点参与共识, 加快了共识速度. DPoS[9]是最早提出的委员会共识, 其在 PoS 的基础上, 通过股权授予的方式选举委员会, 减少了共识节点数量, 并通过拜占庭容错共识 (practical Byzantine fault tolerance, PBFT)[10]对区块进行验证, 加快了共识速度. 但在 DPoS 中, 区块的打包和验证仅由 21 个委员会节点完成, 中心化程度很高. 并且委员会节点只代表各自持股人的利益而不是全网节点的意见, 委员会可靠性较差, 易发生腐败问题. 此外, 股权授予过程会耗费大量时间, 系统无法及时淘汰委员会中的恶意节点. 近年有工作[11,12]分别从选举公平性和共识速度上对 DPoS 进行改进, 但仍未解决委员会共识中存在的两个关键问题: (1) 选举出更能代表全网共识结果的委员会节点. (2) 当委员会中出现恶意节点时, 更快地完成对委员会的迭代, 及时淘汰恶意节点.

混合共识[13]部分解决了委员会共识存在的问题, 其与委员会共识的相同点为二者均选举委员会代替全网节点对区块进行验证, 不同点为混合共识中的区块打包由全网节点完成, 而委员会共识中的区块打包仅由委员会节点完成. 即在混合共识中存在两层共识: 底层共识指由全网节点决定出块节点, 上层共识指由委员会决定区块的正确性. Byzcoin[14]是混合共识的代表工作, 通过设计开放共识组实现了委员会的快速迭代, 全网可以快速就委员会选举结果达成共识. 但 Byzcoin 只在算力维度上保障委员会节点的可靠性, 委员会中仍可能存在部分算力较高的恶意节点, 一旦发生较严重腐败问题导致恶意节点比例超过拜占庭共识容忍度即 1/3, 共识将无法完成, 系统将瘫痪. 在 Byzcoin 之后, Solida[15]和拟态区块链[16]分别从委员会选举安全性和共识切换随机性方面对混合共识进行改进, 进一步 Omniledger[17]将混合共识与分片结合旨在提高吞吐量, 但仍未解决混合共识中存在的两个关键问题: (1) 在算力维度的基础上, 对委员会节点的验证行为进行评估, 选出更能代表全网共识结果的委员会节点, 降低腐败问题出现的可能性. (2) 降低腐败问题对系统的影响, 确保当委员会内部出现较严重的腐败问题时, 系统不会瘫痪.

针对上述问题, 本文提出一种带有预验证机制的区块链动态共识算法 (dynamic blockchain consensus with pre-validation, DBCP), 做出如下关键贡献.

(1) 设计预验证机制, 实现全网节点对委员会的信誉评估, 选举出更可靠的委员会节点. 进一步设计了基于聚合签名的预验证共识算法 (pre-validation consensus based on aggregate signature, PCAS), 在加快委员会预验证速度的同时降低了信誉管理的存储开销.

(2) 设计动态共识机制, 通过区块预分类和共识切换, 实现区块验证权的动态转移, 若委员会发生严重腐败问题, 则由全网节点对区块进行验证, 并淘汰给出错误意见的委员会节点, 避免了系统瘫痪, 有效降低了腐败问题对系统的影响. 若委员会通过迭代达到高可信状态, 则将区块验证权交由委员会, 全网节点将认同委员会的共识结果并不再对区块进行验证, 进而加快区块共识速度.

(3) 对 DBCP 的可扩展性和安全性进行了理论分析和实验. 在可扩展性分析部分证明了 DBCP 中的委员会可以较快完成迭代并淘汰恶意节点; 预验证共识算法 PCAS 在共识速度和存储开销方面优于 PBFT; DBCP 造成的额外存储开销很小, 约 1%. 在安全性部分证明了 DBCP 可以有效抵抗双花攻击、女巫攻击以及委员会腐败问题. 最后通过实验验证了 DBCP 的吞吐量较比特币提升两个数量级且与 Byzcoin 相近; DBCP 可在一个出块周期内快速应对委员会腐败问题, 避免因委员会腐败造成系统瘫痪, 安全性优于 Byzcoin.

本文第 1 节介绍有关委员会共识和混合共识中应对委员会腐败问题的相关工作. 第 2 节介绍 DBCP 的架构以及预验证机制与动态共识的实现细节. 第 3 节对 DBCP 安全性、可扩展性、应用场景进行理论分析. 第 4 节对

DBCP 的迭代速度、共识速度、存储开销、吞吐量以及对恶意攻击的应对效果进行仿真实验和评估. 第 5 节总结全文.

1 相关工作

委员会共识和混合共识通过选举委员会, 达到减少共识节点数量、提升系统吞吐量的目的. 但委员会规模较小, 可靠性较差, 易被收买或攻击, 导致发生委员会腐败问题, 对系统安全性造成严重影响. 本节主要介绍委员会共识和混合共识中应对委员会腐败问题的相关工作.

目前针对委员会共识的优化工作, 大多将研究点放在提高委员会的共识速度和委员会节点可靠性上. Xu 等人提出一种基于模糊集合的委员会共识[11], 在进行委员会选举时, 允许节点投反对票, 从而降低恶意节点被选为委员会节点的概率, 提高委员会的可靠性. Li 等人提出可扩展的多层 PBFT 网络[12], 通过将节点分层分组降低 PBFT 的通信复杂度, 提高委员会的共识速度. 上述工作分别从委员会选举公平性和共识速度上对委员会共识进行改进, 但没有解决委员会迭代慢的问题, 也未实现对委员会节点的信誉评估. Lai 等人通过建立节点信誉评估模型改进委员会共识[18], 将委员会节点的信誉值与投票权重相对应, 以此削弱恶意节点的话语权, 但其提出的信誉评估模型中所涉及的相关指标的真实性并未得到保障, 导致评估结果可信度较低.

针对委员会的迭代速度及抗算力攻击能力, 研究者取得了一定进展. PeerCensus[19]是最早提出的较完整的混合共识方案, 在 PeerCensus 中, 委员会采用 PBFT 共识验证区块, 新节点要成为委员会节点必须提交一份工作量证明, 提交的工作量证明越多, 在委员会中话语权越高. PeerCensus 将委员会节点的话语权与算力挂钩一定程度上提高了委员会的可靠性, 但 PeerCensus 无法将委员会的大小维持在一个合理范围内, 而且采用的 PBFT 共识存在共识速度慢, 通信开销大的问题. 随后 Kogias 等人提出 Byzcoin[14], 解决了 PeerCensus 的问题. Byzcoin 采用基于生成树的 cosi 聚合签名[20]加快委员会的共识速度, 进一步通过设置时间窗口控制委员会大小, 只有在当前时间窗口内成功挖矿的节点才能成为委员会成员并获得一定比例的股份. Solida[15]在 Byzcoin 的基础上将 PBFT 共识引入委员会选举中, 即节点在提交工作量证明的基础上需要经过委员会的 PBFT 共识才可被选入委员会, 进一步提高了委员会的可靠性. Omniledger[17]将 Byzcoin 改进为 ByzcoinX, 并通过分片提高吞吐量, 但分片会导致算力稀释, 一定程度上削弱了委员会的可靠性, 类似的还有 Mvcom[21], SSHC[22]. 以上方案都实现了委员会的自动迭代, 并且通过 PoW 在算力维度上保证委员会节点的可靠性, 被称为基于 PoW 和 PBFT 的混合共识[23].

在其他混合共识优化工作中, Repchain[24]首次提出信誉链和交易链的双链架构, 将信誉作为影响因素引入委员会选举中, 信誉越高的节点被选入委员会的概率越大, 提高了委员会的可靠性. 但 Repchain 中引入的信誉值是由委员会节点相互评估产生, 并不能代表全网节点的意见, 不具有较高的可信度. Huang 等人提出 zkRep[25], 在基于信誉的区块链上引入隐私保护机制, 从而使攻击者更难识别具有高信誉价值的验证节点, 一定程度上降低了腐败问题发生的可能性但同时增加了共识复杂度. Xu 等人提出拟态区块链[16], 在区块链中引入 PoW、PoS 和 PBFT 这 3 种共识, 实现了异构共识机制, 在共识随机性上增强了区块链的安全性. 但拟态区块链的安全性基于参与共识的恶意节点不超过 1/3 的前提, 无法应对较严重的腐败问题.

上述应对委员会腐败问题的相关工作中存在 3 类问题: (1) 选举委员会节点时只考虑节点算力, 未考虑节点行为. (2) 部分工作提出信誉评估模型, 但无法保证信誉值的可靠性和真实性. (3) 当委员会内部出现较严重的腐败问题即恶意节点超过 1/3 时, 系统将瘫痪, 且无法自我恢复.

为解决上述问题本文通过预验证机制实现全网节点对委员会的信誉评估, 并将评估指标记录到区块链中保证评估结果可信不可篡改. 当委员会发生较严重腐败问题时, DBCP 将通过动态共识将区块验证权从委员会转移到全网节点, 并根据全网节点的意见对委员会进行迭代, 淘汰恶意节点, 降低腐败问题对系统的影响.

2 系统设计

本节从 3 方面介绍 DBCP 的细节, 分别是系统架构、预验证机制和动态共识. 在系统架构部分, 通过实例介绍 DBCP 中区块的打包和共识流程. 在预验证机制部分, 从委员会管理机制和预验证共识 PCAS 出发, 介绍委员会

节点的信誉计算、迭代、激励和共识流程. 在动态共识部分, 从区块预分类和共识切换出发, 介绍区块预分类算法的计算流程以及共识切换的条件.

2.1 系统架构

DBCP 是一种动态共识, 与混合共识类似, DBCP 中同样存在两层共识, 其动态性体现在决定区块正确性的上层共识在预验证共识和全网共识之间动态切换. 底层共识的挖矿难度也会随上层共识的切换动态调整, 当上层共识为预验证共识时, 挖矿难度较低; 当上层共识为全网共识时, 挖矿难度较高. 预验证共识采用本文在预验证机制中提出的 PCAS 共识, 即节点只需通过验证委员会给出的预验证意见来决定区块的正确性; 全网共识采用 PoW 中验证区块的方式, 即节点需要验证区块中随机数和所有交易的正确性. DBCP 的系统架构如图 1 所示, 主要由两部分组成: 预验证机制和动态共识.

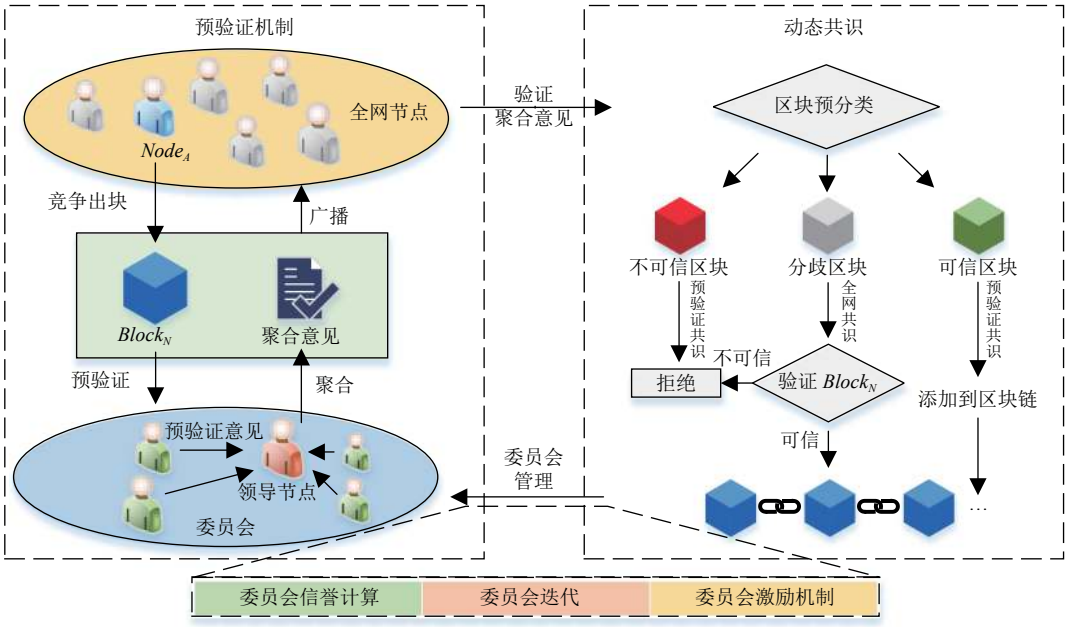


图 1 DBCP 的系统架构

预验证指委员会在区块全网广播前, 预先对区块的正确性进行验证并形成聚合意见的过程. 如图 1 左半边所示, 节点 $Node_A$ 成功打包区块 $Block_N$, 并将 $Block_N$ 发给委员会, 委员会中的节点将对 $Block_N$ 进行预验证, 并生成预验证意见 $\langle preview, hash, sign\rangle$ 发送给委员会中的领导节点. 预验证意见中的 $preview$ 可以取 0、1 分别代表委员会节点对区块的评估意见为反对或支持, $hash$ 代表被评估区块即 $Block_N$ 的哈希, $sign$ 是委员会节点对预验证意见的签名. 领导节点在收到委员会节点的预验证意见后将进行聚合, 生成针对 $Block_N$ 的聚合意见 $\langle msg, hash, aggregate\ sign\rangle$, 随后聚合意见将和 $Block_N$ 一起广播给全网节点. 聚合意见中的 msg 代表预验证意见的汇总结果, 形如 msg = 10X11101 , $msg[i]$ 取 0、1、X 分别代表序号为 i 的委员会节点对 $Block_N$ 的评估意见为支持、反对或缺失, $aggregate\ sign$ 代表领导节点将委员会节点签名聚合后生成的聚合签名.

动态共识指节点在收到区块和聚合意见后, 根据委员会的信誉和聚合意见动态选择共识算法来验证区块的过程. 如图 1 右半边所示, 全网节点首先对 $Block_N$ 的聚合意见进行验证, 并根据聚合意见和当前委员会的信誉对区块进行预分类. 预分类完成后, 不可信区块将被拒绝验证和广播; 分歧区块将由全网节点进行二次验证, 并根据验证结果选择是否将其添加到链上; 可信区块将被全网节点免验证并直接添加到链上.

当有一定数量的新区块出现在主链上, 会触发委员会管理. 如图 1 下方所示, 委员会管理是 DBCP 运行的基础, 分为信誉计算、迭代和激励 3 部分. 委员会管理机制通过将委员会意见与全网共识结果作对照, 计算委员会节

点信誉值, 信誉计算完成后将进行迭代, 淘汰掉低信誉节点、选入新节点, 并根据委员会节点信誉值的变化情况对委员会节点进行激励.

2.2 预验证机制

DBCP 通过预验证机制选举可靠的委员会节点并快速达成预验证共识. 预验证机制包括两部分: 委员会管理机制和基于聚合签名的预验证共识 PCAS. 其中委员会管理机制包含 3 部分: 信誉计算机制、迭代机制和激励机制.

2.2.1 委员会信誉计算机制

信誉计算是委员会管理的基础, 为选举可靠的委员会节点提供评估依据. 影响信誉值的关键因素是委员会节点是否给出正确的预验证意见, 故 DBCP 以全网共识的结果作为对照, 认为给出的预验证意见与全网共识结果一致的节点为可靠的委员会节点.

计算委员会节点的信誉值需将委员会节点意见与全网共识结果进行对照, 为使对照过程自动完成并保证对照结果可信不可篡改, DBCP 将对照过程与区块打包结合在一起. 如图 2 所示, DBCP 以比特币区块为基础对区块结构进行改造, 规定节点在打包区块时需要加入对照意见组和对照意见组的哈希. 对照意见组中包含若干条对照意见$\langle\langle msg,hash,aggregate sign\rangle,view\rangle$, 每条对照意见包含两部分: 一部分是聚合意见$\langle msg,hash,aggregate\ sign\rangle$, 即委员会对 $Block_N$ 的关联区块的预验证结果; 一部分是参考意见 $view$, 即 $Block_N$ 的打包节点 $Node_A$ 对 $Block_N$ 的关联区块的验证结果. $Block_N$ 的关联区块包括 $Block_N$ 的父区块以及其他与 $Block_N$ 处于相同高度且经过预验证的区块. $Block_N$ 关联区块包含的对照意见可以反映委员会在 $Block_N$ 被打包期间所给出预验证意见的正确性.

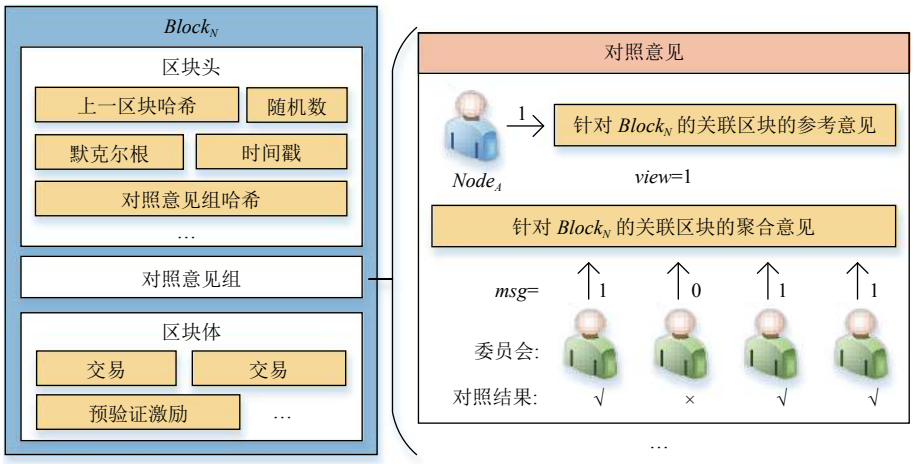


图 2 区块结构

为就对照意见达成全网共识, 全网节点在验证区块的同时需要验证区块中的对照意见. 设 $Block_N$ 中包含的某条对照意见为 CS_R, 设 CS_R 中包含的 $hash$ 即 $CS_R \to hash$ 所对应的区块为 $Block_R$, $CS_R \to view$ 代表 $Block_N$ 的打包节点 $Node_A$ 对 $Block_R$ 的验证结果, $CS_R \to msg$ 代表委员会对 $Block_R$ 的预验证结果. 全网节点验证对照意见的过程分为两部分: (1) 验证 CS_R 的有效性, 即通过验证签名的合法性来判断 CS_R 是否被修改. (2) 验证 $CS_R \to view$ 的正确性, 即验证 $CS_R \to view$ 是否与自身对 $Block_R$ 的验证结果一致. 如果验证通过, 则将区块添加到自己的链上.

当 $Block_N$ 出现在主链上, 说明 $Block_N$ 被全网节点认可, 即代表 $Block_N$ 中的对照意见被全网节点认可. 因此, 对照意见中的 $view$ 即为全网节点的意见, 通过将 $view$ 与 msg 对比即可筛选出给出的预验证意见与全网共识结果不一致的恶意节点, 并降低其信誉值.

信誉值的计算规则为: 设委员会的迭代周期为 T, 即每完成 T 个区块的预验证进行一次委员会的信誉计算. 设委员会为 $V^m = \{V_0, V_1, \ldots, V_{m-1}\}$, 共有 m 个委员会节点, 全网节点数为 n; 称委员会第 i 次迭代后系统处于第 i 个时

代; 委员会节点 V_x 在第 i 个时代开始时信誉值为 $R^i_{V_x}$; 节点初始信誉设为 C, 即 $R^0_{V_x}=C$; 奖励常量为 r_1 ($r_1>0$), 惩罚常量为 r_2 ($r_2>0$), 最大宕机次数为 f_{th} ($f_{th}>0$); 第 i 个时代产生的 T 个区块中总共包含 l ($l\geqslant T$) 条对照意见. 按照先后顺序对对照意见进行标号, 记第 i 个时代中包含的第 d 条对照意见为 CS^i_d; $CS^i_d\rightarrow msg$ 代表汇总的预验证意见, $CS^i_d\rightarrow msg[x]$ 代表委员会中序号为 x 的节点 V_x 给出的预验证意见, $CS^i_d\rightarrow view$ 代表该对照意见所在区块的打包节点给出的参考意见.

对于每条对照意见, 若委员会节点给出的预验证意见与区块打包节点给出的参考意见一致, 委员会节点的信誉将增加 r_1; 若不一致, 委员会节点的信誉将先减半再减去 r_2. 对信誉值减半可以达到信誉越高作恶惩罚越高的效果, 进而降低高信誉节点的作恶意愿.

有两种情况会导致 V_x 的意见缺失致使 $CS^i_d\rightarrow msg[x]$ 的值为 X: (1) 委员会节点宕机, 没有及时将预验证意见发送给领导节点. (2) 领导节点是恶意节点, 在进行意见聚合时故意不聚合部分委员会节点的意见, 从而达到损害部分委员会节点信誉值的目的.

由于每轮预验证会随机选取领导节点, 故领导节点很难连续作恶, 并且节点宕机是一个长期事件, 即若某节点因为硬件或网络故障宕机, 则该节点大概率会连续多次在预验证过程中宕机. 设 V_x 在第 i 个时代最多连续 $F^i_{V_x}$ 没有给出预验证意见, 若 $F^i_{V_x}$ 大于 f_{th}, 则说明 V_x 出现宕机问题, 需要将其信誉减半再减去 r_2.

信誉计算需要遍历该迭代周期内产生的所有对照意见共 l 条, 并遍历每条对照意见中包含的共 m 个委员会节点的意见, 最后需要对 m 个委员会节点计算其意见缺失情况, 故算法复杂度为 $\mathrm{O}(lm+m)$, 近似为 $\mathrm{O}(lm)$, 具体细节见算法 1.

算法 1. 委员会节点信誉计算.

输入: 第 i 时代委员会节点的信誉值 $R^i=\{R^i_{V_0},R^i_{V_1},\ldots,R^i_{V_{m-1}}\}$, 对照意见 $CS^i=\{CS^i_1,CS^i_2,\ldots,CS^i_l\}$;
输出: 第 $i+1$ 时代委员会节点的信誉值 R^{i+1}.

1. **for** $V_x\in V^m$ **do** //初始化
2. $F^i_{V_x}=0$, $f^i_{V_x}=0$, $R^{i+1}_{V_x}=R^i_{V_x}$
3. **end for**
4. **for** $CSP\in CS^i$ **do** //遍历每条对照意见
5. **while** $j=0;j<m;j++$ **do** //遍历委员会节点意见并与区块打包节点的参考意见进行对照
6. **if** $CSP\rightarrow msg[j]\neq X$ **then**
7. **if** $f^i_{V_j}>F^i_{V_j}$ **then**
8. $F^i_{V_j}=f^i_{V_j}$ //更新 V_x 在 i 时代的最大意见连续缺失次数 $F^i_{V_x}$
9. **end if**
10. $f^i_{V_j}=0$
11. **if** $CSP\rightarrow msg[j]==CSP\rightarrow view$ **then**
12. $R^{i+1}_{V_j}=R^{i+1}_{V_j}+r_1$ //提高给出正确意见的委员会节点的信誉值
13. **else**
14. $R^{i+1}_{V_j}=R^{i+1}_{V_j}-\dfrac{R^i_{V_j}}{2}-r_2$ //降低给出错误意见的委员会节点的信誉值
15. **end if**
16. **else**
17. $f^i_{V_j}++$ //累加 V_x 在 i 时代的意见连续缺失次数 $f^i_{V_j}$
18. **end if**
19. **end while**

20. **end for**
21. **while** $k=0;k<m;k++$ **do** //遍历委员会节点的意见缺失情况
22. **if** $F_{V_k}^i > f_{th}$ //对宕机节点进行惩罚
23. $R_{V_k}^{i+1} = R_{V_k}^{i+1} - \frac{R_{V_k}^i}{2} - r_2$
24. **end if**
25. **end while**
26. **return** R^{i+1}

2.2.2 委员会迭代机制

系统初始化阶段会在全网随机选取或通过推举的形式产生第 1 代委员会, 之后每打包 T 个区块触发一次委员会迭代, 淘汰信誉值小于阈值 R_{th} 的节点并从委员会节点候选池中选入新节点, R_{th} 决定了系统对低信誉委员会节点的容忍度. 委员会节点候选池定义为主链上的区块的打包节点, 该类节点算力较高, 可降低选举过程被女巫攻击的风险.

迭代过程如图 3 所示, 迭代之前根据区块中的对照意见进行信誉计算, 并统计信誉值小于 R_{th} 的节点的数量 z, 如果 z 不为零则开始迭代. 设主链上高度为 h 的区块的打包节点为 $Node_h$, 首先以迭代区块的父区块中的随机数作为种子通过线性同余法或可验证随机函数 (verifiable random functions, VRF)[26]产生长度为 z 且不重复的随机数序列 $r^z=\{r_1,r_2,\ldots,r_z\}$, 则 $\{Node_h \mid h\in r^z\}$ 为新当选的委员会节点. 若新当选节点与现有委员会节点重复, 系统会产生新随机数直到选出 z 个不同的委员会节点, 进而保证委员会节点的多样性.

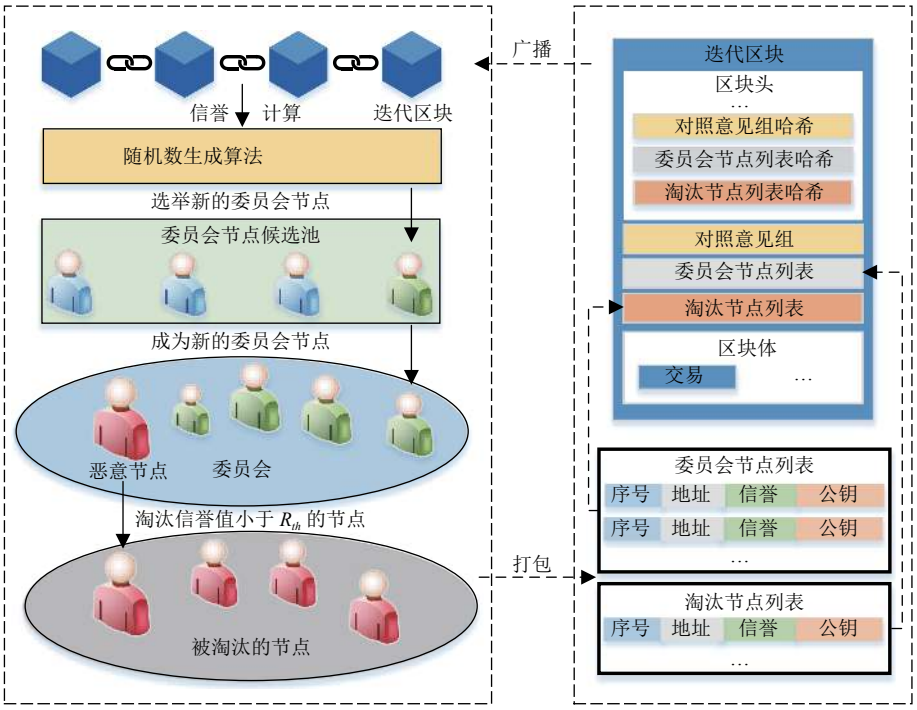


图 3 委员会的迭代过程

为使迭代过程自动完成并保证迭代结果可信不可篡改, DBCP 同样将委员会迭代与区块打包结合在一起, 设计迭代区块. 如图 3 右边所示, 迭代区块中除包含对照意见组及对照意见组的哈希外, 还包含委员会列表, 淘汰节点列表及列表的哈希. 委员会列表用于向全网公示新一代委员会节点的相关信息, 包括委员会节点的序号、地址、

信誉值和公钥. 淘汰节点列表用于记录在该次迭代中被淘汰的节点.

2.2.3 委员会激励机制

为提高委员会节点积极性, 激励委员会节点给出更高质量的预验证意见, DBCP 在每个迭代周期结束后根据委员会的信誉变化对委员会节点进行激励. 激励来自交易的手续费, 通过智能合约按照激励规则分发给委员会节点.

交易在发起时, 会将一部分手续费转向激励合约的地址作为预验证收益, 预验证收益分为基础收益和信誉收益两部分, 并在激励合约的管理下发放给委员会节点. 设第 i 个时代产生的预验证总收益为 Y^i, 委员会节点 V_x 获得总收益为 $E^i_{V_x}$, $E^i_{V_x}$ 由基础收益 $B^i_{V_x}$ 和信誉收益 $C^i_{V_x}$ 两部分组成, 如公式 (1).

$$E^i_{V_x} = B^i_{V_x} + C^i_{V_x} \tag{1}$$

基础收益 $B^i_{V_x}$ 根据委员会节点在第 i 个时代的信誉值增量, 即预验证意见的正确率, 分配给委员会节点, 从而激励节点给出更高质量的预验证意见, 可根据公式 (2) 计算出.

$$B^i_{V_x} = \frac{Y^i}{2}\sum_{g=0}^{m-1}\frac{R^i_{V_x} - R^{i-1}_{V_x}}{R^i_{V_g} - R^{i-1}_{V_g}} \tag{2}$$

信誉收益 $C^i_{V_x}$ 根据第 i 个时代开始时委员会节点的信誉占委员会总信誉的比例分配给节点, 从而降低高信誉节点的作恶意愿, 可根据公式 (3) 计算出.

$$C^i_{V_x} = \frac{Y^i}{2}\sum_{g=0}^{m-1}\frac{R^{i-1}_{V_x}}{R^{i-1}_{V_g}} \tag{3}$$

2.2.4 基于聚合签名的预验证共识算法 PCAS

聚合签名可以把多个参与方的公钥和签名合并为一个公钥与签名, 并无法从合并后的签名推导出合并前的信息, 仅需对聚合签名进行一次验证即可完成对多个参与方签名的验证. 聚合签名可以有效降低存储空间和验证过程中的网络流量成本, 尤其对签名频次较低但验证频次较高的业务场景有显著效果. 预验证共识是一个由少量节点签名并由全网节点验证的场景, 利用聚合签名可有效减少预验证共识的通信和存储开销.

本文基于 BLS 聚合签名[27]设计 PCAS 共识算法, 如图 4 所示, 分为 6 个阶段: 请求阶段、领导节点选举阶段、意见收集阶段、签名聚合阶段、确认阶段和全网广播阶段, 具体流程如下.

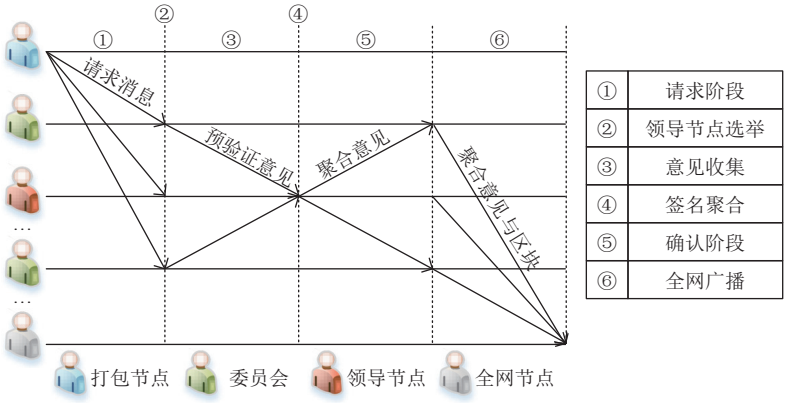


图 4 PCAS 的共识过程

区块的打包节点向委员会广播一条请求消息$\langle Block, sign, v\rangle$, 用于传送区块内容和通知新一轮共识的开始, 其中 $Block$ 指需要被委员会验证的区块, $sign$ 指区块打包节点对该条请求消息的签名, v 指当前共识所处的视图编号. 收到请求消息后, 委员会节点以该轮被预验证的区块中的随机数作为种子通过线性同余法或 VRF 产生 $[0,m)$ 的随机数 $Nonce_v$, 委员会中节点序号为 $Nonce_v$ 的节点成为视图 v 的领导节点, 其他委员会节点会将预验证意见发送给当前视图的领导节点. 领导节点在收到请求消息后计时一段时间 t_1, 超过 t_1 时间后将不再接收预验证意见, 并

将已收到的预验证意见聚合为聚合意见, 广播给委员会节点. 委员会节点在发送预验证意见后开始计时一段时间t_2, 如果在t_2时间内收到领导节点的聚合意见并验证通过, 委员会节点将结束对该阶段的共识并把区块和聚合意见一同广播到全网. 如果在t_2时间内没有收到领导节点的聚合意见, 则认为领导节点宕机, 共识将返回到领导节点选举阶段, 产生下一个视图的随机数$Nonce_{v+1}$, 选举新的领导节点.

在 PCAS 中, 涉及节点间通信的阶段为: 请求阶段、意见收集阶段、确认阶段和全网广播, 该 4 个阶段的通信复杂度分别为$O(m)$、$O(m)$、$O(m)$、$O(n)$. 在 PBFT 中, 该 4 个阶段的通信复杂度分别为$O(m)$、$O(m^2)$、$O(m^2)$、$O(nm)$. 委员会在确认阶段已完成共识, 全网广播阶段目的是向全网节点广播区块和委员会的共识结果, 故在 PCAS 中, 委员会达成共识的通信复杂度为$O(3m)$, 近似为$O(m)$; 整个预验证流程的通信复杂度为$O(3m+n)$, 由于委员会节点数量m一般很小, 且远远小于全网节点的数量n, 故可近似为$O(n)$. 在 PBFT 中, 委员会达成共识的通信复杂度为$O(m+2m^2)$, 近似为$O(m^2)$; 整个预验证流程的通信复杂度为$O(m+2m^2+nm)$, 近似为$O(nm)$, 故 PCAS 在通信复杂度上优于 PBFT.

此外, 在 PCAS 中, 节点只需存储一条与普通签名大小相同的聚合签名即可记录委员会节点的评估意见, 可以较大减少存储空间占用; 在 PCAS 中, 委员会节点可以投反对意见, 相较于 cosi 聚合签名, PCAS 可以更全面地记录委员会节点的验证行为.

2.3 动态共识

腐败问题会影响委员会的信誉和预验证结果, 导致委员会无法正确代表全网节点进行验证, 故系统需要根据委员会的信誉和预验证结果对共识进行动态切换. 本节提出动态共识, 包括区块预分类和共识切换两部分. 区块预分类用于根据委员会信誉状态和预验证意见对区块进行预分类, 共识切换用于根据预分类结果动态切换用于验证区块的上层共识.

2.3.1 区块预分类

根据委员会的信誉状态及其给出的聚合意见可以将区块分为 3 类: 可信区块、分歧区块和不可信区块. 影响区块分类结果的因素包括: 支持率、反对率和评估得分. 支持率指委员会中投支持票的节点占委员会节点总数的比例, 反对率指委员会中投反对票的节点占委员会节点总数的比例, 评估得分指根据委员会的信誉状态和投票结果计算出来的区块得分.

设委员会对区块$Block_h$的聚合意见为AS_h、支持率α_h、反对率β_h、给出的评估得分为Q_h. 设支持率阈值α_t、拒绝率阈值β_t、评估得分上限阈值Q_1、评估得分下限阈值Q_2、$Block_h$的类型K_h. K_h的值为−1、0、1 时分别代表$Block_h$为不可信区块、分歧区块和可信区块. 若委员会节点V_x对$Block_h$给出支持意见, 则$Block_h$的评估得分Q_h将加上V_x的信誉值$R^i_{V_x}$, 若给出反对意见则减去$R^i_{V_x}$, 若委员会节点V_x的意见缺失, 则不计入. 此外, 在信誉得分的基础上引入通过率和拒绝率作为判断依据, 即区块在$Q_h > Q_1$的基础上需再得到一定比例的委员会节点的支持即$\alpha_h > \alpha_t$才能被分类为可信区块, 从而避免出现少数高信誉恶意节点通过控制信誉得分控制预验证结果的情况. 区块预分类算法见算法 2, 每次预分类需遍历聚合意见中包含的m个委员会节点的意见, 故算法复杂度为$O(m)$.

算法 2. 区块预分类算法.

输入: 委员会对$Block_h$的聚合意见AS_h, 打包$Block_h$时所处的时代i;
输出: $Block_h$的类型K_h.

1. 将α_h, β_h, Q_h, K_h置 0
2. **while** $j=0; j<m; j++$ **do** //遍历聚合意见中包含的每个委员会节点的意见
3. **if** $AS_h \to msg[j]=$ '1' **then**
4. $Q_h = Q_h + R^i_{V_j}$
5. $\alpha_h = \alpha_h + \dfrac{1}{m}$

6. **end if**
7. **if** $AS_h \to msg[j] = \text{‘0’}$ **then**
8. $Q_h = Q_h - R^i_{V_j}$
9. $\beta_h = \beta_h + \dfrac{1}{m}$
10. **end if**
11. **end while**
12. **if** $Q_h > Q_1$ and $\alpha_h > \alpha_t$ **then**
13. $K_h = 1$
14. **end if**
15. **if** $Q_h < Q_2$ and $\beta_h > \beta_t$ **then**
16. $K_h = -1$
17. **end if**
18. **return** K_h

2.3.2 共识切换

在完成区块预分类后, 全网节点会根据预分类结果进行共识切换, 可以分为以下 3 种情况.

(1) 节点在收到不可信区块时, 将直接拒绝该区块并不再广播, 从而避免无意义的带宽消耗.

(2) 节点在收到分歧区块时, 会对区块中随机数和交易的正确性进行二次验证, 并根据二次验证结果选择是否将区块加到链上. 因为分歧区块需要在全网范围内进行二次验证, 故共识速度较慢. 为避免分叉, 需要通过上调挖矿难度来降低出块频率, 故在基于分歧区块挖矿时, 挖矿难度近似于比特币, 此时系统吞吐量较低.

(3) 节点在收到可信区块时, 会判断是否连续出现了 ε 个可信区块. 即对于可信区块 $Block_N$, 需要判断从 $Block_{N-\varepsilon}$ 到 $Block_N$ 是否都为可信区块. 由于算力竞争存在弱一致性, 区块的正确性需要一定时间确认, 故在将全网共识切换到预验证共识 PCAS 之前, 需要 ε 个可信区块来确保经由全网共识验证的区块已经被算力充分确认. 如不满足该条件则继续采用全网共识来验证区块; 如满足该条件, 则称该可信区块为 $Block^{\varepsilon}_N$, 并切换到 PCAS 共识来验证 $Block^{\varepsilon}_N$. 由于节点在收到 $Block^{\varepsilon}_N$ 后无需对区块的正确性进行验证, 只需要验证聚合意见的有效性, $Block^{\varepsilon}_N$ 的验证时间将大大缩短, 共识速度将加快, 进而可以大幅下调以 $Block^{\varepsilon}_N$ 作为父区块挖矿的难度. 若基于 $Block^{\varepsilon}_N$ 打包的下一个区块仍是可信区块则可以继续保持低挖矿难度, 维持高吞吐量. 若在 $Block^{\varepsilon}_N$ 后出现了分歧区块, 则说明委员会中存在恶意节点导致预验证意见出现分歧. 为保证系统安全性, 共识将立即从预验证共识 PCAS 切换到全网共识, 由全网节点对该区块进行验证, 并根据全网节点给出的对照意见对委员会进行迭代, 淘汰委员会中的恶意节点.

3 性能分析

本节将对 DBCP 的安全性、可扩展性和应用场景进行分析. 在安全性方面将分析 DBCP 抵抗双花攻击、女巫攻击和委员会腐败的效果; 可扩展性方面将分析 DBCP 的吞吐量, DBCP 中委员会的迭代收敛速度以及 DBCP 的存储开销; 应用场景方面将分析 DBCP 在公链和联盟链中的性能表现.

3.1 安全性分析

DBCP 中存在两种共识分别是 PoW 和 PCAS. Tian 等人[28]分析了区块链中常见的攻击方式有双花攻击、女巫攻击, 并且 PoW 共识已被证明可较好地应对双花攻击和女巫攻击[29], 因此本节主要分析预验证共识 PCAS 在面对双花攻击、女巫攻击、委员会腐败问题时的表现.

(1) 双花攻击指恶意节点拥有 50% 以上算力, 挖到区块后不进行广播, 而是创造一个不公开且长度大于主链的分支区块链, 当恶意节点成功将主链上的代币提现后, 再广播分支区块链. 由于分支区块链长度大于主链, 主链

将回滚, 被提现的代币将返回到恶意节点账号中, 从而完成双花. 当系统处于预验证共识时, 由于可信区块 $Block_N^e$ 的正确性已被委员会确认, 全网节点不会再认可其他与 $Block_N^e$ 处于相同高度 N 的区块, $Block_N^e$ 所在链不会回滚, 故 DBCP 可以抵抗双花攻击.

(2) 女巫攻击指恶意节点通过创建多个帐户身份或节点从而控制网络. 委员会节点候选池定义为成功打包过区块的节点, 故攻击者伪造的节点必须拥有一定算力并成功打包过区块才可能成为委员会节点, 仅伪造多个身份无法控制委员会, 故 DBCP 可以抵抗女巫攻击.

(3) 委员会腐败指恶意节点通过收买或攻击委员会节点, 在委员会中获得一定的话语权, 进而影响预验证结果. 为降低委员会节点的作恶意愿, 系统在信誉计算部分提高了对高信誉节点作恶的惩罚, 在激励机制部分提高了高信誉节点的收益. 并且在对区块进行预分类时, 在信誉得分的基础上引入通过率和拒绝率作为判断依据, 避免出现少数高信誉节点掌控预验证结果的情况. 一旦委员会中发生腐败问题导致意见出现分歧, 系统将通过动态共识将区块验证权从委员会转移到全网节点, 由全网节点对区块进行验证, 并把预验证共识结果和全网节点共识结果对照, 筛选出腐败节点并淘汰, 故 DBCP 可以快速应对委员会腐败问题.

3.2 可扩展性分析

3.2.1 DBCP 的吞吐量

系统吞吐量受制于区块共识速度, 只增加区块体积或出块频率而不提高共识速度无法提高吞吐量, 反而会造成过多叉链, 影响系统安全性. 本节从区块共识速度的角度研究对比 DBCP、PoW、DPoS 和 Byzcoin 的吞吐量上限.

设区块在全网节点间广播的耗时为 t_p、区块被全网节点验证的耗时为 t_v、区块在委员会内部广播的耗时为 t_w、区块被委员会验证的耗时为 t_c、全网节点验证委员会签名的耗时 t_s. 则 PoW 等全网共识算法的区块共识时间包括 t_p 和 t_v 两部分. DPoS 等委员会共识算法的共识时间包括 t_w 和 t_c 两部分. Byzcoin 等混合共识算法的共识时间包括 t_p、t_c 和 t_s 这 3 部分. 当 DBCP 通过全网共识验证区块时, 区块的共识时间包括 t_p、t_c、t_s 和 t_v 这 4 部分. 通过实验计算出当委员会节点数量为 1 000, 全网节点数量为 100 万时, t_c 与 t_s 的和约 3 s; Decker 测得的比特币区块在 90% 节点同步需要 2.4 min[30], 故 t_c 与 t_s 的和远小于比特币区块全网同步时间, 此时吞吐量上限近似于 PoW 共识. 当 DBCP 通过预验证共识验证区块时, 区块共识时间的组成与混合共识相同, 故可以通过将混合共识的共识速度与委员会共识和全网共识进行对比来分析 DBCP 处于预验证共识时的吞吐量上限.

混合共识与 DPoS 等委员会共识的关键不同点为在委员会共识中区块的出块和验证都由委员会负责, 区块只需在委员会内部同步. 而在混合共识中, 全网节点负责出块, 委员会只负责验证, 区块仍需要大量时间在全网同步, 导致混合共识的吞吐量上限低于委员会共识.

设基于 PoW 共识的比特币网络吞吐量为 P_{bitcoin}. 对于比特币, 区块完成全网共识需要时间为 $t_p + t_v$. 对于混合共识, 区块完成全网共识需要时间为 $t_c + t_p + t_s$. 故混合共识的吞吐量 P_{hybrid} 可以由公式 (4) 给出.

$$P_{\text{hybrid}} = P_{\text{bitcoin}} \frac{t_p + t_v}{t_c + t_p + t_s} \tag{4}$$

影响 t_p 的主要因素是网络带宽, 影响 t_v 的主要因素是区块大小, 影响 t_c 和 t_s 的主要因素是委员会大小和委员会内部采用的共识算法. 上文指出 t_c 与 t_s 的和远小于比特币区块全网同步时间, 故公式 (4) 可以简化为公式 (5). 可以看出混合共识的吞吐量上限主要取决于区块验证耗时 t_v 与区块广播耗时 t_p 的比值.

$$P_{\text{hybrid}} = P_{\text{bitcoin}} \left(1 + \frac{t_v}{t_p}\right) \tag{5}$$

在比特币网络中, 大部分时间用于验证区块及其包含的交易[31]即 $t_v > t_p$, 故混合共识的吞吐量上限较比特币有显著提升. 在实验部分将进一步对比分析 DBCP、Byzcoin 和 PoW 的吞吐量.

3.2.2 委员会的迭代收敛速度

在系统运行初期, 委员会信誉值较低, 且可能存在一定比例恶意节点, 导致委员会对区块给出的评估得分和支持率会偏低, 大部分正确区块会被分类为分歧区块, 影响系统吞吐量. 随着委员会不断迭代, 委员会信誉和诚实节点比例逐步提高, 更多正确区块被分类为可信区块, 由于可信区块不需要被全网验证, 系统吞吐量将达到较高的状

态. 因此, 委员会能否快速收敛到诚实节点比例较高的高可信状态, 直接影响到系统的可扩展性, 下面将探讨影响委员会收敛速度的因素.

影响委员会迭代收敛速度的关键因素是恶意节点历史算力占全网节点历史算力的比例δ. 委员会节点候选池由成功挖矿的节点组成, 故候选池中恶意节点的比例近似为δ. 设第p次迭代后委员会中恶意节点的比例为θ_p, 初代委员会中恶意节点的比例为μ. 假设恶意节点进入委员会后立即作恶, 则第p次迭代淘汰出的恶意节点数量约为$m\theta_{p-1}$, 并从候选池中选入等量的新委员会节点, 选入的新节点中恶意节点的数量为$m\theta_{p-1}\delta$, 故第p次迭代后委员会中恶意节点的比例为$\theta_{p-1}\delta$. 综上所述, 公式 (6)、公式 (7) 成立.

$$\theta_0 = \mu \tag{6}$$

$$\theta_p = \theta_{p-1}\delta(p > 0) \tag{7}$$

由公式 (6)、公式 (7) 可推出公式 (8).

$$\theta_p = \mu\delta^p \tag{8}$$

设当$\theta_p \leqslant \gamma$即委员会中恶意节点比例小于$\gamma$时, 委员会达到高可信状态, 求出$p \geqslant \log_\delta \frac{\gamma}{\mu}$, 即$\log_\delta \frac{\gamma}{\mu}$次迭代后委员会收敛到高可信状态. 设单个区块的全网共识时间为t_{pr}, 则一个迭代周期消耗的时间为Tt_{pr}. 则委员会完成迭代达到收敛需要的时间t_{iter}可由公式 (9) 计算出.

$$t_{iter} = Tt_{pr}\log_\delta \frac{\gamma}{\mu} \tag{9}$$

全网节点数量n越大, 区块广播和验证的时间越长, 委员会节点数量m越大, 委员会内部的共识时间越长,

所以t_{pr}正相关于n和m. 区块链是开放式网络, n的大小无法调控, 可以通过调整m和T的大小来调整委员会迭代速度. m越小, 预验证共识消耗的时间越短, 迭代速度越快, 但过少的委员会节点会导致系统的中心化程度较高. T越小迭代速度越快, 但会增加迭代区块分叉的概率.

假定$\gamma = 0.05$、$\delta = 0.5$、$\mu = 0.5$、$T = 50$. 通过公式 (9) 可以计算出经过约 166 次区块共识后, 委员会将完成迭代达到收敛, 如不出现内部腐败问题, 系统将一直处于高吞吐量状态.

3.2.3 DBCP 的存储开销

预验证机制在区块中引入对照意见等信息会带来额外的存储开销, 本节将通过计算分析 DBCP 额外开销的大小. 设一个区块中平均包含u条对照意见, 系统平均每次迭代淘汰w个委员会节点. 系统采用 BLS 聚合签名和 SHA256 哈希算法, 故对照意见组的哈希占 64 字节, 对照意见中*aggregate sign*占 33 字节, *hash*占 64 字节[26]. 由于预验证结果有 3 种取值, 故需要 2 个位的空间即$\frac{1}{2}$字节来表示, 故*view*占$\frac{1}{2}$字节, *msg*占$\frac{m}{2}$字节, 通过计算得到每条对照意见占$(\frac{m}{2}+97.5)$字节. 同样可通过计算得出迭代区块中的委员会列表及其哈希占$(68m+64)$字节, 淘汰节点列表和其对应的哈希占$(68w+64)$字节. 普通区块带来的额外存储开销S_{cb}可由公式 (10) 计算得出.

$$S_{cb} = u\left(\frac{m}{2}+97.5\right)+64 \tag{10}$$

迭代区块带来的额外存储开销S_{ib}可由公式 (11) 计算得出.

$$S_{ib} = S_{cb}+68(m+w)+128 \tag{11}$$

迭代区块约每T个区块出现一次, 故平均单个区块带来的额外存储开销S_p可由公式 (12) 计算得出.

$$S_p = \frac{((T-1)S_{cb}+S_{ib})}{T} \tag{12}$$

假定$T = 50$、$u = 3$、$w = 1$、$m = 100$, 代入公式 (12) 计算出 DBCP 中平均一个区块的额外存储开销约 647 字节. 比特币区块大小约 0.5 MB[30], 在该条件下 DBCP 中的区块大小相较比特币仅增加了 0.12%.

3.3 应用场景分析

DBCP 是针对大规模、去信任区块链网络中委员会腐败问题设计的动态共识, 包括预验证和动态共识两个核心机制. 本节将分析上述机制的运行前提, 并总结 DBCP 的适用场景.

(1) 在预验证机制中, 节点通过在区块中添加对照意见来评估委员会, 故要求区块打包权归属全网节点, 且要

求系统的底层共识具有高安全性, 以保证区块不可篡改和预验证结果的可靠性.

(2) 在动态共识中, 节点需要根据预验证意见动态调整验证区块的策略, 故要求节点可以快速完成区块验证策略的切换, 以减少对系统性能影响.

以比特币和以太坊为代表的大规模公链网络, 底层共识一般采用 PoW, 安全性强, 区块打包权归属全网节点, 且节点可以自主选择区块验证策略, 故 DBCP 可以较好地应用于上述场景. 但 DBCP 的运行效果受节点规模影响, 节点规模越大, 区块验证权在转移到委员会后, 验证区块所需的时间缩短越多, 吞吐量提升越大, 而以太坊中验证节点数量少于比特币, 故 DBCP 在比特币中的性能表现优于以太坊.

在以 Hyperledger Fabric 为代表的联盟链网络中, 区块的打包仅由 Orderer 节点负责, 普通节点无法打包区块. 其次在 Hyperledger Fabric 中, 验证区块的策略被称为背书策略, 背书策略的修改需要背书节点重新部署链码, 会影响业务正常运行, 故不满足 DBCP 运行的前提条件. 此外, 联盟链网络对节点有准入限制, 并非完全去信任, 节点出现腐败问题的可能性较低, 故 DBCP 不适用于联盟链场景.

4 实　验

本文通过 Golang 语言实现了 DBCP 的出块、广播、验证和委员会选举过程, 并在 CPU 为 Intel(R) Core(TM) i7-10700, 内存为 16 GB 的主机上进行仿真实验, 相关实验参数设定如下.

(1) 网络中存在恶意节点, 恶意节点可能会恶意篡改、丢弃消息; 恶意节点可能会掌握一半以上的哈希算力, 但不会长期掌握一半以上的算力.

(2) Decker 测得比特币网络中 90% 节点带宽在 3M 以上, 50% 节点带宽在 33M 以上[30], 故实验中将节点上下行带宽的默认值设定为 20M, 并在 20–60M 区间内进行对比实验.

(3) Antoni 通过初步实验检测到比特币网络中有超 11 万个对等节点[32], 故实验中将全网节点数量的默认值设定为 10 万并在 10 万–100 万区间内进行对比实验, 委员会的大小默认为 100.

本节从 DBCP 中委员会的迭代收敛速度, PCAS 的共识速度, DBCP 的存储开销, DBCP 的吞吐量, DBCP 的安全性 5 个方面开展实验, 进一步说明 DBCP 相较 Bitcoin、Byzcoin 的优势.

4.1 DBCP 中委员会的迭代收敛速度

本节将通过实验研究恶意节点历史算力对收敛速度的影响. 实验结果如图 5 所示, δ 表示恶意节点历史算力占历史总算力的比值, 横轴为时间, 纵轴为委员会中诚实节点的比例. 设 $\gamma = 0.05$ 即诚实节点比例超过 95% 时, 委员会完成迭代. 设定全网节点的数量 $n = 100000$、委员会节点数量 $m = 100$、迭代周期 $T = 50$, 可以观察到 δ 越大, 委员会迭代速度越慢, 但都可在 100 h 内完成迭代.

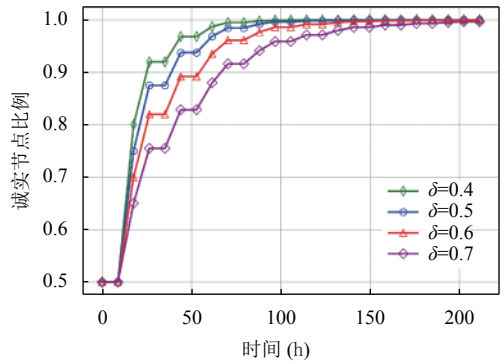
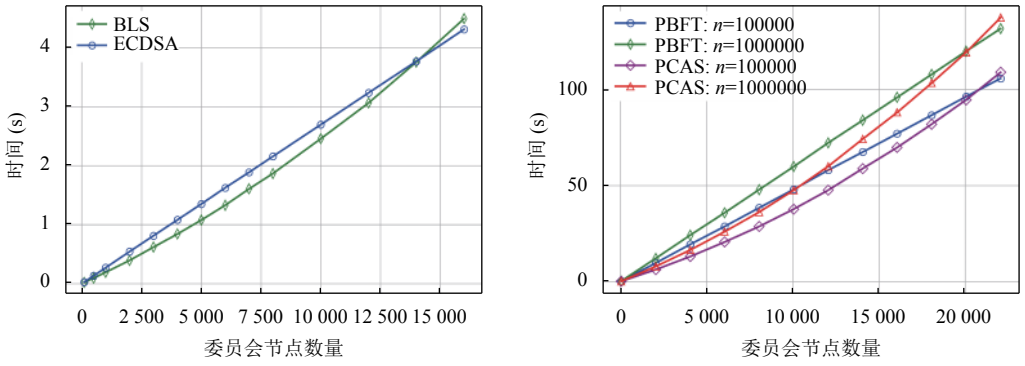


图 5　委员会的迭代收敛速度

4.2 PCAS 的共识速度

本节将通过实验对比基于 BLS 聚合签名的 PCAS 共识和基于椭圆曲线数字签名算法 (elliptic curve digital

signature algorithm, ECDSA) 的 PBFT 共识的共识速度, 签名算法的实现基于 Go 语言高级加密算法库 DEDIS. 如图 6(a), 首先测试 BLS 和 ECDSA 的验证速度, 横轴为参与签名的委员会节点数量, 纵轴为验证签名所需的时间. 由于 ECDSA 是一种非聚合签名, 验证签名的次数与参与签名的节点数量相同, 故随着节点数量增多, ECDSA 所需的验证时间是线性增长的. 而 BLS 呈现出非线性, 在参与签名的节点数量较少时, 可以更快地完成验证, 但随着节点数量增多, BLS 的验证速度将逐渐慢于 ECDSA.



(a) 委员会节点数量对签名验证速度的影响 (b) 委员会节点数量对共识速度

图 6 委员会节点数量对 PCAS 的签名验证速度和共识速度的影响

进一步对 PCAS 和 PBFT 的共识速度进行了实验, 共识速度包括 3 部分: 签名速度、广播速度和验证速度. 如图 6(b), 横轴为参与签名的委员会节点的数量, 纵轴为完成共识所需的时间. 经测试表明, 共识速度中占比最大的为验证速度, 可以看出图 6(b) 的走势接近图 6(a), 且 PCAS 的共识速度在委员会节点数量少于 20 000 时是快于 PBFT 的. 参考 DPoS、Byzcoin 的委员会设置即 DPoS 为 21 个委员会节点, Byzcoin 平均为 500 个委员会节点, 并兼顾系统去中心化程度和吞吐量, 将委员会的节点数量设置在 100–1000 之间较为合适, 在该区间内 PCAS 的共识和验证速度更快.

4.3 DBCP 的存储开销

本节将通过实验说明 PCAS 相较 PBFT 在存储开销上具有优越性. 设定 $T = 50$、$u = 3$、$w = 1$, 横轴为委员会的节点数量, 纵轴为单个区块的存储开销. 如图 7 所示, 在采用 PCAS 共识时, 由于 PCAS 使用聚合签名, 随着委员会节点数量变多, 存储开销不会明显变大. 在采用 PBFT 共识时, 由于 PBFT 采用非聚合签名, 委员会签名会在区块中占据较大空间, 随着委员会的节点数量增多, 存储开销会明显变大, 每增加 100 个委员会节点, 存储开销约增加 4%.

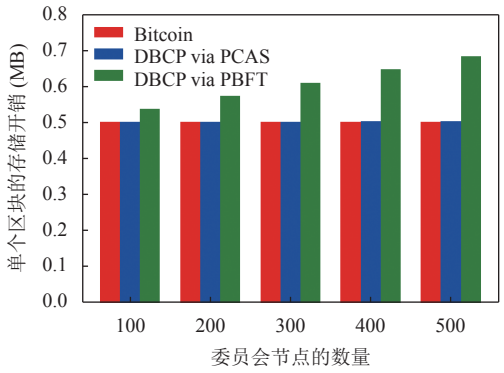
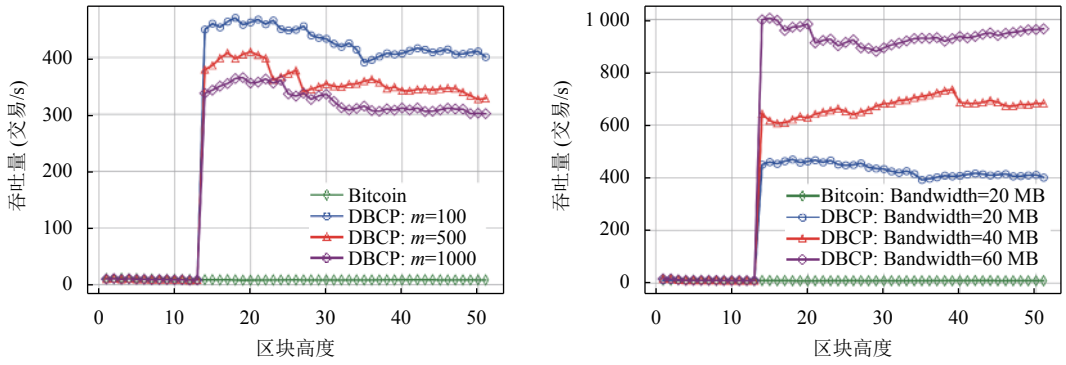


图 7 委员会节点数量对 DBCP 的额外存储开销的影响

4.4 DBCP 的吞吐量

本节将通过实验研究委员会大小和带宽对吞吐量的影响. 如图 8 所示, 横轴为区块高度, 区块高度为 H 代表

第 H 个区块完成打包的时刻, 纵轴为吞吐量, Bandwidth 代表带宽. 在区块高度小于 15 时, DBCP 处于委员会迭代阶段, 全网节点采用 PoW 共识验证区块, 吞吐量近似于比特币, 当迭代完成后, DBCP 将区块验证权转移至委员会, 采用预验证共识验证区块, 区块的验证时间缩短, 吞吐量较大提升, 约为 Bitcoin 的 70–150 倍.



(a) 委员会节点数量对吞吐量的影响 (b) 网络带宽对吞吐量的影响

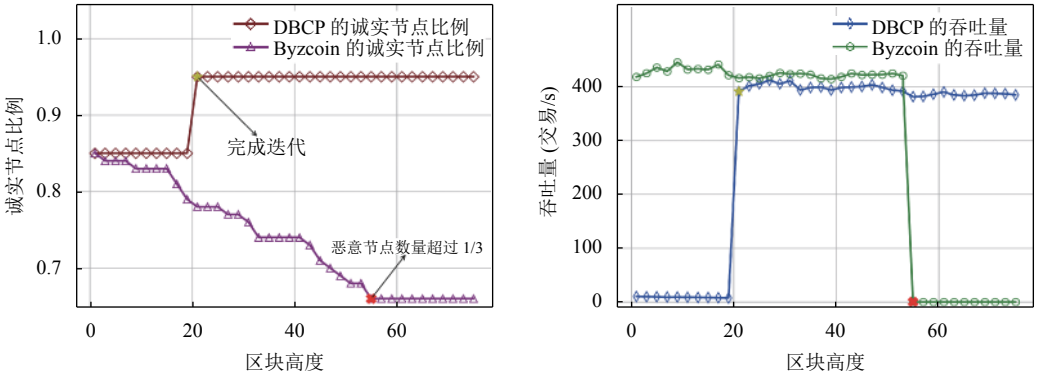
图 8 委员会节点数量与网络带宽对吞吐量的影响

如图 8(a), 设定 Bandwidth = 20 MB 保持不变, 随着委员会节点数量变少, 预验证共识的速度将加快, 系统吞吐量提高, 故委员会节点数量与 DBCP 的吞吐量呈负相关关系. 如图 8(b), 设定 $m = 100$ 保持不变, 随着系统带宽变高, 区块的广播速度将加快, 系统吞吐量提高, 故系统带宽与 DBCP 的吞吐量呈正相关关系.

4.5 DBCP 的安全性

本节从算力攻击和委员会腐败两个方面对比 DBCP 和 Byzcoin 的安全性.

算力攻击的相关设置: 设定 $T = 20$ 、 $m = 100$, 恶意节点掌握 1/3 的算力, 委员会中初始有 15 个恶意节点. 设定 $\gamma = 0.05$ 即 DBCP 在诚实节点比例超过 95% 时完成迭代. 如图 9 所示, 横轴为区块高度, 区块高度为 H 代表第 H 个区块完成打包的时刻, 图 9(a) 纵轴为委员会中诚实节点比例, 图 9(b) 纵轴为系统吞吐量. 如图 9(a), DBCP 在区块高度为 20 时完成迭代, 使诚实节点比例提高到 95% 以上, 此时如图 9(b), DBCP 进入高吞吐量阶段. 如图 9(a), 由于恶意节点掌握 1/3 算力, 故每次出块, 恶意节点有 1/3 的概率进入 Byzcoin 委员会, 导致 Byzcoin 在区块高度为 55 时, 委员会中恶意节点的比例超过 1/3, 超过了拜占庭类共识的容忍度, 此时如图 9(b), Byzcoin 不再出块.

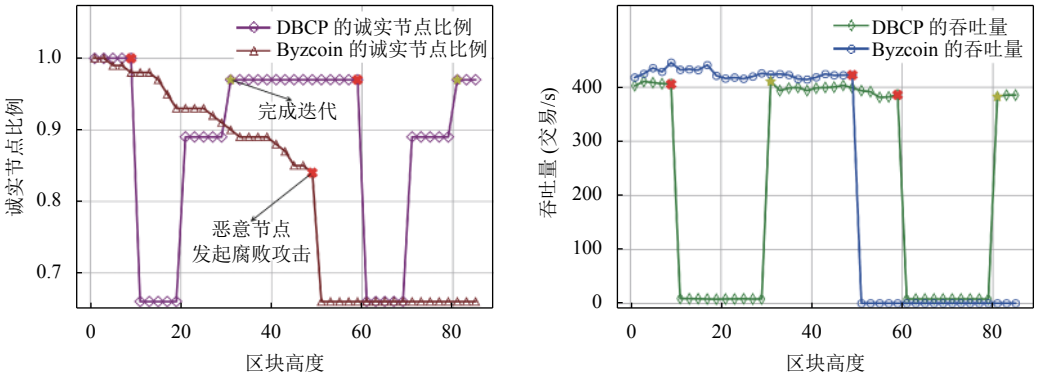


(a) 算力攻击对诚实节点比例的影响 (b) 算力攻击对吞吐量的影响

图 9 算力攻击对 DBCP 和 Byzcoin 的影响

委员会腐败相关设置: 设定 $T = 10$ 、 $m = 100$, 恶意节点掌握 1/3 的算力, 委员会初始化时无恶意节点. 设定 $\gamma = 0.05$ 即 DBCP 在诚实节点比例超过 95% 时完成迭代, 腐败攻击会导致委员中恶意节点比例提升到 1/3 以上.

如图 10(a) 所示, 恶意节点在区块高度为 10、60 时对 DBCP 发起腐败攻击, 在 50 时对 Byzcoin 发起腐败攻击. 当 DBCP 受到腐败攻击后, 会通过动态共识将区块验证权转移到全网节点, 由全网节点对区块进行验证, 故如图 10(b), DBCP 在区块高度为 10 和 60 时, 共识速度变慢, 吞吐量降低. 但 DBCP 经过一段时间的迭代后, 如图 10(a), 诚实节点比例在区块高度为 30 和 80 时, 重新回到 95% 以上, 共识切换到预验证共识, 共识速度加快, 吞吐量也恢复到较高水平. 然而 Byzcoin 由于采用拜占庭类共识, 无法容忍恶意节点超过 1/3 的腐败问题, 如图 10(b), 在区块高度为 50 时, 系统不再出块, 且无法自动恢复.



(a) 委员会腐败对诚实节点比例的影响　　(b) 委员会腐败对吞吐量的影响

图 10　委员会腐败对 DBCP 和 Byzcoin 的影响

实验表明, DBCP 可以快速有效地应对算力攻击和委员会腐败问题, 在吞吐量与 Byzcoin 相似的前提下, 较大的提高了系统的安全性.

5 总　结

本文提出一种带有预验证机制的区块链动态共识算法 DBCP, 通过预验证机制完成了全网节点对委员会的信誉评估, 选举出更可靠的委员会节点, 进一步设计基于聚合签名的预验证共识, 降低了信誉管理的存储开销. 通过动态共识机制, 全网节点可以根据委员会的信誉和预验证意见灵活调整共识算法, 针对可信区块采取预验证共识, 实现可信区块的快速验证, 进而提高系统的吞吐量; 针对分歧区块采取全网共识, 并根据全网节点的共识结果淘汰给出错误意见的委员会节点, 有效地降低委员会腐败问题对系统的影响. 经实验验证, DBCP 的吞吐量较比特币提升两个数量级且与 Byzcoin 相近, 可在一个出块周期内快速应对委员会腐败问题, 安全性优于 Byzcoin.

References:

[1] Hou BB, Chen F. A study on nine years of Bitcoin transactions: Understanding real-world behaviors of Bitcoin miners and users. In: Proc. of the 40th IEEE Int'l Conf. on Distributed Computing Systems. Singapore: IEEE, 2020. 1031–1043. [doi: 10.1109/ICDCS47774.2020.00091]

[2] Wang S, Ouyang LW, Yuan Y, Ni XC, Han X, Wang FY. Blockchain-enabled smart contracts: Architecture, applications, and future trends. IEEE Trans. on Systems, Man, and Cybernetics: Systems, 2019, 49(11): 2266–2277. [doi: 10.1109/TSMC.2019.2895123]

[3] Xu H, Zhang L, Onireti O, Fang Y, Buchanan WJ, Imran MA. BeepTrace: Blockchain-enabled privacy-preserving contact tracing for COVID-19 pandemic and beyond. IEEE Internet of Things Journal, 2021, 8(5): 3915–3929. [doi: 10.1109/JIOT.2020.3025953]

[4] Chi JC, Li Y, Huang J, Liu J, Jin YW, Chen C, Qiu T. A secure and efficient data sharing scheme based on blockchain in industrial Internet of Things. Journal of Network and Computer Applications, 2020, 167: 102710. [doi: 10.1016/j.jnca.2020.102710]

[5] Shen M, Liu HS, Zhu LH, Xu K, Yu HB, Du XJ, Guizani M. Blockchain-assisted secure device authentication for cross-domain industrial IoT. IEEE Journal on Selected Areas in Communications, 2020, 38(5): 942–954. [doi: 10.1109/JSAC.2020.2980916]

[6] Wu YL, Dai HN, Wang H, Choo KKR. Blockchain-based privacy preservation for 5G-enabled drone communications. IEEE Network, 2021, 35(1): 50–56. [doi: 10.1109/MNET.011.2000166]

[7] King S, Nadal S. PPCoin: Peer-to-peer crypto-currency with poof-of-stake. 2012. https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf

[8] Xia Q, Dou WS, Guo KW, Liang G, Zuo C, Zhang FJ. Survey on blockchain consensus protocol. Ruan Jian Xue Bao/Journal of Software, 2021, 32(2): 277–299 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6150.htm [doi: 10.13328/j.cnki.jos.006150]

[9] BitShares-Core Contributors. Delegated proof of stake (DPOS). Bitshares Documentation, 2022: 24.

[10] Castro M, Liskov B. Practical byzantine fault tolerance. In: Proc. of the 3rd Symp. on Operating Systems Design and Implementation. New Orleans: USENIX Association: 1999. 173–186. [doi: 10.5555/296806.296824]

[11] Xu GX, Liu Y, Khan PW. Improvement of the DPoS consensus mechanism in blockchain based on vague sets. IEEE Trans. on Industrial Informatics, 2020, 16(6): 4252–4259. [doi: 10.1109/TII.2019.2955719]

[12] Li WY, Feng CL, Zhang L, Xu H, Cao B, Imran MA. A scalable multi-layer PBFT consensus for blockchain. IEEE Trans. on Parallel and Distributed Systems, 2021, 32(5): 1146–1160. [doi: 10.1109/TPDS.2020.3042392]

[13] Pass R, Shi E. Hybrid consensus: Efficient consensus in the permissionless model. IACR Cryptology ePrint Archive, 2016, 2016: 917.

[14] Kogias EK, Jovanovic P, Gailly N, Khoffi I, Gasser L, Ford B. Enhancing bitcoin security and performance with strong consistency via collective signing. In: Proc. of the 25th USENIX Conf. on Security Symp. Austin: USENIX Association, 2016. 279–296. [doi: 10.5555/3241094.3241117]

[15] Abraham I, Malkhi D, Nayak K, Ren L, Spiegelman A. Solida: A blockchain protocol based on reconfigurable byzantine consensus. In: Proc. of the 21st Int'l Conf. on Principles of Distributed Systems. Lisbon: Leibniz-Zentrum für Informatik, 2016.

[16] Xu MX, Yuan C, Wang YJ, Fu JH, Li B. Mimic Blockchain—Solution to the security of blockchain. Ruan Jian Xue Bao/Journal of Software, 2019, 30(6): 1681–1691 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5744.htm [doi: 10.13328/j.cnki.jos.005744]

[17] Kokoris-Kogias E, Jovanovic P, Gasser L, Gailly N, Syta E, Ford B. Omniledger: A secure, scale-out, decentralized ledger via sharding. In: Proc. of the 2018 IEEE Symp. on Security and Privacy. San Francisco: IEEE, 2018. 583–598. [doi: 10.1109/SP.2018.000-5]

[18] Lai YX, Bo ZX, Liu J. Research on sybil attack in defense blockchain based on improved PBFT algorithm. Journal on Communications, 2020, 41(9): 104–117 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2020170]

[19] Decker C, Seidel J, Wattenhofer R. Bitcoin meets strong consistency. In: Proc. of the 17th Int'l Conf. on Distributed Computing and Networking. Singapore: ACM, 2016. 13. [doi: 10.1145/2833312.2833321]

[20] Syta E, Tamas I, Visher D, Wolinsky DI, Jovanovic P, Gasser L, Gailly N, Khoffi I, Ford B. Keeping authorities "honest or bust" with decentralized witness cosigning. In: Proc. of the 2016 IEEE Symp. on Security and Privacy. San Jose: IEEE, 2016. 526–545. [doi: 10.1109/SP.2016.38]

[21] Huang HW, Huang ZY, Peng XW, Zheng ZB, Guo S. MVCom: Scheduling most valuable committees for the large-scale sharded blockchain. In: Proc. of the 41st IEEE Int'l Conf. on Distributed Computing Systems. Washington: IEEE, 2021. 629–639. [doi: 10.1109/ICDCS51616.2021.00066]

[22] Liu YZ, Liu JW, Wu QH, Yu H, Hei YM, Zhou ZY. SSHC: A secure and scalable hybrid consensus protocol for sharding blockchains with a formal security framework. IEEE Trans. on Dependable and Secure Computing, 2022, 19(3): 2070–2088. [doi: 10.1109/TDSC.2020.3047487]

[23] Xiao Y, Zhang N, Lou WJ, Hou YT. A survey of distributed consensus protocols for blockchain networks. IEEE Communications Surveys & Tutorials, 2020, 22(2): 1432–1465. [doi: 10.1109/COMST.2020.2969706]

[24] Huang CY, Wang ZY, Chen HX, Hu QW, Zhang Q, Wang W, Guan X. RepChain: A reputation-based secure, fast, and high incentive blockchain system via sharding. IEEE Internet of Things Journal, 2021, 8(6): 4291–4304. [doi: 10.1109/JIOT.2020.3028449]

[25] Huang CY, Zhao YJ, Chen HX, Wang X, Zhang Q, Chen YJ, Wang HX, Lam KY. ZkRep: A privacy-preserving scheme for reputation-based blockchain system. IEEE Internet of Things Journal, 2022, 9(6): 4330–4342. [doi: 10.1109/JIOT.2021.3105273]

[26] Micali S, Rabin M, Vadhan S. Verifiable random functions. In: Proc. of the 40th Annual Symp. on Foundations of Computer Science. New York: IEEE, 1999. 120–130. [doi: 10.1109/SFFCS.1999.814584]

[27] Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing. Journal of Cryptology, 2004, 17(4): 297–319. [doi: 10.1007/s00145-004-0314-9]

[28] Tian GH, Hu YH, Chen XF. Research progress on attack and defense techniques in block-chain system. Ruan Jian Xue Bao/Journal of Software, 2021, 32(5): 1495–1525 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6213.htm [doi: 10.13328/j.cnki.jos.006213]

[29] Zhang R, Preneel B. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security. In: Proc. of the 2019 IEEE Symp. on Security and Privacy. San Francisco: IEEE, 2019. 175–192. [doi: 10.1109/SP.2019.00086]

[30] Croman K, Decker C, Eyal I, Gencer AE, Juels A, Kosba A, Miller A, Saxena P, Shi E, Sirer EG, Song D, Wattenhofer R. On scaling

decentralized blockchains. In: Proc. of the 2016 Int'l Conf. on Financial Cryptography and Data Security. Christ Church: Springer, 2016. 106–125. [doi: 10.1007/978-3-662-53357-4_8]

[31] Decker C, Wattenhofer R. Information propagation in the Bitcoin network. In: Proc. of the 2013 IEEE P2P. Trento: IEEE, 2013. 1–10. [doi: 10.1109/P2P.2013.6688704]

[32] Donet Donet JA, Pérez-Solà C, Herrera-Joancomartí J. The Bitcoin P2P network. In: Proc. of the 2014 Int'l Conf. on Financial Cryptography and Data Security. Christ Church: Springer, 2014. 87–102. [doi: 10.1007/978-3-662-44774-1_7]

附中文参考文献:

[8] 夏清, 窦文生, 郭凯文, 梁赓, 左春, 张凤军. 区块链共识协议综述. 软件学报, 2021, 32(2): 277–299. http://www.jos.org.cn/1000-9825/6150.htm [doi: 10.13328/j.cnki.jos.006150]

[16] 徐蜜雪, 苑超, 王永娟, 付金华, 李斌. 拟态区块链——区块链安全解决方案. 软件学报, 2019, 30(6): 1681–1691. http://www.jos.org.cn/1000-9825/5744.htm [doi: 10.13328/j.cnki.jos.005744]

[18] 赖英旭, 薄尊旭, 刘静. 基于改进PBFT算法防御区块链中sybil攻击的研究. 通信学报, 2020, 41(9): 104–117. [doi: 10.11959/j.issn.1000-436x.2020170]

[28] 田国华, 胡云瀚, 陈晓峰. 区块链系统攻击与防御技术研究进展. 软件学报, 2021, 32(5): 1495–1525. http://www.jos.org.cn/1000-9825/6213.htm [doi: 10.13328/j.cnki.jos.006213]

侯凯祥(1998－), 男, 硕士, 主要研究领域为区块链共识, 工业物联网.

周晓波(1984－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为无线网络, 数据中心网络, 车联网.

邱铁(1980－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为 5G 智慧物联网, 工业物联网, 智能化分布式区块链.

池建成(1994－), 男, 博士生, 主要研究领域为工业物联网, 区块链.

徐天一(1989－), 男, 博士生, 工程师, CCF 专业会员, 主要研究领域为物联网, 区块链, 数据挖掘.