

DOI:10.16644/j.cnki.cn33-1094/tp.2022.07.003

# 边缘计算中基于区块链的访问控制机制

罗金喜<sup>1</sup>, 王 璞<sup>2</sup>

(南京邮电大学通信与信息工程学院, 江苏 南京 210003)

**摘要:** 针对边缘计算中数据共享的信任关系难建立、隐私难保证和中心框架的单点故障等问题,提出一种基于区块链的访问控制机制。利用区块链的智能合约来管理访问权限和审计数据,依赖于区块链的去中心化和不可篡改特性解决单点故障和节点信任问题。为了保证数据的隐私性,上传到第三方服务器上的数据采用AES-128算法加密,而解密密钥则通过SGX技术构建安全程序进行共享。性能分析表明,该方案满足扩展性要求,能进行大型数据共享。

**关键词:** 边缘计算; 区块链; 访问控制; AES-128算法; SGX技术

中图分类号:TP309

文献标识码:A

文章编号:1006-8228(2022)07-12-05

## Blockchain-based access control mechanism in edge computing

Luo Jinxi<sup>1</sup>, Wang Jun<sup>2</sup>

(School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210000, China)

**Abstract:** In order to solve the problems of establishing trust relationship, ensuring privacy and single point of failure of the central framework in edge computing, a blockchain-based access control scheme is proposed. Using smart contracts of the blockchain, access rights are managed and data is audited. Relying on the decentralized and immutable nature of the blockchain, single point of failure and node trust problems are resolved. To ensure data privacy, data uploaded to third-party servers is encrypted with the AES-128 algorithm, and the decryption keys are shared through the SGX technology to build a security program. Performance analysis shows that the solution meets the scalability requirements and can perform large data sharing.

**Key words:** edge computing; blockchain; access control; AES-128 algorithm; SGX technology

## 0 引言

边缘计算<sup>[1]</sup>作为5G时代的核心技术之一,通过将终端设备的数据放入边缘服务器以进行存储或计算,可以解决终端存储容量有限、计算能力不足的问题。边缘计算系统中的边缘服务器节点是众多终端的汇聚点,这些终端包括智能移动终端、传感器、摄像头等物联网设备。若将终端收集、产生或存储的一些数据共享,那么数据资源将得到最大化利用。

然而边缘网络中不同个体或机构间缺乏合作的保障和信任关系,难以形成安全可靠的数据共享局面,因此急需一个适用于边缘网络的访问控制机制。传统的访问控制策略包括基于角色的访问控制(Role-Based Access Control, RBAC)<sup>[2]</sup>、基于属性的访问控制

(Attribute Based Access Control, ABAC)和基于风险的访问控制(Risk Based Access Control, Risk-BAC)<sup>[3]</sup>。上述访问控制方案均通过一个权威的中心机构来验证访问权,因此面临着中心化所带来的问题,即可能发生单点故障。而且访问控制有三个重要的安全要素:认证、授权和审计,如果由一个中心服务器来提供访问控制,那么恶意的中心服务器可能会认证、授权不具有相关权限的用户,从而导致数据泄露问题。审计负责记录系统中所有已发生的访问控制事件,如果中心服务器拥有对审计的完全控制权,则很难防范恶意的服务器篡改审计数据。

为了解决上述问题,本文利用区块链上的智能合约来管理访问权限和审计数据,依赖于区块链的去中

收稿日期:2021-11-22

作者简介:罗金喜(1996-),男,江西人,南京邮电大学硕士研究生,主要研究方向:区块链技术。

通讯作者:王璐(1975-),女,江苏人,博士,副教授,硕士生导师,主要研究方向:下一代网络技术。

心化和不可篡改特性解决单点故障和节点信任问题。为了保证数据的隐私性,上传到第三方服务器上的共享数据采用AES-128算法加密,而解密密钥则通过SGX(Software Guard Extensions)<sup>[5]</sup>技术构建安全程序进行共享。

## 1 系统模型和安全模型

### 1.1 系统模型

如图1所示,本文所考虑的边缘计算系统由云服务器、边缘服务器和众多终端组成。由于本文方案的底层区块链网路为以太坊<sup>[5]</sup>,而以太坊是一种公有区块链,任何节点可以随时加入和退出网络,因此边缘计算系统中的任意设备均可成为区块链节点。本文根据上述各元素在访问控制中的作用,将其分为以下角色:数据所有者、第三方服务器、区块链网络和数据请求者。

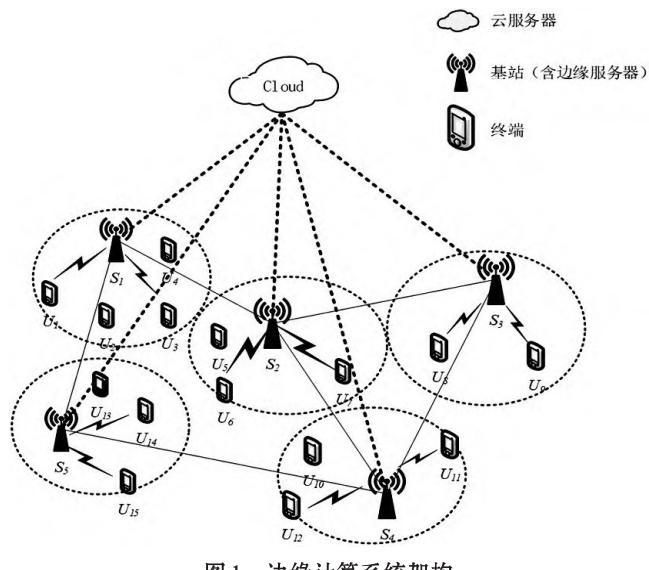


图1 边缘计算系统架构

**数据所有者:**共享数据的拥有者,可以是网络中的任意节点。

**第三方服务器:**主要由边缘服务器和云服务器组成,为数据共享提供存储和计算资源,本方案的存储模式也是分布式的,数据所有者可以自由选择系统中注册的服务器作为存储数据的第三方服务器。

**区块链网络:**包含交易和智能合约的区块链,由边缘网络中所有设备共同维护,任意节点可根据业务需要灵活地加入。

**数据请求者:**是共享数据的请求者。

### 1.2 安全模型

本文采用了文献[6]中定义的访问控制列表模型来

实现权限管理。访问控制列表(Access Control List, ACL)模型是一种以对象为中心的方法,为每个对象O定义了一个列表L,即O的访问控制权,这个列表列举了所有拥有O的访问权限的主体,这个列表还指定了授予主体的访问权限(例如,读和写)。为了将ACL映射到我们的方案,对于每个对象O,也就是存储在第三方服务器上的数据,本文定义了如表1所示的列表,记录主体(数据请求者)所拥有的权限的相关信息。这个列表存储在区块链中。

表1 访问控制列表

Subject	operation	permission	Grant-time
user1	read	0X52d65…	2021-07-01
user2	write	0	2021-07-01
user3	write	0X4a515k…	2021-07-05

## 2 一种基于区块链的访问控制框架

本文基于区块链技术设计了一种适用于边缘计算系统的访问控制框架。该框架通过访问控制列表ACL模型实现访问控制,而此模型的权限列表则由区块链上的智能合约管理,因此可以解决传统访问控制中的集中化问题。

区块链上资源宝贵,因此共享数据需存储到第三方服务器,为了防止第三方服务器泄漏数据,本方案将数据的共享分割为两部分,即加密数据和解密密钥的共享方式并不相同。第三方服务器通过执行以太坊合约验证请求者访问权限,根据返回结果决定加密数据的共享,而解密密钥则通过SGX技术进行管理。

### 2.1 智能合约系统框架

本文提出的智能合约框架包含三种合约,ACC合约、ACP合约和AUD合约。

#### 2.1.1 ACC合约

对于数据所有者而言,该合约是ACP合约的生产工厂,即ACP合约只能通过调用ACC合约提供的createACP函数进行部署。而数据请求者则可以通过数据对象的唯一id索引所需的相关信息,包括ACP合约地址、原始数据hash摘要(数据请求者用于验证解密后数据的正确性)、密钥分发节点、第三方服务器等。该合约主要包含以下几个函数。

- createACP:该函数接受数据对象的相关参数,并在区块链上部署一个ACP合约,同时把相应的信息存入此合约维护的列表中。

- `retrieveObject`: 该函数接受数据对象 ID, 返回获取数据所需的相关信息。

### 2.1.2 ACP 合约

该合约用于管理数据对象的访问权限。每个 ACP 合约中都维护着一个列表, 结构如表 1 所示。该列表记录了其他终端用户的访问权限, 其中许可字段存储的是数字签名(用于在 Enclave 程序中验证请求者对对称密钥的获取权限), 第三方服务器通过判断返回的许可是否为零来决定用户的访问权。

ACP 合约所有者可以对访问控制列表的内容进行增删改查的操作; 第三方服务器可以通过交易发起验证用户权限的请求, 相应的访问日志会被记录在 AUD 合约中。合约所有者地址、第三方服务器地址等属性由合约的构造函数进行初始化。ACP 主要提供了以下函数来管理访问控制。

- `addRight`: 该函数接收新的访问控制策略信息, 并将此项控制策略添加到访问控制列表中。
- `updateRight`: 该函数接收需要更新的策略信息, 并更新访问控制列表中的策略。
- `validateRight`: 该函数接收访问控制所需的信息, 返回访问结果, 同时调用 AUD 合约中的 `setLog` 函数, 将此次访问信息存储到日志中。

### 2.1.3 AUD 合约

该合约由 ACP 合约的构造函数部署到区块链上。AUD 合约维护一个审计所需信息的列表, 此列表是为了将数据使用情况存储到区块链上进行审计。该合约包含以下几个函数。

- `accessLog`: 该函数接收数据访问者公钥, 并返回该访问者的历史访问日志。
- `setLog`: 该函数接收访问日志信息, 并将此日志信息的情况存入 AUD 合约维护的审计列表中。

## 2.2 接入控制

整个接入控制过程可分为共享数据上传和共享数据下载两部分。

### 2.2.1 共享数据上传

首先数据所有者通过交易执行区块链上的 ACC 合约中的 `createACP` 函数, 将共享数据注册, 并部署管理该共享数据访问控制的 ACP 合约, ACP 合约初始化时也会部署一个记录日志信息的 AUD 合约, 最后将返回 ACP 合约地址给数据所有者。

为了保障数据的隐私性, 本方案中在将数据上传

到第三方服务器前, 首先需要对其进行加密。为了降低加密、解密数据所消耗的计算资源, 本文采用对称加密技术, 即数据所有者随机选择一个 128 位的加密密钥, 并使用 AES-128 算法对数据进行加密。然后, 数据所有者将加密的数据与 ACP 合约地址一起发送给第三方服务器。发送 ACP 合约的地址是为了让第三方服务器调用并执行该合约的 `validateRight` 函数, 以验证数据请求者的访问权限。同时第三方服务器还要获得 ACP 合约的所有者公钥, 以此验证数据所有者的签名。如果签名有效, 第三方服务器将把已加密的数据存储到他们的设备中。

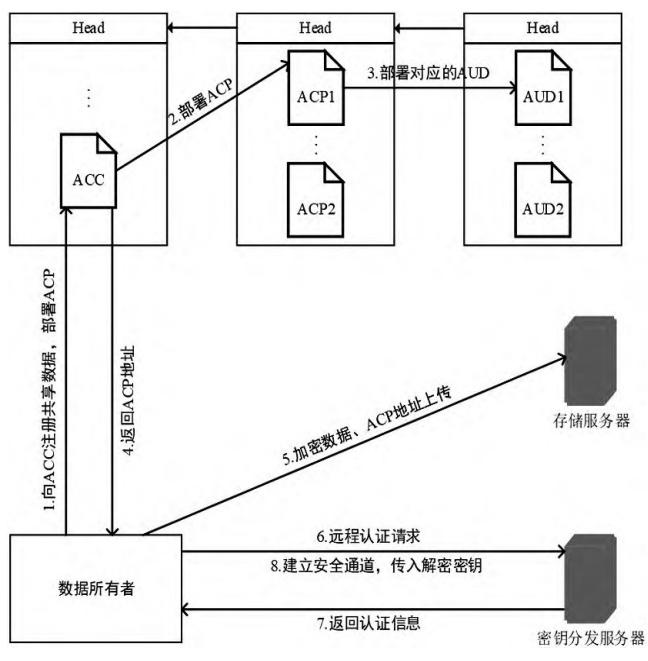


图 2 数据上传流程

数据成功上传到第三方存储服务器后, 数据所有者通过英特尔认证服务(Intel Authentication Service, IAS)验证密钥分发节点中实例化的 Enclave 的完整性, 包括内部数据和代码。此外, 验证报告还包含 Enclave 产生的公钥, 数据所有者用该公钥加密需分发的对称密钥, 从而保证其安全地传入了 Enclave 中。为了保证系统的鲁棒性, 应该有多个密钥分发节点, 以确保在任何时间内系统都能正常工作。

### 2.2.2 共享数据下载

当数据请求者需要访问某项数据时, 为了保证数据的安全性, 首先调用 ACC 合约中的 `retrieveObject` 函数, 根据数据唯一 ID 获得实现访问所需的相关信息, 如第三方服务器地址、密钥分发节点和原始数据 hash 摘要。

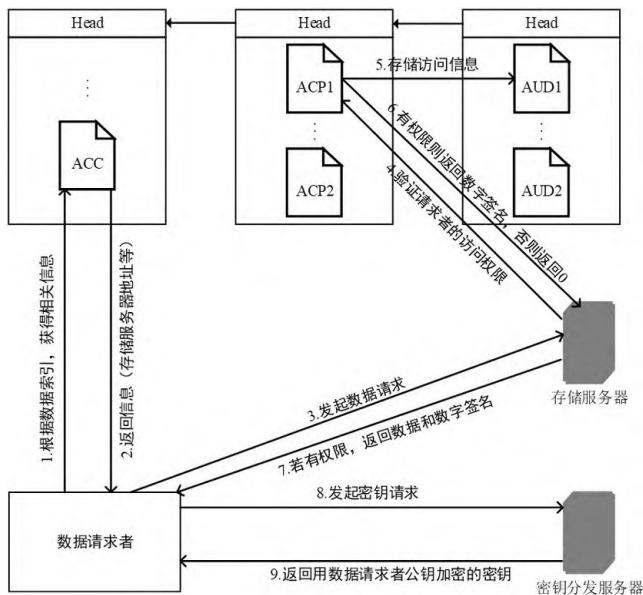


图3 数据下载流程

数据请求者随后向第三方服务器发送一个请求，该请求包含所需数据对象 ID、自身公钥以及对访问所需的相关数据的签名(数字签名使用 ECDSA)。第三方服务器可以通过提供的公钥验证用户的身份，如果签名是有效的，那么将继续验证用户的访问权限。它向区块链发送一个交易，以执行 ACP 合约 validateRight 函数来验证访问权，如果返回的是非零的数字签名，则表明用户有权访问数据，那么就会向用户提供加密的数据和返回的数字签名，此数字签名是数据所有者对预定义消息(根据消息结构生成)的签名，该消息结构如表2所示。

表2 预定义消息结构

字段	含义
$PK_{do}$	数据所有者公钥
$PK_{dr}$	数据请求者公钥
$T$	截止时间
$ID$	请求数据的 ID

为了解密第三方服务器传来的数据，数据请求者向密钥分发节点发起密钥请求，该请求包含上述数字签名  $\sigma$  和预定义的消息  $m$ 。密钥分发节点中的 Enclave 程序首先通过数字签名  $\sigma$  验证消息  $m$  的完整性，如果  $m$  有效，且当前的系统时间小于截止时间，则证明该用户拥有访问权。然后根据  $m$  中的请求数据的  $ID$  索引数据所有者存入的解密密钥，并用数据请求者的公钥  $PK_{dr}$  对其进行加密，将结果返回数据请求者，Enclave 程序的具体流程如图 4 所示。最终数据请求者通过自己的私钥解密传回的结果获得解密密钥。

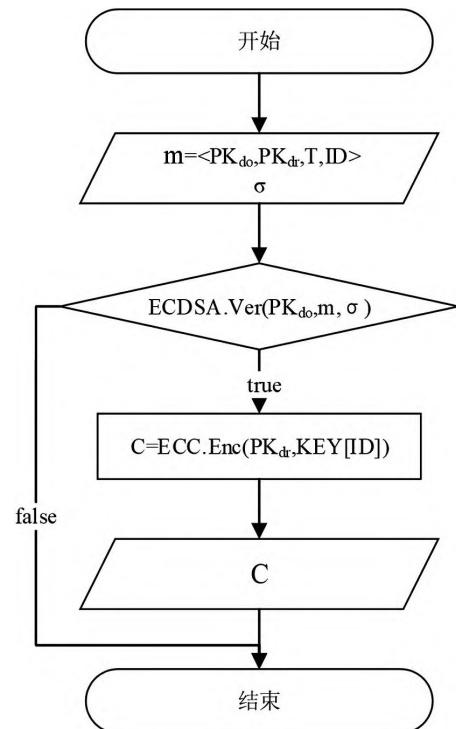


图4 Enclave程序流程图

### 3 性能分析

此方案的性能高度依赖于底层区块链平台，区块链系统的 hash 算力、网络架构、智能合约的复杂性、任务数量、区块生成率和网络带宽都将极大地影响了此方案的性能。

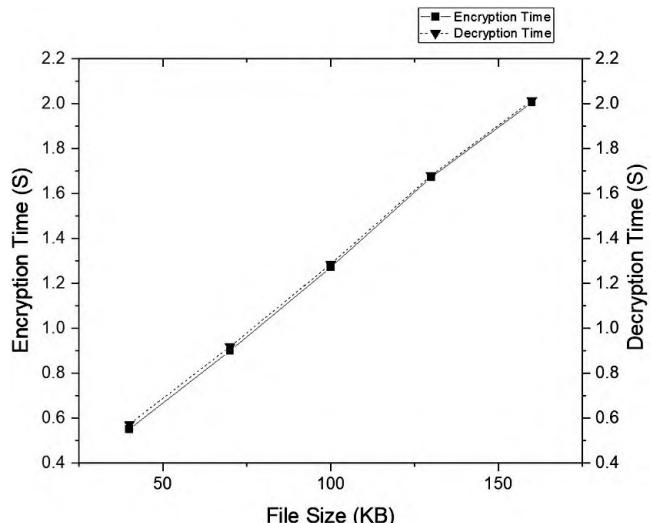


图5 加/解密开销与文件大小的关系

因此本节主要对初始化阶段和执行阶段的额外开销进行扩展性分析。在初始化阶段，所有的额外开销和以下操作有关系：智能合约被创建/部署到区块链、对 Enclave 进行远程认证并传入解密密钥和用

AES-128 算法对原始数据进行加密。其中前两个操作的时间花销与共享数据的大小无关,在一个固定值附近波动,而最后一个操作与共享数据的大小有着紧密的关系,本文在个人电脑上使用 AES-128 算法进行了性能测试,结果如图 5 所示,加密时间随着共享数据的大小增加而线性增加,满足扩展性要求,而每一份共享的数据对象只需要初始化一次,因此本文提出的方案的初始化开销是可以接受的。

在执行阶段,所有的额外开销由访问权验证交易、从 Enclave 中获取解密密钥和用对称密钥解密数据的时间组成。其中,交易完成时间由区块链的区块生成时间决定,又解密密钥固定为 128 位,即正常网络情况下密钥请求时间基本固定,最终的数据解密时间与文件的大小关系为图 5 所示的线性关系,满足扩展性要求。综上,相对于现有的集中式数据共享平台,本方案虽然会有性能上的损失,但总体时间成本可控,能够满足大型数据共享的要求。

#### 4 结束语

本文针对边缘计算网络中数据共享的隐私、信任和单点故障的挑战,提出了一个基于智能合约的方案,该方案可以为边缘计算系统提供一个去中心化的访问控制机制。而且根据上述的性能分析可知,相对

于传统的数据共享系统,本文提出的方案虽然在性能上有一定损失,但安全性却大大增加,能抵抗恶意的服务提供商攻击和中间人攻击。本方案的性能高度依赖于底层区块链平台,因此,下一步重点研究的内容是探索适合该方案框架的区块链扩容技术,进一步提高方案的实用性。

#### 参考文献(References):

- [1] Marjanović M, Antonić A, Žarko I P. Edge computing architecture for mobile crowdsensing[J]. IEEE Access, 2018, 6: 10662–10674
- [2] Cruz J P, Kaji Y, Yanai N. RBAC-SC: Role-based access control using smart contract[J]. IEEE Access, 2018, 6: 12240–12251
- [3] 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
- [4] 王鹏,樊成阳,程越强,等. SGX 技术的分析和研究[J]. 软件学报, 2018, 29(9): 2778–2798
- [5] Wood G. Ethereum: A secure decentralised generalised transaction ledger[J]. Ethereum project yellow paper, 2014, 151(2014): 1–32
- [6] Goodrich M T, Tamassia R. Introduction to computer security[M]. London, UK: Pearson, 2011.



(上接第 11 页)

浙江工业大学, 2010

- [3] 王胜鹏, 史凡钦等. 基于图像识别的电梯群控算法研究[J]. 科学技术创新, 2021, 4: 86–90
- [4] 钟永祥. 面向新型建筑智能化平台的电梯群控系统调度方法研究[D]. 合肥: 安徽建筑大学, 2021
- [5] 闫秀英, 郭普静, 范凯兴. 基于 ABC-SA 混合算法的群控电梯优化调度[J]. 计算机测量与控制, 2020, 28(8): 107–111
- [6] 卞晓晨. 基于蚁群算法优化的电梯群控研究[D]. 沈阳: 沈阳理工大学, 2019
- [7] 蔡奇志, 苗莹霞, 等. 基于粒子群优化神经网络的电梯群控算法[J]. 国外电子测量技术, 2019, 38(5): 114–119
- [8] Wang Shu, Fei Cindy Y., Chen Jerred. Smart dispatching and optimal elevator group control through real-time occupancy-aware deep learning of usage patterns[J]. Advanced Engineering Informatics, 2021, Volume 48: 421–427
- [9] Shunji Tanaka, Daiki Hoshino, et al. Group control of

multi-car elevator systems without accurate information of floor stoppage time. Flexible Services and Manufacturing Journal, 2016, 28(3): 461–494

- [10] Emre Oner Tartan; Cebraii Ciftlikli. A Genetic Algorithm Based Elevator Dispatching Method For Waiting Time Optimization[J]. IFAC-Papers OnLine, 2016, 49(3): 424–429
- [11] 魏君燕. 基于目的层调度的新型电梯群控系统的研究[D]. 杭州: 浙江工业大学, 2013
- [12] 张扬. 改进群体智能优化算法及其应用[D]. 南京: 南京邮电大学, 2020
- [13] 赵伟, 王志磊, 李晓理, 等. 基于客流分析的电梯群控仿真系统研究[J]. 控制工程, 2015, 22(5): 826–830
- [14] Bao Ding, Yong-Ming Zhang, et al. A hybrid approach for the analysis and prediction of elevator office building[J]. Automation in Construction, 2013, 35: 69–78

