计算机科学
COMPUTER SCIENCE

基于信誉的区块链分片共识方案

王梦楠，黄建华，邵兴辉，麦勇

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210800227

基于信誉的区块链分片共识方案

王梦楠[1]　黄建华[1]　邵兴辉[1]　麦　勇[2]
1 华东理工大学信息科学与工程学院　上海 200237
2 华东理工大学商学院　上海 200237
(mnwang0105@163.com)

摘　要　分片是一种解决区块链扩容问题的技术,但是分片可能会导致恶意节点更容易集中在单个分片内,从而阻碍整个系统的安全运行。文中提出了一种基于信誉的区块链分片共识协议,通过建立信誉机制来衡量节点行为,促使节点遵循协议,并通过基于信誉等级的分片方法来减小各分片节点信誉等级分布的差异,防止恶意节点集中在单一分片进行作恶。提出一种验证链和记录链相结合的双链模型,该模型通过交易信息的差异化存储,在扩展区块链存储容量的同时提高了区块链的安全性。将投票份额与节点信誉相关联,同时差异化节点承诺,提出了基于信誉的快速拜占庭容错共识算法,使诚实节点更快达成共识,并减小恶意节点的影响。安全性分析表明,RCBSP能够保证分片内节点分布的合理性和共识过程的安全性,防止双花攻击、无利害关系攻击。实验结果表明,RBSCP在保证安全性的前提下,能够做到低分区时延、低共识时延和高吞吐量。

关键词:区块链;分片;信誉机制;双链模型;共识协议
中图法分类号　TP309

Reputation-based Blockchain Sharding Consensus Scheme

WANG Meng-nan[1],HUANG Jian-hua[1],SHAO Xing-hui[1] and MAI Yong[2]
1 School of Information Science and Engineering,East China University of Science and Technology,Shanghai 200237,China
2 School of Business,East China University of Science and Technology,Shanghai 200237,China

Abstract　Sharding is a technology that solves the problem of blockchain capacity expansion. However,sharding may make it easier for malicious nodes to be concentrated in a single shard,thus hindering the safe operation of the entire system. This paper proposes a reputation-based sharding consensus protocol(RBSCP),which establishes a reputation mechanism to measure node behavior and encourage nodes to follow the protocol. The reputation level-based sharding method reduces the difference in the reputation level distribution in different shards,so as to prevent malicious nodes from concentrating on a single shard to do evil. A double-chain model combining verification chain and record chain is proposed. Through the differentiated storage of transactions,the storage capacity of the blockchain is expanded while the security of the blockchain is improved. By associating the voting shares with the node reputation and differentiating the node commitments,a reputation-based fast Byzantine fault tolerance (RFBFT) algorithm is proposed,which enables honest nodes to reach consensus faster and reduces the impact of malicious nodes. Security analysis shows that RBSCP can guarantee the rationality of node distribution in shards and the security of consensus process,and prevent double spend attack and nothing at stake attack. Experimental results show that RBSCP can achieve low sharding latency,low consensus latency and high throughput under the premise of ensuring security.

Keywords　Blockchain,Sharding,Reputation mechanism,Double-chain model,Consensus protocol

1　引言

近年来,区块链的研究和应用受到广泛关注,在金融、供应链、物联网等领域提出了不少解决方案。但目前区块链技术仍处于探索阶段,研究的重点在于通过改进共识算法来提升区块链的交易吞吐量、扩展区块链容量、保障区块链安全[1]。目前公有链的主流共识算法为工作量证明(Proof of Work,PoW)[2]和权益证明(Proof of Stake,PoS)[3]。其中,PoW的优点在于共识简单,节点数越多安全性越好,但PoW需要消耗算力来争夺出块权,造成吞吐量低,消耗能源过大,并且随着技术升级会引发算力不断集中,导致产生较大的安全隐患。PoS解决了PoW的能耗问题,提高了共识效率,但

到稿日期:2021-08-25　返修日期:2022-03-01
基金项目:国家自然科学基金(61472139)
This work was supported by the National Natural Science Foundation of China(61472139).
通信作者:黄建华(jhhuang@ecust.edu.cn)

又出现了无利害关系攻击、币龄累积攻击等问题。另一种典型的共识算法是实用拜占庭容错算法(Practical Byzantine Fault Tolerance,PBFT)[4],其优点是时延小,吞吐量较高,是商业应用的主流共识算法。但 PBFT 通信复杂度较高,随着节点数的增加,通信量呈指数增长,造成很大的网络开销,网络扩展受到极大限制。

分片(Sharding)技术为打破区块链僵局提供了一个解决方案。分片的主要思想是将节点划分到不同的分区,分而治之,使各个分区能够独立处理交易,并行执行共识,分区越多,吞吐量越大。经典的工作诸如 ELASTICO[5] 和 Zilliqa[6] 都在进行分片的尝试,采用 PoW 算法将网络节点进行分片,此方法的优势是简单、方便,但是 PoW 难度值的设置难以在效率和安全性方面取得平衡。OmniLedger[7] 采用了 RandHound 方案来产生无偏随机数,虽然能够保证分区的随机性,但是分片后单一分片内节点数量会减少,可能会导致恶意节点集中在单个分片内发动 51%的攻击,进而影响整个区块链系统的安全。

针对目前分片方案以及共识算法的不足,本文提出了基于信誉的分片共识方案(Reputation-based Sharding Consensus Protocol,RBSCP)。该方案将信誉机制作为系统运行的基础,对不同节点赋予不同影响力,以削弱恶意节点的作恶能力,同时将区块链存储与共识投票分离,保证了节点分片共识的不可预测性,提高了共识的安全性。本文的主要贡献如下:

(1)提出了地址分片和信誉分片两种分片相结合的分片方案,将节点的区块存储与参与共识分开,避免重新分区时造成大量的数据迁移。

(2)提出了基于信誉等级的分片方法,通过对节点进行初评级,避免恶意节点集中在单一分片造成分片接管问题,并显著提高了分片效率。

(3)设计了包含记录链和验证链的双链模型,通过交易的差异化存储以及共享验证信息,扩展了区块链的存储容量,保证了区块链的安全性。

(4)提出了基于信誉的快速拜占庭容错共识方案(Reputation-based Fast Byzantine Fault Tolerance,RFBFT),该方案通过将投票权重与信誉值相关联,以带权重的秘密共享方案完成节点的聚合签名,保证了共识的公平性,并引入可信执行环境 TEE(Trusted Execution Environment)来保证秘密分发的安全性,削弱了恶意节点的影响力,在降低通信复杂度的同时,加快了共识效率,让诚实节点更快达成共识。

2 研究现状

2.1 共识算法

共识算法是保障区块链性能和安全性的关键,一直受到学术界的广泛关注。最经典的共识算法为比特币所使用的工作量证明(PoW),但其吞吐量低,易出现分叉,攻击者只要掌握足够多的算力就可以发动 51% 的攻击并接管区块链。Eyal 等提出了 Bitcoin-NG[8],该算法根据区块的用途将区块分为用来选择出块者的关键块和用来包含交易的微块,提高了吞吐量,但依然无法避免 PoW 机制的缺陷,当矿工的收入来源即交易费越来越少时,矿工会选择不再维护区块链一致性而导致公地悲剧[9]。为了解决 PoW 的高能耗问题,避免公地悲剧,基于权益证明(PoS)的共识算法被提出。最早引入 PoS 的项目是 PPCoin,它通过节点持有的权益及持有时长计算币龄,币龄越大越容易出块,但节点可能会为了出块权长时间离线累计币龄,没有充分的激励很难保持节点在线维护区块链,权益较小的节点也可能会发动无利害关系攻击。委托权益证明(Delegated Proof of Stake,DPoS)[10] 是对 PoS 的改进,DPoS 选出委员会轮流出块,降低了计算成本,但会导致中心化问题。

除了以上基于证明的共识算法以外,另一类经典共识算法是拜占庭容错算法。最经典的实用拜占庭容错协议 PBFT 支持每秒数万次的事务处理,但是由于副本节点之间需要互相通信,时间复杂度达到 $O(n^2)$,可扩展性较差,因此不能直接用于公有链。为了降低通信复杂度,研究人员提出了许多的拜占庭容错协议改进方案。可扩展拜占庭容错协议(Scalable Byzantine Fault Tolerance,SBFT)[11] 采用了聚合签名的方式,领导者通过收集节点的签名恢复一个总的门限签名来降低通信量。FastBFT[12] 通过将可信执行环境与轻量级秘密共享相结合的方式来实现消息聚合,由于该方法不需要公钥操作,因此降低了消息聚合过程中的计算量和通信开销。Proteus[13] 将共识过程与验证过程分别分配给委员会以及其他常规节点完成,通过构建子集共识的方法来降低通信量。然而,这些改进后的 BFT 共识协议中恶意节点仍然拥有和正常节点相同的话语权,并且无法识别恶意节点并将其排除,可能会导致恶意节点联合破坏系统安全性。

为了识别区块链节点的性质,一些学者将信任度引入共识算法,把节点信任值作为判断节点行为的依据,以保证区块链的安全。T-PBFT[14] 中采用 EigenTrust[15] 来评估节点行为,从而可以选出高质量节点组成共识组以保障共识过程的安全。但如果 EigenTrust 评估过程由不可靠的用户来执行,则可能导致信誉评估不准确,这种不准确性在识别一组用户时会增加忽略恶意用户的风险。CertChain[16] 是一种基于可靠性等级的共识和激励机制,它考虑了经济利益和不当行为,但其是为认证机构(CA)量身定制的,因而无法解决可扩展性问题。文献[17]将出块难度与信誉相结合,但挖矿产生的能耗问题仍无法避免,并且难度值设置过低易发生分叉,过高又容易导致出块权集中。文献[18]和文献[19]将信誉机制和 PBFT 相结合,选取一部分信誉较高的节点进行共识,从而降低通信量。文献[20]提出了基于信誉的拜占庭容错 RBFT 算法,该算法将节点被选为主节点的概率与信誉值相结合,区分了不同信誉节点的话语权;同时,设定了信任状态的转换,删除信誉低的故障节点以维护区块链稳定。然而其共识过程的通信复杂度较高,可扩展性较差,无法应用到公有链环境中。

2.2 基于分片的协议

区块链存在“不可能三角”,难以同时兼顾安全性、可扩展性以及去中心化。目前,分片的方案打破了这一困境,但同时也带来了新的问题,如分区的随机性以及分区内的安全性等。目前主流的分片方法为基于 PoW 的方式,如 ELASTICO 和

Zilliqa。虽然 PoW 分片算法简单,但计算难度值难以设置,难度值较小可能会使节点通过多次计算进入对己有利的分片,从而引发安全性问题,难度值较大则会造成能源浪费,引发效率问题。

分片后,各分片内的节点数量大幅减少,单分片内恶意节点更容易发动 51%攻击,造成单分片接管,因此,分片虽然能提升区块链吞吐量,但也造成了分片脆弱性问题。为了防止分区后节点勾结,OmniLedger 和 Rapidchain[21]定期进行重分区,但是其导致的数据迁移将产生巨大的带宽消耗。Monoxide[22]提出了基于分片的异步共识组,并提出 Chu-ko-nu 连弩挖矿,允许矿工同时参与多个链的出块,避免了单个节点的算力在分片内占比过大,使单个分片内恶意节点发动 51%攻击的门槛提升至全网算力的 51%,但由于其采用 PoW 进行出块,仍然无法兼顾安全和能耗。文献[23]为了避免随机分片导致的单分片恶意节点集结,基于遗传算法(Genetic Algorithm,GA),根据节点的信誉值进行分区,使得每个分区的信誉值差异最小化,实现了一个比较完美的信誉平衡。但是在分片过程中可能会耗费很长的适应和计算时间,造成区块链系统分片过程效率低下。

综上所述,目前缺少一种完整、系统的分片共识方案,既实现快速、随机且安全的分片,避免分片后单个分片节点减少导致的安全隐患,又使片内共识算法能兼顾共识效率、安全性以及公平性。

3 RBSCP 分片共识协议概述

本文提出了基于信誉的分片共识协议 RBSCP,用信誉值评估节点行为,将划分的信誉等级作为分片的依据,使各分区信誉等级分布近似,减少恶意节点聚集带来的安全隐患,从而保障分区内共识的安全性。本文采用双链区块链模型,通过差异化交易存储来扩展区块链的存储容量。

3.1 系统设置与结构

RBSCP 中将节点的存储和共识区分开,共设置了两种分片方式:地址分片和信誉分片。地址分片根据节点的地址进行计算,将其划分成一系列分片。节点进行地址分片后被称为某分片的原节点,仅存储该分片内的交易,从而实现交易的差异化存储,扩展整个区块链的存储容量。信誉分片是将节点共识的投票权划分到各个分片,决定节点在哪个分片内参与共识,也称为某分片的共识节点。信誉分片是将节点按照信誉等级进行划分,通过"先初评估,再随机均分"的方法保证各分片的节点信誉等级分布差异的平衡性,实现投票权分布的不可预测性。前者实现了区块链的扩容,后者则保证了各分片内共识节点的数量平衡以及组成的不可预测性和安全性。通过将存储和共识投票分离的方式,可以在区块链规模扩展的情况下,保证单个节点的存储和计算负担合理且均衡。

RBSCP 中设计了两种角色:使用者和持信者。使用者主要通过使用区块链基础设施完成转账或者合约调用,持信者则需要运行 RBSCP 参与共识,对区块链的安全负责。当新节点加入区块链系统时,若想成为持信者,首先需要向系统公有账户缴纳超过一定数额的押金,从而加入信誉列表 $RpList$ 中。该列表内每一项元素的内容包括持信者的公钥地址、押金数、IP 地址、信誉值、参与共识次数及作恶次数。由于公有账户只能转进,不能主动转出,因此当持信者想要退出时,需要构造一个交易请求。节点退出后,其押金以及获利等将延迟返还,以防止节点频繁进出网络。通过缴纳押金,一方面可以防止女巫攻击,另一方面则作为奖惩机制的经济基础。正常节点共识后可以获得与其押金成正比的奖励,而恶意节点被举报后将给予押金削减的处罚。如果恶意节点的作恶导致共识失败,则其押金被扣除并作为补偿奖励给正常共识的节点。理性的节点考虑到自身利益将会自觉地遵守规则,维护区块链系统的稳定。

网络初运行时,需对网络进行地址分片,即对持信者 p_i 的地址计算 $LSB_k(hash(address_{p_i}))$,此公式对地址进行哈希运算以后,取后 k 位来决定节点所处的分区号。在将持信者划分到多个独立的分片以后,各个分片独立并行地处理分片内的交易。在每一个 epoch 各分区共识开始之前,需要根据持信者所处信誉等级进行信誉分片,决定节点在哪个分区有投票权,这一过程为信誉分片。

在区块链结构上,本文采取了双链模型,包含验证链和记录链。各分区以地址分片为基础,存储本分区内的交易记录,形成各分区独有的记录链,由本分片内部存储。记录链中的区块头以及验证信息形成验证块,经过组合以后形成验证链,其中主要记录了相关验证信息(如 merkel 根)以及各分片共识投票信息等,全网保持一致,由所有持信者存储。双链模型可以通过差异化交易记录存储,减轻节点的存储压力,从而扩展区块链存储容量。各分区对双链的存储示意图如图 1 所示。

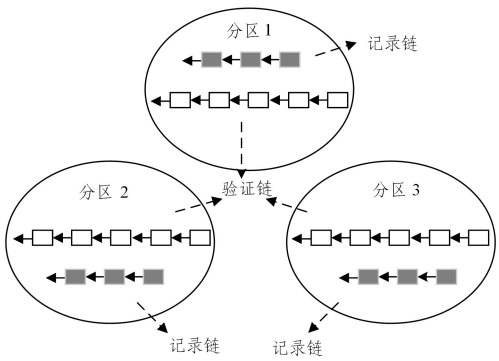

图 1 双链模型

Fig. 1 Double-chain model

在网络模型方面,假设网络是半同步的,通信时延是任意但有限的。在攻击模型方面采用理想模型,即参与者会理性地根据自身利益来决定是否发起攻击。本文假设全网恶意节点信誉值比重小于 1/3,在此假设下,信誉分片后的每个分片内,恶意节点的信誉比小于 1/3,可以正常进行共识。

3.2 RBSCP 方案运行概述

RBSCP 以纪元(epoch)作为大周期运行,将每个 epoch 内划分的多个时隙(slot)作为产生区块的小周期。每个 epoch 开始时,根据信誉列表 $RpList$ 中持信者所处的信誉等级划分分片,从而确定该轮 epoch 持信者可以在哪个

分区具有投票权。

假设所有持信者经过地址计算后被划分进 $n(n\geqslant 3)$ 个分片。根据功能的不同，分片分成创块区、组合区和信誉管理区3种类型。创块区有 $n-2$ 个，负责独立并行处理交易，产生交易块记录本分区内的交易，存储本分区内的记录链。组合区只有1个，主要负责将创块区提交的验证块进行验证和组合，然后将组合后形成的终验块进行全网的广播上链，所形成的验证链将由全网节点存储。信誉管理区也只有1个，主要负责处理各分区提交的信誉报告，从而对 *RpList* 进行计算和管理，并负责在每一个 epoch 结束时形成下个 epoch 的分区策略。RBSCP 的周期运行示意图如图2所示。3种分区的分层共识结构如图3所示。

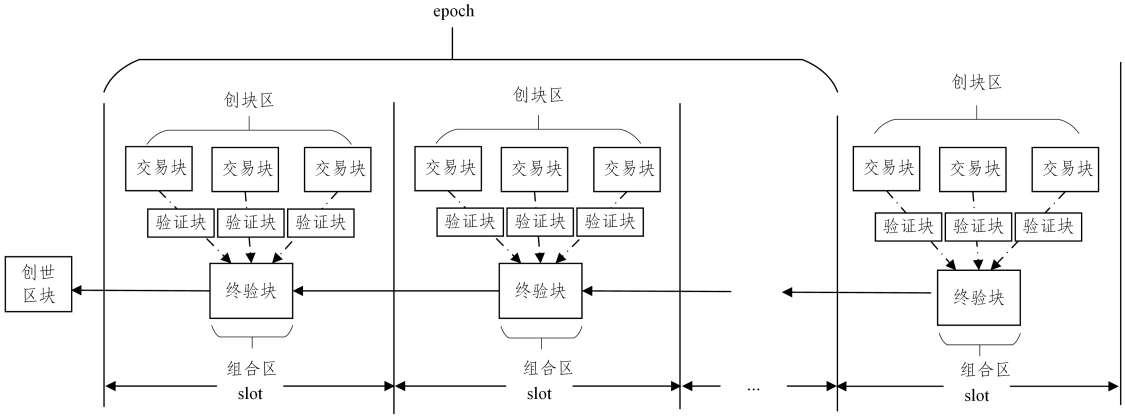

图2 RBSCP 运行示意图

Fig. 2 RBSCP operation diagram

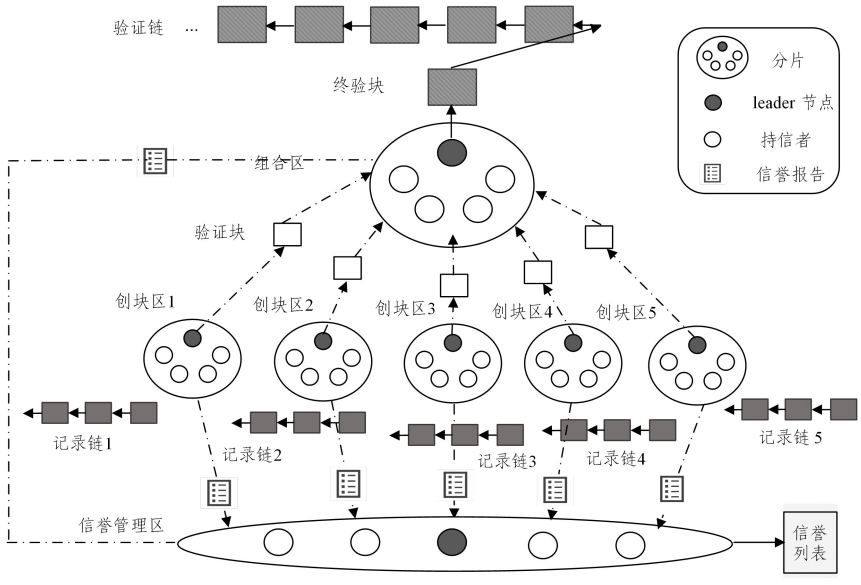

图3 分层共识示意图

Fig. 3 Hierarchical consensus diagram

在每个 epoch 内，持信者的操作如下：

(1)更新信誉列表 *RpList* 和完成信誉分片。由于每一轮 epoch 结束时重新评估了每个持信者的信誉值，并且存在持信者退出网络或者新节点加入网络的可能，因此新一轮 epoch 需要更新 *Rplist*。在每一个 epoch 开始时，各持信者获得和信誉值成正比的投票权重，并按照上轮信誉管理区得出的分区策略分配到本轮参与共识的分区。信誉分片完成后，各个持信者的 TEE 和同一分片内参与共识成员的 TEE 进行相互远程验证，并且在它们之间建立起安全通信通道。

(2)创块区运行一个基于信誉的快速拜占庭共识算法产生交易块。创块区使用 RFBFT 算法进行共识，在每个 slot 分片成员选举一个 leader 来提议区块，使用带投票权重的秘密共享方案进行投票。共识完成后，leader 提交验证块到组合区，并提交一份信誉报告至信誉管理区。如果出现新节点向公有账户缴纳押金请求成为持信者，那么在验证成功后，需要在信誉报告中指出该节点缴纳押金的情况，从而使信誉管理区可以在新一轮 epoch 更新 *RpList* 时将其加入。

(3)组合区产生终验块发布至全网。组合区接收到各个创块区提交的验证块以后，需要把验证块组合形成终验块存储在验证链中。共识过程与创块区类似，组合区先选举 leader，由 leader 验证创块区发送的验证块并且打包组合成终验块。分区内达成共识后，leader 将终验块广播到整个区块链网络中，

由所有持信者保存。同时，leader 需要提交一份信誉报告到信誉管理区。

(4)信誉管理区更新节点信誉值。信誉管理区的 leader 在收到其他分区的信誉报告后，需要根据终验块中的验证信息对信誉报告的内容进行核验，并计算持信者的信誉值，添加新持信者信息，标记申请退出的持信者。leader 将更新过的 *RpList* 以及各分区的信誉报告打包发送给本分区的持信者，通过 RFBFT 共识形成交易块，由本分区存储。

步骤(1)为每个持信者在每个 epoch 之初快速完成，步骤(2)－步骤(4)在每一个 slot 内相继完成，以 slot 为周期循环，直到整个 epoch 结束。在本轮 epoch 最后一个 slot 结束时，信誉管理区将本轮 epoch 的信誉列表进行广播，作为下一轮 epoch 内各分区的第一笔交易进行打包存储。

4 信誉机制

目前基于 PoS 提出的共识方案将押金作为权益份额的决定手段，而把信任度评估仅作为辅助奖惩的依据。但由于经济因素是可控的，投入更多的押金意味着更大的权益或者更大的出块可能性，因此成为权益人的门槛设置并不简单。准入门槛设置过低可能会使恶意者通过投入更多的押金换取更高的权益，门槛设置过高可能使一些想积极参与共识的节点无法加入，因此以经济作为权益决定手段存在一定的弊端。本文提出以信誉作为评价标准来决定一个持信者的投票权重，将缴纳押金作为辅助手段，以防范女巫攻击，并将其作为奖惩机制的经济基础。

4.1 信誉值及信誉等级设置

本文设置了用于记录持信者行为的信誉列表 *RpList*，主要记录持信者当前信誉值、参与共识次数以及作恶次数。信誉值通过信誉函数进行计算，该函数的设计应满足以下几点要求：

(1)信誉值评估和时间有关。通过调整当前轮信誉值以及以往信誉值所占比重，可以体现其是更看重近期行为或是更倚重节点以往的表现。

(2)信誉值的增减速度可根据实际情况具体设置。即奖励系数和惩罚系数可以变动，一般为防止节点信誉过高造成的权力集中，节点信誉值的增长应有节制，而作恶时应快速进行缩减惩罚。

(3)不同行为节点信誉增减幅度应差异化。假如持信者存在恶意行为导致信誉下降，则信誉下降的速度与节点以前的行为相关，比如作恶比例大的节点信誉值应当比作恶比例小的下降更快。同理，正确共识比例大的节点信誉增长更快。

节点的行为定义有以下两种：

(1)正确行为。持信者对于区块的签名在本方案中表现为提供门限秘密共享的秘密碎片，在参与秘密重构的过程中提供正确的秘密碎片，则认为其行为是正确的。

(2)恶意行为。对于分区内的 leader，恶意行为指其提议的区块未能得到足够的秘密碎片用于重构秘密以及发布伪造的重构结果。对于持信者而言，恶意行为指其提供伪造的秘密碎片。leader 和持信者的行为均可被监督和验证，一经举报，则信誉管理区可要求 leader 的 TEE 提供原始秘密及秘密碎片的 hash 值以供验证。

具体的信誉度量公式如下：

$$S_t=\begin{cases}0, & p=0,n=0,t=0\\ S_{t-1}+\alpha\times\dfrac{p}{p+n}\times\delta_t-\beta\times\dfrac{n}{p+n}\times\varphi_t, & t>0\end{cases} \tag{1}$$

$$R_t=\frac{1}{1+e^{-S_t}} \tag{2}$$

其中，R_t 指节点在第 t 个 epoch 的信誉值，取值范围为(0,1)，S_t 为影响因子，α 表示奖励系数，β 表示惩罚系数，p 表示正确行为数，n 表示错误行为数。p 和 n 由信誉管理区依据各分区提交的信誉报告来统计，并更新 *RpList* 以记录各持信者新的信誉值。在第 t 轮共识中节点表现正常，则 δ_t 为 1，φ_t 为 0；如果作恶则取值相反。一般奖励系数 α 比惩罚系数 β 小很多，目的是防止持信者信誉增加过快而造成权力过度集中，而作恶将加大惩罚力度使信誉快速下降，从而促使持信者坚持理性共识。从式(1)和式(2)可以看出，持信者初始状态下，信誉值可设为 0.5，意为该节点将来进行正确共识的可能性与作恶的可能性各占一半。

R_t 公式的原型为 sigmoid 函数，从初始状态开始的增长或减少都呈现先快后慢的趋势，在 S_t 趋近于正无穷或负无穷时，函数变化趋势会趋于平缓。由于 sigmoid 函数中期增长较快，节点信誉值可能会在一两轮内就达到优秀等级，使得恶意节点只需伪装少数几轮便可以获得较高的权限进行作恶。为了防止信誉计算的初期信誉值增长过快[24]，需要通过偏重系数 k 来对影响因子 S_t 进行修正。修正后的影响因子记为 S_t^c，修正公式如下：

$$S_t^c=\begin{cases}0, & p=0,n=0,t=0\\ k\times S_{t-1}+(1-k)\times\left(\alpha\times\dfrac{p}{p+n}\times\delta_t-\beta\times\dfrac{n}{p+n}\times\varphi_t\right), & t>0\end{cases} \tag{3}$$

其中，偏重系数 k 的取值范围为(0,1)。如果 k 的值设置较大，说明与当前轮次相比，持信者的信誉计算结果更加看重持信者从前的行为；反之则说明更看重节点当前轮次的行为。k 对于信誉值的影响在于，k 越大，对当前轮次正常行为的影响占比更小，会抑制本轮正常行为的奖励值增长。由于理论上节点的长期行为相比当前轮次的行为更具有参考价值，故一般令 $k>0.5$。而 k 对于初期信誉增长的调节效果详见 7.1 节信誉调节实验部分。

为了区分不同信誉值的节点的权限，更加快速地管理持信者，按照信誉值的范围将持信者划分成 5 种信誉等级，如表 1 所列。

表 1 信誉等级划分

Table 1 Classification of reputation level

信誉等级	信誉值区间
优秀	$(\alpha,1.0)$
良好	$(0.5,\alpha]$
一般	$(\beta,0.5]$
较差	$[0.1,\beta]$
无效	$(0,0.1)$

为了防止节点持续作恶，在信誉计算式的基础上设置了

0.1 的阈值。当节点信誉下降至 0.1 以后还在继续作恶，信誉值将被直接置为 0，不能再继续参与共识。但从原则上来说，可以给予节点改正的机会。

本文设计了一种信誉恢复机制，如果节点想要继续参加共识，必须交纳双倍定金，并且进入信誉恢复阶段。该阶段节点的信誉会缓慢恢复，但信誉值需要恢复到不小于 0.1 的阈值才能继续参加共识。设置信誉恢复阶段的目的是使之前作恶的节点在一段时间内无法参与共识，拉长其可能作恶的时间间隔，降低作恶频率。假设一个 epoch 的时长为 T，记最后一次共识的信誉值为 R_f，则每经过 T 时间周期，信誉值低于 0.1 的节点的信誉值将上升 r，但不能超过 0.1。这里 r 为恢复系数，决定了节点信誉恢复持续的轮数，理论上的范围为 (0,0.1)。为了降低有作恶史的节点可能作恶的频率，r 值一般设置较小。信誉恢复的公式为：

$$R_r^i=\min\left\{0.1,R_f+r\times\frac{t}{T}\right\},0<r<0.1 \tag{4}$$

4.2 信誉分片

本文提出了基于信誉等级的分片方法，通过信誉值来划分信誉等级，再对各信誉等级的节点进行随机均匀地划分。

RBSCP 基于信誉值将节点划分成 5 种信誉等级，其中等级为无效的节点无法参加共识，其余 4 种信誉等级的节点可以参加信誉分片。由于设置了信誉等级对节点的行为进行初评级，信誉等级为优秀的节点将被认为最有可能继续正确参加共识，信誉等级为一般的节点则被认为其正常共识的可能要大于作恶，以此类推。以初评级为基础进行信誉等级的均分，可以使各分区之间信誉等级分布近似，每个分区的信誉等级分布与总体分布近似，避免恶意节点集中在某个分区内造成的单分片接管。

信誉分片由信誉管理区来完成，信誉管理区将节点按照信誉等级随机均分。在一个 epoch 结束以后更新 $RpList$，将最后一轮 slot 产生的所有 reply secret 进行哈希运算，产生一个无偏随机数 $random$，计算 $k_i=hash(address_{P_i}\mid random)$ 作为节点 P_i 的排序依据。将 $RpList$ 按照信誉等级划分子表，然后将 k_i 按大小排序。假设某个信誉等级共有 h 个节点，将其划分到 n 个分区，另记 $x=h \bmod n$。在不能整除的情况下，为保证各分区内信誉等级分布情况接近，则编号前 x 个分区内划分 $\left\lfloor\frac{h}{n}\right\rfloor+1$ 个节点，其他分区内划分 $\left\lfloor\frac{h}{n}\right\rfloor$ 个节点。

将每个信誉等级的节点随机均分到各个分片，由此决定节点在哪个分片具有投票权。信誉分片的示意图如图 4 所示。

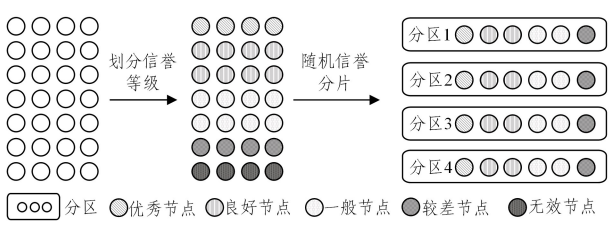

图 4 信誉分片

Fig. 4 Reputation sharding

信誉分片能够保证各分片内节点的等级分布近似相同，如果单分片内出现恶意节点可以联合作恶成功的情况，则说明参与共识的大多数都是恶意节点，系统已经崩溃。文献[23]中基于遗传算法 GA 分片是根据个体进行评估的方法，而 RBSCP 基于信誉等级进行分片则是按集体进行评估，虽然最终分片的结果不能达到 GA 方法的精确度，但误差是可以接受的，且能够更快达到分片目的。

5 RFBFT 共识算法

为了提高分片的共识效率，并保障安全性，本文提出了各分片内使用的共识算法——基于信誉快速拜占庭容错共识算法 RFBFT。通过引入中国剩余定理将信誉与权重相关联，实现将带权重的门限秘密共享方案作为节点聚合承诺，加快速度，并通过差异化诚实和恶意节点的投票权重，削弱恶意节点的影响力，保证共识的公平性。共识中采用了可信执行环境来保障秘密共享过程的可信和安全性。

5.1 可信执行环境(TEE)

可信执行环境(TEE)是在分离内核上运行的防篡改的处理环境[25]。TEE 之所以“可信”，是因为其隔离性和安全存储使其可以在不被常规操作系统干扰的情况下进行可靠的运算，安全地执行应用程序。它不仅能够保证所执行代码的真实性，还能保证运行时状态的完整性；同时，也可以保证存储在持久内存中的数据、代码以及运行时状态的机密性。TEE 还允许远程验证者通过远程认证确定设备的当前配置和行为，向第三方证明其可信度。因此 在 RFBFT 中，假设 TEE 是安全可信的，也就是说，TEE 只可能崩溃而不可能存在拜占庭行为。

5.2 leader 选举

在每个 slot 之初选举 leader 时，本文设计了基于信誉的 follow-the-satoshi 算法以保证 leader 选举的随机性和不可预测性。该算法最初在活动证明(Proof of Activity，PoA)[26]中被提出，通过将权益人被选为 leader 的概率与其所持权益相关，使 leader 的选举无法预测。因为在 follow-the-satoshi 中，每轮的 leader 都是不确定的。虽然权益人的权益越大，就越有可能被选为 leader，但由于选取的随机性，使得攻击者无法知道 leader 的顺序而发起针对性的攻击，更加能保证 leader 选举的安全性。

在 RFBFT 中，引入信誉机制的 follow-the-satoshi 的实现如下：

(1)选取信誉等级为优秀的持信者，按公钥地址顺序排列，按信誉占比划分到[0,1]区间。

(2)取上一个 slot 产生的终验块哈希值后 s 位作为小数位，由其所处的区间决定本轮 slot 的 leader。

设上一个 slot 终验块哈希值后 s 位为 N，所有信誉等级为优秀的持信者共有 h 个，它们的信誉和为 R，被选为 leader 的第 i 位持信者($1\leqslant i\leqslant h$)满足式(5)：

$$\sum_{j=1}^{i}\frac{R_j}{R}\leqslant\frac{N}{10^s}\leqslant\sum_{j=1}^{i+1}\frac{R_j}{R} \tag{5}$$

例如，按公钥顺序排列的信誉等级为优秀的验证者列表为(P_1,0.8)(P_2,0.95)(P_3,0.88)(P_4,0.92)(P_5,0.85)，信誉占比区间如图 5 所示。取上一个 slot 哈希值的后 s 位，如为 57237，则取信誉区间在(0.39773,0.59773]的 P_3 作为新的 leader。

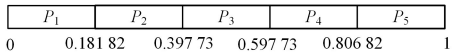

图 5 信誉区间

Fig. 5 Division of reputation

通过对 follow-the-satoshi 引入信誉值，使得 leader 的选举与信誉相关联。节点信誉值越大，就越可能被选为 leader。这里可能性大仅是相对而言，因为信誉等级为优秀的节点信誉值区间在$(x,1)$，换算到占比区间内相差不会很大，因此能避免某一个节点被选为 leader 的次数过多，保证 leader 选举不可预测。

5.3 聚合承诺

在 BFT 类共识方案中，节点投票权重相同，恶意节点的权限与其他节点并无差别，导致恶意节点如果联合作恶，则成功的几率更大。

RFBFT 引入了带权重的门限秘密共享方案。该方案以带权重的动态可验证多秘密共享机制[27]为基础，对节点行为进行信誉评估后，赋予其和信誉成正比的投票权重，然后以共享秘密的重构作为节点的聚合承诺。这种差异化节点话语权的方案可以减小恶意节点在共识过程中的影响力，使诚实的节点能更快达成共识，保证共识的公平性。

方案中所使用的参数如表 2 所列。

表 2 带权重的门限秘密共享参数

Table 2 Threshold secret shared parameters with weight

参数	含义
p	一个大素数
Z_p	有限域
D	分发者
P	参与者
R	重构者
P_i	第 i 个参与者$(1\leqslant i\leqslant n)$
R_i	第 i 个参与者的信誉值
w_i	第 i 个参与者的权重
$hash(\ )$	单向哈希函数
t	秘密重构门限

在共识开始之前，需要对节点信誉值进行权重转换。假设有 n 个持信者，持信者 $i(1\leqslant i\leqslant n)$的信誉值与权重的转换公式如下：

$$w_i=\lceil \rho\times R_i \rceil \tag{6}$$

其中，ρ 为转换系数，可以根据实际情况设置，ρ 值越大，持信者的权重区分度越大，划分的权重值就越大，反之，则权重相差很小。为了保证投票权重的区分度，ρ 值需能使权重与信誉等级相关联，则 $\rho>4$。

本文引入中国剩余定理，并将权重与信誉等级关联，实现了用于聚合承诺的带权重的(t,n)门限秘密共享方案。实现过程包括秘密分发和重构两个阶段，下面分别进行描述。

(1)秘密分发阶段

Step 1 秘密分发者 D 根据总的权重值确定秘密重构门限 t，t 的计算式如下：

$$t=\left\lceil \frac{2}{3}\sum_{i=1}^{n} w_i \right\rceil \tag{7}$$

其中，D 生成两个随机秘密 s_i，并发布每个秘密的加密哈希 $hash(s_i)$。秘密生成多项式如下：

$$f(x)=s_i+\sum_{j=1}^{t-1} a_j x^j \tag{8}$$

可以看出，$f(0)=s_i$。

Step 2 D 选取 n 个不同的随机数 m_i，计算同余方程组：

$$\begin{aligned}
&f(x)\equiv a_{10}+a_{11}x+a_{12}x^2+\cdots+a_{1(w_1-1)}x^{w_1-1} \bmod (x-m_1)^{w_1}\\
&f(x)\equiv a_{20}+a_{21}x+a_{22}x^2+\cdots+a_{2(w_2-1)}x^{w_2-1} \bmod (x-m_2)^{w_2}\\
&\cdots\\
&f(x)\equiv a_{n0}+a_{n1}x+a_{n2}x^2+\cdots+a_{n(w_n-1)}x^{w_n-1} \bmod (x-m_n)^{w_n}
\end{aligned} \tag{9}$$

其中，$(x-m_i)$与$(x-m_j)(1\leqslant i\neq j\leqslant n)$是互素的，系数向量 $\boldsymbol{a}_i$ 可表示为：

$$\boldsymbol{a}_i=(a_{i0},a_{i1},a_{i2},\cdots,a_{i(w_i-1)}) \tag{10}$$

参与者 P_i 的秘密份额由 m_i 和 $\boldsymbol{a}_i$ 共同组成，记为：

$$\boldsymbol{p}_i=(m_i,a_{i0},a_{i1},a_{i2},\cdots,a_{i(w_i-1)}) \tag{11}$$

分发者 D 还需计算一个验证值：

$$V_i=hash(\boldsymbol{p}_i) \tag{12}$$

D 将秘密份额 p_i 发送给参与者 P_i，并公布验证值 V_i。

(2)秘密重构阶段

参与者通过揭示自己的秘密份额来表达自己的承诺。假设参与秘密重构的参与者一共有 k 个，构成集合 $P_r=\{P_1,P_2,\cdots,P_k\}(1\leqslant k\leqslant n)$，秘密重构步骤如下：

Step 1 参与秘密重构的参与者 P_i 发送持有的秘密份额给重构者 R，重构者需要验证秘密份额的有效性。

$$hash(\boldsymbol{p}_i')=V_i \tag{13}$$

若上式成立，则说明份额是有效的，否则认为节点作恶，可以以交易的形式进行举报。

Step 2 重构者 R 验证集合 P_r 的权重和是否达到门限，即判断

$$\sum_{j=1}^{k} w_j=w\geqslant t \tag{14}$$

满足则可以进行重构。

Step 3 根据中国剩余定理对 $f(x)$进行重构。重构者计算：

$$f^*(x)=\sum_{j=1}^{k}\frac{M}{(x-m_j)^{w_j}}\times e_j\times(a_{j0}+a_{j1}x+a_{j2}x^2+\cdots+a_{j(w_j-1)}x^{w_j-1}) \bmod M \tag{15}$$

其中，

$$M=\prod_{i=1}^{k}(x-m_i)^{w_i} \tag{16}$$

$$\frac{M}{(x-m_j)^{w_j}}\cdot e_j\equiv 1 \bmod (x-m_j)^{w_j},1\leqslant j\leqslant k \tag{17}$$

由于 $f(x)$ 和 $f^*(x)$ 均为同余方程组的解，且 $\deg(f^*(x))<w$，根据 $w\geqslant t$ 以及中国剩余定理性质，在$[0,w]$内解是唯一的，则有 $f(x)=f^*(x)$。根据重构出的 $f^*(x)$，计算 $f^*(0)$即可得到秘密。

门限秘密共享方案中，由于使用了 TEE 作为辅助，重构秘密时，对于参与者提供的秘密份额不需要进行重构运算和验证，重构者可直接将该秘密份额进行 hash 运算，然后与 TEE 中加密后的秘密份额进行对比验证，则可以在 O(1)的复杂度内识别该参与者是否存在恶意行为。因此本方案中，计算量集中在多项式取模、求逆元过程中。

秘密份额生成过程中，多项式取模复杂度与门限 t 有关，通过快速傅里叶变换可以在 $O(t\log t)$ 的复杂度内实现求解。秘密重构过程中，多项式求逆元复杂度与模的幂数 w 有关，利用快速傅里叶变换和倍增算法，可在 $O(w\log w)$ 的复杂度内实现求解。

5.4 共识过程

图 6 给出了各分区内 RFBFT 的共识过程，其中 A 为使用者，S_p 为 leader 节点，S_i 为参与共识的持信者。

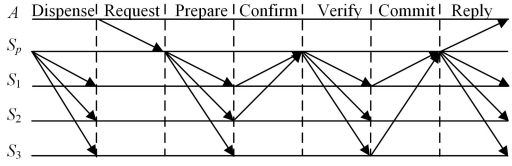

图 6 RFBFT 共识过程

Fig. 6 RFBFT consensus process

下面以创块区为例来说明共识过程的各个阶段，共识过程包括 Dispense，Request，Prepare，Confirm，Verify，Commit 和 Reply 这 7 个阶段。

(1)在 Dispense 阶段，S_p 生成两个随机秘密并发布每个秘密的加密哈希。S_p 根据本分区内的节点投票权重之和决定一个门限值 t，为每个持信者生成一个 m_i，生成两个随机秘密 $verify\ secret$ 和 $reply\ secret$，并发布每个秘密的加密哈希，即 $hash(verify\ secret)$ 和 $hash(reply\ secret)$。假定有 n 个 S_i，S_p 将把每个秘密分成 n 个份额，向每个 S_i 发送一个份额。

为了避免 S_p 利用生成秘密的权力进行作恶，m_i 和两个秘密将由 S_p 的 TEE 产生，在秘密拆分后发送到各 S_i 的 TEE 中。此时 S_i 的 TEE 内有其秘密份额，但 S_i 无法知晓。

(2)在 Request 阶段，由使用者向区块链网络发送交易。持信者需要对交易进行有效性验证，检查交易是否属于本分区、签名是否正确、余额是否充足等。通过了有效性验证后的交易将被放入交易缓冲池。

(3)在 Prepare 阶段，S_p 节点从交易缓冲池中取出交易并打包，产生一个初始交易块。S_p 将初始交易块广播给本分区参与共识的持信者，请求 verify secret 的重构。

(4)在 Confirm 阶段，只有 S_p 的 prepare 消息正确，S_i 的 TEE 才会将秘密份额释放给 S_i，S_i 通过揭示其秘密的份额来表明其承诺。S_p 收集所有此类份额以重建秘密，其代表所有副本的聚合承诺。S_p 收集份额，并通过份额公开的 hash 值验证其有效性，然后有效地达到权重门限的份额，从而进行秘密重构。

(5)在 Verify 阶段，S_p 将重构后的 verify secret 多播给所有 S_i，后者可以根据相应的哈希对其进行验证。

(6)在 Commit 阶段，S_i 在验证秘密之后，将 reply secret 的秘密份额发送给 S_p。使用相同的方法聚合来自所有主动的 S_i 的回复消息，S_p 经过份额验证并在权限满足门限条件后重新构造 reply secret。

(7)在 Reply 阶段，S_p 将重构后的 reply secret 以及 verify secret 的相关信息记录到区块内，形成最终的交易块，并发送给本分区的原节点进行存储，其结构如图 7(a)所示。并形成如图 7(b)结构的验证块，提交给组合区。

图 7 交易块及验证块结构

Fig. 7 Structure of transaction block and verification block

组合区的共识过程与创块区类似，经过基于信誉的 follow-the-satoshi 算法选出 leader，在 prepare 阶段 leader 产生一个初始验证块，然后经过两轮秘密重构，产生如图 8 所示结构的最终验证块。leader 将此验证块广播到整个区块链网络，由所有持信者保存。

图 8 终验块结构

Fig. 8 Structure of final verification block

创块区和组合区在本分区共识完成以后，向信誉管理区发送信誉报告。信誉管理区同样使用 RFBFT 完成共识，产生用于记录信誉报告的交易块。信誉管理区的交易块的结构与创块区的交易块结构类似，区别在于其存储的内容为信誉报告而不是交易。但信誉管理区不产生验证块，而是在每一个 epoch 最后一轮 slot 共识结束以后，将更新后的信誉列表 $RpList$ 发送至全网，作为下一个 epoch 的第一笔交易进行记录。如果新的 epoch 有持信者对 $RpList$ 进行了造假，则其他

持信者可以轻易验证出来。如果有持信者对更新后的 $RpList$ 存疑，则可以要求信誉管理区展示记录链，根据交易块中的信誉报告以及全网的验证链对更新的 $RpList$ 进行验证。

在共识过程中如果出现 leader 的提议错误，未能正确出块，创块区将等待下一轮 slot，由新的 leader 提议交易块，而组合区和信誉管理区在出现错误后将重新选举 leader，直到提议出正确的终验块和 $RpList$。

在 RFBFT 共识过程中主要是 leader 和其他持信者之间进行一对多的通信，而持信者之间无须进行通信，因此通信复杂度仅与持信者个数有关，即 RFBFT 通信复杂度为 $O(n)$。

6 安全性分析

6.1 聚合承诺的安全性分析

本文使用带权重的秘密重构方案作为共识过程中节点的聚合承诺，需要满足正确性和安全性。正确性指权重大于门限 t 即可恢复 $f(x)$，重构出秘密；安全性则指任意权重和小于 t 的参与者无法构造出正确的 $f(x)$，不能得到秘密信息。下面对这两点性质进行证明。

(1)正确性

定理 1[28] 设 $q_1(x),q_2(x),\cdots,q_k(x)$ 为 $K[x]$ 内互素且次数大于等于 1 的多项式，任给 $f_1(x),f_2(x),\cdots,f_k(x)\in K[x]$，必存在 $f(x)\in K[x]$，使得 $f(x)\equiv f_i(x) \bmod (q_i(x))$ $(i=1,2,\cdots,k)$，并且 $f(x)$ 关于 $q(x)$ 是唯一的。其中 $q(x)=q_1(x)q_2(x)\cdots q_k(x)$。

假设共有 k 个参与者参与秘密重构，构成集合 $P_r=\{P_1,P_2,\cdots,P_k\}(1\leqslant k\leqslant n)$。并且有：

$$\sum_{j=1}^{k} w_j = w \geqslant t \tag{18}$$

P_i 向重构者 R 发送自己的秘密份额，

$$\boldsymbol{p}_i=(m_i,a_{i0},a_{i1},a_{i2},\cdots,a_{i(w_i-1)}) \tag{19}$$

则重构者 R 根据这 k 个秘密份额可以构建出一个同余方程组。

$$\begin{aligned}
&f^*(x)\equiv a_{10}+a_{11}x+a_{12}x^2+\cdots+a_{1(w_1-1)}x^{w_1-1} \bmod (x-m_1)^{w_1}\\
&f^*(x)\equiv a_{20}+a_{21}x+a_{22}x^2+\cdots+a_{2(w_2-1)}x^{w_2-1} \bmod (x-m_2)^{w_2}\\
&\cdots\\
&f^*(x)\equiv a_{k0}+a_{k1}x+a_{k2}x^2+\cdots+a_{k(w_k-1)}x^{w_k-1} \bmod (x-m_k)^{w_k}
\end{aligned} \tag{20}$$

根据中国剩余定理可得：

$$f^*(x)=\sum_{j=1}^{k}\frac{M}{(x-m_j)^{w_j}}\times e_j\times(a_{j0}+a_{j1}x+a_{j2}x^2+\cdots+a_{j(w_j-1)}x^{w_j-1}) \bmod M \tag{21}$$

其中 $M=\prod_{i=1}^{k}(x-m_i)^{w_i}$，$e_j$ 满足 $\frac{M}{(x-m_j)^{w_j}}\times e_j=1 \bmod (x-m_j)^{w_j}$，$1\leqslant j\leqslant k$。由于 $\deg(M)=w\geqslant t,\deg(f(x))=t$，根据定理 1 可知，$f(x)$ 关于 M 是唯一的，而 $f(x)$ 和 $f^*(x)$ 都是上述同余方程组的解，因此 $f(x)=f^*(x)$，方案正确性得证。

(2)安全性

假设共有 y 个参与者参与秘密重构，构成集合 $P_n=\{P_1,P_2,\cdots,P_y\}(1\leqslant y\leqslant n)$，并且有：

$$\sum_{j=1}^{y} w_j = w_n < t \tag{22}$$

在重构者 R 收到 P_n 的秘密份额后，重构得：

$$g(x)=\sum_{j=1}^{y}\frac{M}{(x-m_j)^{w_j}}\times e_j\times(a_{j0}+a_{j1}x+a_{j2}x^2+\cdots+a_{j(w_j-1)}x^{w_j-1}) \bmod M^* \tag{23}$$

其中，$M^*=\prod_{i=1}^{y}(x-m_i)^{w_i}$，$e_j$ 满足 $\frac{M^*}{(x-m_j)^{w_j}}\times e_j\equiv 1 \bmod (x-m_j)^{w_j}$，$1\leqslant j\leqslant y$。由于 $\deg(M^*)=w_n<t,\deg(f(x))=t$，根据定理 1 可知，$f(x)\neq g(x)$，有 $f(x)=g(x)+\theta\times M^*$。因此权重和小于 t 的参与者集合无法重构秘密。

6.2 信誉分区的安全性分析

目前的分片方法基本是随机的分片，如基于 VRF 和 MPC 产生的无偏随机数。虽然节点无法预测被分入的分片，保证了一定的安全性，但是由于分片后节点数目减少，恶意节点更容易集结。为了说明这一点，本文通过实验模拟了随机分片情况下，恶意节点数增加导致的分片内出现异常的概率变化。实验设置节点数为 120 个，恶意节点从 0 增加到 60，即恶意节点占比从 0 增加到 1/2，共设置了 4 个分区。随机分片中，当分片内恶意节点超过总数 1/3 时，则认为该分片无法正常进行共识；信誉分片中，当分片内恶意节点的信誉值之和占比超过总和的 1/3 时，则认为该分片无法正常进行共识。每种恶意节点数情况下都进行了 100 次模拟，结果如图 9、图 10 所示。

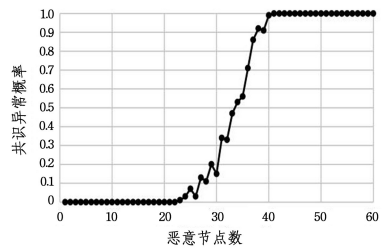

图 9 随机分区的异常概率

Fig. 9 Anomaly probability of random sharding

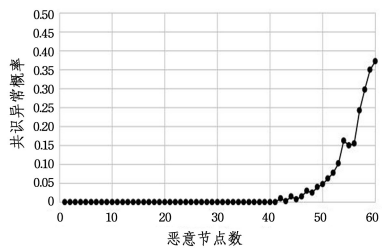

图 10 信誉分区的异常概率

Fig. 10 Abnormal probability of reputation sharding

从图 9 和图 10 的结果可以看出，随机分片中，随着恶意节点数的增加，分片内恶意节点占比大于 1/3 的概率随之增大。从恶意节点数为 20 个即占比为 1/6 往后，共识异常的概率快速上升；占比接近 1/3 时，共识异常概率接近 1；而占比超过 1/3 时，共识异常的概率为 1。而信誉分片情况下，即使全网恶意节点占比高达 1/3，但共识异常概率仍为 0；当恶意节点大于 1/3 而小于 2/3 时，异常概率波动上升，但速度较慢且低于 40%。这是因为本文提出的信誉分片方法通过将

节点划分等级后进行随机均分，使各分区内节点信誉等级分布近似，即分片内部的等级分布与全网的等级分布近似。为了防止出现分区内超过 1/3 恶意节点而无法正常共识的情况，本文提出的 RFBFT 为其上了第二重保险，即区分节点的投票权重，使占比为 1/3 的恶意节点权重总和无法超过 1/3，依旧可以正常进行共识。即便恶意节点发送了错误的秘密碎片，leader 节点仍然可以轻易地利用 TEE 存储的秘密碎片 hash 值验证出来，从而在信誉报告中对恶意节点进行举报，同时也不会影响正常节点的信誉评估。

6.3 区块链安全威胁防范

6.3.1 双花攻击防范

双花攻击(Double Spend Attack)指攻击者发送一笔交易 1，当该交易被写入区块 A 以后，攻击者从 A 块前面的块上制造分叉，令同一笔钱存在于交易 2 中，最终使得分叉链的的高度高于主链，此时主链将被替代，交易 1 将被撤销，从而使一笔钱被花费两次。在一些以 PoW 或 PoS 为共识算法的区块链中，防范双花攻击难度较大，只要攻击者拥有足够多的算力或者权益，就可以制造分叉进行双花攻击。

在 RBSCP 中，如果某分区的 leader 是恶意的，提议了区块 A 和 B，将其发送给分区内不同的持信者，那么 A 和 B 收到的秘密份额的权重和无法同时超过门限 t(t 假设为 2/3 权益和)，则最多只可能有一个区块出块成功；并且如果持信者发现 leader 提议了多个区块，或者持信者没能正确出块，则可以以交易的形式进行举报，依据信誉机制扣除信誉值并且削减押金。

6.3.2 无利害关系攻击防范

无利害关系攻击(Nothing at Stake Attack)是 PoS 类协议的挑战，指当区块链出现分叉时，所持权益很少的节点选择在多条链上进行出块。因为分叉不会消耗节点的资源，即便分叉可能会造成币值降低，但由于节点所持有权益很少，因此对其影响较小。

RBSCP 中设置了押金机制，在防范女巫攻击的同时，将节点行为与押金绑定，如果节点作恶将会造成信誉值下降，使押金削减甚至直接罚没。同时由于节点信誉与其投票能力相关，使得尝试分叉、作恶的节点投票权重越来越低，其影响也越来越小。理性的节点为了避免押金被削减、投票权重减小，会保持正常的共识，避免分叉，维护区块链的安全。

7 实验结果及分析

实验平台为一台 64 GB 内存 Dell 服务器，实验的软硬件环境如表 3 所列。实验对信誉机制进行模拟，并通过对比系统在采用不同分片方式和不同区块链模型的区块链架构时的性能、共识时延、存储量，来验证方案的正确性和可行性。

表 3 实验软硬件环境配置

Table 3 Experimental software and hardware configuration

软硬件	配置
Docker Engine	版本 18.09.0
CPU	Intel Xeon(R)E5-2407@2.2GHz
内存/GB	64
操作系统	Ubuntu server 14.04

7.1 信誉调节实验

信誉机制的模拟主要是为了验证信誉变化是否满足设计需求，共进行了两组实验。实验一验证信誉公式中的 k 对信誉初期增长速度的控制，实验二通过对某节点共识行为进行信誉值计算，以验证信誉公式的有界性，其增减是否表现为缓慢增加和快速缩减，并验证连续作恶对节点信誉的惩罚程度。

实验一将式(3)中的参数假设为 $\alpha=3$，$\beta=9$，信誉等级为优秀的信誉值范围为[0.8,1)，对 k 分别取值为 0.1，0.3，0.7，0.9 进行测试。信誉值增长曲线如图 11 所示。

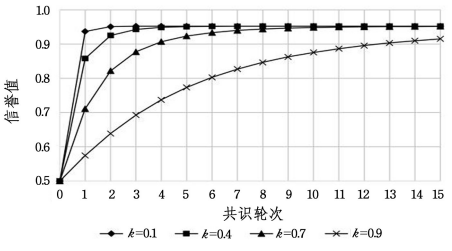

图 11 k 对于信誉值初期增长控制测试

Fig. 11 Control of k on initial growth of reputation value

从图 11 可以看出，$k=0.1$ 和 0.3 时，信誉值在第一轮就突增至 0.8 以上，达到了优秀的信誉等级。而随着 k 的增加，曲线增长趋于平缓，达到优秀的信誉等级需要更多的共识轮数。这是因为 k 值决定了信誉值评估是更依赖于以往的共识行为还是当前轮次的共识行为。k 越大，说明更加看重以往的共识表现，当前轮次的正常行为对信誉值的影响就越小，故曲线增长更平缓。由于以往累计的共识行为比当前一次共识行为更具参考价值，因此建议 k 值大于 0.5。

实验二将式(3)中的参数假设为 $k=0.7$，$\alpha=3$，$\beta=9$，对某一个节点的行为进行信誉曲线的绘制，其结果如图 12 所示。一共进行了 37 轮共识，其中，第 10，15，19，25，27，31，32，33，34，35，36 轮节点存在作恶行为，信誉值快速下降，其 1 次作恶下降的信誉值大约需要后续正常工作 3 轮以上才可以恢复，满足快速下降缓慢回升的要求，限制恶意节点的作恶频率。如果恶意节点连续作恶，如第 31－36 轮，信誉将快速下降并在第 36 轮直接置 0，成为无效节点。理性模型下的节点为避免信誉值快速下降，会维持正确共识。

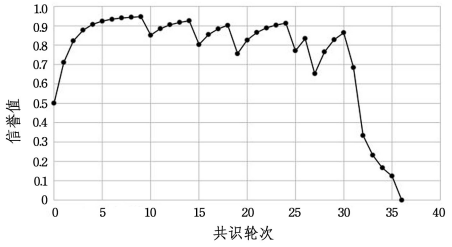

图 12 信誉变化模拟

Fig. 12 Reputation change simulation

根据信誉恢复机制，成为无效节点的持信者有机会继续参加共识，但需要缴纳双倍押金并进入信誉恢复阶段。信誉恢复公式中的 r 设置为 0.015，恢复过程如图 13 所示。在第 36 轮成为无效节点以后，进入信誉恢复过程，在第 43 轮信誉

恢复为 0.1 后才可继续参加共识。接下来的信誉增减将以该节点以往的行为记录为基础进行。信誉恢复阶段作为先前作恶的部分惩罚，限制了该节点在一段时间内无法参与共识，如果该节点为恶意节点，则抑制了其作恶频率。

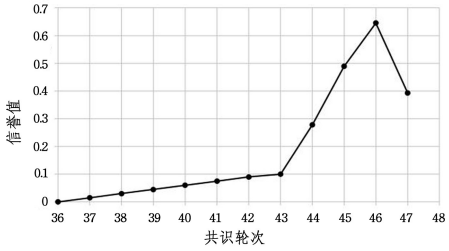

图 13 信誉恢复模拟

Fig. 13 Reputation recovery simulation

7.2 分区时延

分区时延实验主要是将本方案与基于 PoS 和基于 PoW 这两种分区方式进行比较，这里进行比较的是 RBSCP 的信誉分区方式。RBSCP 在进行信誉分区时，首先要将信誉列表 *RpList* 按照信誉等级划分子表，将子表内节点地址与随机数进行哈希运算后按大小顺序排列，再按照分区个数成组均匀划分，所以 RBSCP 的分区时延主要体现在信誉列表的计算上。基于 PoS 的分片方式通过将权益人对自己的公钥以及一个随机数进行哈希运算，根据计算结果的对应规则，进入所属分区，所以其分区时延主要是哈希运算时长。在基于 PoW 的分区方式中，节点需要计算一个满足难度值要求的随机数，并根据该随机数的后几位得到自己所属的分区，所以其时延主要体现在随机数计算的时长上。

假设信誉分区的分区数为 4 个，在分区以前，随机生成 f 个节点的信誉值，其中 f 从 10 增加到 120，间隔为 10。由于信誉分区每次仅进行一次，而基于 PoW 和 PoS 的分区需要每个节点都计算各自的分区，故信誉分区时延为所有节点分区一次的时长，基于 PoW 以及 PoS 的分区时延取所有节点完成最终分区的时长。本实验在各节点数量下均进行 30 次测试，分别取平均值。

分区时延对比实验测试了以上 3 种分区方式的时延，在分区数不变的情况下，进行了 12 组实验，实验结果如图 14 所示。

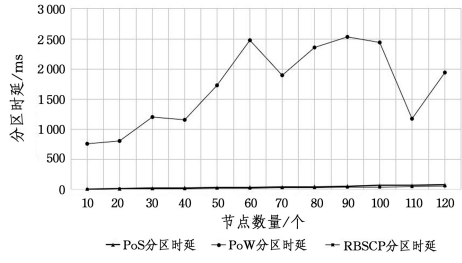

图 14 分区时延对比

Fig. 14 Sharding delay comparison

由实验结果可以看出，RBSCP 的分片时延明显小于 PoW，这是因为 PoW 是概率性运算，在同等的困难度和算力下，对不同的难题需要不同的计算时长，且耗费大量算力导致时延较高。基于 PoS 的分片方式只需进行哈希运算，所以时延很小且相对稳定。而 RBSCP 的时延主要消耗在信誉列表的随机计算上，成组地划分分区的方式使分区时延较短，时长接近 PoS。与 PoS 相比，RBSCP 由于信誉等级的初评估使得恶意节点不会因为随机分区而集结在某个分片，在保证时延较短的同时提升了分区的安全性。

7.3 性能测试分析

性能测试主要包括单分区内共识时延和多分区的吞吐量对比测试。因为 leader 接收到交易的时长以及打包的时长较短，在此忽略不计，所以在单分区的共识时延方面，RFBFT 的共识时延指信誉分区以后 leader 进行秘密份额生成、分发，持信者向 leader 提供秘密份额以及 leader 进行两轮秘密重构所需的时长。由于聚合承诺采用的门限秘密共享算法需要节点的信誉值作为权重，每轮实验随机生成了节点的信誉值，取信誉总和的 2/3 并取整作为门限 t。PoW 实验中，采用的工作量证明函数为 SHA256。RFBFT 与主流 PoW 以及 PBFT 的共识时延对比结果如图 15 所示。

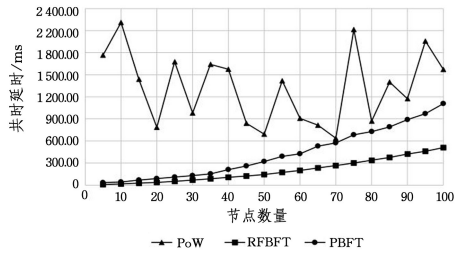

图 15 共识时延

Fig. 15 Consensus delay

由实验结果可以看出，PoW 的共识时延不稳定，波动较大且时延较长，这是 PoW 的概率性计算导致的结果。而 PBFT 共识时延呈现出较快的近线性增长，原因是 PBFT 中每个副本节点都需要和其他节点进行通信，导致通信复杂度为 $O(n^2)$。随着分区内节点数量的增加，节点间通信量快速增加，共识时延迅速增长。相较于前两者，RFBFT 共识时延较短，仅呈现出缓慢的近线性增长，这是因为 RFBFT 通信复杂度仅为 $O(n)$，远小于 PBFT。虽然共识算法采用了基于门限秘密共享的聚合承诺代替节点签名，慢于 PBFT 中的签名过程，但是聚合承诺的安全性更高，并且由于其区分了正常节点和恶意节点的话语权，可使共识更加公平。同时，相较于节点通信的时延，聚合承诺导致的时延增加影响较小。因此随着分区内节点数的增加，相比 PBFT，RFBFT 的共识时延优势会越来越大。

吞吐量(Transactions Per Second，TPS)指单位时间内区块链的交易处理量，计算式如下：

$$TPS=n\cdot\frac{Tx_s}{t} \tag{24}$$

其中，n 为总的分区数，Tx_s 为创块区产生交易块内的交易处理量，t 为多个分区的平均共识时延。吞吐量对比实验中设置每个分区内有 10 个节点，共进行 20 组实验。因为每个区块内的交易数不会影响到实验结果中的趋势，为了便于仿真，本文假设每个区块只包含两笔交易。分区内使用 PoW，

PBFT 和 RFBFT 的吞吐量，对比结果如图 16 所示。

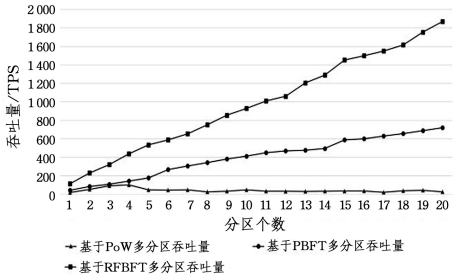

图 16 吞吐量对比实验

Fig. 16 Throughput comparison experiment

在各分区内节点数量相同的情况下，RFBFT 表现出了最好的吞吐量结果。PoW 算力消耗大，共识时延很长，导致吞吐量很小。相对来说，PBFT 具有较好的可扩展性。随着分区数的增加，吞吐量呈近线性增长，但这仅限于单一分区内节点数目较少的情况，因为 PBFT 的通信复杂度为 $O(n^2)$，随着分区内节点数量的增加，PBFT 的这种优势会越来越弱，可扩展性较差。相比 PBFT，采用 RFBFT 的 RBSCP 则表现出更好的吞吐量性能，增长更快，吞吐量为采用 PBFT 的分片方案的 2.2～2.5 倍。这是因为 RFBFT 中的通信量小，通信复杂度为 $O(n)$，并且这种差距在单分区内节点数更多的情况下将更为明显。

7.4 存储量测试

RBSCP 在区块链模型上采用了双链模型，每个分区内的持信者仅存储与本分区相关的交易块以及需要全网统一存储的终验块，从而减少了本地的存储量。存储量测试主要对采用双链模型的分区存储以及不分区情况下的存储量进行对比。假设每个分区提交 5 个区块，每一个区块中区块头和区块体的比例为 1∶3。在分区数增加时，分区存储和不分区存储的存储量对比如图 17 所示。从实验结果可以看出，分区的情况下节点所需存储降低了 40%～70%，也就是说，采用双链模型的分区方案在扩容方面将容量提升了 40%～70%。这是因为每个分区拥有独立的记录链，仅需存储与本分区有关的交易块，网络规模越大，分区越多，存储量越大。而全网统一存储的并非是完整的区块链而是包含验证信息的验证链，从而减小了节点存储压力，扩展了区块链的存储容量。

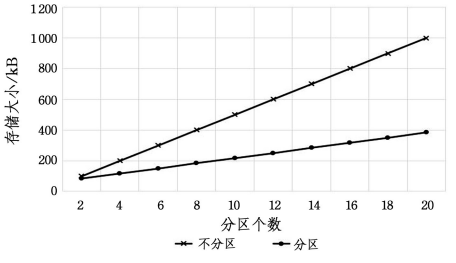

图 17 存储量对比实验

Fig. 17 Storage capacity comparison experiment

7.5 与其他方案比较

本文将 RBSCP 与其他主流的区块链方案进行了比较，结果如表 4 所列。可以看出，分片的共识方案在吞吐量方面相比不分片的 Bitcoin 和 PPCoin 更有优势，随着节点数的增多，分片方案表现出更好的可扩展性。对比经典的分片方案 ELASTICO 和 RapidChain，RBSCP 具有单分片控制管理，相比随机分片更加保障了分片的安全性；并且由于采用了地址分片和信誉分片的双分片模式，区分了存储和共识过程，避免了重分片时产生的数据迁移。相较于采用了连弩挖矿而同样避免了单分片接管的 Monoxide，RBSCP 的共识算法更具优势，不仅避免了算力浪费和能源消耗，还降低了共识时延、提升了吞吐量。对于容错性，由于本文的容错与节点信誉值相关，当节点信誉值相同时，容错存在下限 1/3；为了计算上限，本文假设恶意节点信誉值均值为 0.3，诚实节点均值为 0.75，则上限约为 55.6%。因此相较于其他 BFT 类共识算法，本文方案容错性表现更好。

表 4 RBSCP 与其他方案比较

Table 4 Comparison of RBSCP with other solutions

	核心算法	拜占庭容错	单分片接管控制	数据迁移
Bitcoin	PoW	<1/2	—	—
PPCoin	PoW+PoS	<1/2	—	—
PBFT	BFT	<1/3	—	—
ELASTICO	PoW+PBFT	<1/4	无	有
RapidChain	PoW+BFT	<1/3	无	有
Monoxide	PoW	<1/2	连弩挖矿	—
RBSCP	RFBFT	约<55.6%	信誉分片	无

结束语 本文提出了基于信誉的分片共识协议 RBSCP，解决了现有的分片方案中单分区内恶意节点集结引发的安全问题。RBSCP 以信誉机制为基础评估节点以往的行为，并用信誉值表示其将来表现正常的概率，从而决定其在共识过程中可以担当的角色以及投票的权重。RBSCP 采用了双链模型，每个分区有自己独立的记录链，仅存储与本分区有关的交易，而包含验证信息的验证链则由全网统一存储，这种差异化存储的双链模型减少了节点本地存储量，扩展了全网的存储容量。在分区方式方面，本文设计了地址分区与信誉分区两种分区方法，地址分区后的节点存储本分区的记录链，信誉分区决定节点具有投票权的分区。本文提出的信誉分区是“一初评估，二随机均分”的方法，保证分区后各分区的信誉等级分布相近，防止恶意节点在某一分区集结，保障了分区共识安全。基于信誉权重的快速拜占庭容错共识协议 RFBFT 将节点信誉与其共识话语权相关联，通过带权重的门限秘密共享来完成聚合承诺，减小了恶意节点对共识结果的影响。安全性分析证明了 RFBFT 中聚合承诺的正确性和安全性，能够抵御双花攻击和无利害关系攻击。实验结果表明 RBSCP 在兼具安全性的同时，能够保持较小的分区时延和较高的吞吐量。

分片技术十分复杂，并且区块链的可扩展性、安全性、去中心化理论上相互牵制。本文方案并不完善，还需进一步的研究，研究内容主要有：减少跨分片交易，实现高效率的跨分片交易和交易验证，进一步减小共识过程中的通信开销，结合实际应用进行方案的落地等。

参 考 文 献

[1] PUTZ B, PERNUL G. Detecting Blockchain Security Threats [C] // 2020 IEEE International Conference on Blockchain (Blockchain). Rhodes, Greece, 2020: 313-320.

[2] NAKAMOTO S. Bitcoin: A Peer-to-Peer Electronic Cash Sys-

tem [EB/OL]. https://bitcoin.org/bitcoin.pdf.
[3] KING S,NADAL S M. PPcoin:Peer-to-peer crypto-currency with proof-of-stake[EB/OL]. https://www.semanticscholar.org/paper/PPCoin%3A-Peer-to-Peer-Crypto-Currency-with-King-Nadal/0db38d32069f3341d34c35085dc009a85ba13c13.
[4] CASTRO M,LISKOV B. Practical byzantine fault tolerance [C]// Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI). New Orleans, 1999: 173-186.
[5] LUU L,NARAYANAN V,ZHENG C,et al. A Secure Sharding Protocol for Open Blockchains[C]// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York:ACM,2016:17-30.
[6] THE ZILLIQA TEAM. The ZILLIQA Technical Whitepaper [EB/OL]. https://docs.zilliqa.com/whitepaper.pdf.
[7] KOKORIS-KOGIAS E,JOVANOVIC P,GASSER L,et al. OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding[C]// 2018 IEEE Symposium on Security and Privacy(SP). San Francisco,CA,2018:583-598.
[8] EYAL I,GENCER A E,SIRER E G,et al. Bitcoin-NG:A scalable Blockchain protocol[C]// Proceedings of 13th USENIX Symposium on Networked Systems Design and Implementation. Santa Clara,CA,USA,2016:45-59.
[9] MANNING D T,TAYLOR J E,WILEN J E. General Equilibrium Tragedy of the Commons[J]. Environmental & Resource Economics,2018,69(4):1-27.
[10] LUO Y,CHEN Y,CHEN Q,et al. A new election algorithm for DPos consensus mechanism in blockchain[C]// 2018 the 7th International Conference on Digital Home(ICDH). IEEE, 2018: 116-120.
[11] GUETA G G, ABRAHAM I, GROSSMAN S, et al. SBFT: A Scalable and Decentralized Trust Infrastructure[C]// 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks(DSN). Portland, OR, USA, 2019: 568-580.
[12] JIAN L,WENTING L. Scalable Byzantine Consensus via Hardware-Assisted Secret Sharing[J]. IEEE Transactions on Computers,2019,68(1):139-151.
[13] JALALZAI M M,BUSCH C,RICHARD G G. Proteus:A Scalable BFT Consensus Protocol for Blockchains[C]// 2019 IEEE International Conference on Blockchain(Blockchain). Atlanta, GA,USA,2019:308-313.
[14] GAO S. T-PBFT:An EigenTrust-Based Practical Byzantine Fault Tolerance Consensus Algorithm[J]. China Communications,2019,16(12):111-123.
[15] KAMVAR S D,SCHLOSSER M T,GARCIA-MOLINA H. The EigenTrust Algorithm for Reputation Management in P2P Networks[C]// Proceedings of the 12th International Conference on World Wide Web. 2003:640-651.
[16] CHEN J,YAO S,YUAN Q,et al. Certchain:Public and efficient certificate audit based on blockchain for tls connections[C]// IEEE INFOCOM 2018—IEEE Conference on Computer Communications. IEEE,2018:2060-2068.
[17] KE W E,SUN R,CHEN C M,et al. Proof of X-repute blockchain consensus protocol for IoT systems[J]. Computers & Security,2020,95:101871.
[18] LAO L,DAI X H,XIAO B,et al. G-PBFT:A location-based and scalable consensus protocol for IoT-blockchain applications [C]// IEEE International Parallel and Distributed Processing Symposium. Piscataway,NJ:IEEE Press,2020:664-673.
[19] CAI W J,JIANG W,XIE K,et al. Dynamic reputation-based consensus mechanism:Real-time transactions for energy blockchain[J]. International Journal of Distributed Sensor Networks, 2020,16(3):1-13.
[20] LEI K,ZHANG Q,XU L,et al. Reputation-Based Byzantine Fault-Tolerance for Consortium Blockchain[C]// 2018 IEEE 24th International Conference on Parallel and Distributed Systems(ICPADS). 2018:604-611.
[21] ZAMANI M,MOVAHEDI M,RAYKOVA M. RapidChain: Scalingblockchain via full sharding[C]// Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018:931-948.
[22] WANG J,WANG H. Monoxide:Scale out blockchains with asynchronous consensus zones[C]// Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation. 2019:95-112.
[23] YUN J,GOH Y,CHUNG J M. Trust-Based Shard Distribution Scheme for Fault-Tolerant Shard Blockchain Networks[J]. IEEE Access,2019(7):135164-135175.
[24] HUANG J H,XIA X,LI Z C,et al. Proof of Trust:Mechanism of Trust Degree Based on Dynamic Authorization[J]. Journal of Software,2019,30(9):2593-2607.
[25] SABT M,ACHEMLAL M,BOUABDALLAH A. Trusted Execution Environment:What It is,and What It is Not[C]// 2015 IEEE Trustcom/BigDataSE/ISPA. IEEE,2015:57-64.
[26] BENTOV I,LEE C,MIZRAHI A. Proof of activity:Extending Bitcoin's proof of work via proof of stake [J]. SIGMETRICS Performance Evaluation Review,2014,42(3):34-37.
[27] ZHANG M W,CHEN M W,XIE H T. Weighted Dynamic and Verifiable Multi-Secret Sharing Scheme[J]. Journal of Cryptologic Research,2016,3(3):229-237.
[28] HE B T. The proof and application of the Chinese remainder theorem on $k[x]$[J]. Science Technology and Engineering, 2010,10(24):5965-5966.

WANG Meng-nan, born in 1996, master, is a member of China Computer Federation. Her main research interests include blockchain and so on.

HUANG Jian-hua, born in 1963, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include computer networks, information security and blockchain.

(责任编辑:何杨)