软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(9):4223−4240 [doi: 10.13328/j.cnki.jos.007274] [CSTR: 32375.14.jos.007274]

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

面向跨信任域互联网场景的拜占庭容错访问控制架构*

韩 将[1,2], 张振峰[2], 刘雨果[1,2], 胡可欣[2], 何双羽[2]

[1](中国科学院大学, 北京 100049)

[2](中国科学院 软件研究所 可信计算与信息保障实验室, 北京 100190)

通信作者: 韩将, E-mail: hanjiang@iscas.ac.cn

摘 要: 工业界现用的访问权限控制技术愈来愈难以应对广域互联网场景下部署的分布式系统的访问控制问题, 特别是跨多个信任域部署的大型信息系统在地理分布上不断分散化, 造成防护弱点不断增加. 基于共识的访问控制策略共享技术能够使跨信任域部署的访问控制节点安全快速地达成一致决策. 首先提出面向多节点的基于共识的访问权限控制模型, 提出强安全高性能的访问控制引擎共识算法 Super-Dumbo. 该算法突破 Dumbo2 共识协议的性能瓶颈, 优化消息广播、随机掷币、共识算法设计等关键步骤的设计, 减少数字签名验证等计算开销、有效提升带宽利用率, 从而在吞吐量和延迟时间等性能方面取得大幅提升, 满足 CBAC 访问控制模型对底层共识算法低延迟、大吞吐量的性能要求.

关键词: 分布式访问控制; 拜占庭容错; 异步共识协议; 基于共识的访问控制

中图法分类号: TP309

中文引用格式: 韩将, 张振峰, 刘雨果, 胡可欣, 何双羽. 面向跨信任域互联网场景的拜占庭容错访问控制架构. 软件学报, 2025, 36(9): 4223–4240. http://www.jos.org.cn/1000-9825/7274.htm

英文引用格式: Han J, Zhang ZF, Liu YG, Hu KX, He SY. Access Control Structure Based on Byzantine Fault Tolerance in Cross-trust-domain Internet Scenarios. Ruan Jian Xue Bao/Journal of Software, 2025, 36(9): 4223–4240 (in Chinese). http://www.jos.org.cn/1000-9825/7274.htm

Access Control Structure Based on Byzantine Fault Tolerance in Cross-trust-domain Internet Scenarios

HAN Jiang[1,2], ZHANG Zhen-Feng[2], LIU Yu-Guo[1,2], HU Ke-Xin[2], HE Shuang-Yu[2]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)

[2](Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In the industrial field, currently used access permission control technologies are increasingly struggling to address access control issues of distributed systems deployed in wide-area internet scenarios. This situation is particularly exacerbated when dealing with large-scale information systems distributed across multiple trust domains, thereby engendering an escalating proliferation of vulnerabilities. Consensus-based access control policy sharing technologies can facilitate the secure and expeditious attainment of consensus decisions among access control nodes deployed across trust domains. This study first proposes a consensus-based access permission control model for multiple nodes and presents the Super-Dumbo consensus algorithm for access control engines, which features robust security and high performance. Super-Dumbo surmounts the performance bottlenecks of Dumbo2 by optimizing the design of key steps encompassing message broadcasting, random coin toss procedures, and consensus algorithm constructs. Notably, it reduces computational overhead such as digital signature verification, thereby effectively enhancing bandwidth utilization. This achieves a substantial improvement in performance metrics, such as throughput and latency, aligning seamlessly with the performance prerequisites of the CBAC access control model, which demands low latency and high throughput from the underlying consensus algorithm.

Key words: distributed access control; Byzantine fault tolerance; asynchronous consensus protocol; consensus-based access control

---

* 基金项目: 国家重点研发计划 (2022YFB2701600)
收稿时间: 2023-11-14; 修改时间: 2024-05-13, 2024-07-05; 采用时间: 2024-08-22; jos 在线出版时间: 2024-12-25
CNKI 网络首发时间: 2024-12-26

1 引 言

访问权限控制技术作为保护信息系统安全的重要技术手段, 能够防止恶意攻击者接入受保护的系统资源, 限制正常用户能够使用的系统资源的范围和方式. 典型的访问控制体系框架如图 1 所示, 包含设计空间和运行空间, 设计空间偏左对最小化授权, 运行空间偏右最小化策略授权. 典型的访问控制系统包括基于访问控制列表 (access control list, ACL)、自主访问控制 (discretionary access control, DAC)、基于角色的 (role based access control, RBAC) 以及基于规则的 (rule based access control, RuBAC) 等. 有效的访问权限控制能够降低数据泄露、非法入侵、身份泄密等安全风险, 提升信息系统的安全性. 在实践中, 访问权限控制被广泛应用在数据库管理[1]、网络应用程序接口 (Web API)[2]、大数据分析[3]、云存储[4]、物联网[5,6]等实际生产场景, 在敏感数据和关键信息基础设施的保护中起到了极其关键的作用[7−9].

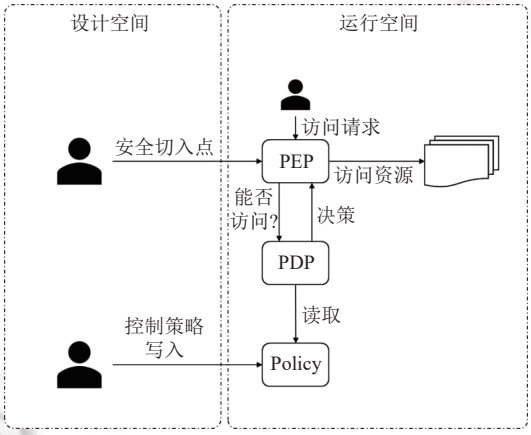


图 1 典型的访问控制系统整体框架

然而, 工业界现有的访问权限控制技术愈来愈难以应对广域互联网场景下部署的分布式系统的访问控制问题, 特别是跨多个信任域部署的大型信息系统在地理分布上不断分散化, 造成防护弱点不断增加[10], 导致传统的访问控制机制频繁失效, 出现了层出不穷的数据泄密事件. 2018 年 3 月, 剑桥数据公司被曝在 2016 年美国大选期间, 利用脸书应用在无用户授权的情况下访问了 5 千万用户的个人数据, 并对用户进行了行为建模分析[11]; 脸书公司 INSTAGRAM 应用在 2019 年 5 月被曝由于数据存储权限管理不善导致泄露近 5 千万用户数据[12]; 2019 年, 知名云存储公司 Box 被曝存在分享链接扫描漏洞, 可能导致了包括苹果公在内的 90 余家企业的内部数据泄露[13]; 2022 年 6 月, NSA 被曝通过木马等各类恶意程序隐蔽嗅探了西北工业大学的核心运维数据, 控制了多个重要业务系统并窃取了用户数据[14]. 国内外频发的安全事件充分说明了开放互联网环境下的分布式信息系统正面临着不断扩大的攻击平面和越发严重的安全风险. 这也对分布式系统的访问权限控制技术提出了愈来愈高的功能性和安全性要求, 特别在跨多信任域场景下, 不同信任域可能采用不同的访问控制策略甚至不同的访问控制模型, 如何在不同信任域之间快速安全地同步用户授权决策, 对授予用户的权限达成一致共识, 就成为必须解决的关键问题.

针对分布式访问控制问题, 已有工作虽然尝试设计了各类不尽相同的分布式访问控制机制[15−26], 初步达成了分布式部署的策略决策点 (policy decision point, PDP) 之间的一致决策, 但普遍采用较弱的安全模型, 即假设所有的分布式 PDP 节点均能诚实地执行预设策略, 无法有效应对部分 PDP 节点被攻击者入侵的情况. 具体来说, 在互联网部署的多信任域环境中, 某些信任域可能由于系统错误配置或遭遇社会工程学攻击而更加脆弱, 部署其中的 PDP 节点也因此更容易被敌手入侵甚至直接控制. 此时, 现有的分布式访问控制机制便面临着严重的安全隐患: 攻击者可以利用成功入侵的少数几个 PDP 节点, 向分布式访问控制机制发动主动攻击 (也被称为拜占庭攻击), 导致用户承受更长的授权时间、降低用户体验, 或者破坏不同信任域之间对用户权限的共识, 甚至操控并恶意提升特定用户的访问权限. 比如, 2020 年 11 月, 著名 CDN (content delivery network) 网络运营商 Cloudflare 公司分布式

Web API 后台的少数节点触发了严重的拜占庭攻击, 造成整个系统用户访问速度的显著降低, 导致 Cloudflare API 服务受损长达 6 h 33 min.

可见, 如何使跨信任域部署的分布式 PDP 节点安全快速地达成一致决策, 从而防止攻击者仅通过入侵部分安全防护较弱的 PDP 节点便影响整个系统的访问权限正常授予, 仍然是分布式访问控制机制设计领域亟待解决的迫切问题. 针对此问题, 本文首先提出了基于共识的访问权限控制 (consensus-based access control, CBAC) 模型, 随后基于异步拜占庭容错共识技术 Dumbo BFT[27]给出了强安全高性能的 CBAC 访问控制引擎算法, 对底层的共识技术进行了多重设计优化, 最后还开展了 CBAC 访问控制引擎算法的软件原型实现和大量测试验证. 具体来讲, 本文具有以下 3 点贡献.

- 第一, 为解决了不同信任域之间难以安全地对用户权限达成一致的问题, 提出了基于共识的访问权限控制模型 (CBAC), 严格定义了 CBAC 模型的安全性. 即使部分 PDP 节点被敌手完全控制, 仍然满足: 即时性 (timeliness), 表明诚实节点在期望常数交互轮之后能够达成决策; 统一性 (unitarity), 表明所有诚实 PDP 节点的决策都是相同的; 正确性 (correctness), 表明用户最终被 CBAC 模型授予的访问权限不会高于大多数诚实节点的本地决策. 本文还给出了 CBAC 模型的实现框架, 将 CBAC 的安全性归约到了共识算法的终止性、有效性和一致性, 并在异步网络下证明了 CBAC 模型的正确性定义已经达到最优, 即不可能在异步网络下设计算法保证分布式授予的用户访问权限小于或等于所有诚实节点的本地决策.
- 第二, 围绕提出的 CBAC 访问控制模型, 提出了强安全高性能的 CBAC 访问控制引擎算法 Super-Dumbo. 围绕小飞象异步拜占庭容错共识算法 (Dumbo BFT)[27]的性能瓶颈, Super-Dumbo 对消息广播、随机掷币、共识算法设计等关键步骤的设计进行了优化. 在安全性方面, Super-Dumbo 实现了经典的异步拜占庭容错公共子集 (asynchronous common subset, ACS) 的安全性, 在容忍攻击者入侵 1/3 的 PDP 节点的同时, 还解决了攻击者可能控制 PDP 间通信的问题, 从而在异步网络下保障了 CBAC 访问权限控制模型的安全性. 在性能方面, Super-Dumbo 相比之前的小飞象算法, 大大减少了数字签名验证等计算开销、有效提升了带宽利用率, 从而在吞吐量和延迟时间等性能方面取得大幅提升, 满足了 CBAC 访问控制模型对底层共识算法低延迟、大吞吐量的性能要求.
- 第三, 开发并开源了基于 Super-Dumbo 共识算法的 CBAC 访问权限控制模型的软件原型实现, 代码量超过 10 000 行. 在服务器环境对上述原型实现进行了部署和测试, 通过控制节点之间的带宽和延迟时间等参数模拟了广域网部署环境, 并从节点规模、网络条件等多维度进行了性能测试, 包括在 4–64 个不同数量节点的情况下与小飞象等现有异步共识算法进行了全面对比. 实验结果表明 Super-Dumbo 相比现有异步共识算法取得了显著的确认延迟降低和吞吐量提升, 验证了基于 Super-Dumbo 算法的 CBAC 访问控制模型的实用性.

第 2 节介绍之前的分布式访问权限控制技术以及难以应对拜占庭攻击行为和异步网络环境的缺点. 第 3 节提出基于共识的访问权限控制模型 (CBAC) 的安全目标, 刻画了异步网络环境和拜占庭攻击等安全风险, 证明 CBAC 安全性可以归约到异步公共子集算法 (即一类特殊的异步拜占庭容错共识算法). 第 4 节针对 CBAC 的性能要求, 提出了 Super-Dumbo 异步公共子集算法, 解释 Super-Dumbo 中有效提升异步共识效率的设计思想. 第 5 节介绍实验设定和测试结果.

2 相关工作

访问控制策略是一个经典的计算机科学的问题, 已经有长期的研究发现, 传统的访问控制策略包括了以下几类方式[28−30]. 自主访问控制 (discretionary access control, DAC)[15], DAC 基于拥有权的概念, 其中用户对其自己的资源和设备拥有完全控制权, 并可以确定其他用户对这些资源和设备的权限. 尽管提出了许多变种, 但 DAC 通常被视为基于身份的访问控制模型, 其中访问权限是根据用户的身份分配的. 提出了各种实施 DAC 的方法, 包括访问矩阵、授权表、访问控制列表 (ACL) 和能力列表. 强制访问控制 (mandatory access control, MAC)[16], 与 DAC 不同, MAC 依赖于一组系统规则, 而不是对象所有者的自行决定. 这些规则通常是根据与主体和对象相关的安全标签定义的. 基于角色的访问控制 (RBAC)[7,17], RBAC 依赖于角色的概念, 以简化组织内访问权限的规范和管理.

角色包括执行某项工作职能所需的权限集. 用户被分配到角色并继承分配给这些角色的权限. 角色通常组织成角色层次结构, 定义了权限在角色之间的继承关系. 基于组织的访问控制 (organization based access control, OrBAC)[18,19], OrBAC 基于 3 个主要概念来指定访问控制策略, 即组织、具体级别和抽象级别, 以及上下文. 组织是一组结构化的活动实体. 类似于其他访问控制模型, 具体的授权是根据主体、操作和对象来指定的, 定义了用户可以 (或不能) 在对象上执行的操作. 基于属性的访问控制 (attribute based access control, ABAC)[20−26], ABAC 是一种通用的访问控制模型, 其中访问权限受制于主体、对象、操作和环境的属性. 策略和访问请求是根据属性名称/值对来定义的. 策略适用于请求是通过将请求中的属性与策略中的属性进行匹配来确定的. ABAC 模型通常提供了合并不同利益相关者制定的策略以及解决这些策略可能引发的冲突的构建机制.

但传统的访问控制研究主要着眼于对用户权限的属性考虑, 之前的研究并不关注于跨域环境下, 多访问控制服务器在协同工作中可能因网络错误或是被恶意攻击而导致权限分配不一致的问题. Jemel 等人[31]提到了集中式访问控制系统中存在的一些问题. 中央机构负责控制访问, 因此存在单点故障的风险. 基于属性的加密方法也存在一些问题, 如来自私钥生成器 (private key generator, PKG)[32]的隐私泄漏以及前面提到的单点故障. 目前在多个管理域中管理访问控制的解决方案效率不高. 根据 Paillisse 等人[33]观察到静态方法不具备可扩展性和粒度, 基于 PKI (public key infrastructure) 的系统难以管理. 他们建议将访问策略分布和记录在一个经许可的区块链中.

为降低部分 PDP 节点失效或被敌手入侵时的危害, 近年来学术界和产业界先后提出了一系列分布式访问权限控制技术. Cruz 等人[34]设计了一个基于角色的访问控制平台, 利用以太坊区块链和 Solidity 智能合同在多个组织中使用. 他们实施了一个智能合同来初始化角色和挑战-响应协议, 以验证角色的所有权和用户身份验证. Hardjono 等人[35]提出的 ChainAnchor 是一种解决共享权限区块链中身份和访问控制问题的方案. ChainAnchor 共识方法在包含所有身份信息的数据库中查找交易发件人的公钥, 并基于此强制进行访问控制. 用户的身份完全匿名, 不能被系统中的任何人披露. 但目前这些方案依然存在性能瓶颈, 尽管最近有关改进区块链性能的研究[36,37], 但基于区块链的解决方案的性能仍然无法与当前的中心化解决方案竞争. 正如我们所看到的, 大多数研究将其所提出的系统的性能与其他基于区块链的平台进行比较, 而不是与当前的解决方案进行比较.

可以看到, 之前的工作的目的主要在于提升多个 PDP 节点综合利用各自信息, 从而更加准确地判断用户访问权限, 但在拜占庭攻击或异步网络环境下可能导致正常用户无法获取权限、或导致恶意用户获得额外权限, 因此具有严重的安全风险. 本工作针对性地解决了拜占庭攻击和异步网络攻击的问题, 与前述工作相比, 具有以下突出特点: 第一, 首次提出了能够容忍拜占庭攻击的分布式访问控制模型, 即基于共识的访问控制模型 (CBAC); 第二, 给出了 CBAC 在异步网络环境的高效实现, 并采用可证明安全性分析方法, 论证了 CBAC 在异步网络环境下的安全性可以归约到异步公共子集算法.

3 基于共识的访问权限控制 (CBAC) 的安全模型

3.1 系统模型

如图 2 所示, 考虑系统由若干的用户、PDP 节点、PEP (policy enforcement point) 节点、待访问资源对象等实体组成. 每个 PDP 节点对应一个 PEP 节点, 并和若干待访问资源处在同一信任域内, 控制用户对信任域内的资源进行访问. 用户访问某个资源的权限 R 由 0–K 之间的某个整数量化, 越大的 R 代表着更多的访问或操作权限. 比如, $R=0$ 代表用户完全无权访问资源, 而 $R=K$ 代表用户拥有该资源读取、修改、执行等所有的使用权限. 注意在 CBAC 访问控制机制中, 用户对某资源的访问权限 R 是由系统中所有 PDP 节点通过网络交互而共同决策的, 并最终由被访问资源所在信任域的 PEP 节点强制执行.

CBAC 访问控制机制中每个参与实体的具体介绍如下.

- 用户: 系统的主体, 通过发起请求来访问受保护的资源. 用户可以是人、机器、程序等.
- PDP 节点: 策略决策点, 负责处理用户访问请求, 根据用户身份和相关策略决定用户对资源的访问权限. 每个 PDP 节点都与一个 PEP 节点相对应, 共同构成一个信任域.

● PEP 节点: 策略执行点, 负责强制执行 PDP 节点所决定的访问策略. PEP 节点与 PDP 节点相对应, 共同构成一个信任域.

● 待访问资源对象: 系统中的受保护资源, 包括数据、文件、服务等. 每个资源对象都有一个唯一的标识符, 用户可以通过该标识符来请求访问资源.

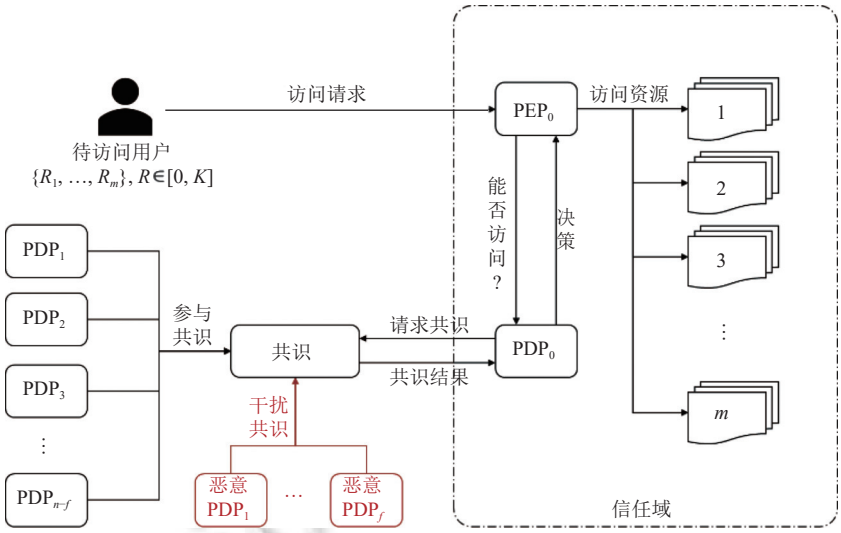


图 2 CBAC 的安全系统模型

在 CBAC 访问控制机制中, 用户的访问权限是由系统中所有 PDP 节点通过网络交互而共同决策的. 具体步骤如下.

● 用户向 PEP 节点发起访问请求, 同时提供身份证明信息.

● PEP 节点将用户的访问请求和身份证明信息转发给对应的 PDP 节点.

● PDP 节点根据用户的身份和相关策略, 以及系统中其他 PDP 节点传递过来的信息, 共同决定用户的访问权限.

● PDP 节点将用户的访问权限结果通过网络传递给 PEP 节点.

● PEP 节点根据 PDP 节点的决策结果, 强制执行访问策略, 允许或拒绝用户的访问请求.

CBAC 访问控制机制具有以下优点.

● 安全性高: 通过多个 PDP 节点的共同决策, 可以有效防止恶意用户的攻击和非法访问.

● 灵活性高: 可以灵活地定义和更新访问策略, 适应不同的业务需求和安全场景.

● 效率高: 通过集中式的策略管理和分布式的访问控制, 可以有效地提高系统的性能和效率.

3.2 威胁模型

在 CBAC 访问控制系统中, 我们考虑腐化 PDP 节点和跨信任域通信网络作为安全威胁. 具体而言, PDP 节点和通信网络的腐化行为可以分别建模如下.

● f-out-of-N 攻击者: 考虑攻击者可以腐化 N 个信任域中的至多 f 个, 并统一控制这 f 个腐化信任域的 PDP 节点开展任意在多项式时间内可计算的攻击行为. 在本文中, 敌手腐化的 PDP 节点称为恶意节点, 未被敌手腐化的 PDP 节点称为诚实节点.

● 异步的跨信任域通信网络: 为刻画 PDP 节点之间不稳定的广域互联网场景, 采用标准的认证异步网络模型, 即 PDP 节点之间的消息经过认证后无法被网络攻击者篡改, 但可能被网络攻击者长时间延迟, 因此无法估计 PDP 节点间消息传输延迟时间的上界.

3.3 安全目标

本文中 CBAC 的安全目标指在 f-out-of-N 攻击者和异步网络环境下实现以下 3 点安全性质.

● 即时性 (timeliness): 当所有诚实 PDP 节点都接收到某条用户访问请求后, 任意的诚实节点都能在期望常数轮交互后对该请求授予访问权限.

● 统一性 (unitarity): 针对每一个用户请求, 所有诚实 PDP 节点最终授予的访问权限是相同的.

● 正确性 (correctness): 对任意一个用户请求, 诚实 PDP 节点们最终一致授予的访问权限一定等于 (或低于) 至少 $N-2f$ 个诚实 PDP 节点本地独立决策的授予权限.

关于上述安全性质, 做出以下说明: (1) 即时性和统一性的定义都是直接和自然的, 前者保证了诚实用户的访问请求能够被快速处理, 后者保证了诚实的 PDP 节点无法被攻击者影响而对恶意用户授予不同的权限, 可见即时性和统一性共同保障了分布式的 PDP 节点能够快速地对用户请求达成一致决策; (2) 正确性的定义存在微妙之处, 因此本文定义的正确性是 CBAC 在异步网络环境下可实现的最强定义.

4 基于共识的访问权限控制系统

4.1 基本系统框架

如图 3 所示, 系统分为用户域、信任域和共识域. 用户域包含所有向系统发出请问请求的计算实体, 简称用户, 其中存在恶意的用户. 信任域包含 PEP、PDP 和可访问资源实体. PEP 和 PDP 对应的策略 (policy) 由共识域取代, 共识域以 Super-Dumbo 为共识协议, 网络中的所有 PDP 实体为参与协议方, 其中包含被恶意用户控制的腐化 PDP 节点. 用户在向系统发送访问请求后, 会由 PEP 向 PDP 询问是否访问, PDP 向所有 PDP 节点发起共识协议, 并基于共识结果做出决策, 返回给 PEP, 最终决定是否接受用户的访问请求.

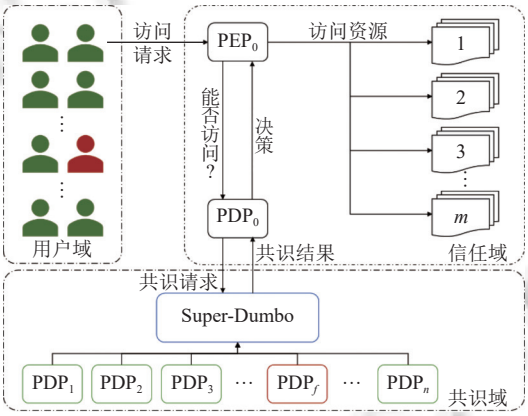


图 3 基于共识 (Super-Dumbo) 的访问权限控制系统框架

4.2 Super-Dumbo: 实现 CBAC 的高性能异步共识协议

拜占庭共识问题由 Lamport 等人[38]于 1982 年首次提出, 旨在讨论如何在不可信的网络环境中, 各诚实计算节点能够高效地达成一致的、正确的共识. 解决该问题的协议被称为共识协议, 而适用于更差的网络环境中, 即异步网络环境中的共识协议被称为异步共识协议. Dumbo BFT[27]是首个应用于区块链的、实用的异步共识协议, 其优秀的安全性和高效性也能够适用于访问控制中.

本文中 CBAC 使用的 Super-Dumbo 基于 Dumbo BFT 中的 Dumbo2 协议, 分别使用多可靠广播实例的 k-RBC、可预测随机源置换优先的 Opti-MVBA 和循环等待的单调外部验证断言, 对其广播阶段吞吐量、共识阶段随机置换和外部验证断言进行优化, 在吞吐量和延迟上均实现较大改进.

本节将先分别介绍 k-RBC、Opti-MVBA 和循环等待的单调外部验证断言 3 处优化细节, 最后对 3 处优化的

安全性进行证明.

4.2.1 广播阶段吞吐量提升: k-RBC

在 PRBC (provable reliable broadcast) 中, 发起广播的节点仅通过一个 PRBC 实例完成交易信息的广播, 这使得对于信道的利用率过低, 从而限制了每一轮可完成共识的交易吞吐量, 并带来了额外的广播延迟. 为了提高信道利用率, 本文提出多可靠广播 (reliable broadcast, RBC) 实例的 k-RBC, 即发起广播的节点广播信息时, 将消息分割成 k 个等长或近乎等长的消息分片, 通过 k 个 RBC 实例进行消息广播. 这里需要注意的一点是, 这里的优化后的广播算法是 k-RBC 而非 k-PRBC, 这是由于共识阶段验证的优化, 导致 PRBC 中的门限签名算法冗余, 从而从 PRBC 简化为 RBC, 使得每一次广播都少一次门限签名算法, 具有 $O(n^2)$ 的时间复杂度的降低 (这里假设使用 ECDSA 数字签名算法构建门限签名算法).

k-RBC 的具体算法如算法 1 所示.

算法 1. 节点 P_l 上的 k-RBC 算法.

1. **if** $P_l == P_{\text{sender}}$ 且收到输入交易 v **then**
2. 将输出交易 v 等分为 k 份, 为 $\{v_i\}_k$
3. **for** $v_i \leftarrow \{v_i\}_k$ **do**
4. 将交易 v_i 分成 n 个分片 $\{s_j\}_n$
5. $\{s'_j\}$ = ErasureCoding($\{s_j\}$) ▷ 利用纠删码对发送数据进行压缩
6. h = MerkleTree($\{s'_j\}$).GetRootHash()
7. **for** $s_j \leftarrow \{s_j\}_n$ **do**
8. b_j = MerkleTree(s'_j).GetTreeBranch() ▷ 利用 Merkle 树验证交易信息是否被篡改
9. 发送消息 $\text{VAL}\left(h, b_j, s'_j, i\right) := \{\text{VAL}, h, b_j, s'_j, i\}$ 给节点 P_j
10. **upon** 收到来自 P_{sender} 的 $\text{VAL}(h, b_j, s'_j, i) := \{\text{VAL}, h, b_j, s'_j, i\}$ 消息 **do**
11. 多播消息 $\text{ECHO}(h, b_j, s'_j, i) := \{\text{ECHO}, h, b_j, s'_j, i\}$
12. **upon** 收到来自 P_j 的 $\text{ECHO}(h, b_j, s'_j, i) := \{\text{ECHO}, h, b_j, s'_j, i\}$ 消息 **do**
13. **if** CheckValidMerkleBranch(b_j, h, s'_j) **then**
14. 保存消息
15. **else**
16. 丢弃消息
17. **upon** 收到来自 $n-f$ 个不同节点的有效 ECHO 消息 **do**
18. 从中采样 $n-2f$ 个 ECHO 消息中的 s''_j
19. 重新计算 h' = MerkleTree($\{s''_j\}$).GetRootHash(), 若出现 $h \neq h'$, 则中止
20. **if** 还未发送消息 $\text{READY}(h, i) := \{\text{READY}, h, i\}$ **then**
21. 多播消息 $\text{READY}(h, i)$
22. **upon** 收到 $f+1$ 个有效的 $\text{READY}(h, i)$ 消息 **do**
23. **if** 还未发送消息 $\text{READY}(h, i) := \{\text{READY}, h, i\}$ **then**
24. 多播消息 $\text{READY}(h, i)$
25. **upon** 收到 $2f+1$ 个有效的 $\text{READY}(h, i)$ 消息 **do**
26. 等待 $n-2f$ 个 ECHO ECHO 消息后, 恢复 v_i
27. **if** V 中不存在 v_i **then**
28. V.add(v_i)
29. **if** $|V| == k$ **then**

30. v = JoinSegment(V) ▷ 按照 i 的顺序将交易分片进行组合
31. 向 P_{sender} 返回 Finish(v) := {Finish, v} 消息 ▷ Finish 消息用于判断广播是否结束
32. **return**
33. 本地存储完成广播的交易集合 $V = V \cup v$

a) 当节点作为广播发送方广播消息时, 先将交易信息切分成 k 份, 并对其中的每一份 $v_i \leftarrow \{v_i\}_k$ 进行单独广播和消息应答. 为了降低通信复杂度, 发送方将交易信息切片 v_i 按照参与协议的节点数进行切分为 $\{s_j\}_n$, 通过纠删码对数据进行压缩, 并将每一个 $s_j \leftarrow \{s_j\}_n$ 作为叶结点构建 Merkle 树 (哈希树). 对于发送方发送给节点 P_j 的 VAL 消息, 其中包含 Merkle 树的根哈希值 h、Merkle 树第 j 个分支的哈希值 b_j、交易分片 s_j 以及分片所在交易切片 v_i 的索引值 i.

b) 当广播发送方完成发送后, 所有节点开始监听消息: 接收方若接收到来自广播发送方的 VAL 消息, 则将 VAL 消息重新封装成 ECHO 消息, 并向协议方节点进行多播; 协议方若接收到来自其他协议方广播的 ECHO 消息, 将检查消息中的根哈希值 h、分支哈希值 b_j 和交易分片 s_j, 以判定消息内容是否被恶意篡改. 若通过检查, 则保存消息, 若未通过, 则抛弃消息.

c) 协议方若累计接收到对同一根哈希值 h 和交易切片索引值 i 的 $n-f$ 个不同且有效的 ECHO 消息时, 则采样 $n-2f$ 个消息, 并重新生成根哈希值 h', 与原根哈希值 h 进行对比. 若两根哈希值一致, 且未广播过对于根哈希值 h 和交易切片索引值 i 的 READY 消息, 则将此消息进行广播; 否则中止广播.

d) 协议方若累计接收到 $f+1$ 个对于同一根哈希值 h 和交易切片索引值 i 的 READY 消息, 且未广播过对于根哈希值 h 和交易切片索引值 i 的 READY 消息, 则将此消息进行广播 (算法 1 第 26–28 行); 协议方若累计接收到 $2f+1$ 个对于同一根哈希值 h 和交易切片索引值 i 的 READY 消息, 则等待 $n-2f$ 个 ECHO 消息, 并从中恢复出交易切片 v_i, 并将切片和索引值进行保存. 若保存的不同交易切片数量达到 k 个, 则按照索引值拼接出原交易信息, 保存在本地, 完成 k-RBC.

4.2.2 共识阶段随机置换优化: Opti-MVBA

MVBA (multi-value validated Byzantine agreement) 中引入随机置换, 这是防止敌手预先知道共识的顺序, 以通过延迟提交, 完成对 ABA (asynchronous Byzantine agreement) 性能的攻击, 即使得 ABA 达成共识的轮数增加[39]. Dumbo2 中使用公共掷币生成随机种子, 再进行置换. 虽然根据公共掷币的不可预测性, 在未执行公共掷币前, 敌手预测生成的随机数的概率 P 和生成随机数长度 n, 满足:

$$|P-\frac{1}{2^n}| \leqslant \varepsilon \tag{1}$$

其中, ε 是一个可忽略值, 这样确保了敌手无法提前得知共识的顺序, 但使用普通的置换, 敌手依旧很难得知共识的顺序. 所以可以将公共掷币换成更为简单的哈希函数, 虽然哈希函数可以使用预处理手段提高预测中随机种子的概率, 但完全预测中随机种子依旧需要较高的时间代价, 并且由于哈希值的生成的时间复杂度 O(n), 小于公共掷币的时间复杂度 O(n^2), 在性能上更具优势. 然而, 我们无从得知敌手的具体能力, 所以通过对 ABA 共识轮数进行计数, 在达到阈值后转换成随机置换完成共识.

优化后的共识阶段算法 Opti-MVBA 为: 各节点通过一致广播对提交的交易发起共识请求, 收到请求的节点通过循环等待对提交的交易进行验证. 完成验证后, 对进行共识的交易顺序进行伪随机置换, 之后进入 ABA 共识; 若 ABA 共识轮数超过阈值 τ, 则重新以公共掷币为种子进行随机置换, 再进入 ABA 共识.

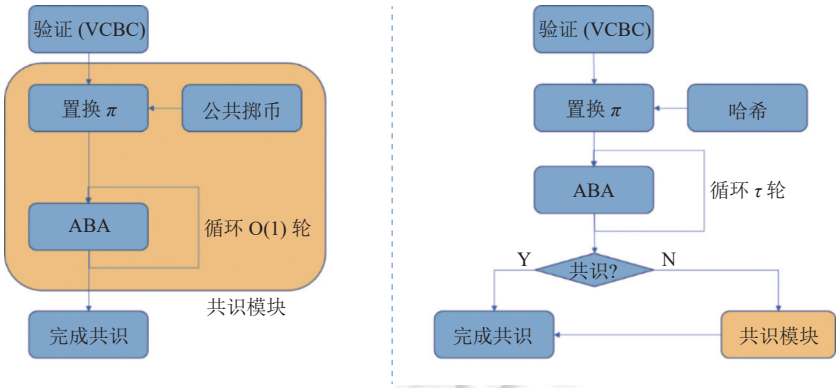
Opti-MVBA 优化前后的流程示意图如图 4 所示, 具体的优化算法如算法 2 所示.

a) 广播发送方向其他协议方发送包含交易 v 的 SEND 消息, 并等待其他协议方返回的 ECHO 消息. 当发送方接收到 $2f+1$ 个不同协议方的 ECHO 消息后, 向其他协议方广播 Finish 消息.

b) 当协议方收到来自发送方的 SEND 消息后, 返回 ECHO 消息. 当协议方收到来自发送方的 Finish 消息后, 若在 k-RBC 阶段存储的本地交易 V 中存在该交易 v, 则输出 v; 否则循环等待, 直到本地交易 V 加入交易 v, 再输

出 v.

c) 在 ABA 共识阶段中, 先以固定值 (如共识的 Session ID) 的哈希值为随机种子, 进行置换后, 进入 ABA 共识. 若 ABA 共识轮数超过阈值 τ, 则使用公共掷币生成随机种子, 再进行置换后, 进入 ABA 共识. 当 ABA 共识输出 1 后, 返回对应交易.



(a) MVBA (b) Opti-MVBA

图 4 MVBA 和 Opti-MVBA 的流程示意图

算法 2. 节点 P_l 上的 Opti-MVBA 算法.

Input: 外部断言 Q; ABA 轮数阈值 τ

CBC 阶段

1. **if** $P_l == P_{\text{sender}}$ 且收到输入交易 v **then**
2. 多播消息 SEND(v) := {SEND, v}
3. **upon** 第 1 次收到来自 P_j 的 ECHO(j) 消息 **do**
4. $DS = DS \cup \{j\}$
5. **upon** $|DS| == 2f + 1$ **do**
6. 多播消息 Finish(v) := {Finish, v}
7. **upon** 第 1 次收到来自 P_{sender} 的 ECHO(j) 消息 **do**
8. 发送消息 ECHO(j)
9. **upon** 第 1 次收到来自 P_{sender} 的 Finish(v) 消息 **do**
10. **if** $Q(v)$ == True **then**
11. 输出 v

ABA 阶段

12. 构造共识序列 $[n] = \{1, 2, \ldots, n\}$
13. 使用可预测随机源置换共识序列得 L = PermutationWithoutCoinTossing($[n]$)
14. $r = 0$
15. **while** $l \leftarrow L[r]$ **do**
16. P_l 将交易 v 加入 ABA_r 中
17. **if** ABA_r 的输出为 1 **then**
18. **return** v
19. **else**
20. $r \leftarrow r + 1$

21. **if** $r \geqslant \tau$ **then**
22. $r = 0$
23. 使用不可随机源置换共识序列得 $L = \text{PermutationWithCoinTossing}([n])$

4.2.3 循环等待断言和 Super-Dumbo

Super-Dumbo 即用以上的 k-RBC 算法和 Opti-MVBA 算法, 替换 Dumbo2 中的广播阶段算法和共识阶段算法. 同时将外部验证断言 Q 进行优化, 优化前需要重述 Dumbo2 中外部验证断言的原理: 此处验证是证明节点 P_i 完成了对其交易的广播, 即至少有 $n-2f$ 个诚实节点收到了来自 P_i 的广播. 为了对这件事进行证明, 在广播阶段, PRBC 使用 $(n-2f,n)$-门限签名算法, 当节点 P_j 在收到 READY 消息后, 使用私钥分片 SK_j 对消息进行签名生成签名分片 ss_j, 并返回给节点 P_i. 这也就意味着当且仅当 P_i 收到了至少 $n-2f$ 个来自其他节点的签名分片, 才能生成对于消息的签名. 当在共识阶段对 P_i 提交的交易进行验证时, 则提交对应的消息签名进行验证, 即外部验证的本质是门限签名算法中的签名验证算法.

签名验证算法依旧是一个 O(n) 复杂度的算法, 而验证的本质是有足够多的节点收到了 P_i 的广播, 所以优化后的外部验证断言 Q 采取直接用本地收到的消息的 Merkle 树的根哈希值进行验证. 对于没有收到消息的节点, 可以进行循环等待, 等到消息后完成验证. 对于没有等到消息的节点, 将处于循环等待状态, 并在本轮共识结束后杀死循环等待进程.

共识阶段的循环等待验证总共减少了 O(n^2) 的外部验证的时间复杂度, 同时由于验证不需要门限签名, 使得广播阶段也省去了门限签名的签名算法和签名分片聚合算法.

其结构示意图如图 5 所示, 具体算法如算法 3 所示.

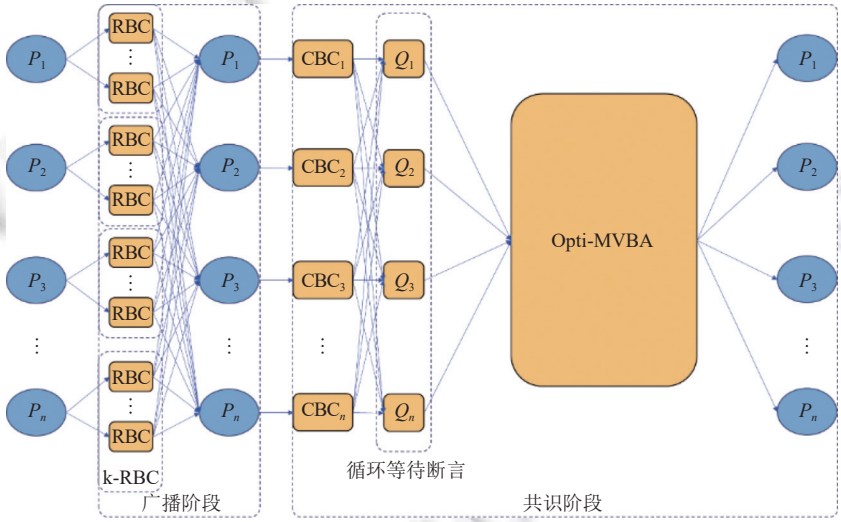


图 5 Super-Dumbo 结构示意图

a) 广播阶段: 节点通过 k-RBC 对提交的交易信息进行广播.

b) 共识阶段: 在完成广播阶段, 即接收到 $n-f$ 个 k-RBC 的 Finish 消息后, 将交易加入 Opti-MVBA 中.

c) 输出阶段: 当 Opti-MVBA 返回交易后, 输出该交易, 完成本轮共识.

算法 3. 节点 P_l 上第 r 轮的 Super-Dumbo 算法.

1. 外部断言 $Q(v) \equiv$ **if** v 在本地交易集合 V 中 **return** True **else** 等待并循环断言 Q
2. **upon** 将交易 v_l 加入共识 **do**

3. 将交易 v_l 输入广播实例 k-RBC$_{(r,j)}$
4. **upon** 首次接收到消息 Finish(v_l) := {Finish, v_l} **do**
5. 进入等待状态, 直到接收到 $n-f$ 个 Finish(v_l) := {Finish, v_l} 消息
6. 将交易 v_l 和外部断言 Q 加入 Opti-MVBA$_r$
7. $\overline{v_l} \leftarrow$ Opti-MVBA$_r$
8. **return** $\overline{v_l}$

4.2.4 安全性分析

在安全性分析中, 需要证明: 1) k-RBC 依旧满足一致性、有效性和总体性; 2) Opti-MVBA 依旧满足一致性、终止性和有效性; 3) Super-Dumbo 依旧满足一致性、有效性和总体性. 并且在完成证明后, 回到第 3.3 节对于安全目标的证明.

引理 1. 当实现 k-RBC 的 RBC 是安全的, 那么 k-RBC 满足一致性、有效性和总体性.

证明:

a) 一致性: 对于某一交易切片 v_i, 当一个诚实节点输出 v_i, 另一个诚实节点输出 v_i', 按照 RBC 的一致性, 可得 $v_i = v_i'$. 当 k-RBC 输出时, 节点会按照交易切片索引值按序进行拼接, 使得一个诚实节点输出 $v = \{v_i\}_k$, 另一个诚实节点输出 $v' = \{v_i'\}_k$, 满足 $v = v'$, 满足一致性.

b) 有效性: 对于某一交易切片 v_i, 当一个诚实节点输入 v_i, 根据 RBC 的一致性, 则存在诚实节点均会输出 v_i. 根据 k-RBC 算法, 将交易切片按序聚合成交易后, 依旧满足当一个诚实节点输入 $v = \{v_i\}_k$, 则存在诚实节点会输出 $v = \{v_i\}_k$, 满足有效性.

c) 总体性: 对于某一交易切片 v_i, 当一个诚实节点输入 v_i, 根据 RBC 的总体性, 所有诚实节点均会输出 v_i. 根据 k-RBC 算法, 将交易切片按序聚合成交易后, 依旧满足当一个诚实节点输入 $v = \{v_i\}_k$, 所有诚实节点均会输出 $v = \{v_i\}_k$, 满足总体性. 证毕.

引理 2. 当 CBC 和 ABA 均安全时, 那么 Opti-MVBA 满足终止性、外部有效性、一致性、完整性.

证明:

a) 终止性: 证明终止性, 即证明当所有的诚实节点输入了经过断言验证的值 v, 则所有诚实节点均将输出值 v.

假设所有的诚实节点输入经过断言验证的值 v, 由于随机置换的策略选取并不会影响 MVBA 的安全性, 仅影响 MVBA 的性能, 所以 MVBA 的终止性不会受到影响, 那么根据 MVBA 的终止性, 所有诚实节点均将输出值 v. 终止性成立.

b) 外部有效性: 证明外部有效性, 即证明如果一个诚实节点输出了值 v, 则有外部断言 Q, 满足 $Q(v)$ = True.

由于断言变为循环等待, 所以此处的断言从 MVBA 的全局、无状态断言变为本地、有状态断言. 此时的断言是单调的, 即随着每个节点内部的状态变化, 断言函数的结果只可能从否变成真, 不能从真变成否. 为了保证本地断言的有效性, 此处要求所有输出值 v, 以及协议节点本地断言集合 $\{Q_i\}$, 满足:

$$\forall Q \in \{Q_i\},\ |\{Q_i\}| \geqslant n-2f,\ Q(v) = \text{True} \tag{2}$$

即至少对 $n-2f$ 个本地断言输出为真. 而 MVBA 的外部断言使用 $(n-2f, n)$-门限签名, 同样需要满足: 对于值 v 的证明, 即有效的签名分片集合 $\{\sigma\}$, 通过断言的条件为:

$$\forall \sigma \in \{\sigma_i\},\ |\{\sigma_i\}| \geqslant n-2f \tag{3}$$

与本地断言的验证条件一致. 所以满足外部有效性.

c) 一致性: 证明一致性, 即证明所有诚实节点的输出一致.

根据 Opti-MVBA 的算法, 由于外部验证和随机置换的优化对 ABA 共识阶段并不会产生影响, 与 MVBA 的 ABA 共识阶段一致, 所以 Opti-MVBA 依然满足一致性.

d) 总体性: 证明总体性, 即证明当部分诚实节点输出了值 v, 则有一些诚实节点输入了值 v.

假设节点 P_i 输出结果 v_i, 根据外部有效性, 输出结果 v_i 满足断言 Q, 即 $Q(v_i)$ = True. 根据终止性, 只有当值 v_i 通过了验证, 所有诚实节点才能输出该值, 则可推出 v_i 是某个 (些) 诚实节点的输入, 证明满足总体性. 证毕.

定理 1. 当 k-RBC 和 Opti-MVBA 均安全时, 那么 Super-Dumbo 满足一致性、有效性和总体性.

证明:

a) 一致性: 证明一致性, 即证明当一个诚实节点输出集合 $V = \{v_i\}$ 时, 所有诚实节点均输出集合 V.

假设节点 P_i 输出了集合 $V = \{v_i\}$, 根据 Opti-MVBA 的外部有效性, 这意味着集合 V 中的所有值均通过了 $n-2f$ 个诚实节点的本地断言. 同时根据 k-RBC 的总体性, 所有诚实节点均在广播阶段输出 v_i, 并且根据 Opti-MVBA 的一致性, 所有的诚实节点的输出均一致, 则所有诚实节点将输出 V. 从而证明一致性.

b) 有效性: 证明有效性, 即证明每个诚实节点的输出集合 V 满足 $|V| \geqslant n-f$ 且至少包含 $n-2f$ 个诚实节点的输入.

假设节点 P_i 输出了集合 $V = \{v_i\}$, 根据 Opti-MVBA 的验证断言可知: 1) 对于每一笔交易 v_i, 均通过了至少 $n-2f$ 个节点的本地验证断言; 2) 集合 V 的大小至少为 $n-f$. 同时根据 k-RBC 的总体性, 节点 P_i 输出的集合 V 的大小至少为 $n-f$.

同时由于在设定中, f 为最大的可能出现拜占庭错误的节点 (腐化节点) 数量, 所以满足输出集合 V 至少来自 $n-2f$ 个诚实节点的输入. 从而证明有效性.

c) 总体性: 证明总体性, 即证明当 $n-f$ 个诚实节点具有输入, 则所有的诚实节点都会产生输出.

根据 k-RBC 的有效性: 诚实节点 P_i 发送的交易 v_i 会被所有诚实节点接收. 现在有 $n-f$ 个诚实节点具有输入, 因此每个诚实节点会至少收到来自 $n-f$ 个不同诚实节点的 Finish 消息, 从而将交易 v_i 作为 Opti-MVBA 阶段的输入. 根据 Opti-MVBA 的一致性和终止性, 所有诚实节点将收到 Opti-MVBA 的相同输出.

同时, 所有的 Opti-MVBA 的输出均满足其外部验证断言. 同时根据 k-RBC 的总体性, 将保证所有诚实节点将接收到 $V = \{v_i\}$. 从而证明总体性. 证毕.

最后, 回归到安全目标, 即系统的即时性、统一性和正确性.

定理 2. 当 Super-Dumbo 满足一致性、有效性和总体性时, CBAC 系统满足即时性、统一性和正确性.

证明:

a) 即时性: 由于 Super-Dumbo 中, k-RBC 的执行轮次为常数, Opti-MVBA 的执行轮数亦为常数, 所以 Super-Dumbo 的执行轮数为常数轮. 故 PDP 节点能够在常数轮次达成共识, 并由于 Super-Dumbo 的有效性, 使得诚实节点都能在期望常数轮交互后对该请求授予访问权限.

b) 统一性: 根据 Super-Dumbo 的一致性, 即所有诚实节点的输出相同, 故根据 Super-Dumbo 的有效性, 那么所有诚实 PDP 节点最终授予的访问权限都是相同的.

c) 正确性: 根据 Super-Dumbo 的有效性, 即输出包含至少 $n-2f$ 个诚实节点的输入, 故诚实 PDP 节点最终一致授予的访问权限一定来自至少 $n-2f$ 个诚实 PDP 节点本地独立决策的授予权限. 证毕.

5 系统实现和实验验证

本文的 Super-Dumbo 算法实现使用 Python, 且算法基于 Dumbo2 进行实现[21]. 在运行算法时, 每个参与协议的节点通过未验证的 TCP 套接字进行通信信道的建立. 所有节点默认为诚实节点, 但在运行前会随机腐化 1/3 的节点.

在公共掷币的实现中, 本文使用 Boldyreva[40]的基于配对的门限签名方案, 使用 MNT224 椭圆曲线构建签名算法. 在门限加密的实现中, 本文使用我们采用 Baek 等人[41]使用 SS512 对称双线性群的门限加密方案实现. 并通过 zfec 库实现 Reed-Solomon 编码.

实验分为 4 个部分: 实验 1 是循环等待断言和 Opti-MVBA 的优化与原 Dumbo2 的性能比较, 实验 2 是 Super-Dumbo 的 k-RBC 在不同的实例数 k 下的吞吐量比较, 实验 3 是比较 Super-Dumbo 与原 Dumbo2 在带宽利用率上

的优势, 实验 4 将分析 Super-Dumbo 中各个部分的时间占用率. 需要特别说明的是, 当 $k=1$ 时, 可以看作广播阶段使用 RBC 或 PRBC, 即 Dumbo2 的广播阶段, 所以实验 2 主要是对 Dumbo2 和 Super-Dumbo 广播阶段进行性能比较. 实验 1 和实验 2, 均进行了小规模 (PC 实验环境, 不多于 10 个仿真验证节点) 和大规模实验 (高性能服务器实验环境, 最多 64 个仿真验证节点), 实验 3 和实验 4 仅进行小规模实验. 小规模实验环境为: 操作系统 Ubuntu 18.04 LTS, CPU 为 AMD Ryzen 7 6800H, RAM 大小 16 GB; 大规模实验环境为: 操作系统 Ubuntu 20.04 LTS, CPU 为 28-core Xeon Platinum 8280, RAM 大小 1 TB. 每个节点实例以线程进行, 通过 Python 库的 gevent. Queue 模拟节点消息缓冲队列, 使用 socket 套接字进行网络通信, 并使用宽带限制和强制延迟模拟异步网络环境, 其中网络环境分为良好的网络环境, 其带宽和延时分别设置为 200 Mb/s 和 50 ms, 以及差的网络环境, 为 50 Mb/s 和 300 ms, 在实验中会按一定规律进行切换. 实验中单笔交易的大小固定为 250 B.

实验 1. 在小规模实验中, 协议方节点数 $n=10$, 腐化节点数 $f=3$, 实验测试轮数为 500 轮. 变量为单批量交易的大小, 从 1 000 笔交易变化到 10 000 笔交易. 实验数据如图 6 所示, 为叙述简洁期间, 基于 Dumbo2 同时进行循环等待断言优化和 Opti-MVBA 优化后的共识协议记为 Dumbo2.5. 数据显示, 在小规模实验中, 即测试机器性能较为有限的情况下, Opti-MVBA 优化明显较于循环等待断言优化更明显, 其中 Opti-MVBA 优化在延迟上较于 Dumbo2 有 13.26%–24.94% 的下降, 验证优化有 1.46%–7.66% 的下降, Dumbo2.5 有 22.59%–27.85% 的下降; 在吞吐量的上, Opti-MVBA 优化有 15.28%–33.23% 的增加, 循环等待断言优化有 1.47%–8.30% 的增加, Dumbo2.5 有 29.18%–38.59% 的增加.

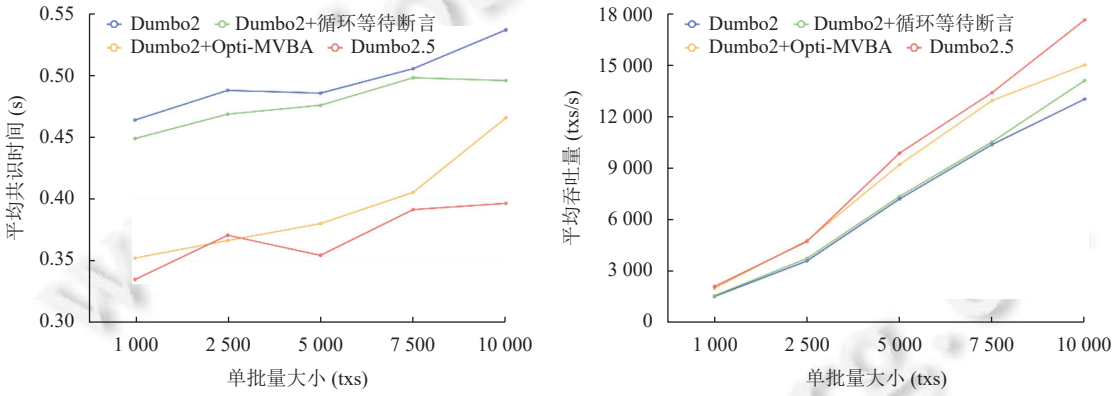


图 6 Dumbo 平均共识时间和平均吞吐量随单批量大小的变化

在大规模实验中, 更注重对协议方规模的实验讨论: 单次交易批量固定为 5 000 笔交易, 实验测试轮数为 500 轮. 变量为参与协议的节点数量, 从 $n=16$ 到 $n=64$, 对应的腐化节点的数量满足 $f=\lfloor(n-1)/3\rfloor$. 实验数据如图 7 所示. 数据显示, 在节点数较多的情况下, 循环等待断言优化比 Opti-MVBA 优化的效果更明显, 具体表现在循环等待断言优化较 Dumbo2 有 7.45%–64.43% 的延迟下降, 而 Opti-MVBA 有 4.57%–15.64% 的延迟下降, Dumbo2.5 有 23.65%–68.08% 的延迟下降; 在吞吐量上, 验证优化有 8.05%–181.13% 的增加, 置换优化有 4.79%–18.54% 的增加, Dumbo2.5 有 30.97%–213.26% 的增加.

实验 2. 在小规模实验中, 协议方节点数 $n=4$, 腐化节点数 $f=1$, 实验测试轮数为 500 轮. 变量为 k-RBC 的 k 值和单批次交易的大小. k 值从 2 到 4, 单批次交易大小从 25 000 笔交易到 500 000 笔交易. 实验数据如图 8 所示. 在实验结果中, 当 $k=3$ 时有更好的表现. 在吞吐量的表现上, 较于 PRBC, $k=2$ 有–0.03%~1.80% 的增加, $k=3$ 有–2.62%~3.57% 的增加, $k=4$ 有–1.92%~3.31% 的增加. 较为明显的是, 当单批量的大小较小时, k-RBC 比 PRBC, 或者是单实例 RBC 的吞吐量更小, 这是因为在单批量大小较小时, k-RBC 的多实例产生的额外开销, 较 k-RBC 带来的网络利用的优化减少的开销更明显, 但在单批量大小较大时有优势.

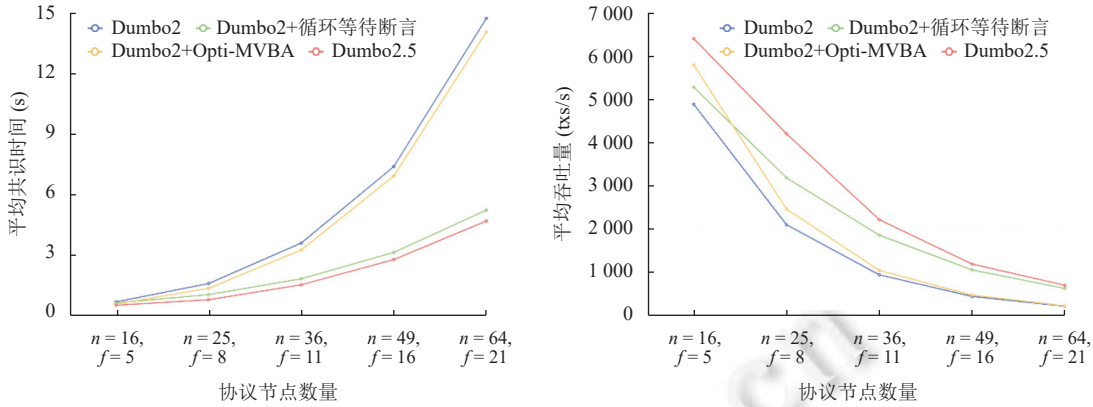


图 7 Dumbo 平均共识时间和平均吞吐量随协议节点数的变化

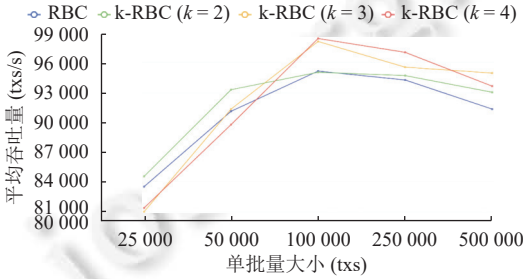


图 8 Super-Dumbo 吞吐量随 RBC 实例数和单批量交易大小变化

在大规模实验中, 协议方节点数 $n=4$, 腐化节点数 $f=1$, 实验测试轮数为 500 轮. k-RBC 的 $k=3$, 单批量交易大小从 100 000 笔交易到 1 000 000 笔交易. 实验结果如图 9 所示, 除开单批量为 100 000 笔交易时出现反常现象, 其余单批量时 k-RBC 的吞吐量均比 PRBC 高 (1.63%–12.95%). 反常现象推测, 在单批量为 100 000 时, k-RBC 实例造成的额外开销 (交易拆分和交易合并) 更大. 在 1% 极值吞吐量中, k-RBC 的 1% 极高吞吐量比 PRBC 更高 (1.23%–24.95%), 1% 极低吞吐量并没有一定规律, 但是极高极低吞吐量的差值 k-RBC 比 PRBC 高出 2.26%–84.79% (除单批量为 500 000 外, 低 6.45%).

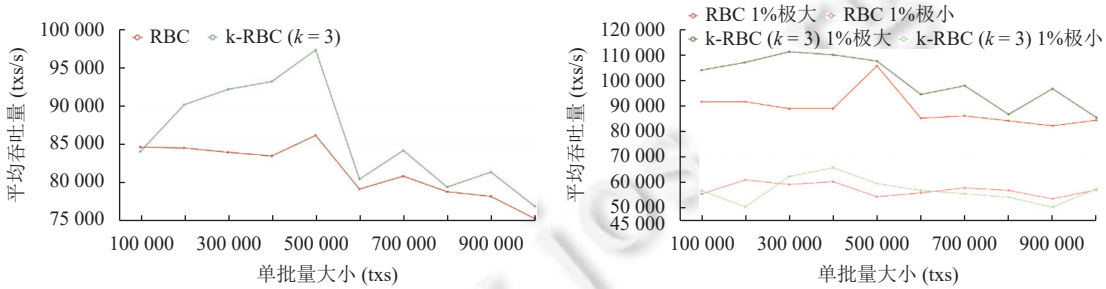


图 9 Super-Dumbo 在 RBC 和 k-RBC (k=3) 下不同单批量大小的平均吞吐量和极值吞吐量

实验 3. 在 Super-Dumbo 和 Dumbo2 的带宽利用率比较实验中, 设定协议方节点数 $n=4$, 10, 16, 腐化节点数 $f=1$, 3, 5, 单批量交易数为 1 000 txs, 2 500 txs, 5 000 txs. 实验测试轮数为至少 100 轮, 在协议方节点数量较少、腐化节点数较少的情况下, 额外提高该轮次的测试执行轮数, 以保证良好的网络情况和差的网络情况均包含在实验中. 实验结果如图 10 所示, 可以看到 Super-Dumbo 的带宽利用率, 在总体上优于 Dumbo2, 尤其是在协议节点数量少、单批量大小较大时更为明显, 有 7.42% 的优势. 但在协议节点数量较多、单批量大小较小时, 这种优势不明显, 降低至 0.68%.

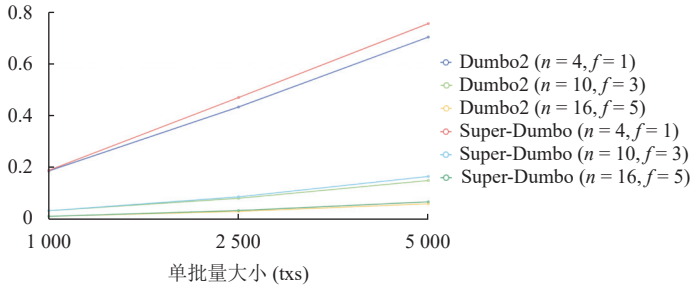


图 10 Super-Dumbo 和 Dumbo2 在不同节点数和单批量交易大小带宽利用率对比

实验 4. 本实验使用 py-spy 对 Super-Dumbo 在不同节点数和单批量交易大小下, RBC 阶段、MVBA 阶段的时间占比进行检测, 节点数和单批量交易大小的设置同实验 3, 实验结果如图 11 所示. 从 RBC 阶段的延迟来看, Super-Dumbo 的延迟在总体上小于 Dumbo2. 尤其在节点数为 10, 交易批次为 1 000 和 2 500 时, Super-Dumbo 的 RBC 阶段延迟分别小于 Dumbo2 的 42.63% (35.73 ms vs. 22.57 ms) 和 37.08% (38.27 ms vs. 25.46 ms). 这显示了 Super-Dumbo 在 RBC 阶段上明显优于 Dumbo2. 从 MVBA 阶段的延迟来看, Super-Dumbo 的延迟全面小于 Dumbo2. 在所有测试场景下, 我们都可以看出 Super-Dumbo 的 MVBA 阶段延迟都小于 Dumbo2, 最高达到了 25.79% (745.35 ms vs. 539.95 ms), 这显示了 Super-Dumbo 在 MVBA 阶段的延迟性能优势. 通过以上数据可得, Super-Dumbo 无论在 RBC 阶段还是 MVBA 阶段, 都能够提供更低的延迟. 证明了 Super-Dumbo 在整体性能上的优势, 这将使得该系统在需要快速响应和高效执行的情况下具有更好的应用前景.

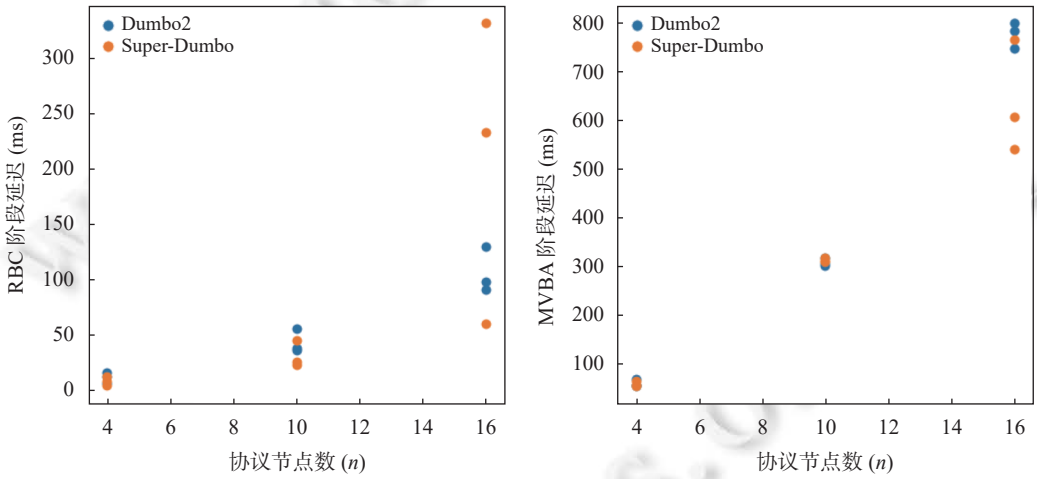


图 11 Super-Dumbo 和 Dumbo2 在不同节点数和单批量交易大小 RBC 阶段和 MVBA 阶段的延迟对比

以上实验证明: 1) 循环等待断言优化和 Opti-MVBA 优化均为共识协议带来较为明显的优化, 但循环等待断言优化在共识节点数量多的情况下有明显优势, 而 Opti-MVBA 在共识节点数量较小时有明显优势. 2) k-RBC 的优化程度取决于单批量交易大小, 当单批量足够大时, k-RBC 优化带来的开销减少才会多于 k-RBC 多实例带来的额外开销. 3) Super-Dumbo 在带宽利用率上较 Dumbo2 更有优势, 使得 Super-Dumbo 的总体性能有较好的提升. 4) Super-Dumbo 在 RBC 阶段和 MVBA 阶段较于 Dumbo2 有更好的性能提升, 尤其是延迟占比大的 MVBA 阶段, Super-Dumbo 有明显的提升. 总体来说, Super-Dumbo 在实验数据上较 Dumbo2 有明显提升.

6 总 结

针对分布式访问控制问题, 本文提出基于共识的访问权限控制模型 (CBAC), 并定义了其安全性, 并给出了具

体的设计思路和方法. 同时, 为了保证提出的算法能够适应未来大型网络环境下的性能需求, 我们提出了一个强安全高性能的 CBAC 访问控制引擎算法 Super-Dumbo, 实现了共识算法的性能优化, 提高了吞吐量和降低延迟, 满足低延迟和高吞吐量的性能需求. 原型验证实验表明, 与小飞象相比在平均时延上, Dumbo2.5 有 22.59%–27.85% 的下降; 在吞吐量的上, Opti-MVBA 优化有 15.28%–33.23% 的增加, 循环等待断言优化有 1.47%–8.30% 的增加, Dumbo2.5 有 29.18%–38.59% 的增加, 综合来看性能均优于原有小飞象协议.

References:

[1] Bertino E, Bettini C, Ferrari E, Samarati P. A temporal access control mechanism for database systems. IEEE Trans. on Knowledge and Data Engineering, 1996, 8(1): 67–80. [doi: 10.1109/69.485637]

[2] Ryutov T, Neuman C, Kim DH, Zhou L. Integrated access control and intrusion detection for Web servers. IEEE Trans. on Parallel and Distributed Systems, 2003, 14(9): 841–850. [doi: 10.1109/TPDS.2003.1233707]

[3] Centonze P. Security and privacy frameworks for access control big data systems. Computers, Materials & Continua, 2019, 59(2): 361–374. [doi: 10.32604/cmc.2019.06223]

[4] Xue KP, Gai N, Hong JN, Wei DSL, Hong PL, Yu NH. Efficient and secure attribute-based access control with identical sub-policies frequently used in cloud storage. IEEE Trans. on Dependable and Secure Computing, 2022, 19(1): 635–646. [doi: 10.1109/TDSC.2020.2987903]

[5] Han DZ, Zhu YJ, Li D, Liang W, Souri A, Li KC. A blockchain-based auditable access control system for private data in service-centric IoT environments. IEEE Trans. on Industrial Informatics, 2022, 18(5): 3530–3540. [doi: 10.1109/TII.2021.3114621]

[6] Ameer S, Benson J, Sandhu R. An attribute-based approach toward a secured smart-home IoT access control and a comparison with a role-based approach. Information, 2022, 13(2): 60. [doi: 10.3390/info13020060]

[7] Sandhu RS, Coyne EJ, Feinstein HL, Youman CE. Role-based access control models. Computer, 1996, 29(2): 38–47. [doi: 10.1109/2.485845]

[8] Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proc. of the 13th ACM Conf. on Computer and Communications Security. Alexandria: Association for Computing Machinery, 2006. 89–98. [doi: 10.1145/1180405.1180418]

[9] Yu SC, Wang C, Ren K, Lou WJ. Achieving secure, scalable, and fine-grained data access control in cloud computing. In: Proc. of the 2010 IEEE INFOCOM. San Diego: IEEE, 2010. 1–9. [doi: 10.1109/INFCOM.2010.5462174]

[10] Hebig RN, Meinel C, Menzel M, Thomas I, Warschofsky R. A Web service architecture for decentralised identity- and attribute-based access control. In: Proc. of the 2009 IEEE Int'l Conf. on Web Services. Los Angeles: IEEE, 2009. 551–558. [doi: 10.1109/ICWS.2009.89]

[11] Xinhuanet.com. Facebook's 50 million user information leaked. 2018 (in Chinese). http://www.xinhuanet.com/world/2018-03/24/c_129836684.htm

[12] CISOMAG. Instagram data breach! 49 million users' sensitive data exposed online. 2019. https://cisomag.com/instagram-data-breach-49-million-users-sensitive-data-exposed-online/#:~:text=May%2023%2C%202019%20Another%20data%20leak%20in%20Facebook%E2%80%99s, accounts%20have%20been%20found%20online%2C%20the%20TechCrunch%20reported

[13] ZDNET. Companies are leaking sensitive files via Box accounts. 2019. https://www.zdnet.com/article/companies-are-leaking-sensitive-files-via-box-accounts/

[14] CCTV. Northwestern Polytechnical University was attacked by the US NSA cyber attack: The US has gradually infiltrated and stolen secrets for a long time. 2022 (in Chinese). https://news.cctv.com/2022/09/27/ARTI1YjUCAzciKAsNQsy1Rxd220927.shtml

[15] National Computer Security Center. A guide to understanding discretionary access control in trusted systems. In: The 'Orange Book' Series. London: Springer, 1987. [doi: 10.1007/978-1-349-12020-8_8]

[16] Upadhyaya S. Mandatory access control. In: van Tilborg HCA, Jajodia S, eds. Encyclopedia of Cryptography and Security. 2nd ed., New York: Springer, 2011. 756–758. [doi: 10.1007/978-1-4419-5906-5_784]

[17] Maulina A, Rasjid ZE. Unified access management for digital evidence storage: Integrating attribute-based and role-based access control with XACML. Int'l Journal of Advanced Computer Science and Applications, 2024, 15(3): 1345–1353. [doi: 10.14569/IJACSA.2024.01503131]

[18] Kalam AAE, Baida RE, Balbiani P, Benferhat S, Cuppens F, Deswarte Y, Miege A, Saurel C, Trouessin G. Organization based access control. In: Proc. of the 4th Int'l Workshop on Policies for Distributed Systems and Networks. Lake Como: IEEE, 2003. 120–131. [doi: 10.1109/POLICY.2003.1206966]

[19] Laamech N, Munier M, Pham C. Translating usage control policies to semantic rules: A model using OrBAC and SWRL. Procedia Computer Science, 2023, 225: 1881–1890. [doi: 10.1016/j.procs.2023.10.178]

[20] Yuan E, Tong J. Attributed based access control (ABAC) for Web services. In: Proc. of the 2005 IEEE Int'l Conf. on Web Services. Orlando: IEEE, 2005. 561–569. [doi: 10.1109/ICWS.2005.25]

[21] Hu VC, Ferraiolo D, Kuhn R, Schnitzer A, Sandlin K, Miller R, Scarfone K. Guide to attribute based access control (ABAC) definition and considerations. Gaithersburg: National Institute of Standards and Technology, 2014.

[22] Shang SY, Wang XH, Liu AD. ABAC policy mining method based on hierarchical clustering and relationship extraction. Computers & Security, 2024, 139: 103717. [doi: 10.1016/j.cose.2024.103717]

[23] Choksy P, Chaurasia A, Rao UP, Kumar S. Attribute based access control (ABAC) scheme with a fully flexible delegation mechanism for IoT healthcare. Peer-to-peer Networking and Applications, 2023, 16(3): 1445–1467. [doi: 10.1007/s12083-023-01486-w]

[24] Liu YF, Zhao B, An Y, Guo JB. DACAS: Integration of attribute-based access control for northbound interface security in SDN. World Wide Web, 2023, 26(4): 2143–2173. [doi: 10.1007/s11280-022-01130-2]

[25] Perez-Haro A, Diaz-Perez A. Attribute-based access control rules supported by biclique patterns. In: Proc. of the 9th Int'l Conf. on Big Data Computing Service and Applications (BigDataService). Athens: IEEE, 2023. 95–102. [doi: 10.1109/BigDataService58306.2023.00020]

[26] Ruan CH, Hu CQ, Li XW, Deng SJ, Liu ZW, Yu JG. A revocable and fair outsourcing attribute-based access control scheme in metaverse. IEEE Trans. on Consumer Electronics, 2024, 70(1): 3781–3791. [doi: 10.1109/TCE.2024.3377107]

[27] Guo BY, Lu ZL, Tang Q, Xu J, Zhang ZF. Dumbo: Faster asynchronous BFT protocols. In: Proc. of the 2020 ACM SIGSAC Conf. on Computer and Communications Security. New York: Association for Computing Machinery, 2020. 803–818. [doi: 10.1145/3372297.3417262]

[28] Bai QH, Zheng Y. Study on the access control model. In: Proc. of the 2011 Cross Strait Quad-regional Radio Science and Wireless Technology Conf. Harbin: IEEE, 2011. 830–834. [doi: 10.1109/CSQRWC.2011.6037079]

[29] Lin C, Feng FJ, Li JS. Access control in new network environment. Ruan Jian Xue Bao/Journal of Software, 2007, 18(4): 955–966 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/955.htm

[30] Wang YD, Yang JH, Xu C, Ling X, Yang Y. Survey on access control technologies for cloud computing. Ruan Jian Xue Bao/Journal of Software, 2015, 26(5): 1129–1150 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

[31] Jemel M, Serhrouchni A. Decentralized access control mechanism with temporal dimension based on blockchain. In: Proc. of the 14th IEEE Int'l Conf. on e-Business Engineering (ICEBE). Shanghai: IEEE, 2017. 177–182. [doi: 10.1109/ICEBE.2017.35]

[32] Ravidas S, Lekidis A, Paci F, Zannone N. Access control in Internet-of-Things: A survey. Journal of Network and Computer Applications, 2019, 144: 79–101. [doi: 10.1016/j.jnca.2019.06.017]

[33] Paillisse J, Subira J, Lopez A, Rodriguez-Natal A, Ermagan V, Maino F, Cabellos A. Distributed access control with blockchain. In: Proc. of the 2019 IEEE Int'l Conf. on Communications (ICC). Shanghai: IEEE, 2019. 1–6. [doi: 10.1109/ICC.2019.8761995]

[34] Cruz JP, Kaji Y, Yanai N. RBAC-SC: Role-based access control using smart contract. IEEE Access, 2018, 6: 12240–12251. [doi: 10.1109/ACCESS.2018.2812844]

[35] Hardjono T, Pentland A. Verifiable anonymous identities and access control in permissioned blockchains. arXiv:1903.04584, 2019.

[36] Anjana PS, Kumari S, Peri S, Rathor S, Somani A. An efficient framework for optimistic concurrent execution of smart contracts. In: Proc. of the 27th Euromicro Int'l Conf. on Parallel, Distributed and Network-Based Processing (PDP). Pavia: IEEE, 2019. 83–92. [doi: 10.1109/EMPDP.2019.8671637]

[37] Dickerson T, Gazzillo P, Herlihy M, Koskinen E. Adding concurrency to smart contracts. In: Proc. of the 2017 ACM Symp. on Principles of Distributed Computing. Washington: Association for Computing Machinery, 2017. 303–312. [doi: 10.1145/3087801.3087835]

[38] Lamport L, Shostak R, Pease M. The Byzantine generals problem. ACM Trans. on Programming Languages and Systems (TOPLAS). ACM, 1982, 4(3): 382–401. [doi: 10.1145/357172.357176]

[39] Cachin C, Kursawe K, Petzold F, Shoup V. Secure and efficient asynchronous broadcast protocols. In: Proc. of the 21st Annual Int'l Cryptology Conf. (CRYPTO). Santa Barbara: Springer, 2001. 524–541. [doi: 10.1007/3-540-44647-8_31]

[40] Boldyreva A. Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-Group signature scheme. In: Proc. of the 6th Int'l Workshop on Theory and Practice in Public Key Cryptography. Springer, 2003, 31–46. [doi: 10.5555/648120.747061]

[41] Baek J, Zheng YL. Simple and efficient threshold cryptosystem from the gap Diffie-Hellman group. In: Proc. of the 2003 IEEE Global Telecommunications Conf. San Francisco: IEEE, 2003. 1491–1495. [doi: 10.1109/GLOCOM.2003.1258486]

附中文参考文献:

[11] 新华网. 脸书 5000 万用户信息泄露. 2018. http://www.xinhuanet.com/world/2018-03/24/c_129836684.htm

[14] 央视网. 西北工业大学遭美国 NSA 网络攻击: 美方逐步渗透、长期窃密. 2022. https://news.cctv.com/2022/09/27/ARTI1YjUCAzciKAsNQsy1Rxd220927.shtml

[29] 林闯, 封富君, 李俊山. 新型网络环境下的访问控制技术. 软件学报, 2007, 18(4): 955–966. http://www.jos.org.cn/1000-9825/18/955.htm

[30] 王于丁, 杨家海, 徐聪, 凌晓, 杨洋. 云计算访问控制技术研究综述. 软件学报, 2015, 26(5): 1129–1150. http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

韩将(1986－), 男, 博士, 工程师, 主要研究领域为零信任网络安全架构, 密码协议.

胡可欣(1991－), 女, 博士, 副研究员, 主要研究领域为区块链技术, 加密货币, 安全协议.

张振峰(1972－), 男, 博士, 研究员, 博士生导师, 主要研究领域为密码学, 数据安全, 抗量子密码, 区块链密码, 网络信任, 隐私保护.

何双羽(1989－), 男, 博士, 助理研究员, 主要研究领域为网络安全, 密码协议安全性分析.

刘雨果(2000－), 男, 硕士生, 主要研究领域为区块链技术, 安全协议.