一种基于区块链的 DNSSEC 公钥验证机制

陈闻宇 [1, 2, 3] 李晓东 [1] 杨 学 [3] 徐彦之 [4]

摘 要 针对中心化域名安全扩展 (Domain name system security extensions, DNSSEC) 架构所导致的信任链复杂性和单边控制模式, 提出了一种去中心化的 DNSSEC 公钥验证机制. 该机制结合区块链结构、密码学累加器和共识算法设计, 创新性地实现使用区块链技术的密钥绑定、轮转和验证操作, 无需中心化权威节点即可使用可信公钥验证域名记录. 进一步分析和实验表明, 所提出的机制在保证密钥管理安全性的同时, 提高了密钥验证的效率.

关键词 域名安全扩展, 公钥基础设施, 区块链, 密码学累加器

引用格式 陈闻宇, 李晓东, 杨学, 徐彦之. 一种基于区块链的 DNSSEC 公钥验证机制. 自动化学报, 2023, **49**(4): 731−743

DOI 10.16383/j.aas.c201082

A Blockchain-based DNSSEC Public Key Verification Scheme

CHEN Wen-Yu[1, 2, 3] LI Xiao-Dong[1] YANG Xue[3] XU Yan-Zhi[4]

Abstract To solve the problem of the complexity of chain-of-trust and the unilateral governance caused by the centralized domain name system security extensions (DNSSEC) architecture, a decentralized DNSSEC public key verification mechanism is proposed. By introducing blockchain structure design, cryptographic accumulator, and consensus algorithm, the proposed mechanism gives radical new key binding, rotation, and verification operations leveraging blockchain technologies enables the use of trustful public key verification without any centralized authorities. Further analysis and experiments show that the proposed mechanism consistently perform the order of magnitude better key verification performance, as well as achieve a good trade-off between key management complexity and security.

Key words Domain name system security extensions (DNSSEC), public key infrastructure (PKI), blockchain, cryptographic accumulator

Citation Chen Wen-Yu, Li Xiao-Dong, Yang Xue, Xu Yan-Zhi. A blockchain-based DNSSEC public key verification scheme. *Acta Automatica Sinica*, 2023, **49**(4): 731−743

域名系统 (Domain name system, DNS) 主要提供域名解析服务, 是互联网关键基础服务之一, 其安全问题受到广泛关注. 目前 DNS 服务遵循 RFC 1034[1] 和 RFC 1035 规范[2], 依赖于中心化层次架构, 缺乏记录数据的校验机制. 如何从验证机制上改进 DNS 安全性, 一直是学术界和工业界的热点问题. 其中, DNS 安全扩展 (Domain name system security extensions, DNSSEC) 方案受到了广泛关注[3].

收稿日期 2020-12-29 录用日期 2021-04-21
Manuscript received December 29, 2020; accepted April 21, 2021
国家重点研发计划专项基金 (2019YFB1804500) 资助
Supported by National Key Research and Development Program of China (2019YFB1804500)
本文责任编委 袁勇
Recommended by Associate Editor YUAN Yong
1. 中国科学院计算技术研究所 北京 100190 2. 中国科学院大学 北京 100049 3. 中国互联网络信息中心 北京 100190 4. 广东粤港澳大湾区国家纳米科技创新研究院 广州 510770
1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190 2. University of Chinese Academy of Sciences, Beijing 100049 3. China Internet Network Information Center, Beijing 100190 4. Guangdong-Hong Kong-Macao Greater Bay Area (GBA) Research Innovation Institute for Nanotechnology, Guangzhou 510770

DNSSEC 通过额外引入验证密钥, 提供数据来源和完整性验证机制[4]. 其密钥验证机制在设计上采用类似中心化的公钥基础设施 (Public key infrastructure, PKI), 支持层次化注册域名和公钥绑定. 为确保记录中公钥的可信性, 权威域名服务器必须正确签发其密钥和签名, 路径上每个 DNSSEC 子域必须被其父域的密钥正确签名, 同时域名解析器必须具备信任链的验证功能. 在这种中心化域名体系下, 公钥验证依赖于逐层上达信任锚的信任链维护, 导致验证过程复杂且不易管理[5]. 另一方面, 作为解析器信任公钥的信任锚被根区管理者掌控, 形成受制于单边控制的互联网治理模式. 信任链管理的复杂性及其衍生问题, 严重影响 DNSSEC 的广泛应用. 从最新统计看, 只有近 4% 的二级域名部署了 DNSSEC[6]. 2017 年的研究[7] 指出, 已部署 DNSSEC 的域名服务中, 31% 的域名未能发布验证所需的所有相关记录, 39% 的域名使用了低强度签名密钥, 只有 12% 请求 DNSSEC 记录的域名解析器实际执行了验证操作.

密钥验证的中心化机制, 是导致 DNSSEC 难以得到有效部署的重要原因之一. 如何实现去中心化验证, 是近期的一个研究热点问题. 区块链技术基于去中心化思想和共识机制, 支持每个节点独立完成信任验证, 为实现去中心化信任提供新的思路[8]. 基于区块链技术实现去中心化验证, 目前研究方法主要可分为两类.

一类方法是使用区块链直接管理域名记录, 通过共识机制确保每个节点具有可信的域名记录, 意在解决中心化架构下域名记录交换所导致的安全风险[9−13]. 但所构建的域名系统与现有域名不兼容, 同时需额外考虑因共识所导致的分叉、攻击和低效等问题. 这导致此类研究所形成系统难以真正得到部署.

另一类方法是基于区块链实现去中心化的 PKI[14−16], 进而考虑为 DNSSEC 提供支持[17−18]. DNSSEC 在设计上可视为一种中心化的树状 PKI, 但 DNSSEC 在密钥管理上具有其独特设计. 首先, DNSSEC 证书格式与 PKI 使用的 X.509 标准存在差异, DNSSEC 存在多种公钥记录类型. 其次, 在密钥管理上, DNSSEC 定义了两种类型公钥分离的设计, 并且密钥和数字签名以特定的资源记录分别管理. 不同于 PKI 中只有一个证书验证机构 (Certification authority, CA) 签发证书, DNSSEC 中每个域名所有者都可作为 CA, 为其子域签发证书.

去中心化 PKI 需要针对 DNSSEC 密钥管理特点进行重新设计, 但是目前尚未有研究考虑基于区块链形成支持 DNSSEC 公钥注册、验证和轮转的整体机制. 由此, 本文提出一种去中心化的 DNSSEC 公钥验证机制. 一方面, 该机制结合区块链结构、密码学累加器和共识算法设计, 创新性地实现使用区块链技术的密钥绑定、轮转和验证操作, 无需中心化权威节点即可使用可信公钥验证域名记录. 避免了现有体系下服务器端密钥信任链管理复杂的问题. 另一方面, 为改进区块链本身的性能, 本文对支持 DNSSEC 密钥管理的区块链做了全新设计. 通过定义区块结构和区分节点类型, 避免全链同步所有数据的代价; 基于实用拜占庭容错 (Practical Byzantine fault tolerance, PBFT) 共识算法提出了两种分组改进策略, 在不降低安全性的同时改进了共识性能; 通过选择密码学累加器, 避免遍历全部区块验证密钥.

综上, 本文的主要创新为:

1) 针对现有的中心化 DNSSEC 体系进行改进, 创新性地实现了一种基于区块链的去中心化可信公钥验证机制, 在提供去中心化的 PKI 密钥注册、轮转和验证操作的同时, 降低了维护 DNSSEC 信任链的复杂性.

2) 针对 DNSSEC 密钥管理操作特性, 改进了共识机制和区块链结构的设计. 在共识上提出了采用聚类和 DNS 结构两种 PBFT 共识算法分组策略, 在结构上提出通过区分节点类型避免全链同步所有数据的代价, 采用密码学累加器无需遍历区块链即可验证密钥绑定. 所设计的共识和结构可保证密钥管理安全性的同时, 提高密钥验证的效率.

3) 通过进一步的分析和实验, 验证了本文机制, 有效地降低了密钥管理的复杂性. 相比现有区块链和 DNSSEC 实现, 所提出的机制在密钥验证性能表现出一致性的改进, 在验证性能和安全可靠性上取得了很好的权衡.

本文结构如下: 第 1 节为对比相关研究; 第 2 节给出基本概念; 第 3 节进行详细阐释; 第 4 节介绍共识算法的改进; 第 5 节给出实验的对比结果; 最后第 6 节对全文进行归纳总结.

1 相关研究

有别于当前“根−权威−递归”的中心化域名服务系统, 去中心化域名服务由于在安全性和可扩展性上的优点而受到关注. 目前在研究领域存在两类平行工作, 如表 1 所示.

一类是使用区块链管理 DNS 域名数据[9−13]. Hari 等[9] 的研究是使用区块链验证 DNS 域名记录的早期工作之一. 该研究提出使用区块链而非 PKI 实现对 DNS 记录的验证. 无需通过第三方实体验证, 由区块链提供对等节点间仅通过公共密钥验证的机制, 避免了第三方篡改记录和密钥的可能性. 基于此理念, 域名币 (Namecoin)[10] 和区块堆栈 (Blockstack)[11] 提出并实现了基于区块链的 DNS 系统, 为设计去中心化 DNS 提供了新思路, 但此类系统独立并不兼容于传统的 DNS, 难以得到大范围部署. 进一步, Liu 等[12] 提出了基于区块链的去中心化 DNS 解析方法, 解决单点故障和域名解析数据篡改问题. Wang 等[13] 提出了使用改进的 PBFT 共识机制实现的节点复制, 提供高效命名解析. 这些工作虽然实现与本文相同的目标, 即提供去中心化可验证的域名记录, 但是技术途径上采用区块链管理域名记录, 而非管理密钥本身.

另一类平行工作提出使用区块链实现去中心化 PKI 系统[14−18]. 在 DNSSEC 标准中, 也采取了类 PKI 机制传递 DNS 服务器公钥. 一些近期相关工作考虑了基于区块链的 DNSSEC 验证, 与本文最

表 1　近期相关研究对比

Table 1　Comparison of most recent related works

研究	研究对象	提出时间	基本方法	针对问题
Hari 等[9]	DNS	2016 年	首次提出使用区块链而非 PKI 验证 DNS 记录	功能性
Namecoin[10]	DNS	2011 年	首个基于区块链的开源 DNS 系统	功能性
Blockstack[11]	DNS	2016 年	提出将域名数据和控制分层的方案, 通过外部存储降低区块链管理域名记录的复杂性	功能性
Liu 等[12]	DNS	2018 年	进一步提出使用去中心化文件管理实现区块链外部存储 DNS 记录	功能性
Blockzone[13]	DNS	2019 年	整体提出了一种基于 PBFT 的 DNS 记录管理机制	安全性
IKP[14]	PKI	2017 年	使用区块链改进 PKI/CA 对异常操作的处理	安全性
CertLedger[15]	PKI	2019 年	引入区块链提高 PKI 的安全性	安全性
Wang 等[16]	PKI	2022 年	设计区块链交易实现 CA 验证功能	功能性
Gourley 等[17]	DNSSEC	2018 年	提出使用特定区块链网络存储 X.509 格式 DNSSEC 证书	安全性
AuthLedger[18]	DNSSEC	2019 年	提出一种使用区块链实现 PKI 签名验证的设计	功能性

相关的近期工作是文献 [17−18]. Gourley 等[17] 采用类似思路, 通过使用 X509 Cloud 区块链网络存储 X.509 证书, 简化了 DNSSEC 密钥管理, 降低 DNSSEC 响应的大小, 进而加速了响应验证. AuthLedger[18] 建议使用区块链改进签名验证过程, 实现 DNSSEC 相应功能. 上述研究并未充分考虑每个节点同步全部区块的实现代价, 对于区块链系统的大规模部署是需要考虑的问题.

从密钥验证的功能实现看, 近期一些研究针对 DNSSEC 服务的独特需求进行了探索性研究. 文献 [5] 从可伸缩性、灵活性、部署和安全等多个方面讨论了现有 DNSSEC 标准中的密钥服务. 引入区块链的去中心化密钥验证同样面对这些方面的问题. 对此, 文献 [19] 考虑了采用密码学累加器的固定大小数据结构, 降低去中心化 PKI 实现中维护智能合约状态的代价. 本文研究采用类似思想, 但通过选择累加器算法并设计操作机制, 简化了累加器见证的本地验证, 更适用于区块链轻节点的操作. 文献 [20] 针对 DNSSEC 密钥轮转问题, 提出采用状态机定义状态自动转移实现自动密钥轮转. 本文工作的差别之处在于结合已定义的区块链操作实现密钥轮转. 文献 [21] 综述了区块链共识最新进展, 文献 [22] 分类研究了 PBFT 改进算法, 本文基于此提出两种新的分组策略. 综上, 目前鲜见研究考虑基于区块链整体支持 DNSSEC 公钥操作的机制.

2　相关概念

2.1　PKI

PKI 是创建、存储、管理、撤销和分发数字证书的基础结构. 数字证书实现“实体−公钥”对的绑定, 用于验证特定公钥是否属于某个实体. 经过多年研究, PKI 技术已进入大规模应用阶段, 为全球信息系统提供了重要的安全支撑作用.

PKI 系统应用分为密钥注册和验证两部分. 在当前普遍采用的中心化模式下, 数字证书由 CA 作为可信第三方签发, CA 将证书的接收、拒收和撤销授权给注册机构 (Registration authority, RA). 实体获取 CA 根证书并生成自身的密钥对, 进而创建包含身份和公钥的 CSR (Certificate signing request) 请求, 发送给 RA. 经 RA 审核的 CSR 被 CA 接收, 由 CA 签发并向实体返回包含数字签名、公钥、身份、有效期等信息证书. 密钥操作形式化定义如下:

定义 1. 密钥操作 $KO = \langle Gen, Sign, Veri\rangle$ 是由以下三项多项式时间复杂度操作组成的三元组:

1) 生成操作 $Gen\,(H, U) \to (pk_u, sk_u)$ 输入选定的哈希函数族 H 执行随机运算, 生成节点 U 的密钥对, 即公钥 pk_u 和私钥 sk_u.

2) 签名操作 $Sign\,(sk_u, R) \to \psi$ 使用私钥 sk_u 对记录 R 签名, 输出数字签名 ψ.

3) 验证操作 $Veri\,(pk_u, R, \psi) \to$ T/F 使用公钥 pk_u 判断数字签名 ψ 是否是记录 R 的有效签名, 输出真 (True, T) 或假 (False, F).

PKI 用户通过“实体−公钥”对验证通信对端实体的正确性, 这需要使用可信的实体公钥. 对此, PKI 构建了称为“信任链”的加密密钥序列. 当前证书中的签名是使用上一级中介的私钥签发的, 依次直至根 CA. 根证书的签名是用根 CA 私钥自签名的, 验证根证书签名使用根 CA 公钥. PKI 依赖方利用预先存储的根证书签名沿信任链逐一验证, 可信地获取通信对端实体的公钥, 进而用于机密性、数据完整性、身份鉴别等各种安全功能.

2.2　DNSSEC

尽管在互联网中 DNS 承担了域名解析这样的

关键服务, 但其本身并不支持加密操作. 域名查询请求和响应报文在网络传输过程中, 存在被中间人攻击和缓存投毒等篡改的风险[7]. 对此, 负责制定 DNS 协议标准的互联网工程任务组 (Internet engineering task force, IETF) 提出了 DNSSEC, 目的是为 DNS 引入更强的验证机制. DNSSEC 并非对 DNS 查询和响应本身进行加密签名, 而是在响应中添加了区域数据签名的散列值.

DNSSEC 通过额外引入新的资源记录, 提供类似 PKI/CA 的层次化公钥验证机制. DNS 公钥 (DNS public key, DNSKEY) 记录指定区域的数字签名公钥, 资源记录签名 (Resource record signature, RRSIG) 记录给出对区域数据的数字签名, 授权签名 (Delegation signer, DS) 记录提供构建信任链的密钥标签、区域密钥摘要及签名和摘要算法等信息. 其中, DNSKEY 提供两种类型公钥: 区域签名密钥 (Zone signing key, ZSK) 用于生成和验证域名记录, 密钥签名密钥 (Key signing key, KSK) 用于签名验证 ZSK.

在 DNS 递归服务器端, 使用哈希函数对查询的 DNS 资源记录集 (Resource record set, RRSet) 计算信息摘要, 并用自身的私钥签名信息摘要得到 RRSIG 数字签名, 然后将 RRSIG、公钥 DNSKEY 和 RRSet 记录一起发送至查询客户端. 客户利用 DNSKEY 对 RRSIG 解密得到摘要, 之后使用与签名方相同的哈希函数计算 RRSet 的摘要, 并将两者摘要进行对比, 如一致则可确认接收到的资源记录并没有被篡改.

为确保公钥本身不被篡改, DNSSEC 同样需要维护类似 PKI 的信任链. 区域公钥本身由其父区域私钥签名, 构成从根区域到当前区域的信任链. 不同于 PKI, DNSSEC 的信任链对应于 DNS 域名层次结构关系. 其中, 根域名 DNS 服务器作为系统的根 CA, 存储顶级域名的签名信息, 顶级域名服务器存储二级域名的签名信息. 在中心化的信任链机制中, 只要域名验证路径上的任何一个节点尚未部署 DNSSEC, 那么域名树的连通性就无法保证, 子树的所有叶子节点就无法使用 DNSSEC 验证. 这意味着客户端需保存多个 DNSSEC 公钥信息, 形成多个信任锚点, 由此导致 DNSSEC 部署难度增大, 可能故障点增多.

2.3 密码学累加器

密码学累加器提供了一种将集合数据表示为固定大小累加值的算法, 并提供一个称为“存在性见证 (Witness)”的小规模数据结构, 验证候选元素在集合中的存在性. 对于规模为 n 的集合, 密码学累加器将 $\mathrm{O}(n)$ 复杂度的存在性操作转为 $\mathrm{O}(1)$ 复杂度[23]. 密码学累加器操作形式化定义如下:

定义 2. 密码学累加器操作定义为五元组 $CAO = \langle caSetup, caAdd, caUpdate, caDel, caBelongs\rangle$, 由下列五项多项式时间操作组成:

1) $caSetup(H) \rightarrow S_0$: 初始化操作, 输入选定的哈希函数族 H 作为参数, 生成表示空集合 S_0 的密码学累加器数据结构.

2) $caAdd(S_i, R_{i+1}) \rightarrow \{S_{i+1}, W_{i+1}^{R_{i+1}}\}$: 集合添加操作, 输入当前密码学累加器状态 S_i、需添加到集合中的新纪录 R_{i+1}, 输出新的密码学累加器状态 S_{i+1}, 以及 R_{i+1} 的存在性证明 $W_{i+1}^{R_{i+1}}$.

3) $caUpdate(W_i^{R_i}, R_{i+1}) \rightarrow W_i'R_i$: 存在性证明更新操作, 输入当前存在性证明 $W_i^{R_i}$、集合中的新纪录 R_{i+1}, 输出为记录 R_i 更新后的存在性证明 $W_i'R_i$.

4) $caDel(S_i, R_i) \rightarrow \{S_{i+1}, W_{i+1}^{R_i}\}$: 集合删除操作, 从当前密码学累加器 S_i 删除当前记录 R_i, 生成新的当前密码学累加器, 以及新的存在性证明 $W_{i+1}^{R_i}$.

5) $caBelongs(S_i, W_i^{R_i}, R_i) \rightarrow \mathrm{T/F}$: 验证操作, 使用密码学累加器 S_i 和当前存在性证明 $W_i^{R_i}$, 验证当前记录 R_i 是否属于集合, 输出真 (T) 或假 (F).

3 基于区块链的 DNSSEC 公钥验证

本文基于区块链的可信记录管理, 提出了一种新的去中心化 DNSSEC“域名−公钥”绑定对验证机制. 不同于传统 DNS 依赖于中心化根节点验证所构成的信任链, 本文机制基于区块链实现密钥基本操作. 域名所有者提交区域验证密钥及签名, 通过验证生成区块记录, 用户查询无需依赖中心化验证, 即可获取状态一致的密钥. 为基于区块链实现密钥验证操作, 需要对区块结构和节点进行重新设计.

3.1 整体结构

在 DNSSEC 中, 用户在查询域名记录的同时, 获取记录的数字签名 RRSIG 和公钥 DNSSKY. 本文提出的公钥验证机制为公钥记录提供了基于区块链的去中心化验证机制, 在保证安全性和验证性能的同时, 无需 PKI 系统中构建信任链的繁琐易错操作. 引入密码学累加器, 密钥绑定对的验证操作不再需要顺序遍历所有区块, 而只需在当前区块中通过存在性操作即可实现, 极大地降低了区块链操作的复杂性. 整体结构如图 1 所示.

在传统的链式区块链设计中, 所有节点均参与区块生成, 并在本地存储维护整个区块链数据. 这

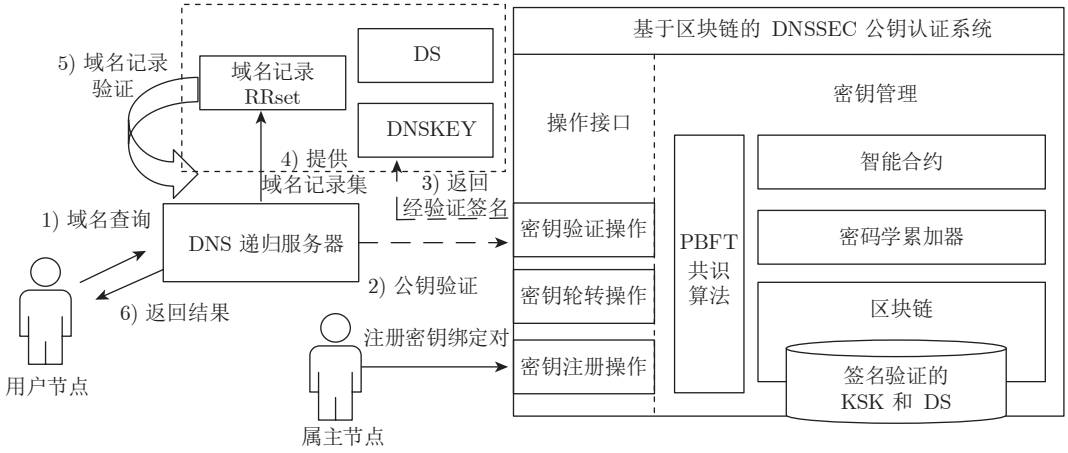


图 1　整体结构图

Fig.1　Overall system structure

种结构导致区块链难以满足 DNSSEC 对高性能验证的要求. 为此, 本文结合 DNS 体系, 设计了 3 种类型的区块链节点, 分别为:

1) 权威节点 (A): 指定为 DNS 体系中指定的权威服务器. 权威节点是安全可信的, 具有唯一的节点标识, 生成节点密钥对. 在区块链建立时, 权威节点将初始化为共识委员会节点, 并预先设定一个权重值.

2) 属主节点 (R): 指定为 DNS 记录的域名所有者, 即维护 DNS 记录的服务器, 通常具有域名解析器. 属主节点可通过提出建议, 达成共识后加入到共识委员会中. 属主节点同样具有唯一的节点标识, 生成节点密钥对.

3) 用户节点 (U): 用户节点是查询 DNS 的用户. 用户节点可验证公钥, 但不参与公钥和密码学累加器的维护, 不参与共识委员会.

如上, 只有类型 $type \in \{A, R\}$ 的节点才能参与公钥的生成和签名, 以及密码学累加器的添加、更新和删除操作. 所有节点 $type \in \{A, R, U\}$ 均可执行公钥验证操作. 节点 u_i 的记录表示为三元组 $A_i = \langle ID_i, Hash(pk_i), type\rangle$, 其中 ID_i 唯一标识节点 u_i, $Hash(pk_i)$ 是节点公钥的哈希值.

3.2　区块链初始化

为实现每个节点无需中心化信任链即可本地验证公钥可信性, 需要在区块中提供共识委员会和密码学累加器信息. 为此, 设计区块结构如图 2 所示.

与其他区块链一样, 区块头部包括区块的时间戳、区块标识以及链接到上一个区块的哈希值. 设计区块链的创世区块中包括所有 k 个权威服务器信息, 即 $\{A_1, \cdots, A_k\}$, 以及初始密码学累加器和存在性证明状态. 随后生成的区块, 头部包括当前共识委员会信息, 以及当前密码学累加器和存在性证明状态.

针对所设计的区块结构, 本文提出算法 1 所示的区块链初始化过程. 算法 1 创建区块中的密码学累加器, 以及由设定权威节点组成的初始共识委员会.

算法 1. 区块链初始化

输入. 哈希函数族 H, 权威节点集合 $\{A_1, \cdots, A_n\}$.

输出. 具有创世区块的区块链.

步骤 1. 对于集合 $\{A_1, \cdots, A_k\}$ 中的每个权威节点, 调用密钥生成操作 $G(H, A_i) \to (pk_{A_i}, sk_{A_i})$.

步骤 2. 指定一个权威节点调用密码学累加器生成操作 $caSetup(H) \to S_0$, 创建初始化累加器状态 S_0.

步骤 3. 调用密码学累加器添加操作 $caAdd(S_{i-1}, R_i)$, 其中 $R_i = (pk_i, sk_i)$, $i = 1, \cdots, k$, 依次将权威节点密钥对添加到状态中 S_i 中, 最终生成累加器初始状态 S_0 和初始见证者 $W_0^{R_0}$.

步骤 4. 当前节点发起生成区块提议 BP_1, 内容包括区块 ID、时间戳、S_0、$W_0^{R_0}$、权威节点账户标识 $A_i = \langle ID_i, Hash(pk_i), type = A\rangle$ 等信息. 当前节点担任领导者发起 PBFT 类共识协议, 其他权威节点作为初始委员会成员, 参与达成共识.

步骤 5. if 多于 $3(m+1)/2$ 个委员会成员验证 $W_0^{R_0}$ 正确并响应投票, **then**

步骤 6. 委员会就 BP_1 达成共识, 区块链初始化完成, 生成创世区块.

步骤 7. else 步骤 2, 重新生成密码学累加器.

3.3　密钥注册

密钥注册实现密钥绑定对在 DNSSEC PKI 中的注册, 通过区块链智能合约实现. 智能合约函数

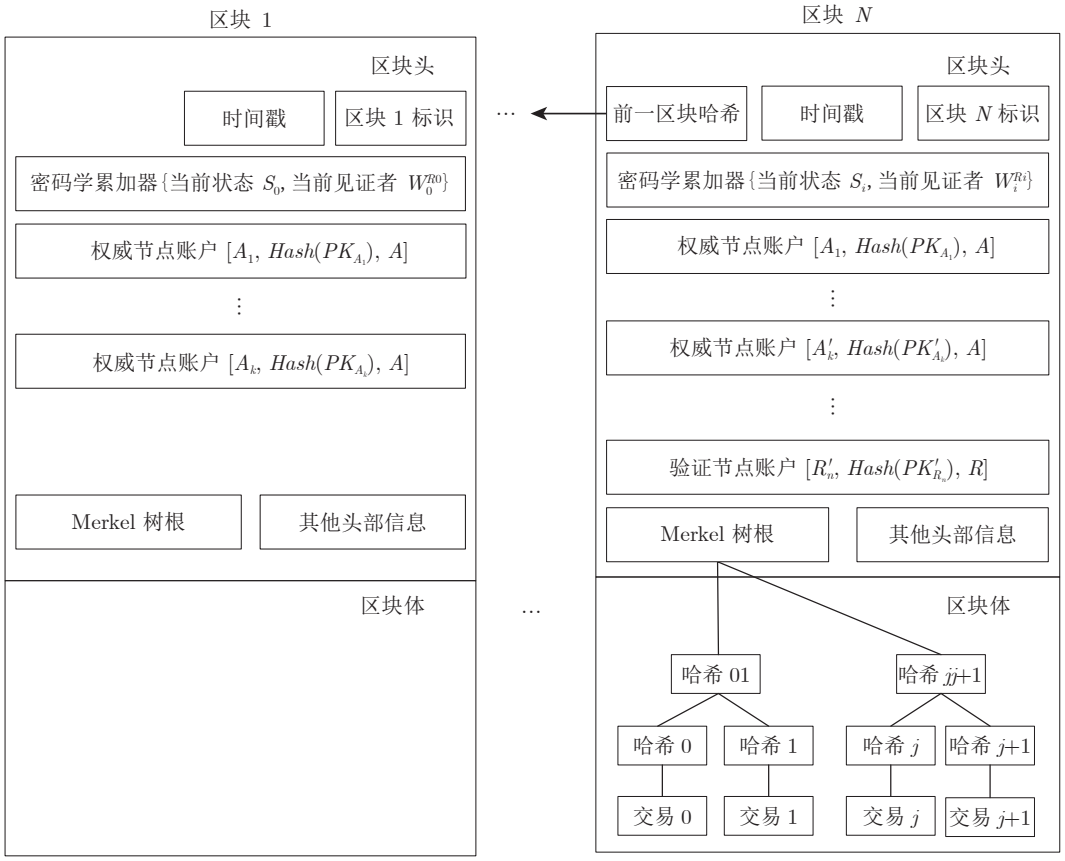


图 2 区块结构图

Fig. 2 Block structure

在被调用时, 会从称作函数选择器的位置开始执行. 函数选择器负责解析消息, 并跳转到相应功能函数. 针对 DNSSEC 密钥注册, 本文设计了两种区块链交易类型, 具体为:

1) 请求绑定交易. 由域名记录的所有者发起, 请求将自身公钥 pk_i 与域名记录 $RRSet_i$ 形成绑定对 $(RRSet_i, pk_i)$.

2) 绑定交易. 由生成公钥的父域发起, 将经根域验证的公钥与域名记录绑定形成验证对, 由智能合约更新密码学累加器, 并加入区块同步到全区块链网络.

密钥绑定操作实现过程如图 3 所示. 具体而言, 域名所有者创建自身域名记录的公钥 DNSKEY 和签名验证 RRSIG. 之后向区块链发起请求绑定交易, 交易中包含域名、DNSKEY 和 RRSIG 信息. 交易中信息将提交智能合约处理. 智能合约首先将信息提交其父域所有者, 并递归直至根域逐层验证父域公钥, 如父域公钥通过验证, 将由父域所有者节点对子域公钥 KSK 生成散列值 DS, 并发起绑定请求, 由智能合约调用密码学累加器的添加操作 $caAdd(S_i, (RRSet_i, pk_i))$ 更新状态和见证, 与签名

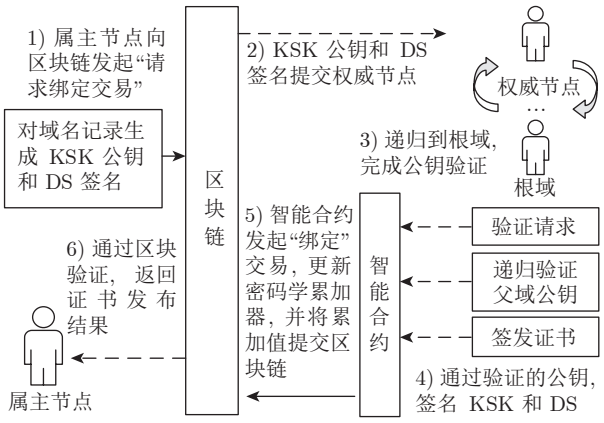


图 3 基于智能合约的公钥绑定对注册操作

Fig. 3 A public key binding pair registration process based on intelligent contract

一并提交区块链, 加入区块同步到全网络, 从而实现对域名的签名验证.

3.4 密钥轮转

密钥轮转实现密钥对的更新, 是 DNSSEC 密钥管理的重要操作. 出于安全考虑, 域名所有者可

定期频繁轮转 ZSK 公钥. 而 KSK 公钥的轮转需要更改父域的 DS 记录, 涉及与整个信任链交互.

本文提出的密钥轮转操作遵循 RFC 4641 定义的预发布方式. 属主节点可同时使用新、旧 KSK 公钥签名, 并在旧 DS 和旧 KSK 记录的最长生存时间 (Time to live, TTL) 到期后删除旧记录. 预发布方法避免需要同时管理两组公钥, 进而降低了域名记录大小. 但预发布需要考虑密钥的时间线管理.

遵循 RFC 7583, 本文定义密钥轮转时间线为创建、发布、激活、去活和移除五个阶段. 具体而言, 创建阶段创建的密钥尚未包含在区域文件中, 未用于对任何记录进行签名. 发布阶段将密钥发布在区域中发布, 但尚未用于签署任何记录. 激活阶段设置已发布密钥为活动状态, 开始签署记录. 去活阶段将已发布的旧密钥状态设为无效, 不再用于签名. 移除阶段域名解析器不再以任何方式维护旧密钥. 由此, 密钥定期自动轮转实现如算法 2 所示:

算法 2. ZSK 密钥定期自动轮转算法

输入. 属主节点, 轮转域名, 旧公钥, 激活时间阈值 θ, 区块链, 累加器 T.

输出. 新公钥绑定.

步骤 1. for $\forall$ 公钥 $pk \in$ 轮转域名:

步骤 2. 判定旧公钥的激活时间 t 是否处于阈值 θ 内;

步骤 3. **if** $t > \theta$, **then**

步骤 4. 属主节点创建新密钥对, 发起请求绑定交易;

步骤 5. **else** 用旧公钥签名生成域名记录的 RRSIG

步骤 6. **if** 请求绑定交易执行成功, **then**

步骤 7. 去活旧公钥, 调用区块链的密钥注册操作, 更新公钥和父域的 DS 记录;

步骤 8. 执行累加器 T 的 $caUpdate$ 操作, 发起生成区块提议 BP_2;

步骤 9. **if** 委员会对 BP_2 达成共识, **then**

步骤 10. 更新所有区块累加器和见证, 激活新公钥;

步骤 11. **else** 返回步骤 4, 重新生成密钥对.

这里需要特别考虑对根域 KSK 的轮转操作. 在去中心化公钥验证中, 由于无需中心化权威即可验证公钥的可信性, 因此支持区域 DNS 自成体系运行. 这种场景下, 区域的根域公钥是整个 DNSSEC 的信任锚 (Trust anchor). 如果解析器无法获取最新的信任锚, 将导致整个信任链不可信, 进而无法解析 DNS 中的任何域名. 不同于传统 DNSSEC 密钥验证, 本文提出的机制通过定期轮询判断根域公钥的可用性, 并在轮转后更新智能合约中的根域公钥记录, 避免了更新整个信任链的代价.

3.5　密钥验证

密钥验证给定域名和公钥的绑定对, 验证该绑定是否可信. 本节设计密钥验证操作的高效性, 主要得益于密码学累加器的采用. 尽管累加器在区块链中广泛用于降低存储复杂性, 但是针对 DNSSEC 密钥验证, 需选取适用于智能合约中累加器状态的累加器.

在 DNSSEC 应用这样大规模部署中, 出于安全考虑, 存在性见证本身也需要设置验证. 由此, 本文工作中采用了 Camacho 等[24] 提出的基于哈希树的累加器. 该累加器设计上使用无冲突散列函数实现平衡二叉哈希树数据结构, 累加器值是哈希树的根节点, 存在性证明是从树上某个节点到根节点的哈希路径. 该设计使得基于哈希树的累加器无需额外维护见证. 不同于基于 RSA 的累加器数据结构大小为 $\mathrm{O}(1)$, 该累加器的数据结构大小为 $\mathrm{O}(\log(n))$, 其中 n 是候选集合大小.

基于密码学累加器的公钥绑定验证如图 4 所示. 由于需要节点构建平衡二叉哈希树作为见证, 而该树结构的空间代价为 $\mathrm{O}(\log(n))$, 因此在空间代价上略高于基于 RSA 的单向哈希累加器. 但该累加器的构造和验证计算复杂度低, 相比基于 RSA 的累加器更易于实现. 实际在区块链中, 新添加全节点可以通过同步交易数据, 实现树结构在本地的重构, 进而实现见证.

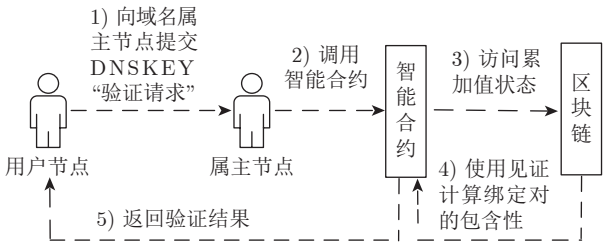


图 4　基于密码学累加器的公钥绑定验证

Fig. 4　Public key binding verification based on cryptography accumulator

如第 3.2 节区块结构设计所示, 通过在区块头部添加累加器和见证信息, 每个达成共识的区块可在节点本地维护自身见证, 即上述平衡二叉哈希树数据结构. 累加器在区块链上的运行机制是:

1) 初始化. 类型 $type \in \{A, R\}$ 的节点通过调用智能合约, 运行初始化操作 $caSetup$, 生成表示空集合 S_0 的密码学累加器数据结构, 即只有根节点的哈希树 T, 作为初始累加值写入区块并向全网广播. 委员会节点接收区块并达成共识, 全网同步累加器状态, 并使用 $caAdd$ 计算各自见证.

2) 添加密钥绑定. 通过验证操作的 (id, pk) 对, 由属主节点调用智能合约, 使用 $caAdd$ 运算添加到累加器, 并更新见证. 属主节点发起"绑定请求"交易添加到区块, 同步给委员会. 委员会节点接收状态后, 使用 $caBelongs$ 操作验证创建的正确性. 若正确, 则使用共识算法投票. 委员会达成后, 区块同步到全网络.

3) 公钥验证. 给定 (id, pk) 对, 用户节点可调用 $caBelongs$ 操作, 使用本地节点中的累加器和见证, 验证公钥是否存在于集合中, 实现公钥的去中心化验证.

3.6 性能分析

本节通过操作的复杂性分析, 对本文提出的 DNSSEC 公钥验证性能做理论上估计, 侧重于验证和轮转操作的计算复杂性.

1) 验证交易时间复杂度为常数 $\mathrm{O}(1)$. 在验证过程中, 每个矿工将交易与自身账本对比, 并在匹配情况下将继续操作, 否则忽略交易. 如果交易尚未在账本中注册, 则验证签名是否按定义正确构建, 进而将包含身份和注册密钥的元组对添加到累加器中, 执行累加器的验证, 确定新的累加器及其新区块的见证. 因为委员会节点在过程中只执行一次读取操作和一次哈希操作. 因此验证操作的时间复杂度为 $\mathrm{O}(1)$.

2) 轮转操作的时间复杂度为 $\mathrm{O}(\log(n))$, 其中 n 为系统中注册的公钥数量. 轮转中, 委员会节点首先验证两个签名均正确, 然后从累加器中删除包含标识和旧密钥的元组, 用新密钥将元组对添加到密钥管理系统, 确定新的累加器及其新区块的见证. 操作的复杂度主要是在区块链区块中查找匹配密钥的代价.

3) 累加器操作中, 添加元素的时间复杂度为 $\mathrm{O}(n\log(n))$, 删除元素的时间复杂度为 $\mathrm{O}(\log(n))$, 更新见证的时间复杂度为 $\mathrm{O}(1)$.

相比传统 PKI 系统, 本文在密钥注册、验证和轮转操作上并未明显增加复杂性. 与 DNSSEC 系统相比, 由于无需维护信任链操作, 因此操作复杂性得以降低. 与区块链分布式账本交易操作相比, 由于引入累加器设计, 降低了可信验证和更新的复杂性.

3.7 讨论

1) 区块链的运行安全和激励机制

传统的 PKI/CA 系统在运行中缺乏提高安全性的激励机制, 并且由于集中式的运行机制, 导致恶意证书报告和排查需手动完成, 这阻止了参与节点做出贡献. 在本文提出的机制中, 可自动对未授权证书进行响应, 激励 CA 正确行为, 并鼓励其他机构报告可能的未授权证书. 激励机制由区块链本身提供, 在本文研究中并未考虑.

从安全性分析, 区块链共识机制通过全部委员会节点参与验证, 未验证的区块将无法添加到链上, 避免错误信息的传播. 在系统设计上, 密钥绑定和轮转需要通过交易将绑定记录 $(RRSet_i, pk_i)$ 写入区块. 这需要矿工节点在本地验证后广播到网络, 通过委员会节点的验证并达成共识. 因此, 具有 n 个全节点的区块链, 可避免 $m < (n+1)/2$ 个节点的错误密钥对绑定, 以及 $m < (n+1)/2$ 个节点的错误密钥轮转. 同样, 区块链可避免 DoS 攻击, 只要超过半数的正常工作节点通过验证, 即可确认区块.

本文进一步设计了两种基于 PBFT 的分组共识机制, 将在第 4 节中展开详述.

2) 是否需要密钥的撤销机制

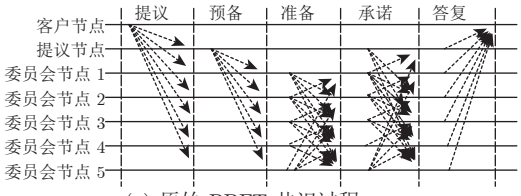
撤销机制一直是密钥管理研究的关注点. 现有 DNSSEC 标准尚未有设计完善的撤销服务. 虽然 RFC 5011 提出了一种多重密钥轮转流程, 同时使用激活和备用密钥, 但复制多份密钥对反而会增加密钥泄露的风险. 在进一步研究中, 将考虑 DNSSEC 密钥撤销机制.

4 基于 PBFT 的共识算法

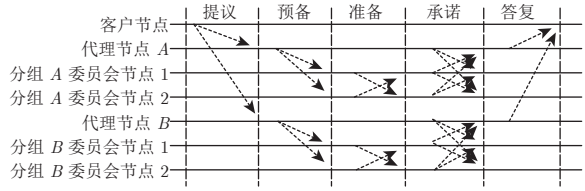
本文提出的基于区块链的 DNSSEC 公钥验证机制, 其安全性取决于区块链所采用的共识算法. 同时, 达成共识的性能也是公钥验证性能的重要影响因素.

实用拜占庭容错 (PBFT) 算法是一种主流的基于投票的算法, 在安全性和性能上提供了权衡, 在区块链中得到广泛使用. 为容忍 f 个恶意节点, 保证新交易区块生成为正确节点所创建, PBFT 共识委员会至少包含 $3f+1$ 个节点, 需要至少 $2f+1$ 个节点达成一致, 才能完成新区块的创建. 相对于 PoW (Proof of work) 类共识, PBFT 能够灵活采用诸如门限和环签名技术保证安全性[25] 的同时, 提高区块链共识的性能.

针对 PBFT 的改进主要针对网络通信开销. 如图 5(a) 所示, n 个委员会节点中的通信开销可达 $\mathrm{O}(n^2)$. 导致委员会超过一定规模后, 共识性能急剧下降. 对此, 本文的解决方法是根据一定策略将委员会划分为多个共识组, 如图 5(b) 所示. 每个组通过采用某种共识机制的可靠性评估后, 在预备、准备和承诺阶段独立达成一致, 进而由组内代理节点统一答复, 形成共识. 委员会分组很大程度上降低



(a) 原始 PBFT 共识过程
(a) The consensus procedure of naive PBFT



(b) 分组策略 PBFT 共识过程
(b) The consensus procedure of PBFT leveraging grouping strategy

图 5　采用分组的 PBFT 共识
Fig. 5　Leveraging grouping mechanism in PBFT consensus

了通信复杂度, 合理的分组机制是改进 PBFT 的关键.

结合 DNS 体系, 本文提出两种 PBFT 分组机制, 在确保安全性同时, 提高达成共识的性能.

第 1 种机制称为聚类分组机制, 采用聚类算法形成共识委员会分组. 首先, 在共识委员会中指定几个权威节点为聚类中心点. 由于权威节点所在的 DNS 权威服务器通常是部署在良好运营环境的可靠节点, 因此是适合的聚类初始中心点. 然后, 采用 KMeans 聚类算法, 定义网络响应时延为距离做聚类运算, 形成委员会分组. 由于聚类中心点并不发生变动, 因此聚类算法迭代可快速收敛. 所形成的分组以聚类中心点为指定的代理节点, 执行分组 PBFT 共识.

第 2 种机制称为域名分组, 考虑 DNS 域名的层次结构, 将同一域名子树的委员会节点分为一组, 并以子树根节点为分组的代理节点. 分组的可靠性考虑各节点先达成组内共识.

对比两种 PBFT 分组机制, 聚类分组考虑了网络延迟特性, 改进了分组内达成共识效率, 但聚类分组需要引入一定额外代价; 域名分组基于预定规则可快速构建委员会分组, 但分组可靠性需通过预先达成共识保证. 本文在实验中将验证不同机制的性能.

5　实验验证

实验通过构建基于区块链的 DNSSEC 公钥验证机制的概念验证 (Proof of concept, PoC) 原型系统, 验证本文提出的公钥验证机制的可行性和有效性.

5.1　实验环境

实验环境采用以太坊和 BIND (Berkeley internet name domain) 域名服务. 以太坊智能合约使用图灵完备的 Solidity 语言编写, 支持分支、循环等程序逻辑, 可定义状态变量、函数以及构造函数. 为部署到以太坊平台, 智能合约源代码编译为以太坊虚拟机 (Ethereum virtual machine, EVM) 运行的二进制执行代码. DNS 环境采用 BIND 9.16 构建的独立域名系统, 分别设置 example.com 权威域名和 sub.example.com 的子域, 提供对 DNSSEC 机制的支持.

5.2　功能验证

密钥验证机制主要部署于属主节点, 即 DNS 域名解析器端, 对客户节点是透明的. 因此功能验证实验通过在客户端使用支持 DNSSEC 的命令操作. 运行 dig 命令查看起始授权机构 (Start of authority, SOA) 记录的结果如图 6 所示.

```
dig example.com +dnssec soa
; <<>> DiG 9.16.3 <<>> example.com
+dnssec soa
;; flags: qr rd ra; QUERY: 1, ANSWER: 2,
AUTHORITY: 0, ADDITIONAL: 1
; example.com. IN SOA
;; ANSWER SECTION:
example.com. 3600 IN SOA ns.icann.org.
noc.dns.icann.org. 2020080310 7200 3600 1209600
3600
example.com. 3600 IN RRSIG SOA 8 2 3600
20200909043739 20200819023700 41461
example.com.
EnQsB5NpGnibrYEuA/9pUQ-JanSOAGAl9Y
7SLUoKhd085JJ4fjH8u5jM2ix/GNmvYArFIksuX
dMwiaAJCEONtpZEiRvOWcqS64lfdP3l2gggmA
p7MnIAMjLdhWJXEnLO9tlAm-CotUcSMFSn
fy3sJRLc/0QX2By6Pm4idOd9dc
bTg=
;; Query time: 164 msec
;; MSG SIZE rcvd: 295
```

图 6　运行 dig 命令查看 SOA 记录的结果
Fig. 6　Running dig command to view the results of SOA records

结果表明已生成 RRSIG 签名并通过验证, 消息大小未发生明显变化. 这表明不同于 PKI 需添加验证证书, 本文机制未引入额外的通信代价.

为进一步验证机制的可行性, 实验开展了中间人攻击测试. 本文将递归服务器发往客户端的解析

应答报文中篡改后发往客户端. 这样客户端向递归服务器发起域名解析请求时, 收到的是被篡改过的应答信息. 启动客户端 DNSSEC 并向递归服务器发起“example.com”域名解析请求, 终端显示“server can′t find example.com: SERVFAIL”, 这表明本地收到篡改的应答包没有通过本地验证, 报文在本地被丢弃导致域名解析失败. 上述操作验证了基于区块链的 DNSSEC 公钥验证机制工作正常.

5.3 共识性能分析

在基于区块链的公钥验证机制中, 密钥注册和密钥轮转都需要达成共识, 因此共识的安全性和性能是影响密钥操作的关键因素之一. 为验证一定网络规模下共识性能, 实验采用 NS2[1] 生成具有 180 个节点的网络拓扑, 并指定其中 10 个为权威节点, 模拟实际物理网络.

在安全性方面, 实验主要针对分组可靠性存在的风险进行测试评估. 不可靠的分组将极大降低 PBFT 共识的容错能力. 实验考虑两种情况, 一种情况是随机指定非权威节点为恶意节点, 另一种情况是指定权威节点为恶意节点. 实验指定共识委员会规模为 30, 聚类分组按 10 个权威节点分为 10 组, 域名分组下随机分为 10 组, 模拟达成 300 次交易, 执行结果情况如表 2 所示. 从实验结果中可见, 域名分组方式和聚类分组方式分别对权威节点作恶和非权威节点作恶表现出更好的容错能力. 但在实际 DNS 环境中, 权威节点由于部署为 DNS 权威服务器, 确保难以成为恶意节点.

表 2 各方式 PBFT 算法容错能力比较 (%)

Table 2 Comparison of fault toleration capability of different PBFT algorithms (%)

PBFT 类型[注1]	无分组[注2]	聚类分组	域名分组
2/0	2	1	2
4/0	5	2	5
10/0	100	17	32
1/2	3	4	9
4/4	13	100	23
10/4	100	100	47

注 1: 表示 30 个节点委员会中“恶意非权威节点个数/恶意权威节点个数”.
注 2: 列 2 ~ 4 给出未达成共识交易次数所占百分比.

在交易处理速度上, 实验的模拟区块链规模从 10 个节点扩展到 180 个节点, 权威节点和分组维持 10 个, 共识委员会相应从 10 个成员扩展到 60 个. 实验结果如图 7 所示. 从图 7 中的实验结果可以看出, 分组方法在交易处理速度方面相较于未分组 PBFT 方案有了明显提高. 特别是随着区块链规模的扩展, 虽然交易时间都呈现上升趋势, 但是两种分组优化方案仍能比原始方案的时间少约 50%, 分组优化方案的交易处理速度提升明显. 聚类分组体现出一致的优于域名分组的性能, 可归为聚类算法实现性能要优于达成共识形成分组.

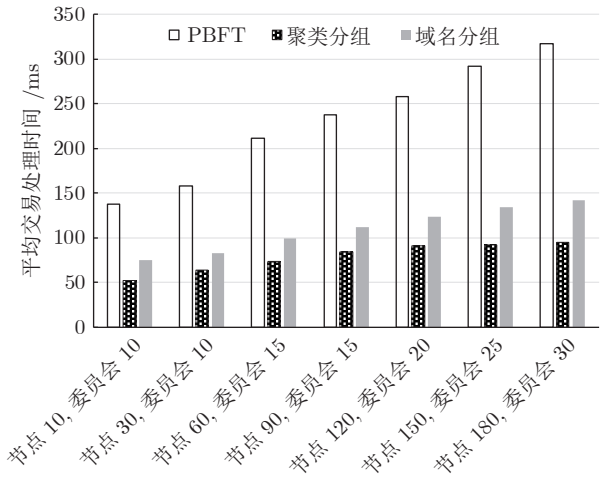


图 7 各 PBFT 算法交易处理速度对比

Fig. 7 Comparison of transaction processing latency of different PBFT algorithms

5.4 密钥操作的实验分析

针对本文提出的密钥注册、密钥轮转和密钥验证操作的有效性, 本节开展进一步实验进行验证. 其中, 共识机制均采用聚类分组方式.

密钥验证操作是本文机制关注的核心问题. 实验采用了两种对比方法, 即遍历区块链查询的传统验证方法, 以及引入密码累加器的验证方法. 方法耗时对比结果如图 8 所示. 传统方法中遍历区块链的时间与区块链高度成线性关系, 本文采用的基于哈希树的累加器进行验证与区块链高度成对数增长, 耗时仅为累加器单步验证运算的时间, 呈现小范围内波动. 此外在实际运行中, 密码累加器的更新和维护需额外增加计算量. 实验表明, 本文引入的基于哈希树的密码累加器, 降低了节点更新维护自身见证的代价, 用户的验证效率相较传统方法得到提升, 并随区块链高度增加而不断提高. 本文提出方法在实现去中心化密钥验证的同时, 取得了一致性的性能改进.

进一步在模拟网络环境中对比本文机制和 DNSSEC 实现, 给出获取一致的公钥情况下对比方法的响应性能, 结果如图 9 所示. 结果可见, 本文机

[1] https://www.isi.edu/nsnam/ns/ns-topogen.html

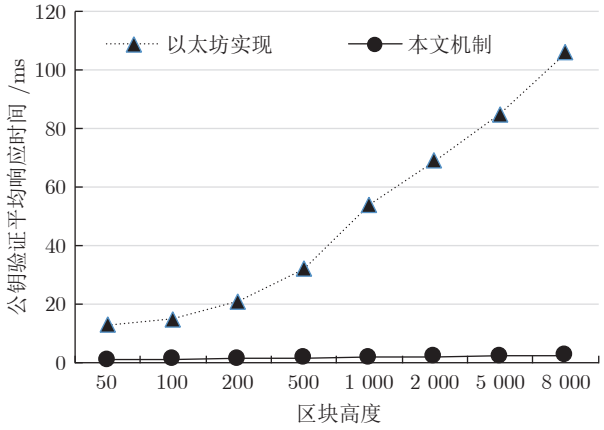


图 8　对比以太坊实现的公钥验证时间

Fig. 8　Public key verification time compared with Ethernet implementation

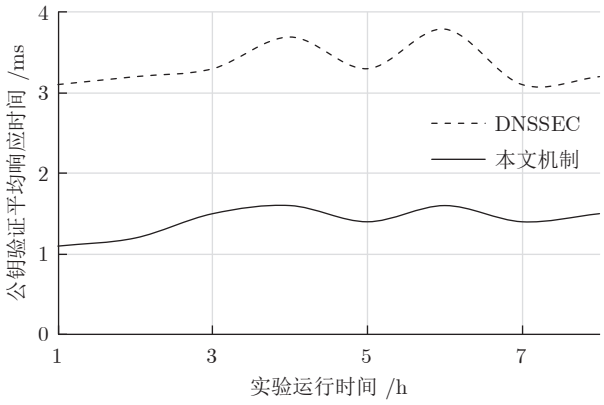


图 9　对比 DNSSEC 的公钥验证时间

Fig. 9　Public key verification time compared with DNSSEC implementation

制表现出优于 DNSSEC 的密钥验证响应性能, 这可归结于本文的去中心化机制只需在本地节点中执行密码学累加器验证操作, 而中心化的 DNSSEC 需要通过信任链上溯至权威节点进行验证. 实验验证了密钥可信验证在性能上的改进. 注意在真实网络环境中, 中心化的 DNSSEC 的响应性能还需考虑网络延迟和安全性因素, 而本文机制可在本地节点完成可信验证, 可进一步表现出更好的对比性能.

对于密钥注册操作, 实验设计采用不同消息大小, 对比传统 DNSSEC 注册时间. 实验结果如图 10 所示, 本文方法的注册时间略高于传统 DNSSEC, 这是由于采用区块链引入的区块验证和共识代价导致的结果. 需要说明的是, 实际应用中密钥注册性能可以通过调整区块链的出块参数进行改变. 实验中设置出块时间为 15 s, 根据以太坊区块链的 GasLimit 设置, 每个区块中交易数约 520 个, 区块大小约 45 KB. 由此估计区块链每秒交易约 35 个.

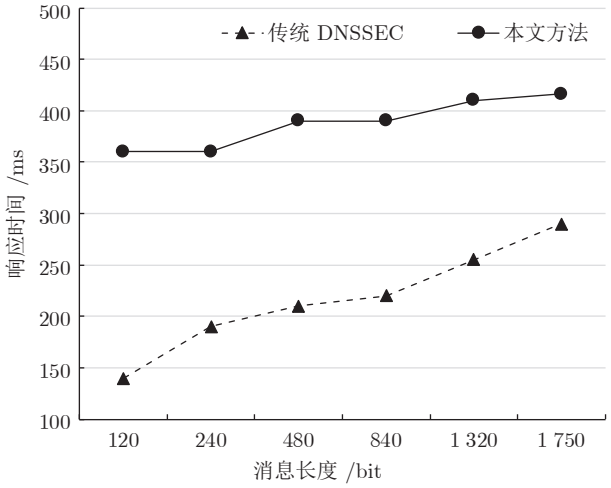


图 10　密钥注册时间对比

Fig. 10　Key registration time comparison

进一步调整区块链参数可以在密钥注册安全性和性能上取得很好的权衡.

对于密钥轮转操作, 本文在实验中考虑子域 KSK 轮转. 实验设计测试 DNSSEC 响应的大小, 如图 11 所示. 通过对响应大小的测试, 可以确定轮转过程中采用了双签机制, 即同时存在新旧 KSK 公钥和签名记录, 并在 TTL 过期后, 撤销旧密钥并维持轮转后新密钥. 该实验验证了双签机制的工作, 以及对父域签名记录 RRSIG 的更新.

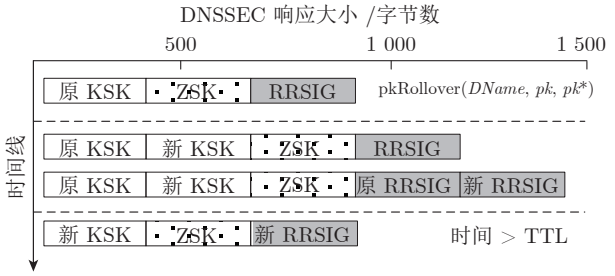


图 11　基于区块链的 KSK 公钥轮转响应示意图

Fig. 11　Schematic diagram of KSK public key rotation response based on block chain

综上, 对密钥操作的实验验证表明, 本文机制可有效地支持密钥的验证、绑定和轮转操作. 基于区块链的密钥验证机制避免中心化 PKI/CA 体系下信任链的构建, 用户查询时通过本地节点中密码学累加器验证操作, 即可验证密钥是否可信. 考虑到密钥注册并非 DNSSEC 体系中的频繁操作, 而本文方法侧重于实现去中心化的密钥验证机制, 因此在注册性能和验证的功能性上的权衡是可以接受的.

6 结束语

DNSSEC 是增强域名服务的重要机制, 但密钥信任链管理的复杂性及其衍生问题, 严重影响 DNSSEC 的普及推广. 基于区块链的 PKI 为解决信任链管理复杂性提供了可行途径, 但需根据 DNSSEC 特性进行改进.

本文针对 DNSSEC 公钥验证的特性, 提出了一种基于区块链的高效密钥操作机制, 通过区块链结构、交易和智能合约操作的设计, 创新性地实现了适用于 DNSSEC 密钥的注册、验证和轮转操作, 避免了中心化体系下管理密钥信任链的复杂度. 进一步实验表明, 本文提出的机制能在保证整体机制安全性的同时, 有效提高了 DNSSEC 公钥验证的性能.

本文提出的机制适用于解决域名安全验证中依赖于中心化 PKI/CA 信任链的问题. 同样适用于解决 DNS over TLS/HTTPS 协议中信任链问题. 另一方面, DNSSEC 本身只是对域名记录的签名验证, 并未提供记录的保密性. 记录和交易的保密性也是区块链领域的近期研究热点问题, 将是我们下一步工作的关注方向.

References

1 Mockapetris P V. Domain names —— Concepts and facilities, STD 13, RFC 1034 [Online], available: https://www.rfc-editor.org/info/rfc1034, November 1, 1987

2 Mockapetris P V. Domain names —— Implementation and specification, STD 13, RFC 1035 [Online], available: https://www.rfc-editor.org/info/rfc1035, November 1, 1987

3 Shulman H, Waidner M. One key to sign them all considered vulnerable: Evaluation of DNSSEC in the internet. In: Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation. Boston, USA: USENIX Association, 2017. 131−144

4 Arends R, Austein R, Larson M, Massey D, Rose S. Resource records for the DNS security extensions [Online], available: https://www.rfc-editor.org/info/rfc4034, March 1, 2005

5 Yang H, Osterweil E, Massey D, Lu S W, Zhang L X. Deploying cryptography in Internet-scale systems: A case study on DNSSEC. *IEEE Transactions on Dependable and Secure Computing*, 2011, **8**(5): 656−669

6 DNSSEC deployment report [Online], available: http://rick.eng.br/dnssecstat/, April 13, 2022

7 Chung T, Van Rijswijk-Deij R, Chandrasekaran B, Choffnes D, Levin D, Maggs B M, et al. A longitudinal, end-to-end view of the DNSSEC ecosystem. In: Proceedings of the 26th USENIX Conference on Security Symposium. Vancouver, Canada: USENIX Association, 2017. 1307−1322

8 Yuan Yong, Ni Xiao-Chun, Zeng Shuai, Wang Fei-Yue. Blockchain consensus algorithms: The state of the art and future trends. *Acta Automatica Sinica*, 2018, **44**(11): 2011−2022
(袁勇, 倪晓春, 曾帅, 王飞跃. 区块链共识算法的发展现状与展望. 自动化学报, 2018, **44**(11): 2011−2022)

9 Hari A, Lakshman T V. The Internet blockchain: A distributed, tamper-resistant transaction framework for the Internet. In: Proceedings of the 15th ACM Workshop on Hot Topics in Network. Atlanta, USA: ACM, 2016. 204−210

10 Namecoin [Online], available: https://namecoin.org/, April 13, 2022

11 Ali M, Nelson J, Shea R, Freedman M J. Blockstack: A global naming and storage system secured by blockchains. In: Proceedings of USENIX Conference on USENIX Annual Technical Conference. Denver, USA: USENIX Association, 2016. 181−194

12 Liu J Q, Li B, Chen L Z, Hou M, Xiang F R, Wang P J. A data storage method based on blockchain for decentralization DNS. In: Proceedings of the 3rd IEEE International Conference on Data Science in Cyberspace (DSC). Guangzhou, China: IEEE, 2018. 189−196

13 Wang W T, Hu N, Liu X. BlockZone: A blockchain-based DNS storage and retrieval scheme. In: Proceedings of the International Conference on Artificial Intelligence and Security. New York, USA: Springer, 2019. 155−166

14 Matsumoto S, Reischuk R M. IKP: Turning a PKI around with decentralized automated incentives. In: Proceedings of the IEEE Symposium on Security and Privacy (SP). San Jose, USA: IEEE, 2017. 410−426

15 Kubilay M Y, Kiraz M S, Mantar H A. CertLedger: A new PKI model with certificate transparency based on blockchain. *Computers and Security*, 2019, **85**: 333−352

16 Wang Z, Lin J Q, Cai Q W, Wang Q X, Zha D R, Jing J W. Blockchain-based certificate transparency and revocation transparency. *IEEE Transactions on Dependable and Secure Computing*, 2022, **19**(1): 681−697

17 Gourley S, Tewari H. Blockchain backed DNSSEC. In: Proceeding of the International Conference on Business Information Systems. Berlin, Germany: Springer, 2018. 173−184

18 Guan Z, Garba A, Li A R, Chen Z, Kaaniche N. AuthLedger: A novel blockchain-based domain name authentication scheme. In: Proceedings of the 5th International Conference on Information Systems Security and Privacy. Prague, Czech Republic: SciTePress, 2019. 345−352

19 Patsonakis C, Samari K, Roussopoulos M, Kiayias A. Towards a smart contract-based, decentralized, public-key infrastructure. In: Proceedings of the 16th International Conference on Cryptology and Network Security. Hong Kong, China: Springer, 2017. 299−321

20 Schaeffer Y, Overeinder B J, Mekking M. Flexible and robust key rollover in DNSSEC. In: Proceedings of the Workshop on Securing and Trusting Internet Names (SATIN 2012). 2012.

21 Liu Yi-Zhong, Liu Jian-Wei, Zhang Zong-Yang, Xu Tong-Ge, Yu Hui. Overview on blockchain consensus mechanisms. *Journal of Cryptologic Research*, 2019, **6**(4): 395−432
(刘懿中, 刘建伟, 张宗洋, 徐同阁, 喻辉. 区块链共识机制研究综述. 密码学报, 2019, **6**(4): 395−432)

22 Zhang Chao, Li Qiang, Chen Zi-Hao, Li Zu-Rui, Zhang Zhen. Medical chain: Alliance medical blockchain system. *Acta Automatica Sinica*, 2019, **45**(8): 1495−1510
(张超, 李强, 陈子豪, 黎祖睿, 张震. Medical chain: 联盟式医疗区块链系统. 自动化学报, 2019, **45**(8): 1495−1510)

23 Benaloh J, de Mare M. One-way accumulators: A decentralized alternative to digital signatures. In: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques. Lofthus, Norway: Springer, 1993. 274−285

24 Camacho P, Hevia A, Kiwi M, Opazo R. Strong accumulators from collision-resistant hashing. *International Journal of Information Security*, 2012, **11**(5): 349−363

25 Sun Hai-Feng, Zhang Wen-Fang, Wang Xiao-Min, Ma Zheng, Huang Lu-Fei, Li Xuan. A robust Byzantine fault-tolerant consensus algorithm against adaptive attack based on ring signature and threshold signature. *Acta Automatica Sinica*, DOI: 10.16383/j.aas.c200694
(孙海锋, 张文芳, 王小敏, 马征, 黄路非, 李暄. 基于门限和环签名的抗自适应攻击拜占庭容错共识算法. 自动化学报, DOI: 10.16383/j.aas.c200694)

陈闻宇 高级工程师, 中国科学院计算技术研究所网络技术研究中心博士研究生. 主要研究方向为互联网基础资源, 网络安全和区块链技术.
E-mail: chenwy2000@163.com
(**CHEN Wen-Yu** Senior engineer, Ph.D. candidate at the Network Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences. His research interest covers internet fundamental resources management, network security, and blockchain technology.)

李晓东 中国科学院计算技术研究所研究员, 清华大学公共管理学院兼职教授. 主要研究方向为互联网基础资源, 大数据分析, 网络安全和互联网治理.
E-mail: xl@ict.ac.cn
(**LI Xiao-Dong** Professor at the Institute of Computing Technology, Chinese Academy of Sciences, and adjunct professor of the School of Public Policy and Management, Tsinghua University. His research interest covers internet fundamental resources management, big data analysis, network security, and internet governance.)

杨 学 中国互联网络信息中心高级工程师. 主要研究方向为互联网基础资源, 大数据和区块链技术. 本文通信作者. E-mail: yangx@cnnic.cn
(**YANG Xue** Senior engineer at China Internet Network Information Center. His research interest covers internet fundamental resources management, big data, and blockchain technology. Corresponding author of this paper.)

徐彦之 广东粤港澳大湾区国家纳米科技创新研究院高级工程师. 主要研究方向为人工智能, 大数据和区块链技术. E-mail: xuyz@cannano.cn
(**XU Yan-Zhi** Senior engineer at Guangdong-Hong Kong-Macao Greater Bay Area (GBA) Research Innovation Institute for Nanotechnology. Her research interest covers artificial intelligence, big data, and blockchain technology.)