计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210100126

基于信任的双层可拓展共识协议

邵兴辉[1]　黄建华[1]　王梦楠[1]　武海霞[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
(18516681679@163.com)

摘　要　共识机制作为区块链技术的核心,决定了区块链系统的性能、可拓展性和安全性。针对当前区块链的性能、可拓展性问题以及维护系统安全所采用的激励机制成本高的问题,提出一种基于信任的双层可拓展共识协议(Trust-based Dual-layer Scalable Consensus Protocol,TDSCP)。首先,通过结构化网络设计了双层协同的信任模型和共识算法,其中,信任模型根据节点信任值决定其能否获得生成区块的权利,避免了高昂的挖矿代价;其次,通过分区内双层共识算法提高共识效率,拓展了参与共识的节点数量,避免了系统中心化问题;最后,结合可验证随机函数和多级图划分算法对节点进行分区,可有效防止恶意节点聚集,减少跨分区交易的数量。实验结果表明,TDSCP提高了区块链系统的可拓展性,其分区内算法共识时延较低,且分区方法明显减少了跨分区交易的数量。

关键词:区块链;共识算法;分区;信任;跨分区交易
中图法分类号　TP393

Trust-based Dual-layer Scalable Consensus Protocol

SHAO Xing-hui[1],HUANG Jian-hua[1],WANG Meng-nan[1],WU Hai-xia[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　As the core of blockchain technology,consensus mechanism determines the performance,scalability,and security of blockchain systems. Aiming at the performance and scalability issues of current blockchains and the high cost of the incentives used to maintain the security of the systems,a trust-based dual-layer scalable consensus protocol (TDSCP) is proposed. First,the trust model and consensus algorithm of dual-layer cooperation are designed through a structured network. The trust model determines whether a node gets the right to generate blocks based on its trustworthiness to avoid the high cost of mining. Secondly,the dual-layer consensus algorithm within the partitions improves consensus efficiency,expands the number of nodes involved in consensus,and avoids the problem of system centralization. Finally,the verifiable random function and the multilevel graph partitioning algorithm are combined for partitioning nodes,which can effectively prevent malicious nodes from gathering and reduce the number of cross-partition transactions. The experimental results show that TDSCP improves the scalability of the blockchain system,the latency of its intra-partition algorithm consensus is lower,and the partition method significantly reduces the number of cross-partition transactions.

Keywords　Blockchain,Consensus algorithm,Partition,Trust,Cross-partition transactions

1　引言

近年来,区块链的应用范围由金融领域进一步向其他领域拓展,一系列基于区块链技术的供应链、医疗保健、能源管理应用已经成功实施[1]。作为一种去中心化和安全可信的技术,区块链在解决数据共享和物联网等领域的信息安全问题中显示出巨大的潜力[2-3],但其自身的服务质量也面临更严峻的挑战,尤其是性能和可拓展性严重制约了区块链的发展。现有的公有区块链如比特币[4]和PPCoin[5]使用工作量证明(Proof of Work,PoW)和权益证明(Proof of Stake,PoS)实现了账本数据的一致性,参与者需要投入较高的算力或权益成本,该类共识机制依赖于加密货币,且存在交易确认时间长、吞吐量低等问题[6],无法满足高性能实时服务的要求。联盟链主要使用实用拜占庭容错(Practical Byzantine Fault Tolerance,PBFT)[7]及其变种算法,该类算法允许网络中存在一定数量的拜占庭节点,并能够以较快的速度完成共识,其不足

到稿日期:2021-01-18　返修日期:2021-05-15
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)

是通信复杂度较高，且存在 Sybil 攻击问题。

目前，提高区块链系统性能和可拓展性的主要方法是分区(Partitioning)。分区技术将网络划分为若干规模相近的子网络，每个子网络并行处理交易，从而实现交易吞吐量随分区数的线性拓展。Elastico[8] 是第一个实现交易处理能力随节点数量增加而同步拓展的分区方案。此后，OmniLedger[9] 分区方案将吞吐量进一步提高至 3 500 tx/s。分区技术实现了区块链网络的横向拓展，但是也带来了新的问题。首先是系统安全性问题，分区后单一分区节点数量的减少增加了拜占庭节点对系统的威胁，而通过代币奖励诚实节点、保持节点在线的方式并不适用于非加密货币的去中心化应用。其次是共识时延问题，PBFT 协议增加共识节点会引发较高的通信复杂度，使网络规模难以扩展。DPoS[10] 通过选举代理节点来减少参与共识的节点数量，以提高共识效率，但这些节点容易受到攻击，导致系统安全性降低，并且会使系统趋于中心化。最后是分区引发的跨区交易问题，过多的跨区交易会降低系统吞吐量、影响交易确认时间，如何减少跨区交易并保证跨区交易的原子性是提高系统性能的关键。针对以上问题，本文提出了一种基于信任的双层可拓展共识协议(Trust-based Dual-layer Scalable Consensus Protocol，TDSCP)，该协议通过分区提高系统的可拓展性，并从以下 3 个方面解决现有分区机制的问题。首先，引入信任机制改进传统的代币激励手段，并实现了分布式网络中节点信任值的验证及存储；其次，提出一种双层共识算法 DBFT(Dual-layered Byzantine Fault Tolerance)，该算法可以拓展单分区内的节点数量，同时保证较高的共识效率；最后，结合可验证随机函数和多级图划分算法设计分区方法，有效避免了 1%攻击，减少了跨区交易的数量。

2 相关工作

2.1 分区共识

目前区块链的扩容方案可分为两层[11]：第一层扩容称为链上扩容，链上扩容通过改进共识算法、改变区块结构等方式优化区块链的数据层、网络层、共识层和激励层，进而提高系统吞吐量。第二层扩容称为链下扩容，链下扩容通过将复杂的计算放到链下进行，从而减轻链上工作负载，提高系统性能。分区[12] 是链上扩容最有效的方式之一，分区技术改进了共识机制，将网络划分为多个分区后，各个分区协同处理交易，在保持区块链去中心化和安全性的同时，提高了系统可拓展性，消除了性能瓶颈。

Elastico[8] 首先提出了区块链分区协议，在每个共识周期，节点通过 PoW 确定所在分区，并以 PBFT 完成分区内部共识，其问题是 PoW 计算和频繁的分区重组会消耗大量的系统资源，造成较高的交易确认延迟。OmniLedger[9] 的初始分区结合了具有抗偏差特性的 RandHound[13] 算法，协议规定每个分区只需处理本分区对应的 UTXO 交易，并使用两阶段的“锁定/解锁”协议保证跨区交易的原子性，但是锁定操作会阻碍交易同步，限制交易的并发。Chainspace[14] 实现了交易和智能合约的计算分区，并提出了跨片交易的原子性协议，但是共识过程中所有分区需通过交互消息判断交易的合法性，由此造成了较高的通信成本。RapidChain[15] 是一个高效的公有区块链分区协议，该协议采用 IDA(Information Dispersal Algorithms)算法[16] 代替传统的 Gossip[17] 协议，减少了每笔交易的数据交换量，但是该交易建立在 UTXO 模型的基础上，一笔交易可能会被拆分到多个分区中进行处理，增加了跨区交易的通信复杂度。

2.2 信任机制

基于信任的共识机制在保证区块链的安全性和提高系统性能上有自己独特的优势[18]，近年来受到学术界的关注。信任涉及交易或交换关系的基础，两个实体可以通过信任预测对方的行为[19]。信任机制在 P2P 网络中已经有较长时间的发展，典型的分布式信任框架有 EigenTrust[20] 和 PeerTrust[21]。EigenTrust 通过迭代计算节点全局信任值，算法复杂度较高[22]。PeerTrust 通过节点间的信任值反馈、交易数量和反馈的信任值计算节点的信任值，能够有效地防御恶意攻击，但是通过反馈得到的信任值趋于主观，不能作为节点准确的信任值。Trust-PBFT[23] 结合 PeerTrust 信任模型计算节点信任值，并根据信任值选举共识节点，进而扩展分布式网络的规模，但是 PeerTrust 无法让所有节点对同一节点的信任值达成一致。CDBFT[24] 是一种基于节点信任的投票机制，根据信任值奖励或惩罚节点，以此增强可信节点的主动性。目前多数基于信任的区块链网络都缺少信任值的验证和存储过程，因而本文在提出节点信任值计算方法的基础上设计了节点信任值的验证和存储方法。

3 TDSCP 分区协议

3.1 系统概述

TDSCP 是一种分区可拓展共识协议，该协议将网络划分为多个分区，各个分区通过并行处理交易提高系统的吞吐量。节点数的增长会扩展分区的数量，处理的交易数量也会随之增加，解决了系统的可拓展性问题。此外，TDSCP 在分区内独立共识，节点只需要存储所在分区的区块链，在减少每个节点存储开销的同时提高了整个网络的账本存储容量，由此实现了区块链的存储扩容。TDSCP 的网络结构如图 1 所示。

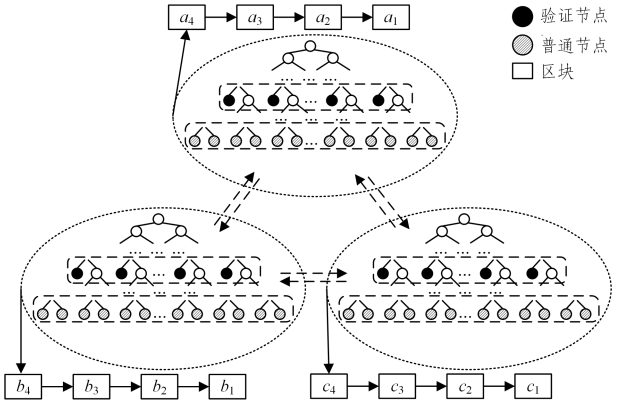


图 1 系统结构

Fig. 1 System architecture

(1)节点身份设置

区块链系统建立在 P2P 模型之上，该模型假设网络中的

节点具有同等的责任。在面向物联网的实际应用场景中，许多节点不具备或者不必有完整节点的功能。TDSCP 为分区中的节点定义了 3 种角色：主节点（leader）、验证节点和普通节点。其中验证节点构成验证层，普通节点构成普通层。leader 节点将收集的网络交易打包成区块发送给验证层，同时将区块头发送给普通层，验证层节点检查区块中交易的有效性，普通层节点检查区块头部的有效性。分角色进行共识可以扩展参与共识的节点数量，保证区块链数据的安全性。

(2)基于信任的共识机制

在工作量证明和权益证明机制中，节点通过增加自身所持资源获得生成区块的权利，由此造成了算力浪费和权益中心化问题，而且这类共识机制往往依赖于加密货币系统。基于信任的共识机制可以降低共识成本，更适用于构建非加密货币的去中心化应用[25]。本文提出了基于节点行为的信任值计算及存储模型，将节点信任值引入验证节点和 leader 节点选举，节点信任值越高，当选验证节点和 leader 节点的概率就越高。信任机制能够减少网络中的拜占庭节点数量，鼓励节点维持良好状态，从而增强单分区的安全性。

(3)分区方法

现有分区方案主要采用随机分区方法以减少因恶意节点聚集造成的单分区接管攻击，但若交互频繁的节点始终处于不同的分区，就会产生大量的跨区交易，影响系统吞吐量及交易确认时间。为了解决该问题，本文采用两种方式进行分区，在网络初期和新节点加入时，系统使用可验证随机函数（VRF）[26]划分分区。当节点的信任值达到一定的阈值后，使用多级图划分算法调整节点所在分区，将交互次数高的节点集合划分到同一个分区，从而减少跨区交易的数量。节点在每次进入新分区后都需要同步当前分区的区块链数据。

3.2 网络结构

TDSCP 在分区内构建了 Kademlia 网络，Kademlia[27]是一种分布式哈希表（DHT）技术，该技术通过结构化的覆盖网络加快了路由速度，减少了节点路由信息存储。Kademlia 的主要设计思想如下：

(1)距离计算

在 Kademlia 网络中，所有节点都是一棵二叉树上的叶子，节点位置由其 ID 的最短前缀唯一确定。Kademlia 使用异或（XOR）计算节点之间的距离，对于 ID 分别为 x 和 y 的两个节点，二者的距离为 $d(x,y)=x\oplus y$，且 $d(x,y)=d(y,x)$。

(2)路由信息配置

在 Kademlia 网络中，每个节点维护 $\log N$（Kademlia 默认 $\log N=160$）个列表，每个列表称为一个 k 桶，k 为系统常量，表示每个桶中节点数目的上限。节点的第 i 个 k 桶记录了与该节点相距 $2^i\sim2^{(i+1)}$ 的一些节点的（IP address，UDP port，Node ID）信息，i 与 k 桶的覆盖距离呈指数关系，因此，每个节点所保存的近距离节点数量多，远距离节点数量少，从而使路由查询是一个收敛的过程。

分区结束后，本文使用 VRF 选举出 leader 节点和验证节点[28]，leader 根据节点计算的 VRF 哈希值为其分配区内 ID，节点以二进制区内 ID 的最短前缀组织成树状拓扑结构。图 2 表示该网络的结构层次，其中 V_1—V_n 为验证节点，P_1—P_m 为普通节点。验证节点前缀更短，所处层次高于普通节点，因此一个验证节点和多个普通节点存在于同一个分支上，通过调整验证节点和普通节点的层次差，可以改变二者的比例。

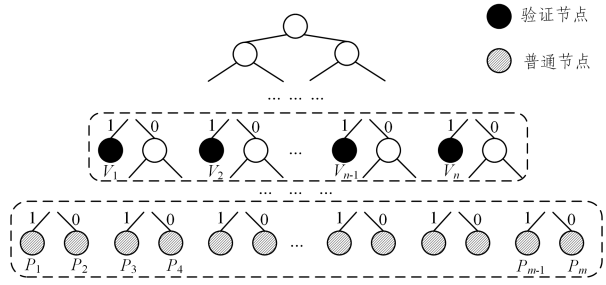


图 2 单分区网络结构

Fig. 2 Single partition network structure

基于上述网络结构，本文使用 Kademlia 路由信息配置方法验证、存储信任值，并利用节点间的距离加快普通层的共识。

(1)传统 Kademlia 网络中 k 桶存储了一系列节点的路由信息，本文更关注节点信任值，所以使用 k 桶存储节点的信任信息，将每个节点验证和存储信任值的个数降到 $\log N$。

(2)普通节点的 k 桶列表里一定会存在数个距离相近的验证节点，在共识过程中普通节点同距离最近的验证节点建立连接，并将签名发送到距离最近的验证节点进行聚合。若验证节点掉线，普通节点可以选择其他距离较近的验证节点。

3.3 信任模型

TDSCP 通过节点行为计算信任值，为了促使节点在线并避免信任中心化问题，其结合动态信任度授权[29]思想对信任值进行修正，并利用 Kademlia 网络验证和存储信任信息来降低分布式网络中维护全局信任值所需的带宽和存储资源。

3.3.1 信任函数

本文使用的信任函数基于 logistic 回归模型，该模型通过改变特征权重控制不同特征对结果的影响，输出值落在 0～1 之间，其优点是计算代价不高、实施高效。信任函数如式(1)所示：

$$T(i)_{cur}=\frac{1}{1+e^{(-\sum_{x=1}^{n}v_x+\alpha\sum_{x=1}^{n}d_x+\beta\sum_{u=1}^{I(u)}c_u w_u)}} \tag{1}$$

该函数以节点的共识行为和交易行为作为特征值。公式中，$T(i)_{cur}$ 是第 i 个节点在当前周期的信任值；n 表示共识的轮次；v_x 表示在第 x 个轮次是否正确参与共识，正确的投票记为 1，恶意投票记为 0；d_x 表示 x 是否进行了恶意投票，α 是对恶意投票的惩罚权重；$I(u)$ 表示当前周期节点的交易总数，c_u 为交易的有效性，w_u 为交易的规模，β 是恶意交易的惩罚权重。由信任函数可以看出，节点只能通过诚实地参与共识来积累信任，而不能通过发起大量交易积累信任，恶意投票和不合法的交易会降低节点的信任值。

节点的历史信任值比当前信任值更能反映节点的长期信任状态，因此节点信任值应由历史信任值和当前信任值两部分计算得出，计算公式如下：

$$T(i)_r^e = f \times T(i)_{cur} + (1-f) \times T(i)_{r-1}^e \quad (2)$$

其中，$T(i)_r$是 i 节点在第 r 个共识轮次得到的信任值，f 是历史信任值在最终信任值中所占权重，$T(i)$函数会使积极节点在数轮共识中信任积累较快。为了避免信任中心化问题，本文使用了文献[29]中的信任修正算法和信任消耗算法来减缓信任值的增长速度，其算法流程如算法 1 所示。

算法 1 信任值修正算法

输入：节点当前信任值 $T(i)_e$，节点历史信任值 $T(i)_{e-1}$

输出：节点信任值 T(i)

1. $\Delta B = B_{current} - B_{previous}$

2. If $\Delta B = 0$

$\delta_h^t T(i) = |T(i)_{h-1}^t - T(i)_{cur}^{(i)}|$

$\xi_h^t T(i) = c \times \delta_h^t T(i) + (1-c) \times \xi_{h-1}^t T(i)$

$f = threshold + (c \times \delta_h^t T(i))/(1 + \xi_h^t T(i))$

$T(i)_h^t = f \times T(i)_{cur} + (1-f) \times T(i)_{h-1}^t$

3. else

$T(i)_{cur} = T(i)_{last} \times e^{-\Delta B}$

算法 1 中，ΔB 表示节点参与共识的间隔，若是连续参与两次共识，则 $\Delta B=0$，此时，信任值增加速率通过 f 进行修正。$\delta_h T(i)$表示信任值偏差，$\xi_h T(i)$为信任值累计偏差。c 值越大，说明历史信任值在当前信任值中的权重衰减越快。为了防止 β 饱和趋近于 1，*threshold* 初始值设为 0.25。当 $\Delta B \neq 0$ 时，节点的信任值会随共识轮次消耗，最后趋近于 0。信用修正算法保证了节点能够积极地参与共识，共同维护网络的安全。

3.3.2 信任值验证

在每轮共识结束后，验证节点使用信任函数计算一定距离内普通节点和验证节点的信任值，并将信任值和节点签名位图提交给 leader，随后 leader 广播所有节点的信任值和签名位图，其中的信任值只有被部分节点验证后才能存储在网络中。验证信任值时，节点被验证并放进 k 桶的概率与其信任值呈正相关。由 Kademlia 中节点的失效概率和在线时长成反比关系可得：节点的信任值和正确共识的次数成正比，信任值越高的节点保留在 k 桶中的可能性越高。在选举验证节点和 leader 节点时，节点的信任值和其出现在 k 桶中的次数同时作为概率选择的权重。该方法在一定程度上避免了信任中心化的问题，节点无需验证、保存全部节点的信任值，且恶意节点无法推断出哪些节点存储了自己的信任值。

3.4 分区内共识算法

在时间上，TDSCP 的运行以周期（epoch）为单位，每个 epoch 包括一个分区阶段和多轮次共识（round）阶段，每个分区在一轮共识后产生一个区块。本文提出了一种分区内双层共识算法 DBFT，DBFT 在验证层和普通层以不同的方法进行共识，验证层由少数信任值高的节点组成，通过运行 PBFT 快速检查完整区块。普通层节点数量较多，该层使用 BLS[30] 签名算法减少共识过程中的通信复杂度。DBFT 以极小的共识时延为代价维持系统的去中心化特性，扩大了分区内参与共识的节点规模，进而增强了单分区的安全性。

DBFT 共识过程如图 3 所示，验证层运行 PBFT 的 pre-prepare，prepare，commit 3 个阶段进行共识，该层节点数较少，PBFT 可以保证较高的共识效率和稳定性。普通层节点数量较多，使用 PBFT 会占用大量的带宽资源，DBFT 通过 BLS 签名算法及 Kademlia 网络改进该层共识。BLS 签名算法可以将多个签名聚合成单个签名从而减少通信复杂度，但是签名的验证和聚合复杂度远高于比特币使用的 ECDSA[31] 签名。SBFT[32] 设置单个收集器聚合签名，当节点数量较多时，签名聚合时延较高。DBFT 利用多个验证节点分散单个收集器的职责，并使用签名的批量验证方法进一步加快共识速度。下面介绍普通层签名算法和共识流程。

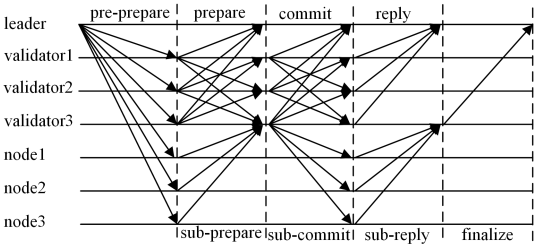


图 3 共识流程

Fig. 3 Consensus process

3.4.1 签名算法

DBFT 使用了 BLS 聚合签名算法，本文对签名算法的输入进行了修改，通过聚合不同消息的签名来避免恶意密钥[30]攻击。BLS 签名算法是基于双线性映射构造的，设$\mathcal{G}$是一个双线性群生成算法，$\mathcal{G}$的输入为一个安全参数 k，输出为一个双线性群$(q, G_1, G_2, G_T, e, g_1, g_2)$，其中，$G_1$，$G_2$，$G_T$为 3 个素数 p 阶群乘法循环群；e 是一个可计算的非退化双线性映射，$e: G_1 \times G_2 \rightarrow G_T$；$g_1$ 和 g_2 分别是 G_1 和 G_2 的生成元。设哈希函数 $H_0: \{0,1\}^* \rightarrow G_2$，$H_1: \{0,1\}^* \rightarrow Zq$，密钥生成算法如算法 2 所示。

算法 2 密钥生成算法

输入：安全参数 k

输出：(pk，sk)

1. $(q, G_1, G_2, G_T, e, g_1, g_2) \leftarrow G(k)$

2. $par \leftarrow (q, G_1, G_2, G_T, e, g_1, g_2)$

3. 密钥生成算法 Kg(par)选择 $sk \leftarrow Zq$

4. 计算 $pk \leftarrow g_2^{sk}$

每个普通节点生成消息摘要时，以消息和公钥作为哈希函数 H_0的输入生成不同的消息摘要 h_i，签名算法如算法 3 所示。

算法 3 签名算法

输入：消息 m，公钥 pk，私钥 sk

输出：(apk_i, h_i)

1. $apk_i \leftarrow \prod pk_i^{H_1(pk_i, \{pk_1, pk_2, \cdots, pk_n\})}$

2. $h_i \leftarrow H_0(m, pk)$

3. $s_i \leftarrow h_i^{sk_i}$

4. $a_i \leftarrow H_1(pk_i, \{pk_1, pk_2, \cdots, pk_n\})$

5. $S_i \leftarrow s_i^{a_i}$

3.4.2 共识流程

(1)pre-prepare 阶段：leader 将完整区块广播给验证层节

点，同时将区块头广播给普通层节点，普通节点验证区块头的有效性。

(2)sub-prepare 阶段：普通节点向距离最近的验证节点发送 sub-prepare〈lockHead, Signature, Valid〉消息，其中，Signature 是普通节点以算法 3 生成的签名 S_i；Valid 的值为 0 或 1，1 表示验证通过，0 表示验证不通过。验证节点等待 Δt 时间后，将收到的签名以式(3)生成聚合签名 $\tilde{\sigma}$，并创建签名者的记录。

$$\tilde{\sigma}=\prod_{j=1}^{n} S_j \tag{3}$$

(3)sub_commit 阶段：验证节点向普通节点发送 sub_commit〈BlockHead, Record, Multisignature〉消息，其中，Record 包含签名者的位图和每个节点的 apk_i 和 h_i，Multisignature 为聚合签名。普通节点通过式(4)对签名进行批量验证。

$$e(\tilde{\sigma}, g_2)=e(h_1, apk_1)\cdots e(h_b, apk_b) \tag{4}$$

(4)sub-reply 阶段：普通节点向验证节点发送 sub-reply〈BlockHead, Valid〉消息，其中 Valid 表示聚合签名是否有效，只有当超过 2/3 数量的节点验证通过后，某个节点的签名才能被证明是有效的。

(5)finalize 阶段：验证节点向 leader 发送 finalize〈Record, Signature〉消息。leader 节点接收到每个验证节点的 finalize 消息后，验证签名数量是否超过节点总数的 2/3，若验证成功，则区块最终被确认。

DBFT 的验证层和普通层共识是并行的，假设共识节点的数量为 n，选举出的验证节点数量为 m，则普通节点的数量为 $(n-m)$。在一轮共识中，验证层节点产生的消息总量为 $2m^2$，普通层在 pre-prepare、sub-prepare、sub_commit 和 sub-reply 阶段分别产生 $n-m$ 个消息，finalize 阶段验证节点发送 m 个消息，产生的消息总量为 $2m^2+4(n-m)+m=4n+2m^2-3m$，当 $n\gg m$ 时，可以认为 DBFT 的通信复杂度为 $O(n)$。DBFT 能够将分区内节点数量拓展至上百个，且共识时延较低，具体分析见 5.2 节和 5.3 节。

4 分区算法

TDSCP 基于可验证随机函数和多级图划分算法对节点分区。在网络初始阶段和新加入节点时，系统无法判断节点的信任度，为了避免恶意节点聚集，通过可信分布式随机数生成源 VRF 划分分区；当某些节点成为可信节点后，通过基于多级图分区的节点分区方法对这些节点进行调整，将其产生的跨区交易降至最少，最小化跨区交易数量不仅可以减少复杂的交易处理开销，而且缩短了交易的确认时间。

4.1 分区引导

初始网络中的节点和新加入的节点通过 VRF 产生一个公开可验证的随机数作为分区依据。在分区之前，节点需缴纳一定数目的押金，目的是增大恶意节点加入网络的代价。当节点能够保持较好的信誉状态时，押金会被退回。具体分区过程如下。

(1)节点使用公钥经过哈希运算得到账户 ID，即：

$$ID=RIPEMD160(SHA256(PK)) \tag{5}$$

(2)节点使用可验证随机函数 VRF 产生一个随机哈希值 y 和对应的证明 π，计算公式为：

$$y=VRF\_hash(ID, SK) \tag{6}$$

$$\pi=VRF\_prove(ID, SK) \tag{7}$$

其中，SK 为节点私钥，y 值是不可伪造的，π 可以让任何节点验证 y 值是否有效。

(3)初始网络中节点需广播 (y, π) 信息，其余节点使用式(8)验证 y 值是否有效，持续一段时间后锁定一个最小的 y 值，产生该值的节点作为 leader。节点向 leader 发送 (PK, ID, y, π) 信息发起加入请求，leader 通过式(9)和式(10)验证节点身份并对节点进行分区，其中 n 为分区个数。

$$y=VRF\_proof\_to\_hash(\pi) \tag{8}$$

$$VRF\_verify(PK, ID, y, \pi) \tag{9}$$

$$SID=y\%n \tag{10}$$

(4)新加入分区的节点先使用式(10)计算自己的目标分区，然后向该分区 leader 发送 (PK, ID, y, π) 信息，leader 使用式(8)和式(9)验证节点身份，决定是否将其纳入分区。

相比于基于 PoW 和基于分布式随机数生成协议的分区算法，TDSCP 的分区引导无需消耗算力、算法效率高，以 VRF 作为随机数生成源，产生随机数时不需要额外的通信，节点收到其他节点的随机数信息后，可以用附带的证明验证该随机数的正确性。在初始分区阶段，假设网络中节点的数量为 n，所有节点需互相发送随机数信息，因此通信复杂度为 $O(n^2)$。系统运行过程中，设新节点的数量为 n_1，此时只需要向目标分区的 leader 发送随机数信息，因此通信复杂度为 $O(n_1)$。TDSCP 的分区算法在分区时的通信复杂度仅为 $O(n^2)$，由于 n_1 远小于 n，因此运行过程中对节点进行分区的通信复杂度很低。

4.2 基于多级图划分的节点分区

图划分是指将一个连通图分割为多个指定规模子图，并使子图之间的边数最小。高质量的图划分可以提高负载平衡、降低通信成本，目前被应用于大规模集成电路设计、数据流划分和任务调度等领域。图划分是一个 NP 完全问题[33]，相应的算法有 Kernighan-Lin[34]、多级图划分算法[35]、基于标签传播[36]的算法等。其中，多级图划分的分区效果较好且算法效率较高，本文将多级图划分方法引入区块链网络的节点分区，以减少跨区交易的数量，实现较高的算法效率。如图 4 所示，多级图划分通过最小化割边数量来减少分区间的连接数量，算法包括 3 个阶段：粗化阶段、初始分区阶段和细化阶段。在粗化阶段，原图通过合并节点得到一个规模较小的图；初始分区阶段将粗化后的图形划分为 k 个分区；细化阶段通过启发式算法减少边缘切割。

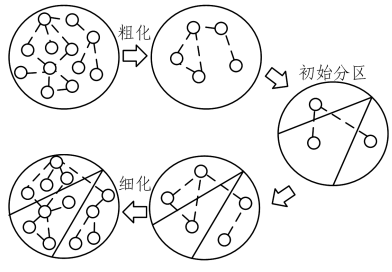


图 4 多级图划分

Fig. 4 Multilevel graph partition

使用多级图划分算法对节点进行分区的一个前提是以基于账户的方式处理交易。比特币的未使用交易输出(Unspent Transaction Output，UTXO)在处理跨分区交易时，一笔交易可能涉及到多个分区，以该方式生成的连通图较复杂，使用图划分算法分区难度较大，而且分区后效益不明显。基于账户的交易中一笔交易最多涉及两个分区，不需要复杂的跨分区通信，有利于生成较高质量的分区。

基于多级图划分的节点分区思想如下：若某个节点的信任值超过设定阈值并且该节点产生了大量的跨分区交易，在epoch末尾，leader使用多级图分区算法对该节点进行调整，以其发起的交易生成连通图。如图5所示，图中顶点表示分区内的节点，边表示交易，边权值表示两个节点间交易的数量。

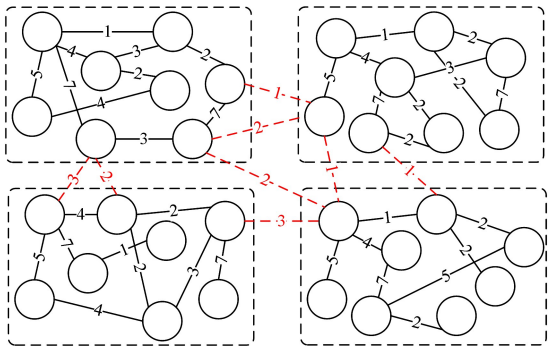


图5 基于多级图分区的节点分区过程

Fig. 5 Node partitioning process based on multi-level graph partition

此时，重分区可以看作对无向加权图分区，通过最小化边缘切割来最大化分区内交易的数量，从而减少跨区交易的数量。算法4描述了节点分区的过程。得到分区结果后，每个分区的leader要交换信息以完成节点的移动，假设分区个数为k，该过程的通信复杂度为$O(k^2)$。

算法4 分区调整算法

输入：Transaction_set

输出：节点所在分区 partitionlist

```
addresslist←Transaction_set
for x in addresslist
  if Num(x)>threshold
    vertexlist.append(x)
graphlist←Graph(vertexlist)
graphlist←ConnectedGraph(graphlist)
for v,a,w in graphlist
  vertex.append(v)
  adjacency.append(a)
  vwgt.append(w)
partitionlist=PartGraphKway( &nVertices,xadj.data(),adjncy.
  data(),vwgt.data(),&nParts).
```

5 实验评估

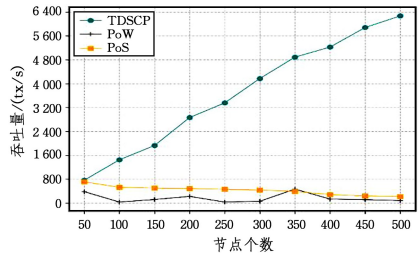
本节从交易吞吐量、共识时延、区块大小对吞吐量的影响和跨区交易数量4个方面对TDSCP协议的性能进行测试，同时将该协议与当前主流的共识算法进行比较，以此验证其有效性和可用性。

实验的硬件环境为一台DELL R320服务器，CPU型号为Intel Xeon(R) CPU E5-2630 @2.40 GHz，内存容量为128 GB；软件环境为64位Ubuntu18.04与Docker Community Edition Version 18.03.1-ce。为了模拟网络中的多节点环境，实验采用了Docker虚拟化技术，每个节点都运行在一个独立的Docker容器中，节点之间通过TCP通信。

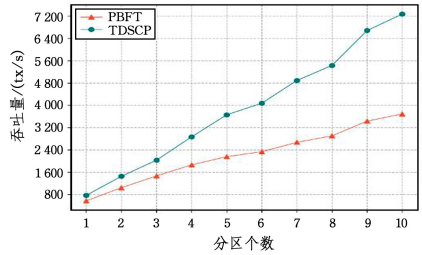
5.1 吞吐量测试

本实验测试TDSCP与当前主流共识算法的吞吐量。实验将区块大小设置为64 kB，并设该区块中包含200笔交易。TDSCP中验证节点和普通节点的数量比为1∶8，由于PBFT适用于节点数量较少的网络，本节实验及5.3节实验中只测试了50个节点时PBFT的吞吐量。

实验结果如图6所示。图6(a)中纵坐标为系统吞吐量，横坐标为节点个数。当节点数由50增长至500时，PoW的吞吐量呈波动趋势，PoS的吞吐量逐渐下降。其原因是运行PoW时，节点寻找*nounce*值的时间不稳定，而运行PoS时，系统开销会随节点数量增加而增大。TDSCP每增加50个节点就会产生一个分区，各分区可以并行处理交易，实现了吞吐量随节点数的线性增长。



(a)



(b)

图6 吞吐量对比

Fig. 6 Throughput comparison

目前多数分区方案是以PBFT作为分区内部共识算法，图6(b)比较了TDSCP与基于PBFT分区系统的吞吐量。实验将分区内节点数设为50，分区数由1递增至10。在分区数和节点数相同的情况下，TDSCP的吞吐量增长速率更快，由此证明了TDSCP具有较好的可拓展性。

5.2 共识时延测试

为了评估DBFT的性能，本实验测试了DBFT在不同网络规模中的共识时延，并与PBFT和SBFT进行对比。如图7(a)所示，当节点数量处于10～100时，PBFT的共识时延增长迅速，DBFT与SBFT都使用了聚合签名来减少通信复杂度，因此共识时延增长较缓慢。当验证节点与普通节点比例

为 1:8 时，DBFT 的共识时延如图 7(a)中实线所示，此时共识时延最低，但是该情况下验证节点数量较少，若验证节点离线会对系统造成较大的影响。为了保证共识的稳定性，将验证节点与普通节点的比例调整为 1:2，结果如虚线所示，此时 DBFT 与 SBFT 的时延较接近。将节点规模扩大至上百以后，如图 7(b)所示，此时 DBFT 的优势较明显，从而证明了双层并行共识和使用的签名方法大大降低了共识时延。

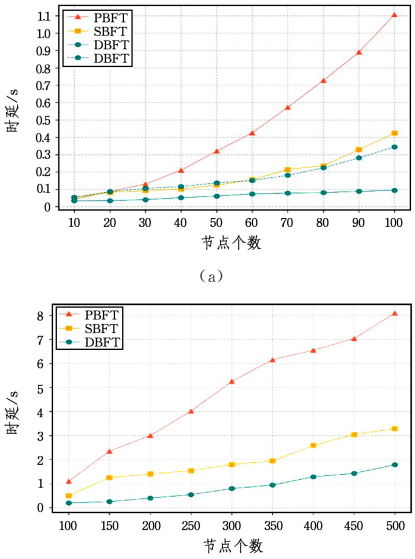


图 7 共识时延

Fig. 7 Consensus delay

5.3 区块大小与系统性能

5.1 节实验中，当区块大小为 64 kB 时，TDSCP 的吞吐量不足 800 tx/s，随着节点数目的增多，吞吐量还会有所下降，而通过增加区块的大小可以提高网络中的数据并发量，进而提高系统吞吐量，因此本节实验测试区块容量对系统性能的影响。

如图 8(a)所示，在单个分区内，当节点数量为 50 时，PBFT 的共识时延随区块的增大而迅速增长，TDSCP 则增长缓慢。图 8(b)为 512 kB，1 MB，2 MB 3 种区块大小的时延测试结果，可以看出，3 种区块的时延都随节点数的增长而线性增加，说明双层共识具有线性的时间复杂度。下面以 200 个节点为例来分析区块大小对共识时延的影响，2 MB 的区块所包含的交易数是 1MB 区块的 2 倍，而共识时延是 1 MB 区块的 1.6 倍；同理，2 MB 区块所包含的交易数是 512 kB 区块的 4 倍，而共识时延是 512 kB 区块的 3.1 倍，这说明在同样的时延下，更大的区块可以处理更多的交易，原因之一是区块容量增大后，区块传播进一步利用了带宽资源；另一个原因是本文的共识算法和网络拓扑减少了区块增长对共识通信的负面影响。

如图 8(c)所示，将分区数增长至 4 个，分区内节点数量设置为 200 个后，随着区块容量的增大，TDSCP 的系统吞吐量逐渐上升。当区块大小为 2 MB 时，分区的吞吐量超过 4000 TPS，此时系统的吞吐量达到峰值，原因是服务器容纳的 800 个节点和网络并发量已经达到设备性能的上限，在实际应用中该吞吐量峰值应该还有较大的提升空间。

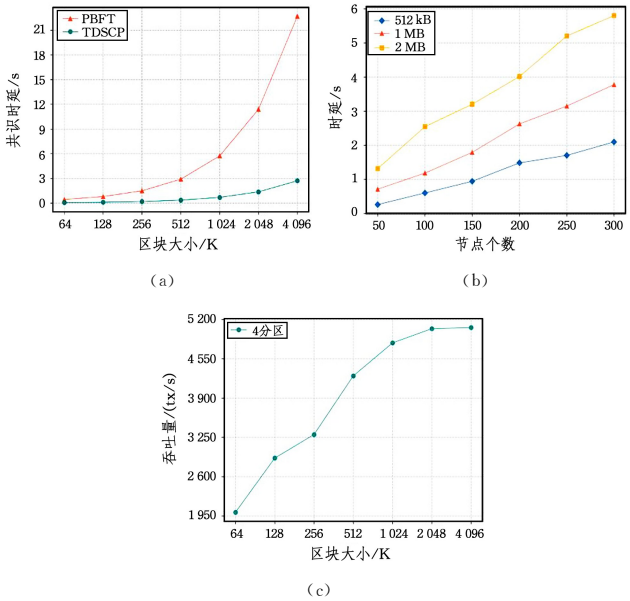


图 8 区块大小对系统的影响

Fig. 8 Impact of block size on system

上述实验证明双层共识可以适应节点数及区块容量的增长，具有较好的可拓展性，此外，通过提高区块的容量，可以在一定程度上提高系统的交易吞吐量。

5.4 跨分区交易

本节实验测试图分区算法对跨区交易数量的减少效果。为了验证多级图划分算法的实用性，实验使用了连续时间内 30000 条以太坊交易数据，首先将这些交易数据按照时间先后分为 a，b 两个集合，在 a 中取出发生了多笔交易的账户集合，该账户集合在集合 a 中发起的交易为交易序列一，在集合 b 中发起的交易为交易序列二；然后以该账户集合和序列一作为图分区算法的输入得到 4 个分区；分区结束后，统计序列二中的区内交易数量。实验结果如图 9(a)所示，图分区算法主要针对频繁交易的账户，因此横坐标为某个账户在集合 a 中发起的交易数量。图中横坐标等于 2 时，直方图的数据含义为：

(1)将 30000 条交易数据划分为 a，b 两个集合，其中，集合 a 发生两笔交易以上的账户有 4579 个，设该账户集合为 c；

(2)账户集合 c 在集合 a 中有 6315 笔交易；

(3)账户集合 c 在集合 b 中有 6270 笔交易；

(4)使用账户集合 c 和交易序列一作为多级图分区算法输入，账户集合 c 最终被划分为 4 个分区，此时，序列二中发生了 2793 笔区内交易；

(5)采用随机分区方法对账户集合 c 进行分区，此时，序列二中发生了 1603 笔区内交易。

图 9(b)分析了图 9(a)中两种分区方式的跨区交易数量占比，随着账户交易数增多，随机分区的跨区交易数量只能维持在约 3/4 的水平，而基于多级图划分的节点分区方法减少了跨分区交易数量。输入数据的增多，更有利于分区算法划分出交易频繁的节点集合。

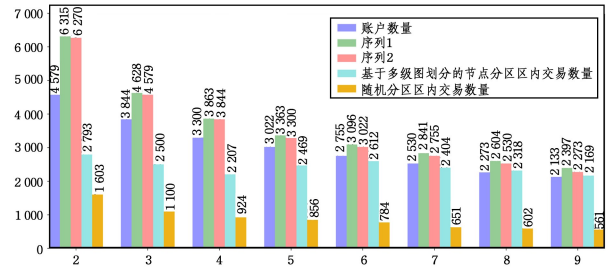


图 9 跨区交易

Fig. 9 Cross-partition transactions

结束语 构建基于区块链的去中心化应用是区块链发展的主要途径，然而当前的共识机制应用场景单一，其性能和可拓展性问题严重限制了区块链应用范围的拓展。本文分析了传统共识机制及分区共识方案，针对分区内多节点共识产生的共识时延问题及分区安全性问题，提出了一种基于信任的可拓展共识协议 TDSCP，TDSCP 在结构化网络的基础上实现了信任模型和共识算法，其中，信任模型的引入改善了当前分区方案缺乏安全性激励措施和激励成本较高的问题，使区块链不再依赖于加密货币。分区内共识算法可以拓展单分区的网络规模，降低共识时延和多节点环境中共识通信占用的带宽资源。此外，TDSCP 使用的分区方法有效保证了初始分区的随机性，基于多级图划分的节点分区方法减少了系统运行过程中跨区交易的数量。实验结果表明，TDSCP 在系统吞吐量和共识时延方面有较明显的优势，而且有效减少了跨分区交易数量。

参 考 文 献

[1] LI Z T, KANG J W, YU R, et al. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2018, 14(8): 3690-3700.

[2] LI J M, ZHAO K, QU T, et al. Research and Analysis of Blockchain Internet of Things Based on Knowledge Graph[J]. Computer Science, 2021, 48(6A): 563-567.

[3] WEI S J, LI S S, WANG J H. A Cross-Domain Authentication Protocol by Identity-Based Cryptography on Consortium Blockchain[J]. Chinese Journal of Computers, 2021, 44(5): 908-920.

[4] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system (2008) [EB/OL]. http://bitcoin.org/bitcoin.pdf.

[5] KING S, NADAL S M. PPcoin: Peer-to-peer crypto-currency with proof-of-stake (2012-08-19) [EB/OL]. https://www.semanticscholar.org/paper/PPCoin%3A-Peer-to-Peer-Crypto-Currency-with-KingNadal/0db38d32069f3341d34c35085dc009a85ba13c13.

[6] MARKO V. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication[C]// International Workshop on Open Problems in Network Security. Cham: Springer International Publishing, 2016: 112-125.

[7] CASTRO M, LISKOV B. Practical byzantine fault tolerance [C]// Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI). USA: USENIX Association, 1999: 173-186.

[8] LUU L, NARAYANAN V, ZHENG C D, et al. A secure sharding protocol for open blockchains[C]// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: Association for Computing Machinery, 2016: 17-30.

[9] KOKORIS-KOGIAS E, JOVANOVIC P, GASSER L, et al. Omniledger: A secure, scale-out, decentralized ledger via sharding [C]// Proceedings of 2018 IEEE Symposium on Security and Privacy (SP 2018). San Francisco, CA, USA: IEEE, 2018: 583-598.

[10] YANG F, ZHOU W, WU Q Q, et al. Delegated Proof of Stake with Downgrade: A Secure and Efficient Blockchain Consensus Algorithm with Downgrade Mechanism[J]. IEEE Access, 2019, 7: 118541-118555.

[11] YU G S, WANG X, YU K, et al. Survey: Sharding in Blockchains[J]. IEEE Access, 2020, 8: 14155-14181.

[12] GENCER A E, RENESSE R V, EMIN G S, et al. Short Paper: Service-Oriented Sharding for Blockchains[C]// Financial Cryptography and Data Security. Springer International Publishing. New York, NY, USA: Association for Computing Machinery, 2017: 393-401.

[13] SYTA E, JOVANOVIC P, KOGIAS E K, et al. Scalable Bias-Resistant Distributed Randomness [C]// Proceedings of 2017 IEEE Symposium on Security and Privacy (SP 2017). IEEE, 2017: 444-460.

[14] AL-BASSAM M, SONNINO A, BANO S, et al. Chainspace: A Sharded Smart Contracts Platform [C]// Proceedings of 25th Annual Network and Distributed System Security Symposium (NDSS 2018). San Diego, CA, USA, 2018: 18-21.

[15] ZAMANI M, MOVAHEDI M, RAYKOVA M, et al. RapidChain: Scaling Blockchain via Full Sharding[C]// Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS 2018). Toronto, ON, Canada: ACM, 2018: 931-948.

[16] RABIN M O. Efficient dispersal of information for security, load balancing, and fault tolerance[J]. ACM, 1989, 36(2): 335-348.

[17] ROBBERT V R, DAN D, VALIENT G, et al. Efficient reconciliation and flow control for anti-entropy protocols[C]// Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware. New York, NY, USA: Association for Computing Machinery, 2008: 1-7.

[18] CHAWLA C. Trust in blockchains: Algorithmic and organiza-

tional[J]. Journal of Business Venturing Insights, 2020, 14: 2352-6734.

[19] LEILA B, SARUNAS G. Trust Mends Blockchains: Living up to Expectations[C]// 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Dallas, TX, USA: IEEE, 2019: 1358-1368.

[20] KAMVAR S D, SCHLOSSER M T, HECTOR G, et al. The Eigentrust algorithm for reputation management in P2P networks[C]// Proceedings of the 12th International Conference on World Wide Web (WWW '03). New York, NY, USA: Association for Computing Machinery, 2003: 640-651.

[21] LI X, LIU L. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(7): 843-857.

[22] ZHANG W, LUO Y, FU S, et al. Privacy-Preserving Reputation Management for Blockchain-Based Mobile Crowdsensing* [C]// 2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2020: 1-9.

[23] TONG W, DONG X W, ZHENG J W, et al. Trust-PBFT: A PeerTrust-Based Practical Byzantine Consensus Algorithm [C]// 2019 International Conference on Networking and Network Applications (NaNA). Daegu, Korea (South): IEEE, 2019: 344-349.

[24] WANG Y H, CAI S B, LIN C L, et al. Study of Blockchains's Consensus Mechanism Based on Credit[J]. IEEE Access, 2019, 7: 10224-10231.

[25] BELLINI E, IRAQI Y, DAMIANI E. Blockchain-based Distributed Trust and Reputation Management Systems: A Survey [J]. IEEE Access, 2020, 8: 21127-21151.

[26] MICALI S, RABIN M, VADHAN S. Verifiable random functions[C]// 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039). New York, NY, USA: IEEE Computer Society, 1999: 120-130.

[27] MAYMOUNKOV P, ERES D M. Kademlia: a peer-to-peer information system based on the XOR metric[M]// Revised Papers from the First International Workshop on Peer-to-peer Systems. Berlin: Springer, 2002: 53-65.

[28] YOSSI G, ROTEM H, SILVIO M, et al. Algorand: Scaling Byzantine Agreements for Cryptocurrencies[C]// Proceedings of the 26th Symposium on Operating Systems Principles. New York, NY, USA: Association for Computing Machinery, 2017: 51-68.

[29] 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.

[30] DAN B, MANU D, GREGORY N, et al. BLS Multi-Signatures With Public-Key Aggregation[EB/OL]. http://theory.stanford.edu/~dabo/abstracts/BLSmultisig.html.

[31] JOHNSON D, MENEZES A, VANSTONE S. The Elliptic Curve Digital Signature Algorithm (ECDSA)[J]. International Journal of Information Security, 2001, 1(1): 36-63.

[32] 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). IEEE, 2019: 568-580.

[33] GAREY M R, JOHNSON D S, STOCKMEYER L. Some Simplified NP-Complete Problems[J]. Theoretical Computer Science, 1976, 1(3): 237-267.

[34] KERNIGHAN B W, LIN S. An Efficient Heuristic Procedure for Partitioning Graphs[J]. Bell Labs Technical Journal, 1970, 49(2): 291-307.

[35] KARYPIS G, KUMAR V. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs[J]. SIAM Journal on Scientific Computing, 1998, 20(1): 359-392.

[36] RAGHAVAN U N, RÉKA A, KUMARA S. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks[J]. Physical Review E, 2007, 76(3 Pt 2): 036106.

SHAO Xing-hui, born in 1996, postgraduate. His main research interests include blockchain and so on.

HUANG Jian-hua, born in 1977, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include computer networks, information security and blockchains.