软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(4):1838−1853 [doi: 10.13328/j.cnki.jos.007485] [CSTR: 32375.14.jos.007485]

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

基于真实源地址验证的轻量共识机制*

徐 易[1], 陈熠豪[2], 王晓亮[3], 徐 恪[2], 李 琦[1]

[1](清华大学 网络科学与网络空间研究院, 北京 100084)

[2](清华大学 计算机科学与技术系, 北京 100084)

[3](首都师范大学 信息工程学院, 北京 100048)

通信作者: 李琦, E-mail: qli01@tsinghua.edu.cn

摘 要: 近年来, 许多研究提出利用共识机制增强网络层安全性. 然而, 现有共识机制存在密钥维护数量多、信任关系传递不灵活和节点身份验证开销大等局限, 难以满足网络层功能的性能需求. 为解决这些问题, 提出一种基于真实源地址验证技术的轻量共识框架. 该框架在多个层次上优化共识效率: 首先, 针对同一地址域内的共识节点, 该框架利用真实地址作为身份识别标志, 通过域内节点共享同一密钥的方式实现密钥聚合, 从而大幅降低所需维护的密钥数量; 其次, 在地址域的粒度上, 该框架构建以真实地址为信任基础的网络信任联盟, 基于前缀树聚合可信地址域, 从而在实现灵活信任传递的同时, 进一步降低所需维护的密钥数量; 最后, 在节点层面, 针对传统共识节点身份验证开销大的问题, 该框架设计基于真实地址和对称密钥的分步验证机制, 从而有效降低共识开销, 实现共识过程轻量化. 仿真实验结果表明, 所提出的轻量共识框架与基于 ECDSA 身份验证的共识机制相比, 平均可提升 70% 共识吞吐量并降低 40% 共识计算开销, 显著提升了共识效率.

关键词: 真实源地址验证; 共识机制; 网络层安全

中图法分类号: TP393

中文引用格式: 徐易, 陈熠豪, 王晓亮, 徐恪, 李琦. 基于真实源地址验证的轻量共识机制. 软件学报, 2026, 37(4): 1838–1853. http://www.jos.org.cn/1000-9825/7485.htm

英文引用格式: Xu Y, Chen YH, Wang XL, Xu K, Li Q. Lightweight Consensus Mechanism Based on Source Address Validation. Ruan Jian Xue Bao/Journal of Software, 2026, 37(4): 1838–1853 (in Chinese). http://www.jos.org.cn/1000-9825/7485.htm

Lightweight Consensus Mechanism Based on Source Address Validation

XU Yi[1], CHEN Yi-Hao[2], WANG Xiao-Liang[3], XU Ke[2], LI Qi[1]

[1](Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China)

[2](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

[3](Information Engineering College, Capital Normal University, Beijing 100048, China)

Abstract: In recent years, many studies have proposed using consensus mechanisms to enhance network layer security. However, existing consensus mechanisms have limitations, such as heavy key maintenance, inflexible trust expansion, and high authentication overhead. To address these issues, this study proposes a lightweight consensus framework based on source address validation. The framework optimizes consensus efficiency at multiple levels: First, among consensus nodes within the same domain, the framework uses authentic IP addresses as identities and achieves key aggregation by sharing the same key among nodes within the domain, thus efficiently reducing the number of keys that need to be maintained. Second, at the domain level, the framework constructs a trusted network alliance based on trust derived from authentic IP addresses and aggregates trusted domains through a prefix tree, thus further reducing the number of keys to maintain while enabling flexible trust expansion. Finally, at the node level, to address the issue of high authentication overhead, the framework designs a two-step authentication mechanism based on authentic IP addresses and symmetric keys, effectively reducing

* 基金项目: 国家自然科学基金 (62132011)
收稿时间: 2024-12-24; 修改时间: 2025-02-06; 采用时间: 2025-06-09; jos 在线出版时间: 2025-11-05
CNKI 网络首发时间: 2025-11-06

overhead and enabling a lightweight consensus process. Simulation experiment results show that the proposed framework can improve consensus throughput by 70% and reduce consensus latency by 40% on average, compared to the consensus mechanism based on ECDSA authentication, significantly improving consensus efficiency.

Key words: source address validation; consensus mechanism; network layer security

近年来, 随着区块链技术的广泛应用和共识机制的不断发展, 利用共识机制增强网络层安全性逐渐成为工业界和学术界关注的热点. 现有研究尝试利用共识机制的分布式信任特点, 以解决资源公钥基础设施 (resource public key infrastructure, RPKI) 等传统网络基础设施中信任集中化所带来的安全问题. 例如, IPchain[1]等方法通过引入 PoS 共识机制[2], 构建分布式的信任架构, 实现防篡改和高容错的互联网号码资源管理. 然而, 当前主流共识机制的开销普遍过高, 难以满足网络层功能的性能需求[3–5]. 以比特币中采用的 PoW 共识机制[6]为例, 其在每条交易上的时间开销高达 0.1 s, 远高于网络层功能的一般时延[7,8]. 这些性能限制严重制约了共识机制在网络层安全性增强场景中的实际应用. 因此, 如何有效降低共识开销、提升共识性能, 使其适用于网络层安全需求, 已成为当前共识研究领域亟待解决的重要问题[9].

导致当前主流共识机制性能薄弱的原因主要有 3 个方面. 首先, 密钥维护开销大. 为确保共识节点的身份真实可信, 主流共识机制通常为每个节点单独维护密钥以进行身份验证. 这导致所需维护的密钥数量众多, 增加了密钥维护成本. 其次, 信任关系传递不灵活. 主流共识机制普遍采用基于非对称加密证书的身份验证方式, 依赖公钥基础设施 (public key infrastructure, PKI) 建立的证书链进行信任传递, 且信任关系仅限于共识节点之间. 这限制了信任关系传递的灵活性, 增加了信任维护的复杂度. 最后, 节点身份验证开销大. 现有主流共识机制通常假设共识节点完全不可信, 忽视了互联网 IP 地址这一潜在信任基础, 因而引入大量安全机制与复杂的身份验证流程. 这导致节点身份验证开销过高, 进而降低了共识机制的整体性能.

真实源地址验证技术为解决上述问题提供了新的思路. 真实源地址验证是一类针对网络层源地址进行真实性验证的技术. 以 Wu 等人[10]提出的源地址验证体系结构 (source address validation architecture, SAVA) 为例, 该架构将网络根据管理范围划分为多个地址域[11], 通过域内和域间等多层次的验证方式, 确保其中的 IP 地址真实可信. 当前, 以 SAVA 为代表的真实源地址验证技术已在互联网上得到了规模部署, 且规模持续增长. 因此, 本文尝试利用真实 IP 地址这一广泛存在的信任基础, 将真实 IP 地址作为共识机制中节点身份验证的信任锚点, 在不损失安全性的同时有效降低共识开销, 从而构建适用于网络层安全需求的轻量级共识机制.

基于上述思路, 本文提出了一种基于真实地址的轻量共识框架. 该框架围绕真实 IP 地址, 在多个层次上优化共识效率. 首先, 在同一地址域内, 该框架利用真实地址作为身份识别标志, 通过域内节点共享同一密钥的方式实现密钥聚合, 从而大幅降低所需维护的密钥数量, 解决了传统共识机制中密钥维护开销大的问题. 其次, 在不同的地址域之间, 该框架构建了以真实地址为信任基础的网络信任联盟, 基于前缀树聚合可信地址域, 从而解决了传统共识机制中信任关系传递不灵活的问题. 最后, 在节点层面, 为解决传统共识机制中节点身份验证开销大的问题, 该框架设计了基于真实地址和对称密钥的分步验证机制, 从而有效降低验证开销, 实现共识过程轻量化. 本文对所提出的轻量共识框架进行了原型系统实现, 对其安全性进行了分析, 并通过仿真性能测试验证了其实际效果. 结果表明, 本文提出的框架能够达到与传统非匿名共识相似的安全性, 且相较基于椭圆曲线签名算法 (elliptic curve digital signature algorithm, ECDSA) 非对称身份验证机制的共识框架, 降低了 40% 的共识计算开销, 增加了 70% 的共识吞吐量, 实现了共识效率的大幅提升.

本文主要贡献如下.

(1) 提出一种基于真实地址的轻量共识框架, 将真实可信 IP 地址作为身份标识, 构建以真实地址为核心的多层次信任机制, 有效降低密钥维护开销、提升信任关系传递灵活性, 并实现轻量化共识过程.

(2) 提出基于真实地址的域内密钥聚合机制, 通过可信地址域内节点共享同一对称密钥, 显著降低密钥数量和管理开销, 提升整体共识效率.

(3) 设计基于真实地址的可信地址域聚合方法, 通过前缀树结构高效聚合可信地址域, 实现灵活的跨域信任扩展, 显著提升共识性能和可扩展性.

(4) 设计并实现所提出框架的原型系统, 在仿真环境中测试了不同节点数量下的共识效率、吞吐量与信任联盟动态更新效率. 实验结果表明, 与基于 ECDSA 身份验证的共识机制相比, 本文提出的框架在共识计算开销上降低了 40%, 在共识吞吐率上提升了 70%, 显著降低了共识开销.

1 相关工作

1.1 利用共识机制的网络层安全增强

当前, 许多互联网基础设施 (如资源公钥基础设施) 采用中心化架构, 依赖于单一或少数可信节点提供网络层的核心功能与服务. 然而, 这种架构导致攻击面高度集中, 不仅容易引发单点故障, 还存在功能实施不透明、审计困难等问题. 相比之下, 共识机制避免了对单一可信方的依赖, 能够实现分布式信任, 使系统在无中心控制的情况下协作运行, 从而提升系统的安全性、公平性与透明性.

因此, 近年来, 许多研究工作尝试利用共识机制解决网络层中的中心化问题, 以增强网络层安全性. de la Rocha Gómez-Arevalillo 等人[12]提出了完全去中心化的信任管理系统 SBTM, 该系统基于工作量证明 (proof of work, PoW) 共识算法, 并通过类似比特币钱包的 SBTMwallet 管理互联网自治系统. Xing 等人[13]提出了 BGPcoin, 这是一种基于权益证明 (proof of stake, PoS) 共识算法的互联网号码资源授权和可信管理方案. BGPcoin 通过在以太坊上运行智能合约来执行和审计资源分配, 为边界网关协议 (border gateway protocol, BGP) 提供可信的路由源验证功能. Paillisse 等人[1]提出了 IPchain, 利用区块链技术和 PoS 共识算法构建分布式信任架构, 提供灵活的信任模型. IPchain 将 IP 地址前缀和互联网自治系统号存储在区块链上, 模仿加密货币对 IP 地址前缀进行管理, 实现了 IP 地址前缀转移、链上 BGP 消息来源验证等功能. 类似地, He 等人[14]提出了 ROAchain. 在 ROAchain 中, 每个互联网自治系统维护了一个全球一致且防篡改的路由源授权 (route origin authorization, ROA) 仓库, 用于验证路由来源的合法性, 从而防止 BGP 前缀劫持攻击. ROAchain 采用了一种新的共识算法, 引入可信值、分片等机制, 以提升共识效率和吞吐. Li 等人[15]提出了 DeBGP, 将每个 BGP 更新消息转化为相应的共识内容, 实现查询验证. 此外, DeBGP 将 AS 分为不同的联盟, 在联盟内执行本地验证, 在联盟间执行基于区块链的验证, 以提升验证效率. Saad 等人[16]提出了 RouteChain, 利用共识机制的源真实性和防篡改特性对抗 BGP 劫持, 并保持 Internet 路由路径的一致视图. RouteChain 根据自治域的地理接近程度进行分组, 构建出双层区块链模型对前缀宣告进行检测.

尽管上述研究强调了共识机制在增强网络层安全性方面的应用潜力, 但当前主流共识机制普遍面临共识开销过大的问题, 严重制约了其在实际网络场景中的应用. 例如, SBTM 采用 PoW 共识算法, 其共识吞吐量仅为每秒 7 条交易; BGPcoin 和 IPchain 采用 PoS 共识算法, 其共识吞吐量分别仅为每秒 15 条和每秒 10 条交易; ROAchain 虽引入了新型共识算法, 其吞吐量也仅为每秒 140 条交易, 远不足以满足网络层功能的性能需求. RouteChain 采用了 PoS、Clique 和 PoET 这 3 种共识算法, 然而吞吐量均不超过每秒 200 条交易. 其他常见的共识算法, 如 Paxos[17–19]、Raft[20]和 PBFT[21]等, 同样面临着共识开销过大的问题.

1.2 真实源地址验证

除了针对网络层协议和基础设施进行安全增强外[22,23], 利用真实源地址验证技术验证 IP 地址也能提升网络层安全性并提供信任基础. 真实源地址验证是一类针对网络层源地址进行真实性验证的技术, 其主要目的是通过验证数据包的源 IP 地址的真实性, 防止伪造源地址的网络攻击. Cao 等人[24]提出了 pSAV, 在控制面利用区块链作为信任基础, 提供服务订购、审核和激励, 在数据面利用 P4 可编程交换机实现灵活高效的源地址验证. Yang 等人[25]提出了 SEC, 采用哈希和查表方法将源地址映射为伪随机标签, 利用标签进行源地址验证. Qin 等人[26]提出了 UniSAV, 对现有源地址验证机制进行了统一抽象, 并为其实施和开发提供了一种简化的方法. 当前较为典型的真实源地址验证技术是由 Wu 等人[10]提出的源地址验证体系结构 (SAVA), 自 2008 年通过 IETF 标准化以来, 已在互联网上得到了规模部署. SAVA 的验证体系分为 3 个层次: 接入网源地址验证、地址域内源地址验证和地址域间源地址验证.

首先, 在接入网层次, SAVA 提供主机粒度的源地址验证能力, 以保证该接入网内的主机之间无法相互伪造地

址[27–29]. 具体而言, SAVA 通过将合法 IP 地址与主机的链路层属性绑定形成绑定锚, 进而匹配数据包中的 IP 源地址与绑定锚, 以验证其真实性. 其次, 在地址域内源地址验证层次, SAVA 确保从接入网流入地址域的流量, 其源地址无法假冒该地址域下的其他接入网地址. 具体地, 通过发送探测报文, SAVA 生成源前缀和入接口的对应关系, 构建源地址验证表, 从而验证地址域内流量的源地址真实性. 最后, 在地址域间源地址验证层次, SAVA 通过部署在边界路由器上的状态机验证源地址. 具体地, 每个源地址域针对不同的目的地址域生成不同的状态机, 以验证所发送的报文是否源于本地址域, 并同时为报文添加标识源地址域的可验证身份标签, 以防止源地址域地址被冒用.

以 SAVA 为代表的真实源地址验证技术为网络层节点的身份验证提供了可信基础. 在一个全面部署真实源地址验证技术的网络中, 伪造源地址的难度及成本显著增加, 从而确保报文的源地址真实可靠且可追溯. 随着 SAVA 部署率的不断提升, 基于 SAVA 确保共识消息源地址真实性逐渐成为可能, 为构建以 IP 地址为信任基础的轻量级共识机制提供了可行的实现路径.

2 基于真实地址的轻量共识框架概览

为有效降低共识开销、提升共识性能, 以满足网络层安全需求, 本文提出了基于真实地址的轻量共识框架. 其核心思路是充分利用真实 IP 地址这一广泛存在的信任基础, 将真实 IP 地址作为共识机制中节点身份验证的信任锚点. 基于此思路, 该框架设计了域内密钥聚合、地址域间聚合和节点轻量共识这 3 个核心过程, 从而有效解决现有机制面临的密钥维护数量多、信任传递不灵活和身份验证开销大的问题.

该框架的运行流程如图 1 所示, 具体包含以下 3 个核心过程: (1) 基于真实地址的域内密钥聚合. 该过程面向同一地址域内的共识节点, 利用真实 IP 地址验证节点身份, 而非依赖节点各自维护的非对称密钥, 因而可将每个共识节点单独的密钥聚合为统一的地址域密钥, 大幅降低密钥协商和维护的复杂度. (2) 基于前缀树的信任联盟域间聚合. 该过程面向不同的地址域, 构建以真实 IP 地址为信任基础的网络信任联盟, 并利用前缀树维护地址域的真实源地址验证技术部署状态, 结合地址域的多种前缀关系完成域间聚合, 从而在灵活传递信任关系的同时, 进一步减少需要维护的密钥数量. (3) 基于 IP 地址身份验证机制的轻量共识. 该过程面向共识节点间的身份验证, 通过真实 IP 地址和对称密钥进行分步认证, 取代传统基于非对称密钥的验证方式, 有效降低共识过程的时间和运算开销, 实现共识过程轻量化.

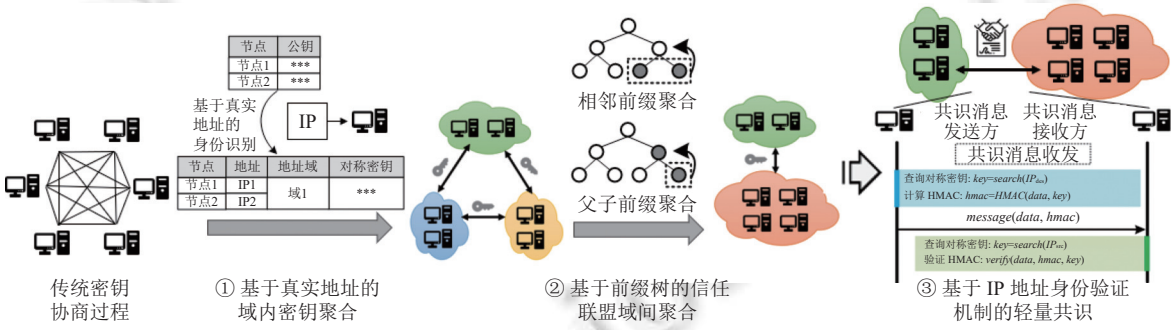

图 1 轻量共识框架概览

本文将在第 3–5 节中分别对域内密钥聚合、地址域间聚合和节点轻量共识作详细介绍.

3 域内密钥聚合

传统共识框架中, 每个共识节点都需单独维护密钥, 带来了很高的密钥维护开销. 为了解决这一问题, 本文提出的轻量共识框架使用 IP 地址代替非对称密钥作为域内节点的身份标识, 从而实现域内密钥聚合. 如图 2 所示, 传统共识框架利用节点公私钥对的唯一性, 将节点身份与公钥绑定, 因此需要为每个节点分别维护非对称密钥. 而本文的轻量共识框架则利用共识过程中节点 IP 地址的唯一性, 通过真实源地址验证技术确保 IP 地址真实可信,

构建 IP 地址和共识节点的可信绑定关系. 在完成 IP 地址与节点的一一对应后, 节点身份的验证仅需依据其 IP 地址所属的地址域, 使用该地址域内共享的一个对称密钥, 而无需为每个节点单独维护非对称密钥. 因此, 这一基于真实地址的域内密钥聚合过程可以大幅降低所需维护的密钥数量.

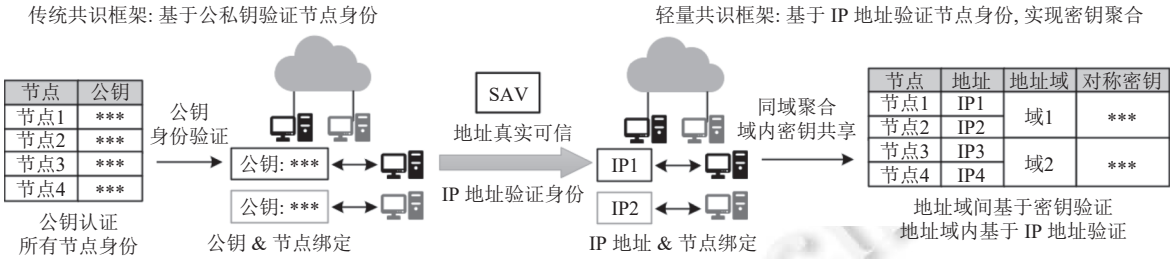

图 2 轻量共识框架密钥聚合过程

具体而言, 针对不同的真实源地址验证技术部署场景, 该域内密钥聚合过程存在两种实施方式.

首先, 在共识环境中所有地址域均部署真实源地址验证技术的情况下, 本文的轻量共识框架采用密钥同域聚合、域内密钥共享的方式降低密钥维护开销. 框架在地址域层面维护地址域与对称密钥的一一对应关系, 以验证共识节点所属地址域的真实性. 地址域对应的对称密钥由域内的所有共识节点共享, 从而取代每个节点原本需要单独维护的非对称密钥. 通过将每个节点的非对称密钥聚合为地址域的对称密钥, 框架实现了域内密钥的高效聚合. 在节点身份验证过程中, 当共识节点位于同一地址域时, 可根据 IP 地址和节点的绑定关系验证节点身份; 当共识节点分属不同地址域时, 则先通过对称密钥验证节点所在地址域的真实性, 进而通过 IP 地址完成节点身份验证.

其次, 针对共识环境中真实源地址验证技术尚未完全部署的情况, 本文的轻量共识框架以地址域为单位, 在维护对称密钥的基础上额外维护非对称密钥, 用以兼容传统基于非对称密钥的身份验证体系. 对于已部署真实源地址验证技术的地址域, 与对称密钥相似, 其公私钥由所有域内共识节点共享, 从而替代每个节点的公私钥, 实现域内密钥聚合. 而对于未部署真实源地址验证技术的地址域, 其节点仍需各自维护独立的公私钥. 对于未部署真实源地址验证技术的地址域内节点, 在其身份验证过程中, 若待验证节点所属的地址域部署了真实源地址验证技术, 将先根据地址域的公私钥验证地址域真实性, 然后通过 IP 地址验证节点身份; 否则, 将依照传统基于非对称密钥的身份验证流程, 通过节点各自维护的公私钥验证身份.

在实际实现中, 本文的轻量共识框架在每个地址域内通过一个密钥维护模块实现上述功能. 该模块的核心功能主要包括域间密钥维护和域内密钥分发, 以确保地址域内所有共识节点共享统一的“密钥-地址域”对应关系. 域间密钥维护功能负责与其他已部署真实源地址验证技术的地址域协商密钥并定期更新, 构建地址域与密钥信息的一一对应关系. 密钥协商过程定期进行, 独立于共识过程, 因而可以有效降低共识过程中的通信开销和时间开销. 域内密钥分发功能负责将更新后的“密钥-地址域”对应关系同步给域内的所有共识节点. 为保证信息一致性, 密钥分发采用了两种机制: 一是主动向共识节点推送更新后的对照表, 二是支持节点按需查询并拉取对照表. 这种设计可确保全地址域内的共识节点记录的一致性, 有效降低因网络故障等原因引发的信息不一致问题, 减少潜在的安全风险.

4 地址域间聚合

传统共识机制依赖证书链进行信任传递, 且信任关系仅限于共识节点之间, 导致信任关系传递不灵活. 为了解决这一问题, 本文提出的轻量共识框架通过引入地址域间的网络信任联盟, 以真实 IP 地址为信任基础, 在地址域之间构建灵活的信任关系, 进而充分利用多种前缀关系, 借助前缀树对地址域进行聚合, 从而有效降低信任维护的开销.

具体而言, 网络信任联盟的实现包括两个关键步骤: 信任基础的构建与地址域的聚合. 在信任基础构建过程中, 信任联盟根据真实源地址验证技术部署情况, 对联盟成员进行筛选和记录, 确保成员所代表的地址域均已完全部署真实源地址验证技术, 从而形成稳健的信任关系网络. 随后, 在地址域聚合过程中, 信任联盟根据地址域间的前缀关系, 采用父子节点聚合或兄弟节点聚合策略, 最大限度地减少联盟内地址域的数量, 从而提高信任管理的整

体效率.

在实际运行中, 网络信任联盟的工作流程分为两个阶段: 激活阶段和动态更新阶段. 激活阶段主要面向已完全部署真实源地址验证技术的地址域. 这些地址域相互验证对方的真实源地址验证技术部署情况, 建立地址域粒度的信任关系及真实地址域前缀表. 构建完成后, 信任联盟基于前缀关系对联盟内的地址域进行聚合. 动态更新阶段则涵盖 3 种情景: 新地址域加入信任联盟、信任联盟的动态维护以及地址域退出信任联盟. 在动态更新阶段, 信任联盟需验证地址域的真实源地址验证技术部署情况, 并基于前缀树重新进行父子节点聚合或兄弟节点聚合. 完成聚合后, 联盟会更新真实地址域前缀表, 确保联盟内的地址域列表及信任关系始终保持一致.

4.1 网络信任联盟激活

在激活阶段, 网络信任联盟采取先建立地址域间信任关系、后进行全局共识的方式, 以维护联盟内统一的信任基础. 具体而言, 拟加入信任联盟的地址域需向其他已完全部署真实源地址验证技术的地址域发起通信请求. 接收方通过验证请求方的真实源地址验证技术部署情况, 以筛选并排除恶意或误操作的地址域. 验证通过, 双方建立域间信任关系. 在地址域间通信结束后, 信任联盟会汇总所有拟加入联盟的地址域信息, 根据 IP 前缀长度从短到长对这些地址域的真实源地址验证技术的部署情况进行共识验证. 通过让所有地址域参与共识过程, 信任联盟确保了真实源地址验证技术部署信息的全局一致性, 从而构建信任基础, 并有效防止恶意或误操作的地址域加入信任联盟.

在实现灵活的信任传递后, 网络信任联盟借助前缀树对联盟内地址域进行聚合. 具体如算法 1 所示.

算法 1. 真实地址域前缀树、前缀表生成算法.

输入: 已经完全部署真实源地址验证技术的 IP 地址域列表 *IP_domain_table*;
输出: 网络信任联盟真实地址域前缀树和前缀表.

```
FOR IP_domain in IP_domain_table:
    FOR bit in IP_domain:
        // 判断是否可以进行父子节点聚合, 若当前地址域存在完全部署了真实源地址验证技术的上级域, 则
           不将其纳入信任联盟
        IF cur_node.has_domain = True:
            cur_node = tree_root
            BREAK
        // 若不存在, 则纳入
        IF bit = IP_domain.end:
            cur_node.has_domain = True
            table.add(IP_domain)
            cur_node.remove(sons)
            table.remove(cur_node.sons)
            // 递归方式判断是否可以进行兄弟节点聚合, 若当前地址域存在相邻地址域可构成更大的地址域,
               则将该更大的地址域纳入信任联盟
            WHILE cur_node.father.ano_son.has_domain = True:
                combine(cur_node, cur_node.father.ano_son)
            END WHILE
    END FOR
END FOR
```

网络信任联盟通过前缀树记录联盟内地址域, 并根据地址域间的前缀关系进行聚合, 在确保信任联盟覆盖率

不受影响的前提下, 最大限度压缩地址域数量. 如图 3 所示, 针对不同的地址域间前缀关系, 聚合方式分为两种: 父子节点聚合和兄弟节点聚合.

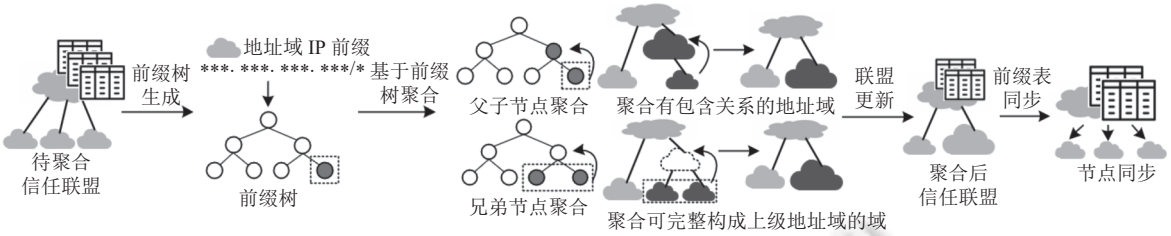

图 3 网络信任联盟中地址域聚合过程

父子节点聚合: 网络信任联盟通过父子节点聚合, 将一个地址域所包含的所有下级地址域全部聚合到该地址域. 具体而言, 在对地址域的真实源地址验证技术部署情况进行共识时, 如果发现该地址域存在已经加入信任联盟的上级地址域, 则该地址域将被舍弃, 即不在前缀树和前缀表中进行记录, 也不允许其加入信任联盟. 只有在前缀树搜索过后确认该地址域不存在已加入信任联盟的上级地址域时, 联盟才会对其真实源地址验证技术部署情况进行共识验证, 验证通过后允许其加入信任联盟. 在该地址域加入信任联盟后, 其下级地址域 (如果存在) 将从信任联盟中移除. 这是因为网络信任联盟认为, 若一个地址域已完全部署真实源地址验证技术, 则该地址域的真实性可以保证所有下级地址域的源地址真实性. 因此, 对于该地址域的下级地址域, 无需进一步验证, 也不允许其单独加入信任联盟.

兄弟节点聚合: 网络信任联盟通过兄弟节点聚合, 将可完整构成上级地址域的相邻地址域聚合为该上级地址域. 具体而言, 对于申请加入联盟的地址域, 信任联盟会检查其在前缀树上的相邻地址域, 并判断能否聚合成更大的上级地址域. 如果可以, 信任联盟会将这些地址域聚合, 并将聚合后的上级地址域加入联盟, 同时移除原有的下级地址域. 兄弟节点聚合的可行性在于, 这些被聚合的地址域均已完全部署真实源地址验证技术, 因此这些地址域内的共识节点无法假冒其他节点的 IP 地址. 经过聚合后, 上级地址域内的每个共识节点的源地址真实性与这些地址域分别部署真实源地址验证技术的效果相同, 因此确保了上级地址域真实可信.

4.2 网络信任联盟动态更新

为保证网络信任联盟的可扩展性和信任基础的实时一致性, 在完成激活后, 联盟便进入动态更新阶段. 具体而言, 网络信任联盟的动态更新可以分成 3 种场景: 地址域加入、地址域动态维护以及地址域退出.

地址域加入: 为保证网络信任联盟的可扩展性, 联盟允许完全部署真实源地址验证技术的地址域申请加入, 并在验证通过后对新地址域进行聚合优化. 具体而言, 如算法 2 所示, 地址域的加入过程分为 3 步: 验证真实源地址验证技术的部署情况、对新地址域进行聚合以及更新信任联盟内记录. 在处理加入申请时, 信任联盟首先通过真实地址域前缀表验证该申请地址域, 检查其是否为现有信任联盟地址域的下级地址域. 如果是, 则对其进行父子节点聚合, 并拒绝其申请; 否则, 联盟发起共识流程, 对该申请地址域的真实源地址验证技术部署情况达成共识. 共识通过后, 进入聚合阶段, 信任联盟通过真实地址域前缀树判断申请加入的地址域是否具备兄弟节点聚合条件. 如果能够进行兄弟节点聚合, 则将相关地址域聚合为上级地址域, 并将聚合后的上级地址域纳入信任联盟, 同时从联盟中移除这些被聚合的地址域. 反之, 若无法进行兄弟节点聚合, 则直接允许该地址域加入信任联盟. 在新地址域加入后, 信任联盟对真实地址域前缀树和前缀表进行更新. 若发现新加入地址域的下级地址域已存在于联盟中, 则将这些下级地址域从联盟中移除. 同时, 在新地址域与联盟内其他现有地址域之间建立信任关系, 确保联盟内的全局信任基础一致.

算法 2. 地址域加入验证算法.

输入: 申请加入的地址域 *IP_domain*;
输出: 地址域准入判断结果, 更新后网络信任联盟真实地址域前缀树和前缀表.

```
FOR bit in IP_domain:
    // 判断是否存在上级地址域
    IF cur_node.has_domain = True:
        approved = FALSE
    IF bit = IP_domain.end:
        cur_node.has_domain = True
        table.add(IP_domain)
        cur_node.remove(sons)
        table.remove(cur_node.sons)
        // 判断是否存在兄弟地址域
        IF IP_domain.father.ano_son.has_domain = True:
            approved = FALSE
            WHILE cur_node.father.ano_son.has_domain = True:
                combine(cur_node, cur_node.father.ano_son)
            END WHILE
        ELSE
            approved = TRUE
END FOR
```

地址域动态维护: 为保证信任联盟内信任基础的准确性和实时性, 信任联盟通过动态维护, 定期检查各地址域的真实源地址验证技术部署情况, 确保前缀树和前缀表的实时更新. 当联盟内地址域的真实源地址验证技术部署情况发生变化时, 信任联盟会发起共识, 对该地址域变化后的部署情况进行验证. 共识完成后, 根据结果更新地址域信息, 同时检查能否进行父子节点聚合和兄弟节点聚合, 并对真实地址域前缀树和前缀表进行相应调整.

地址域退出: 当地址域申请退出信任联盟时, 信任联盟内需对该退出申请发起共识. 共识完成后, 根据结果判断是否将该地址域从信任联盟中移除. 若确认移除, 则取消对该地址域的信任关系, 并更新真实地址域前缀树和前缀表. 为降低管理和维护开销, 信任联盟默认地址域退出时会同时包括其所有下级地址域, 并统一将其从联盟中移除. 若下级地址域希望继续留在信任联盟中, 则需独立向信任联盟重新发起加入请求, 进行准入验证流程.

5 节点轻量共识

为解决传统共识框架中节点身份验证开销过大的问题, 本文框架采用了基于 IP 地址身份验证的轻量共识过程. 该过程通过分步认证方式, 结合真实地址和对称密钥, 实现对节点身份的高效验证, 从而显著降低共识过程的开销. 具体而言, IP 地址身份验证分为两个步骤: 首先, 根据对称密钥判断共识信息源节点所属的地址域; 然后, 通过共识消息中记录的真实源 IP 地址确认源节点身份. 基于这一 IP 地址身份验证机制, 轻量共识过程有效提升了身份验证效率, 同时兼容传统基于非对称加密的身份验证机制.

5.1 IP 地址身份验证机制

IP 地址身份验证机制通过分步认证方式对节点身份进行验证. 对于信任联盟地址域内的共识节点, 验证过程首先基于共识消息的源 IP 地址确定该节点所属的地址域; 然后, 通过密钥维护模块中记录的 “对称密钥-地址域” 对应关系, 验证共识消息中所包含的对称密钥, 从而有效降低身份验证的计算开销. 对于信任联盟地址域外的共识节点, 则采用基于公钥密码体系的非对称加密身份验证机制, 确保身份验证的可扩展性.

具体而言, 本文提出的轻量共识框架通过算法 3 和算法 4 实现了 IP 地址身份验证机制. 在共识消息的收发过程中, 框架利用真实 IP 地址和对称密钥进行节点身份验证, 并确保消息的完整性. 在身份验证时, 算法首先根据报

文的源地址和目的地址, 判断目标共识节点是否为网络信任联盟地址域内的共识节点. 对于联盟内节点所发出的消息, 框架通过 IP 地址判断其所属地址域, 并利用地址域对应的对称密钥完成身份验证, 同时借助哈希值验证报文完整性. 而对于联盟外节点所发出的消息, 框架则采用基于公钥签名的验证方式, 通过签名确认身份, 并通过哈希校验确保报文完整性.

算法 3. IP 地址身份验证算法 (发送方).

输入: 共识消息 *message*、目的 IP 地址 *IP_des*、源 IP 地址 *IP_src*;
输出: 含有共识节点身份信息的共识消息.

```
// 判断目标节点所在地址域是否完全部署真实源地址验证技术
IF has_SAV(IP_des) == TRUE:
    // 采用 IP 地址身份验证机制, 根据目的节点所在地址域的对称密钥计算 HMAC
    domain = search_domain(IP_des)
    secret = search_domain_secret(domain)
    hmac = HMAC(message, secret)
    message = message | hmac
ELSE
    // 采用传统非对称密钥的身份验证机制, 用源节点所在地址域的公私钥对进行签名
    pri_secret = search_domain_pri_secret(IP_src)
    hash = hash(message)
    sig = sign(hash, pri_secret)
    message = message | hash | sig
RETURN message
```

算法 4. IP 地址身份验证算法 (接收方).

输入: 含有共识节点身份信息的共识消息 *message*, 源 IP 地址 *IP_src*;
输出: 验证结果.

```
IF has_SAV(IP_src) == TRUE:
    // 采用 IP 地址身份验证机制, 判断源节点所在地址域, 并根据对应对称密钥验证 HMAC
    domain = search_domain(IP_src)
    secret = search_domain_secret(domain)
    result = verify_hmac(message, secret)
ELSE
    // 采用传统非对称密钥的身份验证机制, 获取源节点对应公钥, 验证签名
    pub_secret = search_node_pub_secret(IP_src)
    result = verify_sig_hash(message, pub_secret)
RETURN result
```

5.2 轻量共识过程

基于 IP 地址身份验证, 本文提出的框架实现了轻量化共识过程. 该过程如图 4 所示, 框架在传统共识过程的基础上, 在共识节点层面采用对称密钥取代传统的非对称公私钥, 并使用 IP 地址身份验证机制取代传统的基于哈希和签名的身份验证机制. 在这一过程中, 框架使用哈希运算消息认证码 (hash-based message authentication code,

HMAC) 技术, 以支持 IP 地址身份验证. HMAC 采用对称密钥对哈希值进行加密, 能确保共识消息的完整性, 同时也满足身份验证的需求. 整个轻量化共识过程中的节点行为可以分为两类: 共识消息发送和共识消息接收.

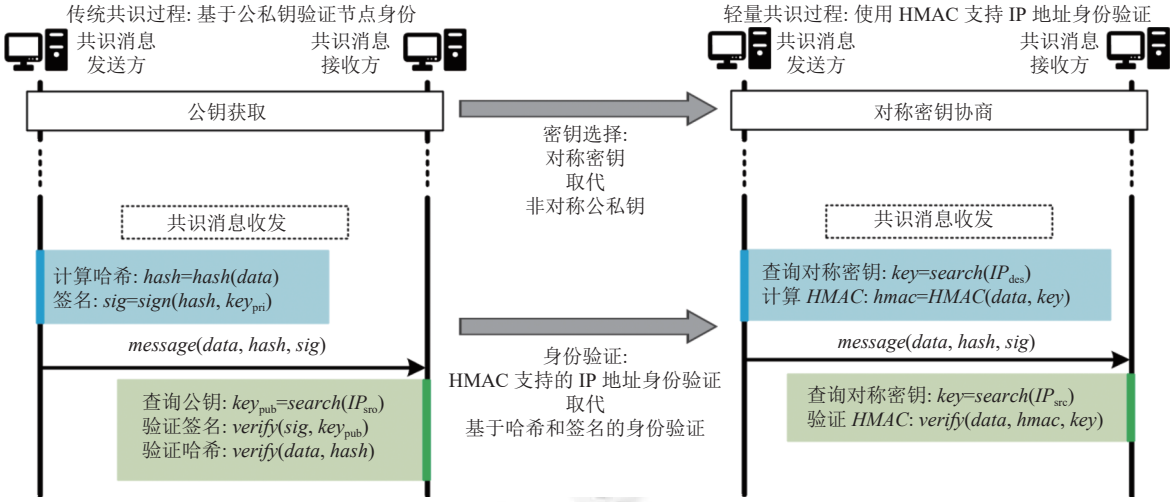

图 4 轻量共识过程中节点行为示意图

共识消息发送: 为降低发送共识消息的通信与计算开销, 在发送共识消息时, 本文框架结合 IP 地址和 HMAC, 在共识消息中附加节点身份信息. 具体而言, 发送共识消息时, 共识节点首先根据目标节点的 IP 地址, 通过密钥维护模块查看“对称密钥-地址域”对应表, 判断目标节点是否为信任联盟内可信地址域的下级地址. 如果目标节点在信任联盟内, 共识节点将使用接收方所在地址域对应的对称密钥, 对共识消息内容生成 HMAC, 并将其附加到消息中发送. 如果目标节点不在信任联盟内, 共识节点则遵循传统共识过程的身份验证流程, 对消息内容进行哈希计算和签名后发送共识消息.

共识消息接收: 在接收共识消息时, 本文框架根据共识消息中附带的 HMAC, 通过对称密钥验证源节点所在地址域, 并根据 IP 地址验证源节点身份. 具体而言, 在接收到共识消息后, 共识节点首先根据消息的源 IP 地址, 通过密钥维护模块的“对称密钥-地址域”对应表查找该地址, 判断消息源节点是否为网络信任联盟内地址域的下级地址. 如果源节点在信任联盟内, 共识节点将使用源节点所在地址域对应的对称密钥, 对共识消息中的 HMAC 进行验证; 验证通过后, 根据共识消息源地址进一步确认源节点身份. 如果源节点不在信任联盟内, 共识节点将遵循传统基于非对称加密的身份验证流程, 对消息中的签名进行验证.

为确保消息完整性, 同时降低验证开销, 本文框架在生成消息的哈希值时采用密钥关联的 HMAC. 具体而言, 在 HMAC 生成过程中, 发送方会根据密钥维护模块分发的“对称密钥-地址域”对应关系, 将收发双方所在地址域协商的对称密钥输入到 HMAC 计算过程中, 确保生成的 HMAC 含有该密钥信息, 从而支持身份验证的功能. 在 HMAC 验证过程中, 接收方会使用相同的对称密钥进行 HMAC 计算, 并对比生成的 HMAC 值与消息中附带的 HMAC 值, 以验证双方密钥的一致性及消息的完整性, 从而完成身份验证. 由于信任联盟内地址域之间的对称密钥是独立协商生成的, 任意两个地址域之间的对称密钥均唯一且互不共享. 这种“对称密钥-地址域”对应关系的唯一性能显著降低节点身份验证的复杂度, 同时确保共识消息的可靠性和安全性, 从而实现轻量化共识过程.

6 安全性分析

本节对所提出的轻量共识框架进行安全性分析. 共识机制的目的是让所有诚实的共识节点达成一致, 因而共识机制的安全性在于确保共识过程和结果的一致性和有效性. 为了保证二者的一致性和有效性, 进而确保共识过

程的安全性, 共识机制需要保证共识过程中共识节点身份验证的准确性, 以及共识消息传递的真实性和完整性, 实现对于诚实节点的共识信息的准确传递.

为衡量轻量共识框架身份验证过程的安全性, 本文对轻量共识框架的 IP 地址身份验证机制进行分析, 认为其可以保证信任联盟地址域内共识节点身份真实可信. 在轻量共识框架中, 信任联盟地址域间互相协商了独特的对称密钥, 构成了“对称密钥-地址域”一一对应关系. 因此, 共识节点可以根据其他节点的 IP 地址判断其所在地址域, 进而根据对应关系获得对称密钥. 通过验证共识消息的 HMAC 中所包含的对称密钥, 根据其独特性, 共识节点可以确认该消息地址域来源是否真实. 在源地址域真实的情况下, 共识节点根据共识消息的源 IP 地址可以确认目标节点身份, 因为在完全部署真实源地址验证技术的地址域中, 地址域内的其他节点无法假冒该节点的 IP 地址发送消息. 轻量共识框架的 IP 地址身份验证机制, 通过对称密钥验证源地址域, 通过 IP 地址验证节点身份, 保证地址域外恶意节点无法获取密钥, 地址域内恶意节点无法假冒 IP 地址, 实现了身份验证准确无误.

为衡量轻量共识框架共识消息的安全性, 本文将轻量共识框架的 IP 地址身份验证机制, 与传统非匿名共识框架的非对称加密身份验证机制进行了对比, 认为两者在共识消息的安全性上相仿. 如表 1 所示, 在保证共识消息的完整性上, 传统非匿名共识模型采用哈希进行验证并采用签名保证哈希算法已知时哈希和消息内容不会同时受到篡改, 基于真实地址的共识模型采用 HMAC 进行验证并保证哈希和消息内容不会受到篡改; 在保证共识消息的真实性上, 传统非匿名共识模型采用签名进行验证, 基于真实地址的共识模型采用真实 IP 地址进行验证. 因此, 基于真实地址的共识模型能够在共识过程中保证共识消息的真实性和完整性, 实现与传统非匿名共识相似的安全性.

表 1 共识消息安全性分析

共识机制	共识消息完整性	哈希算法已知时哈希不受篡改	共识消息真实性
传统非匿名共识机制	哈希	非对称加密	非对称加密
基于真实地址的轻量共识机制	HMAC	HMAC	真实IP地址

为了分析轻量共识框架在部分部署情况下的安全性, 本文对轻量共识框架的 IP 地址身份验证机制进行分析. IP 地址身份验证机制对基于签名进行的身份验证进行了兼容处理, 对于完全部署了真实源地址验证技术的地址域采用 IP 地址身份验证, 对于未部署的地址域采用签名验证. 因此, 共识机制实现了部分部署情况下安全性与签名验证相仿, 在部分部署情况下也能够达到与签名验证相仿的安全性.

7 原型系统实现与测试

为进一步测量基于真实地址的轻量共识框架的共识效率、共识吞吐与网络信任联盟的激活与动态更新的性能, 本文对所提出的轻量共识框架进行了原型系统实现并对其进行了测试. 测试结果表明, 本文的基于真实地址的轻量共识框架相较于常用的基于 ECDSA 和 RSA 的身份验证算法, 通过在共识过程中降低身份验证的开销, 大幅提升了共识过程的吞吐量.

7.1 原型系统实现

本文使用 Golang 1.17.5 实现了所提出的轻量共识框架的原型系统, 并通过仿真的方式对其进行了测试, 测试实验参数如表 2 所示. 测试平台为搭载 Intel E5-2650 v4 @ 2.20 GHz 的处理器、377 GB 内存的 Ubuntu 20.04 服务器, 测试中对每个共识节点采用单独的线程进行仿真. 在网络仿真环境构建方面, 本文采用全连接拓扑结构进行建模, 主要参数设置如下: (1) 节点间通信采用零丢包设置; (2) 节点间单向网络延迟设置为 27.4 ms; (3) 系统性能指标通过分布式日志采集. 该参数设置依据国内某大型运营商提供的实测数据, 其数据表明在网络正常运行情况下, 其骨干网 BGP 节点间通讯具有零丢包特征, 且端到端传输延迟均值为 27.4 ms. 测试框架分为节点仿真模块与网络信任联盟仿真模块, 其中节点仿真模块主要负责模拟节点的行为, 网络信任联盟仿真模块主要负责模拟共识过程、网络信任联盟的激活与维护过程、地址域间对称密钥的协商过程, 以及地址域内密钥分发的过程.

表 2 测试实验参数表

配置	参数
编程软件	Golang 1.17.5
测试平台	Ubuntu 20.04 服务器
处理器	Intel E5-2650 v4 @ 2.20 GHz
仿真方式	多线程并行, 线程代表共识节点
网络拓扑结构	节点间全连接
丢包率	0
网络延迟	27.4 ms
指标采集方法	日志采集

在节点仿真模块中, 节点的行为分为两部分: 节点个体行为与共识行为. 节点个体行为主要针对基于真实地址和对称密钥的 IP 地址身份验证机制的相关功能, 包括本地“对称密钥-地址域”对应关系维护和身份验证两个子功能. 节点个体行为模拟中, 本文采用基于真实地址、SHA256 的 HMAC 对信任联盟内节点进行身份验证, 采用非对称加密的方式交换、维护密钥. 共识行为主要针对网络信任联盟的相关功能, 包括发起、参与共识两个子功能, 主要负责与网络信任联盟仿真模块对接.

在网络信任联盟中, 本文在共识部分采用广泛应用的 PBFT 共识机制对网络信任联盟激活和动态更新的共识过程进行模拟. 共识过程按照 PBFT 共识机制执行, 共识完成后, 采用向所有节点广播同步的方式保证各节点所记录的共识结果一致, 所维护的网络信任联盟信息一致. 对于共识过程的相关指标, 测试过程中通过专门实现的函数从各线程所模拟的共识节点和发起共识请求的用户端中拉取日志进行分析, 共识完成后根据日志内容计算后得到具体测试结果.

7.2 共识效率测试

本文首先测试了在网络信任联盟正常运行的情况下, 部署真实源地址验证技术的地址域内各节点进行共识过程的共识效率. 本文将所提出的基于真实地址的轻量共识框架与在目前使用最为广泛的区块链应用 Bitcoin、Ethereum 中所使用的身份验证机制: 基于 ECDSA 算法的身份验证机制进行了对比, 同时与常用于身份验证的基于 RSA 算法的身份验证机制也进行了对比, 将分别采用这 3 种不同的身份验证机制的情况与不进行身份验证的情况在 PBFT 共识框架下进行共识过程不同负载情况的测试. 测试内容为轻载和平均负载环境下共识过程的计算开销和共识延迟, 计算开销指不考虑网络延迟的情况下完成单个共识所需的平均用时, 共识延迟指完成单个共识所需的平均用时. 测试过程取共识节点数为 4、7、10、13、16、25、32 个节点, 测试结果如图 5、图 6 所示.

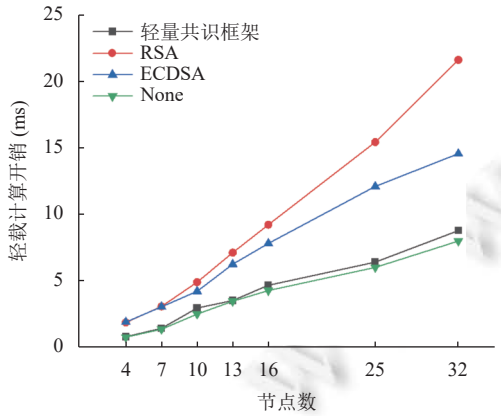

图 5 轻载计算开销测试结果

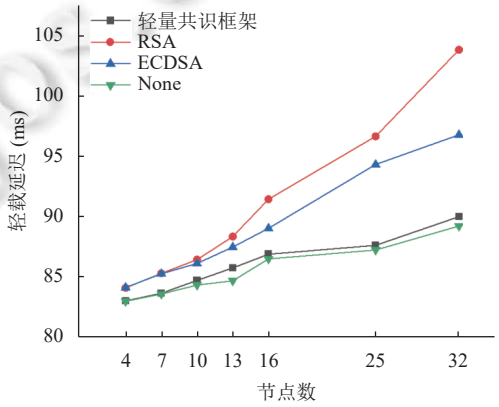

图 6 轻载延迟测试结果

如图 5、图 6 所示, 基于真实地址的轻量共识框架在共识负载较低时降低了共识过程的计算时间开销和整体时间开销, 实现了共识过程的轻量化. 从图 5、图 6 中对于采用不同身份验证算法的共识计算开销和轻载延迟测试结果可以看出, 相较于现在广泛应用的基于 ECDSA 算法的身份验证机制的共识框架, 基于 IP 地址身份验证机制的轻量共识框架在轻载计算开销方面可以实现平均 44.4% 的计算开销降低, 在轻载延迟方面可以实现最大 7.1% 的延迟降低; 相较于基于 RSA 的身份验证机制的共识框架, 轻量共识框架实现了平均 52.5% 的计算开销降低和最大 12.3% 的轻载延迟降低; 即使是相较于不进行身份验证直接进行共识的情况, 轻量共识框架也只增加了平均 7.6% 的额外计算开销和最大 2% 的额外共识开销.

具体而言, 在共识节点数为 7 时, 不进行身份验证操作直接处理发送消息进行共识的情况下, 单节点的轻载计算开销为 0.001 38 s/共识, 本文的轻量共识框架的轻载计算开销为 0.001 44 s/共识, 仅增加了 0.000 06 s, 约 4.3% 的额外计算开销; 采用 ECDSA 算法作为身份验证机制的共识框架的轻载计算开销为 0.003 06 s/共识, 相较于不进行身份验证增加了 0.001 68 s, 约 121% 的额外计算开销; 而采用 RSA 算法作为身份验证机制的共识框架的轻载计算开销为 0.003 09 s/共识, 相较于不进行身份验证增加了 0.001 71 s, 约 124% 的额外计算开销.

如图 7、图 8 所示, 基于真实地址的轻量共识框架在进行一定数量的共识时降低了共识过程的计算时间开销和整体时间开销, 实现了共识过程的轻量化. 从图 7、图 8 中对于采用不同身份验证算法的共识平均负载计算开销和平均负载延迟测试结果可以看出, 基于 IP 地址身份验证机制的轻量共识框架相较于现在广泛应用的基于 ECDSA 算法的身份验证机制的共识框架可以实现平均 40.2% 的计算开销降低和最大 6.9% 的延迟降低; 而相较于基于 RSA 的身份验证机制的共识框架, 轻量共识框架实现了平均 52.6% 的计算开销降低和最大 16.6% 的延迟降低; 即使是相较于不进行身份验证直接进行共识的情况, 轻量共识框架也只增加了平均 4.7% 的额外计算开销和最大 1.7% 的延迟增加.

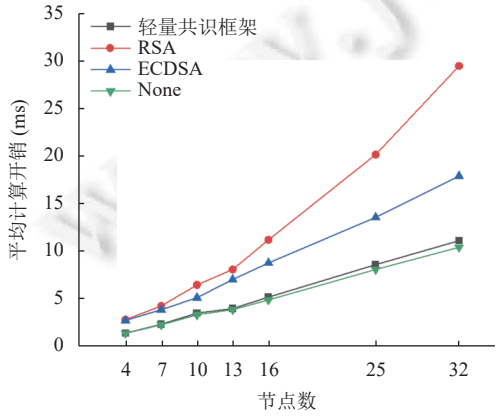

图 7 平均负载计算开销测试结果

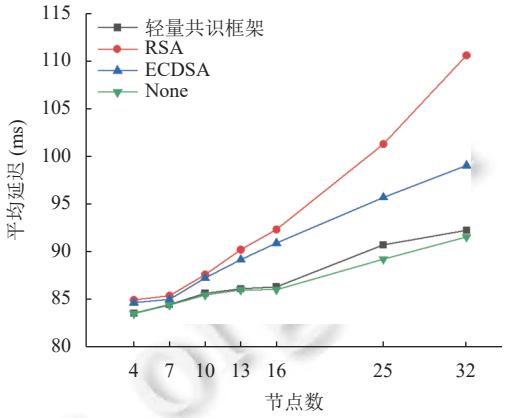

图 8 平均负载延迟测试结果

具体而言, 在共识节点数为 10 时, 不进行身份验证操作直接处理发送消息进行共识的情况下, 共识平均负载计算开销为 0.003 27 s/共识, 本文的轻量共识框架的共识平均负载计算开销为 0.003 45 s/共识, 仅增加了 0.000 18 s 的额外计算开销, 相较于不进行身份验证增加了 5.5% 的额外计算开销; 采用 ECDSA 算法作为身份验证机制的共识框架的共识平均负载计算开销为 0.005 07 s/共识, 相较于不进行身份验证增加了 0.001 80 s, 约 55.0% 的额外计算开销; 而采用 RSA 算法作为身份验证机制的共识框架的共识平均负载计算开销为 0.006 42 s/共识, 相较于不进行身份验证增加了 0.003 15 s, 约 96.3% 的额外计算开销.

7.3 共识吞吐测试

本文同时对网络信任联盟已完成部署并正常运行的情况下, 部署了真实源地址验证技术的地址域内各节点进行共识过程的吞吐量测试实验. 本文将基于真实地址的轻量共识框架所采用的基于真实地址、SHA256 和 HMAC

的身份验证算法, 与在 Bitcoin、Ethereum 中所使用的基于 SECP256k1 曲线的 ECDSA 身份验证算法和 RSA 身份验证算法进行对比实验. 本文将这 3 种不同的身份验证算法在 PBFT 共识框架下进行了共识吞吐量测试, 同时测量了不采用身份验证算法的情况下共识过程的吞吐量, 吞吐量指单位时间内完成的共识数. 测试过程取共识节点数为 4、7、10、13、16、25、32 个节点, 最终测试结果如图 9 所示.

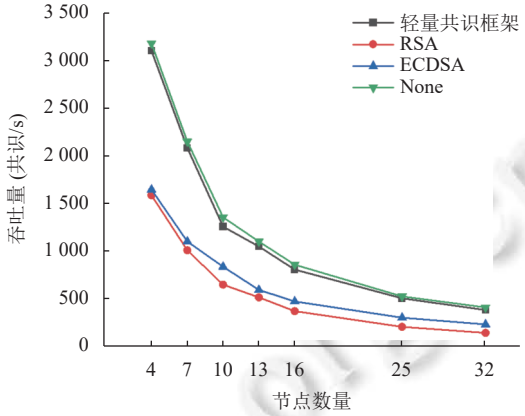

图 9 共识吞吐量测试结果

如图 9 所示, 基于真实地址的轻量共识框架降低了共识的整体开销, 提升了共识吞吐量, 实现了共识过程的轻量化. 从图 9 中的测试结果可以看出, 在共识吞吐量方面, 相较于基于 SECP256k1 曲线的 ECDSA 身份验证算法的共识框架, 基于 IP 地址身份验证机制的轻量共识框架实现了 72.2% 的共识吞吐量的提升; 相较于基于 RSA 的身份验证算法的共识框架, 轻量共识框架实现了 117.7% 的共识吞吐量的提升; 而相较于不进行身份验证的 PBFT 共识框架, 轻量共识框架也仅降低了 4.9% 的共识吞吐量.

具体而言, 在共识节点数为 13 时, 不进行身份验证操作直接处理发送消息进行共识的情况下, 共识吞吐量为 1 102 共识/s, 本文的轻量共识框架的共识吞吐量为 1 050 共识/s, 相较于不进行身份验证降低了 52 共识/s, 下降比例为 4.7%; 采用 ECDSA 算法作为身份验证机制的共识框架的共识吞吐量为 595 共识/s, 相较于不进行身份验证降低了 507 共识/s, 下降比例约 46.0%; 而采用 RSA 算法作为身份验证机制的共识框架的共识吞吐量为 517 共识/s, 相较于不进行身份验证降低了 585 共识/s, 下降比例约 53.1%.

8 总 结

为解决现有共识机制开销过大、难以满足网络层安全需求的问题, 本文提出并实现了基于真实地址的轻量共识框架. 该框架为解决密钥维护开销大的问题, 提出利用真实地址作为身份识别标志, 通过域内密钥共享实现域内密钥聚合降低密钥维护开销, 实现地址域内密钥维护轻量化. 为解决信任关系传递不灵活的问题, 同时进一步降低密钥维护开销, 该框架采用信任联盟维护信任基础, 采用前缀树根据前缀关系聚合地址域, 实现灵活的信任传递和地址域间聚合. 为解决共识过程中节点身份验证开销过大的问题, 该框架提出基于 IP 地址身份验证机制的轻量共识机制, 通过采用分步认证的方式, 基于真实地址和地址域进行身份验证, 降低共识开销, 实现节点共识过程轻量化. 本文实现了基于真实地址的轻量共识框架原型系统, 并与传统的基于 SECP256k1 的 ECDSA 身份验证算法的共识框架和基于其他身份验证算法的共识框架进行了对比, 结果显示本文的框架降低了 40% 的共识计算开销, 增加了 70% 的共识吞吐量, 实现了共识过程的轻量化和高吞吐.

References

[1] Paillisse J, Manrique J, Bonet G, Rodriguez-Natal A, Maino F, Cabellos A. Decentralized trust in the inter-domain routing infrastructure. IEEE Access, 2019, 7: 166896–166905. [doi: 10.1109/ACCESS.2019.2954096]

[2] King S, Nadal S. PPCoin: Peer-to-peer crypto-currency with proof-of-stake. 2012. https://people.cs.georgetown.edu/~clay/classes/fall2017/835/papers/peercoin-paper.pdf

[3] Kaur M, Khan MZ, Gupta S, Noorwali A, Chakraborty C, Pani SK. MBCP: Performance analysis of large scale mainstream blockchain consensus protocols. IEEE Access, 2021, 9: 80931–80944. [doi: 10.1109/ACCESS.2021.3085187]

[4] Zhang L, Li QW. Research on consensus efficiency based on practical Byzantine fault tolerance. In: Proc. of the 10th Int'l Conf. on Modelling, Identification and Control (ICMIC). Guiyang: IEEE, 2018. 1–6. [doi: 10.1109/ICMIC.2018.8529940]

[5] Fu X, Wang HM, Shi PC, Ouyang X, Zhang XH. Jointgraph: A DAG-based efficient consensus algorithm for consortium blockchains. Software: Practice and Experience, 2021, 51(10): 1987–1999. [doi: 10.1002/spe.2748]

[6] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008. https://nakamotoinstitute.org/library/bitcoin/

[7] Miller A, Xia Y, Croman K, Shi E, Song D. The honey badger of BFT protocols. In: Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security. Vienna: ACM, 2016. 31–42. [doi: 10.1145/2976749.2978399]

[8] Pass R, Shi E. Hybrid consensus: Efficient consensus in the permissionless model. 2016. https://eprint.iacr.org/2016/917.pdf

[9] Zhang HJ, Dong YC, Chiclana F, Yu S. Consensus efficiency in group decision making: A comprehensive comparative study and its optimal design. European Journal of Operational Research, 2019, 275(2): 580–598. [doi: 10.1016/j.ejor.2018.11.052]

[10] Wu J, Bi J, Li X, Ren G, Xu K, Williams M. RFC 5210: A source address validation architecture (SAVA) testbed and deployment experience. 2008. https://www.rfc-editor.org/info/rfc5210 [doi: 10.17487/RFC5210]

[11] Li D, Qin LC, Wu JP, Su YY, Xu MW, Shi XG, Gu YN, Lin T. Internet source address verification method based on synchronization and dynamic filtering in address domain. Telecommunications Science, 2020, 36(10): 21–28 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-0801.2020289]

[12] de la Rocha Gómez-Arevalillo A, Papadimitratos P. Blockchain-based public key infrastructure for inter-domain secure routing. In: Proc. of the 2017 Int'l Workshop on Open Problems in Network Security (iNetSec). 2017. 20–38.

[13] Xing QQ, Wang BS, Wang XF. BGPcoin: Blockchain-based Internet number resource authority and BGP security solution. Symmetry, 2018, 10(9): 408. [doi: 10.3390/sym10090408]

[14] He GB, Su W, Gao S, Yue JR, Das SK. ROAchain: Securing route origin authorization with blockchain for inter-domain routing. IEEE Trans. on Network and Service Management, 2021, 18(2): 1690–1705. [doi: 10.1109/TNSM.2020.3015557]

[15] Li J, Cao JH, Yang Y, Liu ZT, Li Q, Wang YY, Meng ZL, Xie RJ, Xu MW. DeBGP: Decentralized and efficient BGP hijacking prevention system. In: Proc. of the 2023 Int'l Conf. on Computer Communications and Networks (ICCCN). 2023. 24–26.

[16] Saad M, Anwar A, Ahmad A, Alasmary H, Yuksel M, Mohaisen D. RouteChain: Towards blockchain-based secure and efficient BGP routing. Computer Networks, 2022, 217: 109362. [doi: 10.1016/j.comnet.2022.109362]

[17] Lamport L. Paxos made simple. 2001. https://lamport.azurewebsites.net/pubs/paxos-simple.pdf

[18] Lamport L. Generalized consensus and Paxos. Technical Report, MSR-TR-2005-33, Microsoft, 2005. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-33.pdf?msockid=18e24034efce61972d8d50a1ebce67e1

[19] Lamport L. Fast paxos. Distributed Computing, 2006, 19(2): 79–103. [doi: 10.1007/s00446-006-0005-x]

[20] Ongaro D, Ousterhout J. In search of an understandable consensus algorithm. In: Proc. of the 2014 USENIX Annual Technical Conf. Philadelphia: USENIX Association, 2014. 305–319.

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

[22] Li Q, Xu MW, Wu JP, Zhang XW, Lee PPC, Xu K. Enhancing the trust of Internet routing with lightweight route attestation. In: Proc. of the 6th ACM Symp. on Information, Computer and Communications Security. Hong Kong: ACM, 2011. 92–101. [doi: 10.1145/1966913.1966927]

[23] Li Q, Hu YC, Zhang XW. Even rockets cannot make pigs fly sustainably: Can BGP be secured with BGPsec. In: Proc. of the 2014 Workshop on Security of Emerging Networking Technologies. San Diego, 2014.

[24] Cao JM, Liu Y, Liu MX, He L, Jia YH, Yang F. pSAV: A practical and decentralized inter-as source address validation service framework. In: Proc. of the 29th IEEE/ACM Int'l Symp. on Quality of Service (IWQOS). Tokyo: IEEE, 2021. 1–7. [doi: 10.1109/IWQOS52092.2021.9521336]

[25] Yang XY, Cao JH, Xu MW. SEC: Secure, efficient, and compatible source address validation with packet tags. In: Proc. of the 39th IEEE Int'l Performance Computing and Communications Conf. (IPCCC). Austin: IEEE, 2020. 1–8. [doi: 10.1109/IPCCC50635.2020.9391554]

[26] Qin LC, Liu LB, Chen L, Li D, Shi YQ, Yang HB. UniSAV: A unified framework for Internet-scale source address validation. In: Proc. of the 2024 Applied Networking Research Workshop. Vancouver: ACM, 2024. 81–87. [doi: 10.1145/3673422.3674888]

[27] Wu J, Bi J, Bagnulo M, Baker F, Vogt C. RFC 7039: Source address validation improvement (SAVI) framework. 2013. https://www.rfc-

editor.org/info/rfc7039 [doi: 10.17487/RFC7039]

[28] Bi J, Wu J, Yao G, Baker F. RFC 7513: Source address validation improvement (SAVI) solution for DHCP. 2015. https://www.rfc-editor.org/info/rfc7513 [doi: 10.17487/RFC7513]

[29] Bi J, Yao G, Halpern J, Levy-Abegnoli E. RFC 8074: Source address validation improvement (SAVI) for mixed address assignment methods scenario. 2017. https://www.rfc-editor.org/info/rfc8074 [doi: 10.17487/RFC8074]

附中文参考文献

[11] 李丹, 秦澜城, 吴建平, 苏莹莹, 徐明伟, 施新钢, 顾钰楠, 林涛. 基于边界路由动态同步的互联网地址域内真实源地址验证方法. 电信科学, 2020, 36(10): 21–28. [doi: 10.11959/j.issn.1000-0801.2020289]

作者简介

徐易, 博士生, 主要研究领域为互联网路由安全.

陈熠豪, 博士生, 主要研究领域为互联网路由安全.

王晓亮, 博士, 讲师, 主要研究领域为移动计算, 互联网安全, 网络体系结构.

徐恪, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为互联网体系架构, 网络安全.

李琦, 博士, 副教授, 博士生导师, CCF 高级会员, 主要研究领域为互联网安全, 人工智能安全, 物联网安全.