

## 多域物联网中基于区块链和权能的访问控制机制

王思源, 邹仕洪

北京邮电大学 网络空间安全学院, 北京 100876

**摘要:** 物联网中的数据通常包含大量的隐私信息, 为了防止设备协同过程中因越权访问造成隐私数据泄露的问题, 针对多域物联网设备协同场景提出了一套访问控制机制。将分布式的基于权能的访问控制 (capability-based access control, CapAC) 与区块链技术相结合, 设计了存储于区块链的权能令牌以及基于智能合约实现的令牌管理合约。根据 CapAC 的决策方式, 设计了基于区块链的令牌验证方法。针对物联网的特性进行优化, 设计了区块链轻量化节点。最后, 搭建了区块链系统并实现了所提出的机制。实验测试结果显示, 相比于中心化的访问控制机制, 该方案不但在大规模的物联网场景下安全准确地执行访问决策而且具有更稳定的处理性能。此外, 轻量化设计能够大幅度地降低节点存储负担。

**关键词:** 区块链; 访问控制; 物联网; 智能合约

**中图分类号:** TP309; TP311.13

**文章编号:** 0255-8297(2021)01-0055-15

## Blockchain and Capability Based Access Control Mechanism in Multi-domain IoT

WANG Siyuan, ZOU Shihong

*School of Cyberspace Security, Beijing University of Posts and Telecommunications,  
Beijing 100876, China*

**Abstract:** Data in Internet of things (IoT) usually contains a large amount of personal privacy information, In order to prevent privacy data leakage due to unauthorized access during device collaboration, this article proposes a set of access control mechanisms for multi-domain IoT device collaboration scenarios. By combining distributed capability-based access control (CapAC) with blockchain technology, this article designs a capability token stored in the blockchain and a token management contract based on smart contracts. According to CapACs access decision-making method, a blockchain-based token verification method is designed. The blockchain lightweight node is optimized for the characteristics of IoT. Finally, a blockchain system is built to implement the mechanism proposed in the article. Experimental test results show that compared to centralized access control mechanisms, this solution can safely and accurately execute access decisions in large-scale IoT scenarios and has more stable processing performance. Lightweight design can greatly

收稿日期: 2020-11-15

基金项目: 国家重点研发计划基金 (No.2018YFB0803602) 资助

通信作者: 邹仕洪, 副教授, 研究方向为移动安全和区块链技术。E-mail: zoush@bupt.edu.cn

reduce node storage burden.

**Keywords:** blockchain, access control, Internet of things (IoT), smart contract

自1999年出现物联网概念以来,物联网领域出现了翻天覆地的变化,智能穿戴、智能家居、工业物联网等概念已走向现实。到2020年底,预计全球物联网设备数量将增长到358亿台。物联网应用的多样化改变了工业制造、智慧生活等领域,同时也带来了安全隐患。物联网产生了海量的数据,其中包含大量的个人隐私,这些隐私信息一旦泄漏就会给用户带来巨大的损失。身份认证和对资源、服务的访问控制是保护物联网设备安全性和隐私性的主要方式。作为在计算机系统中实现安全性的基本机制,访问控制根据特定的安全性模型和策略决定授权主体对客体的通信权限。一套有效的访问控制机制能够满足系统机密性、完整性、可用性等信息安全需求。

物联网不是传感器、通信接口和通信设备的简单叠加,而是不同网络、不同领域中各种各样的仪器设备相互连通与协作。随着物联网规模的不断增长,设备之间协作场景的出现概率越来越高。当前市场上的智能家具没有统一的管理标准,通常某品牌的设备只能通过厂商提供的应用平台与本品牌其他设备交互,导致不同品牌的设备相互隔离而无法有效协作;同时智能家具涵盖了各种类型的家用设备,不同类型的设备也有各自的安全和隐私需求。例如视频监控镜头、智能门锁等对于安全和隐私的要求较高,而智能灯、洗衣机、小家电等对安全和隐私的要求相对较弱。为了保障安全性,不同类型设备之间进行协作时应根据设备的安全和隐私保护特性分域进行管理。

针对多域物联网设备协作的场景,本文在CapAC模型的基础上结合了区块链技术,设计并实现了一套融合区块链和权能的访问控制机制。在该应用场景下,设备厂商、服务提供商等机构具备一定的权威性,可见系统并不完全去中心化,因此可以采用委任权益证明(delegated proof of stake, DPoS)作为区块链的共识算法,采用投票方式选择共识节点以便在去中心化和性能之间达到平衡。借助智能合约实现访问控制权能令牌的管理,根据访问控制系统的需求设计了新的交易结构,将读写操作更加频繁的访问控制决策和日志记录过程,通过交易创建、验证的方式来实现,提高了访问请求处理性能。引入区块链中的轻量化节点,适配物联网中复杂多样的异构设备,以增强系统的可扩展性。

## 1 相关技术

### 1.1 物联网中的访问控制

随着物联网技术和应用的不断发展,物联网从早期依托射频识别(radio frequency identification, RFID)技术的物流网络发展到目前万物皆可连的智慧地球,物联网环境下的访问控制也随之不断发展。由于物联网的特殊性,除了安全和隐私的需求之外,访问控制还应考虑可伸缩性、灵活性、轻量级以适配物联网环境。应用于物联网中的访问控制方法包括基于角色的访问控制(role-based access control, RBAC)、基于属性的访问控制(attributes-based access control, ABAC)、基于权能的访问控制等。

RBAC最初是用来解决大型企业级系统的访问控制问题的<sup>[1]</sup>。RBAC将用户映射到角色,用户即可通过角色享有许可。该模型定义了不同的角色、角色的继承关系、角色之间的联系以及角色所受的限制,动态或静态地规范了用户的行为。文献[2-3]基于RBAC的物联网访问控制,采用万维物联网(Web of things, WoT)的方法在智能设备上实施访问控制策略。文献[4]扩展了RBAC模型,在访问控制决策过程中引入上下文约束。然而,上述研究都只能实现粗粒度的访问控制,却无法精确地指定物联网中的各种资源。若只使用RBAC模型实现细

粒度的访问控制,那么不仅需要存储大量〈用户,角色〉、〈角色,权限〉的信息,而且会产生角色爆炸的问题,不适合作为结构复杂且角色界定模糊的物联网场景的安全策略。

不同于 RBAC 需要管理者提前预设〈角色,权限〉等对应关系,ABAC 是一种动态的访问控制模型,以属性作为访问控制的关键要素。因为属性是主客体内在固有的,所以 ABAC 无需管理者手动输入也可以借助属性发现机制挖掘出独立而完备的属性集合,并通过自动化的属性-权限关联关系发现机制快速挖掘出〈属性,权限〉关系。在物联网中引入 ABAC<sup>[5-6]</sup>,减少了可能导致角色爆炸的规则数量,解决了高度分散的物联网环境中 RBAC 模型的问题。文献 [7] 针对物联网感知层提出了一种基于 ABAC 的有效认证和授权方案,该方案基于属性证书授予用户特定权限,以确保细粒度的访问控制。但是随着设备数量的增加,为不同的域制定统一的访问控制策略会显著增加策略管理的工作量,因此基于属性的 ABAC 模型不适用于大规模分布式物联网。

CapAC 是访问控制矩阵 (access control matrix, ACM) 模型的一种实现方式,而访问控制列表 (access control lists, ACL) 是另一种实现方式。在 ACL 模型中,每个客体都与一个访问控制列表相关联,该列表记录了其他主体对该客体的访问权限。相反地,在 CapAC 模型中,每个主体都与一个权能列表相关联,该列表记录了该主体对其他客体的访问权限。CapAC 模型在物联网环境中不但实现了轻量级<sup>[8]</sup>、分布式<sup>[8-10]</sup>的访问控制,而且支持物联网的动态性和可扩展性<sup>[11-12]</sup>,并已逐渐成为物联网环境下最有前景的访问控制模型之一。为避免使用集中式服务器所带来的单点故障问题,基于分布式设计的 CapAC 在物联网中轻量级的设备上实现访问控制决策。然而,物联网设备因计算能力和存储能力较弱而容易受到恶意用户的攻击,无法成为一个安全的决策实体<sup>[13]</sup>。因此,基于分布式设计的 CapAC 无法解决在不可信环境下的物联网访问控制问题。本文在采用传统的分布式 CapAC 模型的基础上融合了区块链技术,为访问控制提供可信环境,以解决参与决策的物联网设备易被攻击的问题。

## 1.2 区块链技术

区块链最初作为比特币的底层记账系统而为人熟知<sup>[14]</sup>。Nakamoto 结合了 b-money 和 HashCash<sup>[15]</sup>,创建了一个不依赖中心化机构的货币发行、结算和交易验证的电子现金系统。区块链系统中各节点通过一定的共识机制选取具有打包交易权限的出块节点。出块节点需要将新区块的前一个区块的哈希值、当前时间戳、一段时间内的有效交易及其默克尔树根值等内容打包成一个区块向全网广播。由于每一个区块与前续区块都是通过密码学证明的方式链接在一起的,当区块链达到一定长度时,若要修改某个历史区块中的交易内容就必须重构该区块之前的所有区块的交易记录和密码学证明,这对任何节点来说都是难以实现的,因此该方式有效地实现了防篡改。区块链不仅包括链状数据块的结构,还包括 P2P 网络技术、共识机制、密码学技术等一系列技术结合后的产物。以比特币为代表的区块链 1.0 主要应用于数字货币的支付与流通等,能够实现去中心化的数字货币交易支付功能。

2014 年后,外界逐渐认识到区块链技术的价值并将其用于数字货币以外的领域,如分布式身份认证、分布式域名系统、分布式自治组织等。这些应用被称为分布式应用 (distributed applications, DAPP)。以区块链架构完整构建一个 DAPP 是困难的,但不同的 DAPP 会共享很多相同的组件。区块链 2.0 的诞生是为了创建可共用的 DAPP 平台并向开发者提供 BaaS (blockchain as a service) 服务,从而提高交易速度,降低资源消耗。智能合约是一种旨在以信息化方式传播、验证或执行合同的计算机协议。这类协议一旦制定和部署就能实现自我执行和自我验证,而不再需要人为干预。区块链 2.0 具有代表性的项目如 Ethereum<sup>[16]</sup> 和 Hyperledger Fabric<sup>[17]</sup> 等都提供了图灵完备的智能合约,降低了 DAPP 的开发难度。

在典型的区块链网络中,由于每一个节点都能够存储包含全网发生的历史交易记录的完整账本,对个别节点账本数据的篡改、攻击不会影响全网总账的安全性。账本的一致性是通过共识机制实现的。共识机制的核心是选主和记账两部分,在具体操作过程中,每一轮可以分为选主、造块、验证和上链4个阶段<sup>[18]</sup>。共识过程的输入是数据节点生成和验证后的交易或数据,输出则是封装好的数据区块以及更新后的区块链。4个阶段循环往复执行,每执行一轮就会生成一个新区块。工作量证明(proof of work, PoW)、权益证明(proof of stake, PoS)、DPoS、实用拜占庭容错(practical Byzantine fault tolerance, PBFT)等都是区块链系统中常用的共识算法。PoW和PoS都属于证明类共识,PoW是基于矿工的算力完成随机数搜索任务而竞争记账权的,PoS则是基于权益如持有货币数量来竞争记账权的。DPoS是基于某种特定方式选举出一组代表节点,然后代表节点以轮流或再选举的方式取得记账权。PBFT通常用于私有链和联盟链的共识算法,解决了在有限节点情况下的拜占庭问题,这样整个分布式系统可以在容忍小于1/3个无效或者恶意节点的同时保证一定的性能。

近几年,区块链技术和应用都在不断演进变化,区块链核心技术的研究方向包括安全与隐私保护技术<sup>[19-20]</sup>、跨链技术<sup>[21-22]</sup>、分片技术<sup>[23-24]</sup>、分布式数据存储<sup>[25]</sup>、新型共识算法<sup>[26]</sup>、智能合约形式化验证<sup>[27]</sup>等,但迄今为止尚未产生广泛认可的区块链3.0形态。

### 1.3 区块链节点轻量化

随着区块链技术的不断发展演进,区块链已广泛应用于各行各业,但也暴露出了一些问题。截至2019年底,比特币的账本大小已经接近0.5TB,而个人电脑的硬盘大小通常只有1.0~2.0TB。由于物联网中存在设备异构的特性,节点的运算、存储性能差距较大。对于常见的物联网智能家电设备来说,存储完整的区块链账本几乎是无法实现的。因此,在物联网环境中应用区块链技术就需要解决区块链存储的问题。

简单支付验证(simplified payment verification, SPV)<sup>[14]</sup>是应用最为广泛的区块链存储优化方式。SPV节点只需下载区块头而不必下载包含在每个区块中的交易信息,由此产生的不含交易信息的区块链大小只有完整区块链的1/1000,无法获得完整的交易信息,因此在验证交易时所用方法与全节点方法略有不同,还需依赖对等节点按需提供区块链相关部分的局部视图。SPV节点会通过请求得到的默克尔路径定位交易所在区块,并验证区块中的工作量证明,从而证实交易的存在性。在绝大多数实际情况中,具有良好连接的SPV节点是足够安全的,它在资源需求、实用性和安全性之间维持了恰当的平衡。

在近些年出现的基于区块链技术的物联网应用中,节点轻量化也是研究重点之一。对于只需验证消息完整性的场景,存储消息摘要值而不是完整消息就可以有效降低节点负担。文献[28]将完整的车辆检测报告生成摘要值存储在区块链中,实现了应用于联网车辆取证的轻量区块链框架。仅在区块链中存储控制信息,将空间占用较大的数据存储在本地的云服务器中,也可以获得降低节点存储负担的效果<sup>[29]</sup>。在Ethereum中,区块链同时记录了交易信息和余额状态,于是可以通过删除无用历史状态数据达到节省存储空间的目的,而所有的历史状态可以使用交易信息重新计算恢复。类似地,文献[30]设计了一款新型的区块卸载过滤器以避免账本的无限增长,优化了存储却不影响区块链的可追溯性。

## 2 融合区块链和权能的访问控制

### 2.1 系统模型

针对多域物联网下的设备协作场景,本文提出了一种融合区块链和权能令牌的访问控制机制,其系统整体架构分为云层和终端层两部分,如图1所示。

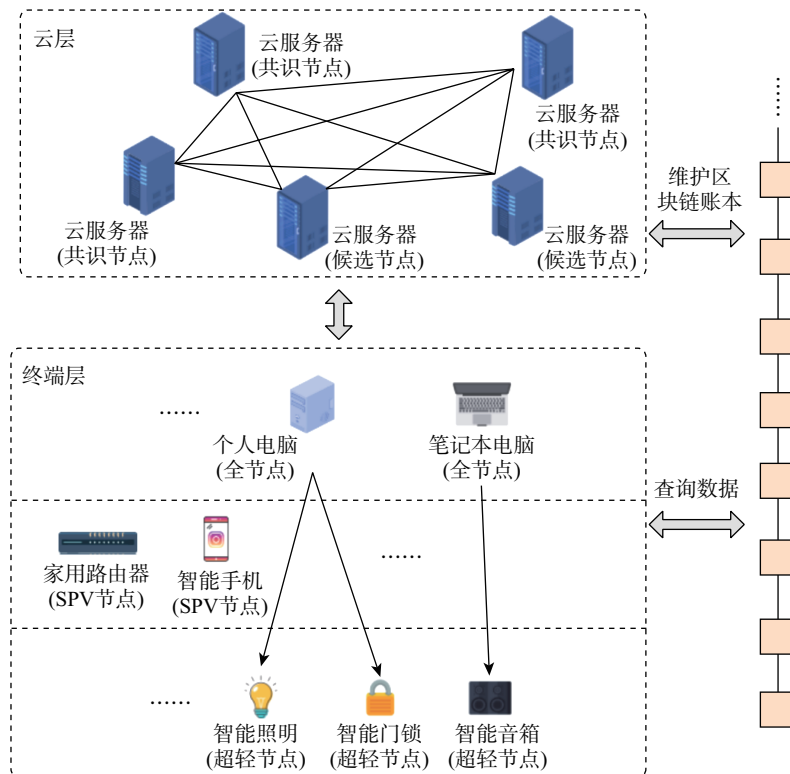


图 1 基于区块链和权能的访问控制系统模型

Figure 1 Blockchain and capability based access control system model

### 2.1.1 云层

该层次下的设备通常是具有高性能且稳定可靠的服务器, 在系统中充当共识节点和候选节点的角色。共识节点负责将包含访问控制记录的交易集构造成区块, 并维护区块链账本。在一个选举周期结束后, 为了保证共识节点都是稳定而可靠的, 本文采用 DPoS 作为系统的共识算法, 根据投票结果从候选节点中重新选择共识节点, 并将不稳定或低效的共识节点降级为候选节点。除区块链交易验证与打包区块之外, 云层中的节点还负责维护网络中的域信息列表。对于存在异常的访问主体, 云层中的节点需要追溯其访问记录, 并通知相应域管理器撤销异常节点的权能令牌。

### 2.1.2 终端层

该层次下的设备包含 3 种类型, 分别是高性能设备、普通设备和低功耗设备。在系统中, 高性能设备作为全节点, 可以进行交易的创建、验证以及查询等操作; 普通设备作为 SPV 节点, 可进行交易创建、SPV 等操作; 低功耗设备需设置域内某一高性能设备作为代理后加入系统。在每个域中, 都需要选举一个高性能设备作为域管理器来负责维护域内设备信息, 因为终端设备接入、异常节点的管理都需要域管理器的参与。

## 2.2 权能令牌存储结构

系统内的所有物联网设备实体在访问控制过程中分为主体和客体。主体是访问的请求者, 客体是待访问的资源, 且所有已注册的实体都应包含一个唯一的身份 ID 与该实体相关联。本

文中实体的唯一身份 ID 是通过区块链账户进行标识的。权能令牌关联了主体、客体、访问权限和约束条件,其基本结构如表 1 所示。每个主体可持有多个权能令牌,每个客体也可授予多个权能令牌。不同于传统 CapAC 将令牌存储在本地,本文将权能令牌附加在交易内并存储到区块链上,借助了区块链的防篡改性和可追溯性。

表 1 权能令牌结构

Table 1 Structure of capability token

字段名	含义
$t_{create}$	访问令牌的创建时间
$t_{end}$	访问令牌的失效时间
$A_{subject}$	访问主体账户,令牌持有者(请求访问设备)的唯一身份标识
$A_{object}$	访问客体账户,令牌颁发者(被访问设备)的唯一身份标识
$O$	主体对客体的访问权限,如可读、可写等
$C$	一组上下文感知信息,如时间或位置等

### 2.3 权能令牌管理智能合约

客体为主体创建权能令牌而实现访问控制过程中的授权。主体在访问过程中携带权能令牌作为具有相应权限的证明,客体验证权能令牌来实现访问控制过程中的鉴权。因此,权能令牌的管理是本文提出的访问控制机制的关键之一。每个终端设备在区块链上拥有一个完整账户,该账户包含一个令牌管理智能合约,而合约包含令牌创建、令牌转移和令牌撤销 3 个对外的函数接口。设备在首次加入区块链时可以为自已设置访问控制策略,完成令牌管理智能合约的创建。

在令牌管理智能合约中定义了令牌的发放策略,未获得客体访问令牌的主体可以调用该接口获取令牌

$$\text{TOKENCREATE}(A_{subject} \in B_{64}, A_{object} \in B_{64}, O \in B_1) \equiv tx_{create} \in B_{64}$$

输入参数包括请求访问的主体账户  $A_{subject}$ 、被访问客体账户  $A_{object}$  以及请求执行的操作  $O$ 。智能合约根据预先定义的策略决定是否发放令牌。若成功则创建一笔包含令牌信息的交易,交易发送方为被访问客体,交易接收方为请求访问主体。创建完成后返回该交易标识  $tx_{create}$ ,交易由共识节点打包到新区块内并附加在区块链上,此时即认为令牌已发放完成。

类似地调用令牌管理合约中的令牌撤销接口,创建一笔包含待撤销令牌信息的交易

$$\text{TOKENDESTROY}(A_{subject} \in B_{64}, A_{object} \in B_{64}) \equiv tx_{destroy} \in B_{64}$$

令牌撤销交易的发送方为被访问客体,接收方为一个固定的令牌撤销账户。若一笔交易选择该账户作为接收方,则表示此交易是一笔令牌撤销交易。

需要转移权能令牌时,原令牌持有主体可调用客体部署的令牌管理合约中的令牌转移接口函数来实现

$$\begin{aligned} \text{TOKENTRANSFER}(A_{subject} \in B_{64}, A'_{subject} \in B_{64}, A_{object} \in B_{64}) \\ \equiv (tx_{destroy} \in B_{64}, tx_{create} \in B_{64}) \end{aligned}$$

输入参数包括当前令牌持有主体账户  $A_{\text{subject}}$ 、待转移主体账户  $A'_{\text{subject}}$  以及被访问客体账户  $A_{\text{object}}$ 。令牌管理合约中预定义的策略允许转移操作, 则会创建一笔包含原令牌的令牌撤销交易  $tx_{\text{destroy}}$ , 同时为转移后的令牌持有主体创建一笔令牌生成交易  $tx_{\text{create}}$ , 从而实现了权能令牌的转移。

## 2.4 访问控制工作流程

### 2.4.1 设备注册

在多域协作的场景中, 每个域都需要保证域内节点的可靠性, 这可以通过设备备案、异常追溯等方式实现。这些方式需要对初次接入的设备进行注册, 其流程如图 2 所示。

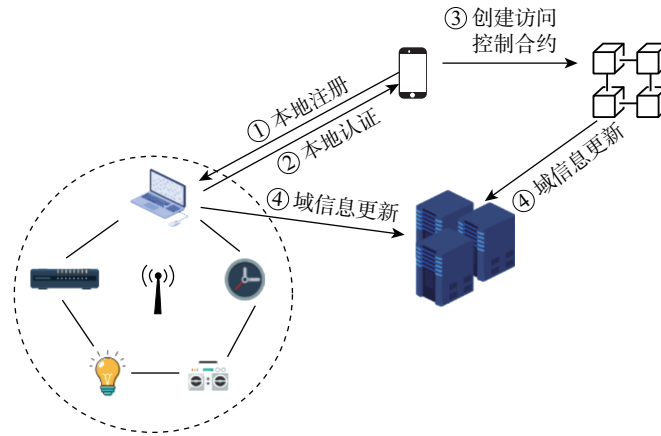


图 2 设备注册流程图

Figure 2 Device registration flowchart

#### 2.4.1.1 本地注册

待注册设备需要生成自己的账户基本信息, 包括用户公钥、用户私钥、账户标识, 再将设备类型、设备编码、账户标识发送给域管理器进行确认。

#### 2.4.1.2 本地认证

域管理器接收到本地注册请求后, 对待接入的设备信息进行验证。验证方式根据实际安全需求选择, 可以是预定义在域管理器内的接入策略或用户手动验证。验证通过后, 域管理器会通知云层节点修改该域的设备列表。

#### 2.4.1.3 令牌管理合约创建

设备完成本地注册后, 需要在区块链上为自己创建令牌管理合约。令牌管理合约需要预先制定权能令牌颁发、转移的相关策略, 并关联账户的交易创建权限。

#### 2.4.1.4 域信息更新

云层节点接收到域管理器的域信息修改请求后, 检测待加入账户是否完成令牌管理合约的创建, 若完成则修改该域的设备列表。至此, 设备完成了整个注册流程, 并成功接入到域内。

### 2.4.2 访问控制决策

在智能生活的场景中, 存在不同的域划分方式, 如根据智能设备的品牌进行划分或根据智能设备的安全级别要求进行划分。无论采取哪种划分方式, 跨域访问的情况都会频繁发生。本文中设备间的访问控制决策的流程如图 3 所示。

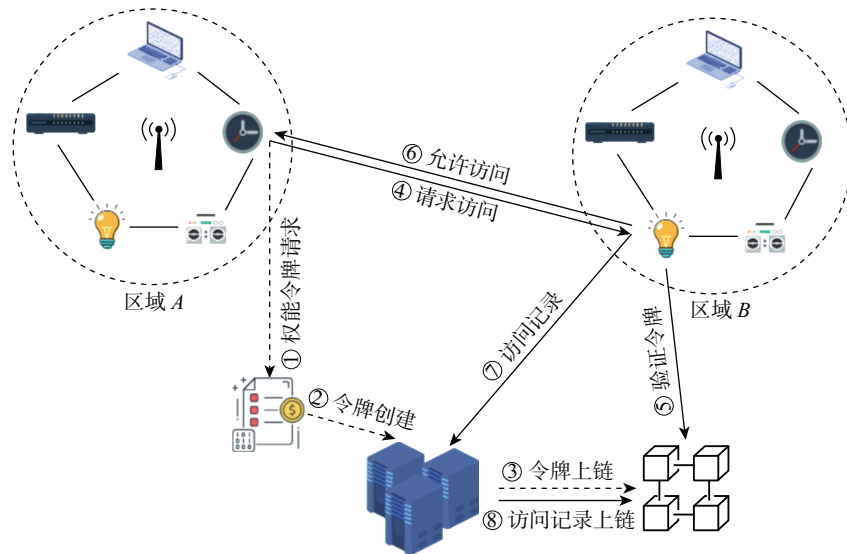


图3 设备访问控制流程图

Figure 3 Device access control flowchart

#### 2.4.2.1 权能令牌请求

当某个设备希望访问其他设备时,请求访问主体需要持有被访问设备颁发的权能令牌,若并未获取过相应令牌则需要先请求该令牌。请求主体使用自身账户调用被访问客体预先部署在区块链上的令牌管理合约申请令牌,合约根据预定义的令牌颁发策略进行决策,若成功则创建相应的权能令牌并存储在区块链上。

#### 2.4.2.2 访问请求构造

请求主体需构造访问请求,完整的访问请求结构如表2所示。

表2 访问请求结构

Table 2 Structure of access request

字段名	含义
$t_{\text{access}}$	主体发起访问请求的时间
$\text{TX}_{\text{create}}$	完整的令牌创建交易信息,用于被访问客体对令牌的验证
$O'$	请求的操作权限,如可读、可写等
sign	请求者对请求信息的签名值,用于被访问客体对请求者身份的验证

首先需要构造访问请求正文  $m$ , 包括访问时间  $t_{\text{access}}$  和包含令牌创建交易  $\text{TX}_{\text{create}}$  以及请求的操作权限  $O'$ ; 然后基于椭圆曲线  $\text{secp256k1}$  的 ECDSA 签名算法对消息正文的摘要值  $e$  进行签名。签名值 sign 用于后续过程中对主体身份及请求消息完整性的验证

$$\text{ECDSAPUBKEY}(k_{\text{private}} \in B_{32}) \equiv k_{\text{public}} \in B_{64}$$

式中,  $k_{\text{private}}$  为 32 字节的访问主体的用户私钥, 由  $k_{\text{private}}$  可以计算出其公钥  $k_{\text{public}}$ , 长度



为 64 字节。公钥用于计算访问请求的签名值

$$\text{ECDSASIGN}(e \in B_{32}, k_{\text{private}} \in B_{32}) \equiv (v \in B_1, r \in B_{32}, s \in B_{32})$$

式中,  $e$  表示访问请求的摘要值, 根据  $e$  和  $k_{\text{private}}$  可以计算出访问请求的签名值  $(v, r, s)$ , 具体过程如算法 1 所示。

**算法 1** ECDSASIGN 签名算法

**输入** 访问请求正文摘要  $e = \text{HASH}(m)$ , 私钥  $k_{\text{private}}$ , secp256k1 公共参数  $T = (p, a, b, G, n, h)$ .

**输出**  $\text{sign} = (v, r, s)$ .

**begin**

**while** true

**do**

$k = \text{random}(), 1 \leq k \leq n - 1$

$(x_1, y_1) = k k_{\text{private}}$

$r = x_1 \bmod n$

**if**  $r = 0$

**continue**

$s = k^{-1}(e + k_{\text{private}}r) \bmod n$

**if**  $s \neq 0$

**break**

**if**  $(x_1, y_1)$  为偶

$v = 27$

**else**

$v = 28$

**return**  $\text{sign} = (v, r, s)$

**end**

本文使用的签名方案不需要中心化的认证机构来保管用户公钥证书。链上公开的用户信息为经过用户公钥单向哈希计算得到的账户地址, 因此在验证签名过程中会用到公钥恢复技术。传统 ECDSA 签名算法返回的签名值只包含  $(r, s)$ , 本文中添加了恢复标识  $v$ , 它指定  $r$  为  $x$  值的椭圆曲线点坐标的奇偶性和有限性。根据恢复标识  $v$  可以确定签名时使用的唯一椭圆曲线点坐标, 从而实现公钥的恢复。

#### 2.4.2.3 访问请求验证

被访问客体接收到访问请求后, 需要先确定请求主体的身份及访问请求的完整性, 这可以通过验证签名值得以实现

$$\text{ECDSARECOVER}(e \in R_{32}, v \in B_1, r \in B_{32}, s \in B_{32}) \equiv k_{\text{public}} \in B_{64}$$

被访问客体根据访问请求消息  $m$  计算出摘要值  $e$  和签名值中的  $(r, s)$ , 推导出请求主体公钥点坐标的可能取值, 并以恢复标识  $v$  确定唯一的公钥点坐标  $k_{\text{public}}$ , 而  $k_{\text{public}}$  经过哈希运算得到的账户标识应当与请求主体一致。采用这种验证方式能够避免引入集中式 CA 机构。先恢复请求主体的公钥, 再以该公钥和消息的哈希值重新计算签名值中的  $r$  并与原签名值进行对比, 若结果一致则签名验证成功。具体验证流程如算法 2 所示。

**算法 2** ECDSARECOVER 公钥恢复算法

**输入** 访问请求正文摘要  $e = \text{HASH}(m)$ , 签名值  $\text{sign} = (v, r, s)$ , secp256k1 公共参数  $T = (p, a, b, G, n, h)$

**输出** 签名验证结果

**begin**

**if**  $r > n - 1$  或  $s > n - 1$

**return** false

**for**  $j$  从 0 到  $h$

$x = r + jn$

$X = \text{INT\_TO\_OCTET}(x)$

$R = \text{OCTET\_TO\_POINT}(X)$

**if**  $R$  为有效点

**break**

公钥对应的可能椭圆曲线坐标点  $Q = r^{-1}(sR - eG)$

根据  $v$  判断真正公钥为  $Q$  或  $-Q$

$p_u = (\text{公钥的 } x \text{ 坐标} \parallel \text{公钥的 } y \text{ 坐标})$

**if**  $\text{HASH}(k_{\text{public}})$  截取后 40 位  $\neq$  请求主体账户 addr

**return** false

$\omega = s^{-1} \bmod n$

$u_1 = e\omega \bmod n$

$u_2 = r\omega \bmod n$

$(x', y') = u_1G + u_2Q$

**if**  $x' == r$

**return** true

**else**

**return** false

**end**

#### 2.4.2.4 访问请求决策

被访问客体根据访问请求中携带的令牌创建交易信息查询区块链, 若成功找到一致的令牌创建交易且交易中的权能令牌包含当前请求的权限, 则允许本次访问并返回相应内容。每次访问请求一旦处理完成, 被访问客体就将访问事件记录到一笔交易中, 并由共识节点将交易打包到新区块内。对访问事件的记录有利于发现并处理存在异常行为的节点。

#### 2.4.3 异常访问追溯

基于权能的访问控制机制将访问请求的验证工作交给具有一定计算能力的被请求者执行, 防止集中式服务器验证带来的单节点故障问题影响整个系统。然而, 在本文所处的多域物联网协作场景中, 网络的复杂程度较高, 设备的抗攻击能力普遍低于企业级服务器的抗攻击能力, 可见发现并处理系统中存在异常行为的节点是很必要的。部署在云层的共识节点和候选节点可以对记录在区块链上的访问日志进行监测, 如图 4 所示。区域  $A$  中的某设备并未获得相应权限的权能令牌, 通过伪造令牌或提供不具有相应权限的令牌申请访问区域  $B$  中的设备, 每次处理访问请求时被访问设备都会创建访问记录交易。当共识节点发现  $A$  区域中的该设备存在超出可接受阈值的异常行为时, 将该异常设备从域设备列表中删除, 并通报给异常

节点所在域的域管理器。同时, 从区块链中查询所有该设备的权能令牌, 调用各个令牌管理合约的令牌撤销接口冻结该异常节点的所有访问令牌。

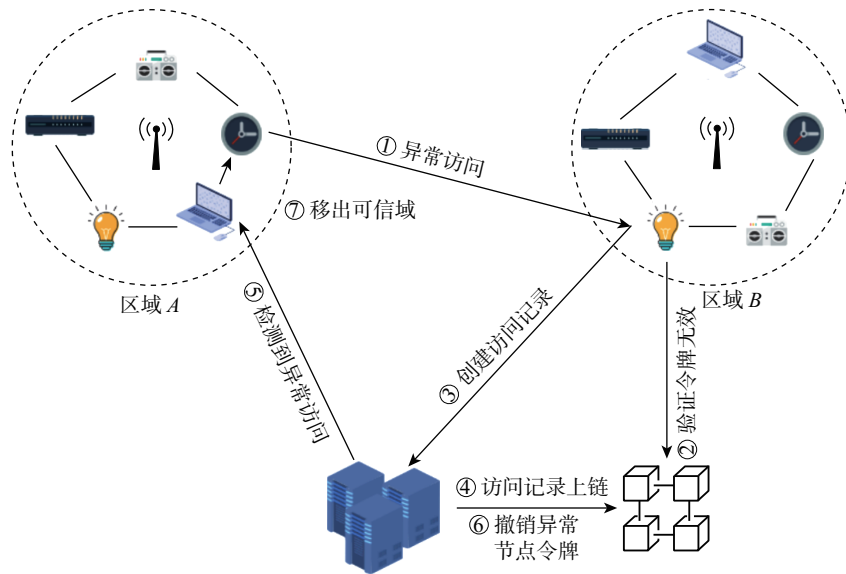


图4 异常访问追溯流程图

Figure 4 Abnormal access trace flowchart

### 3 节点轻量化

在物联网应用场景中, 设备性能相差较大, 而区块链全节点需要存储完整的区块链账本。对于大多数物联网设备来说, 存储完整账本的负担过大, 于是本文引入 SPV 节点和超轻节点, 以适配物联网场景中复杂多样的异构设备, 增强系统的可扩展性。

#### 3.1 SPV 节点

SPV 节点只需存储区块链中每个区块头的内容即可实现对交易的验证。本文参考比特币等数字货币系统中的 SPV 机制, 设计了针对权能令牌的 SPV 机制。在数字货币系统中, SPV 主要用于数字货币支付交易的验证。本文的 SPV 主要用于验证以交易形式存储在区块链上的权能令牌, 其具体流程主要包括以下 2 个阶段: 验证权能令牌是否有效、验证权能令牌是否撤销。

##### 3.1.1 验证权能令牌是否有效

**步骤 1** SPV 节点计算访问请求中附带的令牌创建交易的哈希值, 并发送给区域中的全节点查询默克尔树哈希认证路径。

**步骤 2** 接收到请求的全节点根据该交易哈希值定位到区块, 并返回给 SPV 节点该交易的默克尔树哈希认证路径。

**步骤 3** SPV 节点根据认证路径计算默克尔树根哈希值, 若能从区块头中搜索到该哈希值, 则该交易验证成功, 证明令牌存在于区块链上。

### 3.1.2 验证权能令牌是否撤销

**步骤 1** SPV 节点请求域中的其他全节点搜索包含该令牌的撤销交易, 搜索范围限定在令牌创建交易所在区块及后续所有区块中, 创建者为该 SPV 节点的令牌撤销交易。

**步骤 2** 若未查找到交易则证明令牌依旧生效, 本次访问成功。若查找到交易则返回该交易所在区块的默克尔树哈希认证路径。

**步骤 3** SPV 节点需要 2 次验证, 使用认证路径计算默克尔树根哈希值并定位到区块头, 若成功通过则证明令牌已被撤销, 本次访问不成功。

SPV 机制将区块查询工作交给具备较高性能的全节点完成, 凭借默克尔树结构降低了节点需要存储的区块信息大小, 这样 SPV 节点不必完全信任全节点, 从而实现了简易验证。需要注意的是, SPV 节点随机连接多个全节点, 有利于增加至少连接一个可靠节点的概率。

## 3.2 超轻节点

本文基于支持自定义账户权限和多签名机制, 设计并实现了比 SPV 节点硬件需求更低的超轻节点。超轻节点不存储区块信息, 而只维护一个本地令牌列表。在本地令牌列表中, 存储本节点颁发访问令牌哈希值, 并定期从选择的代理全节点处获取区块链中存储的令牌信息, 更新维护本地令牌列表。每次验证访问令牌时, 首先需要查询本地令牌列表, 若未查询到则进一步通过代理全节点验证。当代理验证成功时, 更新本地令牌列表。

每个超轻节点账户拥有 owner 和 active 两种权限私钥。

### 1) Owner 私钥

该私钥控制账户所有权, 由轻节点自己持有。当轻节点接收到访问请求时, owner 私钥可用于创建访问记录交易。当超轻节点需要切换代理全节点时, owner 私钥可用于管理账户权限, 删除与 active 权限账户的绑定关系。

### 2) Active 私钥

Active 私钥由超轻节点指定的代理全节点持有。当超轻节点接收到访问请求且本地缓存令牌列表未匹配到该权能令牌时, 由代理全节点验证令牌并且使用 active 私钥创建访问记录交易。

## 4 实验分析

本文实验部分应用的区块链系统是基于开源项目 EOS 的二次开发的。如表 3 所示, EOS 采用了 DPoS+BFT 的共识算法, 相比于 Bitcoin 和 Ethereum 采用的 PoW 共识算法, 该共识算法牺牲了部分去中心化程度, 以换取更高的交易吞吐量。相比于 Fabric 2.0 版本中基于故障容错 (crash fault tolerance, CFT) 的 RAFT 共识算法, 该共识算法能够解决拜占庭容错问

表 3 区块链解决方案对比

Table 3 Comparison of blockchain solutions

方案	Bitcoin	Ethereum	EOS	Hyperledger Fabric
共识算法	PoW	PoW	DPoS+BFT	RAFT
智能合约	Script(非图灵完备)	EVM	WASM	ChainCode
SPV 机制	支持	支持	支持	不支持
自定义账户权限	不支持	不支持	支持	支持

题。除 Bitcoin 只支持非图灵完备的交易脚本外, 其他方案都支持智能合约。EOS 区块链也内置了智能合约, 通过 Web Assembly (WASM) 执行开发者提供的合约代码, 使用 clang/llvm 及其 C/C++ 编译器来编译 WASM 代码, 从而构建应用程序。在节点轻量化方面, EOS 既支持 SPV 机制也支持自定义账户和多签名机制, 便于实际搭建 SPV 节点和超轻节点, 于是本文选择了基于 EOS 区块链进行代码实现。

本实验使用 2 台笔记本电脑和 1 台 PC 主机作为系统的云层节点, 配置如表 4 所示。云层节点负责区块链共识和域设备列表维护, 域设备列表通过 EOS 内置的 chainbase 内存数据库存储。终端设备使用 3 种不同配置的 ubuntu 18.04 虚拟机代替, 超轻节点中的本地令牌列表使用轻量的 SQLite 数据库存储。

表 4 节点配置

Table 4 Node configuration

设备名	配置
Laptop	2.6 GHz Intel Core i7 CPU, 16GB DDR4 RAM, Ubuntu 18.04
PC	3.6 GHz AMD Ryzen 7 CPU, 32 GB DDR4 RAM, Ubuntu 18.04
IoT-device-1	4 Core CPU, 8 GB RAM, Ubuntu 18.04
IoT-device-2	2 Core CPU, 4 GB RAM, Ubuntu 18.04
IoT-device-3	1 Core CPU, 1 GB RAM, Ubuntu 18.04

在多域物联网设备协作的场景下, 模拟了一个物联网终端设备从注册到访问请求通过的全部操作, 每次完成访问请求, 共识节点就对该设备执行异常访问追溯的全部操作。按上述流程重复 50 次并取平均值, 设备注册阶段的平均时延为 112 ms, 权能令牌请求的平均时延为 37 ms, 访问请求和访问决策的平均时延为 31 ms, 异常追溯的平均时延为 45 ms。

图 5 对比了基于 RBAC 和 ABAC 实现的访问控制方案与本文访问控制方案的访问请求处理时延。在本地模拟了简易的 RBAC、ABAC 访问控制系统, 设定系统中每个设备均作为客体接收并处理 10 次访问请求, 则 50 个节点会产生 500 条访问请求。随着网络中设备规模的扩大, 所需处理的访问请求也不断增加。从图 5 中可以看到 ABAC 方案的处理时延最小,

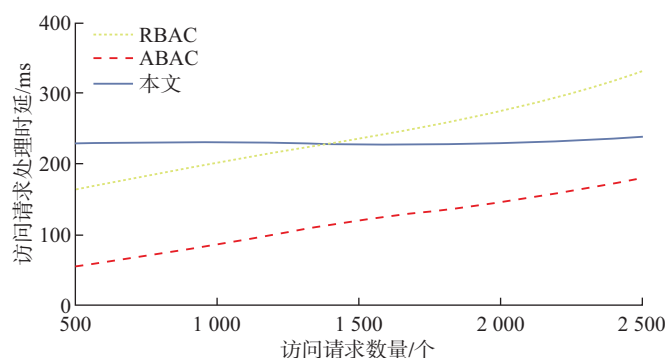


图 5 访问请求处理时延对比

Figure 5 Comparison of access request processing delay

这是因为该方案采用主、客体属性进行决策且属性表规模较小,所以能够更快地查询并决策。本文方案使用了区块链技术,与传统关系型数据库相比查询效率偏低,故单条访问请求的处理时延较大。不同于 RBAC、ABAC 由统一的服务器单点决策,本文将访问控制决策者设置为被访问客体,故各个客体能并行处理访问请求。随着网络中设备规模的增大,访问请求处理时延能够基本保持稳定。由此可以看出,本文提出的方案具有较强的可扩展性,能够适应物联网海量节点的特性。

在实验过程中,全节点产生了大小为 358 MB 的区块链账本文件,而 SPV 节点的账本大小为 67 MB,大大降低了所需的存储空间。超轻节点的本地令牌列表大小仅由授权的主体数量决定,并不会持续增长。若授权主体数量非常庞大导致超轻节点本地令牌列表存储空间不足,则可以通过最近最少使用的 (least recently used, LRU) 方式覆盖其他记录。

## 5 结 语

本文提出了融合区块链技术和权能令牌的访问控制机制,以解决物联网多域协作场景下设备间的访问控制问题。将访问控制的决策交给有一定计算能力的物联网设备处理,相比于集中式的访问控制决策,该方式在物联网的海量节点规模下具有更强的可扩展性。智能合约的执行效率相对较低,于是本文只将智能合约用于权能令牌的管理,而最为频繁的令牌验证操作则通过读取区块链和验证数字签名实现,减少了访问请求处理时间。引入区块链中的 SPV 节点和超轻节点,减轻了低功耗设备的存储负担,以适配物联网场景中复杂多样的异构设备。综上所述,本文利用了物联网海量设备、节点异构、可扩展等特性,能够适用于多域物联网设备协作的场景。

## 参考文献:

- [1] SANDHU R S, COYNE E J, FEINSTEIN H L, et al. Role-based access control models [J]. *Computer*, 1996, 29(2): 38-47.
- [2] DE SOUZA L M S, SPIESS P, GUINARD D, et al. Socrates: a Web service based shop floor integration infrastructure [M]//*The Internet of Things*. Heidelberg, Berlin: Springer, 2008: 50-67.
- [3] SPIESS P, KARNOUSKOS S, GUINARD D, et al. SOA-based integration of the Internet of things in enterprise services [C]//*IEEE International Conference on Web Services*, 2009: 968-975.
- [4] ZHANG G, TIAN J. An extended role based access control model for the Internet of things [C]//*International Conference on Information, Networking and Automation*, 2010, 1: 319-323.
- [5] SMARI W W, CLEMENTE P, LALANDE J F. An extended attribute based access control model with trust and privacy: application to a collaborative crisis management system [J]. *Future Generation Computer Systems*, 2014, 31: 147-168.
- [6] YUAN E, TONG J. Attributed based access control (ABAC) for Web services [C]//*IEEE International Conference on Web Services*, 2006: 74-79.
- [7] NING Y E, ZHU Y, WANG R C, et al. An efficient authentication and access control scheme for perception layer of Internet of things [J]. *Applied Mathematics & Information Sciences*, 2014, 8(4): 1-8.
- [8] MAHALLE P N, ANGGOROJATI B, PRASAD N R, et al. Identity authentication and capability based access control for the Internet of things [J]. *Journal of Cyber Security and Mobility*, 2013, 1(4): 309-348.
- [9] 沈海波, 刘少波. 面向物联网的基于上下文和权能的访问控制架构 [J]. *武汉大学学报 (理学版)*, 2014, 60(5): 424-428.  
SHEN H B, LIU S B. A context-aware capability-based access control framework for the Internet of things [J]. *Journal of Wuhan University (Natural Science Edition)*, 2014, 60(5): 424-428. (in Chinese)

- [10] HERNÁNDEZ-RAMOS J L, JARA A J, MARIN L, et al. Distributed capability-based access control for the Internet of things [J]. *Journal of Internet Services and Information Security*, 2013, 3(3/4): 1-16.
- [11] GUSMEROLI S, PICCIONE S, ROTONDI D. A capability-based security approach to manage access control in the Internet of things [J]. *Mathematical and Computer Modelling*, 2013, 58(5/6): 1189-1205.
- [12] ANGGOROJATI B, MAHALLE P N, PRASAD N R, et al. Capability-based access control delegation model on the federated IoT network [C]//The 15th International Symposium on Wireless Personal Multimedia Communications, IEEE, 2012: 604-608.
- [13] ZHANG Y, KASAHARA S, SHEN Y, et al. Smart contract-based access control for the Internet of things [J]. *IEEE Internet of Things Journal*, 2018, 6(2): 1594-1605.
- [14] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system [R/OL]. 2009[2020-10-15]. <https://bitcoin.org/bitcoin.pdf>.
- [15] BACK A. Hashcash-a denial of service counter-measure [OL]. 2002 [2020-10-15]. [https://www.researchgate.net/publication/2482110\\_Hashcash\\_-\\_A\\_Denial\\_of\\_Service\\_Counter-Measure](https://www.researchgate.net/publication/2482110_Hashcash_-_A_Denial_of_Service_Counter-Measure).
- [16] WOOD G. Ethereum: a secure decentralised generalised transaction ledger [J]. *Ethereum Project Yellow Paper*, 2014, 151: 1-32.
- [17] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger Fabric: a distributed operating system for permissioned blockchains [C]//Proceedings of the Thirteenth EuroSys Conference, 2018: 1-15.
- [18] 袁勇, 倪晓春, 曾帅, 等. 区块链共识算法的发展现状与展望 [J]. *自动化学报*, 2018, 44(11): 2011-2022.  
YUAN Y, NI X C, ZENG S, et al. Blockchain consensus algorithms: the state of the art and future trends [J]. *Acta Automatica Sinica*, 2008, 44(11): 2011-2022. (in Chinese)
- [19] AITZHAN N Z, SVETINOVIC D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams [J]. *IEEE Transactions on Dependable and Secure Computing*, 2016, 15(5): 840-852.
- [20] FENG Q, HE D, ZHADALLY S, et al. A survey on privacy protection in blockchain system [J]. *Journal of Network and Computer Applications*, 2019, 126: 45-58.
- [21] HERLIHY M. Atomic cross-chain swaps [C]//Proceedings of 2018 ACM Symposium on Principles of Distributed Computing, 2018: 245-254.
- [22] SPANOS N, MARTIN A R, DIXON E T, et al. System and method for creating a multi-branched blockchain with configurable protocol rules: U.S. Patent 9608829 [P]. 2017-03-28.
- [23] ZAMANI M, MOVAHEDI M, RAYKOVA M. Rapidchain: scaling blockchain via full sharding [C]//Proceedings of 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018: 931-948.
- [24] DANG H, DINH T T A, LOGHIN D, et al. Towards scaling blockchain systems via sharding [C]//Proceedings of 2019 ACM International Conference on Management of Data, 2019: 123-140.
- [25] BENET J. IPFS-content addressed, versioned, P2P file system [OL]. [2020-10-15]. <https://arxiv.org/abs/1407.3561>.
- [26] GILAD Y, HEMO R, MICALI S, et al. Algorand: scaling Byzantine agreements for cryptocurrencies [C]//Proceedings of the 26th ACM Symposium on Operating Systems Principles, 2017: 51-68.
- [27] KALRA S, GOEL S, DHAWAN M, et al. ZEUS: analyzing safety of smart contracts [C]//Network and Distributed System Security Symposium, 2018.
- [28] CEBE M, ERDIN E, AKKAYA K, et al. Block4forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles [J]. *IEEE Communications Magazine*, 2018, 56(10): 50-57.
- [29] DORRI A, KANHERE S S, JURDAK R, et al. LSB: a lightweight scalable blockchain for IoT security and anonymity [J]. *Journal of Parallel and Distributed Computing*, 2019, 134: 180-197.
- [30] LIU Y, WANG K, LIN Y, et al. LightChain: a lightweight blockchain system for industrial Internet of things [J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(6): 3571-3581.

(编辑: 秦 巍)