Journal of Computer Applications
计算机应用, 2022, 42(7): 2104 - 2111
ISSN 1001-9081
CODEN JYIIDU
2022-07-10
http://www.joca.cn

文章编号:1001-9081(2022)07-2104-08
DOI:10.11772/j.issn.1001-9081.2021040626

# 基于区块链与边缘计算的物联网访问控制模型

张　杰[1],许姗姗[1,2],袁凌云[1,3*]

(1. 云南师范大学 信息学院,昆明 650500; 2. 教育部西部资源环境地理信息技术教育部工程研究中心(云南师范大学),昆明 650500;
3. 教育部民族教育信息化教育部重点实验室(云南师范大学),昆明 650500)

(*通信作者电子邮箱blues520@sina.com)

**摘　要**:边缘计算的出现扩展了物联网(IoT)云-终端架构的范畴,在减少终端设备海量数据的传输和处理时延的同时也带来了新的安全问题。针对IoT边缘节点与海量异构设备间的数据安全和管理问题,并考虑到目前区块链技术广泛应用于分布式系统中数据的安全管理,提出基于区块链与边缘计算的IoT访问控制模型SC-ABAC。首先,提出集成边缘计算的IoT访问控制架构,并结合智能合约和基于属性的访问控制(ABAC)提出并设计了SC-ABAC;然后,给出工作量证明(PoW)共识算法的优化和SC-ABAC的访问控制管理流程。实验结果表明,所提模型对区块连续访问下的耗时随次数呈线性增长,连续访问过程中央处理器(CPU)的利用率稳定,安全性良好。本模型下仅查询过程存在调用合约的耗时随次数呈线性增长,策略添加和判断过程的耗时均为常数级,且优化的共识机制较PoW每100块区块共识耗时降低约18.37个百分点。可见,该模型可在IoT环境中提供去中心化、细颗粒度和动态的访问控制管理,并可在分布式系统中更快达成共识以确保数据一致性。

**关键词**:物联网;边缘计算;区块链;访问控制;属性访问;签名认证

**中图分类号**:TP309; TN929.5; TP391.44　　**文献标志码**:A

# Internet of things access control model based on blockchain and edge computing

ZHANG Jie[1], XU Shanshan[1,2], YUAN Lingyun[1,3*]

(1. *School of Information Science and Technology, Yunnan Normal University, Yunnan Kunming* 650500, *China*;
2. *GIS Technology Research Center of Resource and Environment in Western China, Ministry of Education (Yunnan Normal University), Yunnan Kunming* 650500, *China*;
3. *Key Laboratory of Educational Information for Nationalities, Ministry of Education (Yunnan Normal University), Yunnan Kunming* 650500, *China*)

**Abstract:** The emergence of edge computing has expanded the scope of Internet of Things (IoT) cloud-terminal architecture. With the reduction of transmission and processing delays of massive data on terminal devices, it also brings new security issues. Aiming at the problem of data security and management issues between edge nodes of IoT and massive heterogeneous devices, and considering that blockchain technology is widely used in the security management of data in distributed systems, an IoT access control model Smart Contract for Attribute-Based Access Control (SC-ABAC) was proposed based on blockchain and edge computing. Firstly, an IoT access control architecture integrated with edge computing was proposed, and by combining smart contracts with Attribute-Based Access Control (ABAC), SC-ABAC was proposed and designed. Then, the optimization of Proof of Work (PoW) consensus algorithm and the access control management flow of SC-ABAC were given. Experimental results show that the time consumed by the proposed model increases linearly with the number of times under continuous access to the block, the Central Processing Unit (CPU) utilization rate is stable, and the CPU security is good during the continuous access process. In this model, the time consumption of calling contracts in the query process only increases linearly with the times, and the time consumptions of the strategy addition and judgment process are both constant. And the optimized consensus mechanism has about 18.37 percentage points less time consumption than PoW consensus per 100 blocks. Therefore, the proposed model can provide decentralized, fine-grained and dynamic access control management in the IoT environment, and can reach consensus faster in a distributed system to ensure data consistency.

**收稿日期**:2021-04-20;**修回日期**:2021-07-21;**录用日期**:2021-08-05。　**基金项目**:国家自然科学基金资助项目(61561055);云南省基础研究专项(202101AT070098);云南省万人计划青年拔尖人才项目;云南师范大学研究生创新基金资助项目(ysdyjs2020148)。

**作者简介**:张杰(1997—),男,安徽芜湖人,硕士研究生,主要研究方向:物联网安全、区块链、访问控制、边缘计算; 许姗姗(1994—),女,河南驻马店人,硕士研究生,主要研究方向:湖泊表面水温、传感器; 袁凌云(1980—),女,云南昭通人,教授,博士,CCF会员,主要研究方向:物联网、传感器网络。

**Key words:** Internet of Things (IoT); edge computing; blockchain; access control; attribute access; signature authentication

## 0　引言

物联网(Internet of Things, IoT)是通过互联网等网络方式使大量能感知外部环境的传感器、智能设备相互通信以完成不同的任务,实现“万物互联”。随着物联网技术的逐渐普及,海量设备的接入形成了异构环境,会在短时间内产生大量待处理的数据并通过网络被同时传输到数据中心,网络延时不可避免[1]。而边缘计算[2]为物联网设备海量数据传输过程中高时延问题的解决提供了新的思路。物联网系统边缘节点的存在,增加了系统计算力、降低时延的同时也带来了新的安全风险。当新接入的边缘节点为恶意节点时,可通过并行其他边缘节点进行计算来解密并获取数据中心的用户隐私数据。因此,针对带有边缘节点的物联网系统的访问控制问题成为新的研究热点。

区块链技术已被越来越多地应用于分布式物联网体系中,以提供安全性和隐私性[3]。它在保证数据分布式存储的同时,确保数据的不可篡改,特别适合存储和保护重要隐私数据,降低了物联网设备因数据中心被攻击而导致的用户个人隐私数据泄露或大量数据丢失的风险。而区块链3.0[4]中引入的智能合约则使得通过区块链技术解决访问控制问题有了新的可能。目前通过区块链来解决集成边缘计算下物联网的安全问题,大多是针对隐私保护[5-6]、数据安全[7-8]方面,对物联网边缘节点的访问控制问题的研究并不多见。如Nyamtiga等[9]只针对集成边缘计算的物联网系统中数据匿名性和完整性问题,采用区块链技术实现安全存储IoT数据。Ren等[10]面向工业物联网下大规模数据传输存储问题,用边缘计算处理源头数据再由区块链实现数据的安全存储和管理。国内针对物联网安全数据访问与控制相关的研究仍仅考虑将区块链技术与物联网所结合,如史锦山等[11]提出一种基于区块链的IoT访问控制框架;王思源等[12]提出基于区块链的权能令牌环网来解决物联网越权访问;张建国等[13]提出基于区块链的角色访问控制模型来解决对物联网设备的安全访问。在将边缘计算与区块链集成下的研究中,程冠杰等[14]提出一种基于区块链和边缘计算的物联网数据管理架构来实现数据的安全管理。他们忽视了边缘节点的潜在风险,默认存在的边缘节点为安全可靠,却忽视了物联网规模扩展下,边缘节点在接入物联网时应通过可信机制加以确认,同时也忽视了边缘计算带来的强大计算能力,即针对目前集成边缘计算的物联网体系新范式下,没有考虑边缘节点在访问控制过程中的作用。

针对上述问题,在已有的传统集成区块链与边缘计算体系架构[15]的基础上提出了针对含有边缘计算的物联网系统的访问控制模型。具体地,在物联网三层架构的基础上设计了物联网三层访问控制架构,在基于属性的访问控制机制基础上,将访问控制决策交由区块链中的智能合约负责,而边缘节点负责加密、解密和数据传输,提出了SC-ABAC(Smart Contract for Attribute-Based Access Control)访问控制模型,以实现物联网的有效访问控制和管理。

## 1　基于区块链的物联网访问控制管理机制

### 1.1　集成边缘计算的物联网访问控制架构设计

集成边缘计算的物联网访问控制模型架构分为3个模块:IoT设备管理模块、IoT设备访问控制模块和IoT设备数据管理模块。具体的模型设计如图1所示。

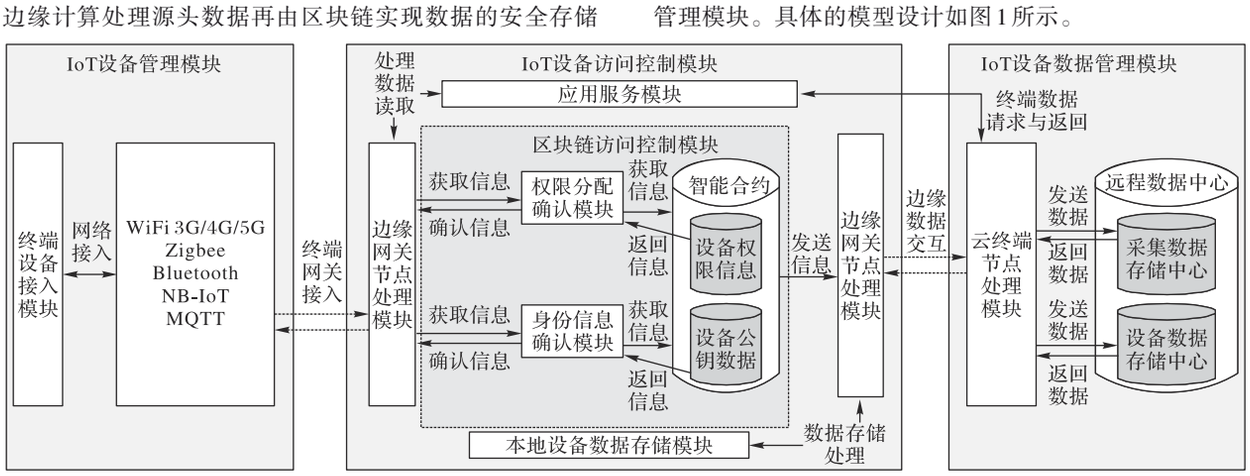


图1　集成边缘计算的IoT访问控制架构设计

Fig. 1　Design of IoT access control architecture integrated edge computing

#### 1.1.1　IoT设备管理模块

IoT设备管理模块将各种IoT异构设备进行接入管理,通过对网络中不同协议的解析来对采集数据进行预处理,并将其打包成统一格式后向IoT设备访问控制模块申请接入系统,进而发送数据至边缘网关节点。

#### 1.1.2　IoT设备访问控制模块

IoT设备访问控制模块集成边缘计算和区块链,为该框架的核心模块,用于实现设备的细颗粒度访问控制管理。

边缘网关节点处理模块是拥有一定计算能力的设备,处于网络中间层(如:智能手机、智能网关等)。主要功能包括:1)加密设备信息;2)解密数据信息;3)验证设备权限;4)动态更改设备权限;5)预处理数据;6)数据转发与获取。设备通过边缘网关节点的身份确认后基于自身的权限才可通过边缘网关节点获取或发送对应数据。

区块链访问控制模块为访问控制模块的核心,通过可编程的智能合约为不同场合的物联网系统提供不同的访问控

制机制。它支持设备的动态权限分配，以实现设备对数据的细颗粒度访问控制。主要功能包括：1）访问控制决策执行；2）设备身份确认；3）设备权限分配。

本地设备数据存储模块用于与边缘网关节点交互，存储预处理的数据，并在需要时将其发送至应用服务模块供用户实时查看。

应用服务模块为上层数据传输接口，供用户对数据进行可视化操作。通过从边缘网关节点和云终端节点获取本地数据和最终处理数据，保证时效性的同时提高数据的精确度。

#### 1.1.3 IoT设备数据管理模块

IoT设备数据管理模块存储：1）由云终端处理后的边缘网关模块预处理数据，供必要时查看；2）由边缘网关模块广播的区块链模块中的设备信息，降低框架风险。

### 1.2 基于智能合约的属性访问控制模型

基于属性的访问控制（Attribute-Based Access Control，ABAC）模型[16]的访问结构由{主体，客体，操作，环境}构成。它通过在对应环境下主体向客体的操作是否包含正确的属性来确定是否向主体提供访问权限。

#### 1.2.1 SC-ABAC模型

在物联网环境中，由于属性是每个主体、客体、操作和环境所固有的，将每种设备和资源的属性与访问权限关联，使得ABAC模型适合管理物联网的简单设备和广泛数据。但对于异构设备的动态接入管理，ABAC采用属性发现机制，并不能按{属性，权限}对接入设备进行精确合适的权限分配。而这会限制异构设备对数据的正常访问和实时处理，给存在边缘节点的场景带来了挑战。

因此本文针对异构设备的动态接入，基于ABAC提出SC-ABAC的访问控制模型，将智能合约与ABAC相结合，通过ABAC对资源和节点进行权限划分，并基于智能合约来确保对应操作的正确执行。

在SC-ABAC中，主体和客体被视为相同对象，这是因为在物联网中每个设备都可以成为为其他设备提供资源的客体，或作为主体从其他设备访问资源。每个物联网设备第一次接入时需通过边缘服务器向区块链注册信息，并通过其固有属性获得对应权限；为了设备自身的安全，将权限与椭圆曲线加密（Elliptic Curve Cryptography，ECC）[17]获得的公钥进行绑定，并在之后的访问和数据传输过程中采用私钥进行加密。而属性的判断则基于智能合约进行实现，并将每次的访问请求和判断结果存入区块链中实现同步。

#### 1.2.2 基于SC-ABAC的智能合约设计

智能合约是一种直接控制区块链内部数据的人为编写的程序脚本[18]，由区块链内的多个用户共同参与实施，在没有第三方的情况下也能控制交易的行为。智能合约可采用调用自定义函数来执行相关命令：如发送某一节点的访问请求或返回结果。

基于SC-ABAC，本文设计了管理合约（Manager Contract，MC）{地址，属性，权限}、判断合约（Judeg Contract，JC）{编号，主体，主体权限，资源，资源权限，结果，时间}、惩罚合约（Punish Contract，PC）{编号，JC结果，结果，时间}和访问控制合约（Access Contral Contract，ACC）{编号，主体，资源，JC结果，PC结果，时间}。

1）MC用于管理相关策略，主体为访问发起人，由对应的MAC地址代替，属性为该对象所固有的特性，权限是以数值进行描述的可操作行为，权限依次提高为读取、写入、管理等。其中自定义函数有：

ManageAdd()用于添加某一对象的权限信息

ManageUpdate()用于更新某一对象的权限信息

ManageDelete()用于删除某一对象的权限信息

QueryData(*Hash*)用于通过*Hash*来获取某一对象的信息

2）JC用于对某一对象申请对某一资源进行操作时的判断，通过对主体权限和资源权限的比较得出结果，包括允许所有操作、可读写、可读和非法访问等。其中自定义函数有：

JudgeFromMC()用于向MC中获取对象的权限信息

JudgeToPC()用于向PC发送判断结果

JudgeToACC()用于向PC发送判断结果

3）PC用于对JC发送的结果进行处理，如对访问越权数据的设备实施惩罚机制，或是对正常访问、发送数据的设备提供奖励机制，以达到分配动态权限的目的。其中自定义函数有：

PublishToACC()用于向ACC发送惩罚结果

4）ACC用于实现对设备和资源进行最终的访问控制，通过JC结果向主体返回判断结果，通过PC结果向MC进行主体和资源的函数操作：

ACCAnswer()将JC结果返回给主体以完成对资源的访问控制

ACCSetMC()通过JC结果调用MC中的函数对对象权限进行修改

基于区块链的访问控制管理机制具体流程如图2所示。

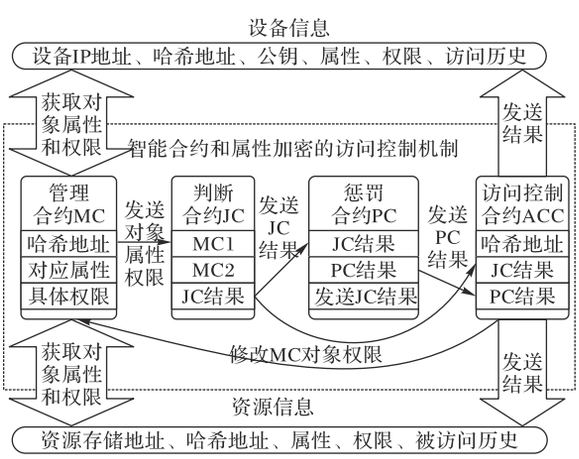


图2 基于SC-ABAC的智能合约实现流程

Fig. 2 Flowchart of smart contract implementation based on SC-ABAC

## 2 设备间访问控制流程设计

本章将详细介绍在基于区块链和边缘计算的物联网访问控制流程，如图3所示，以某一设备A接入边缘服务器向某一资源B发出访问请求为例，其访问控制流程包括系统初始化阶段、注册阶段、访问阶段和授权阶段。

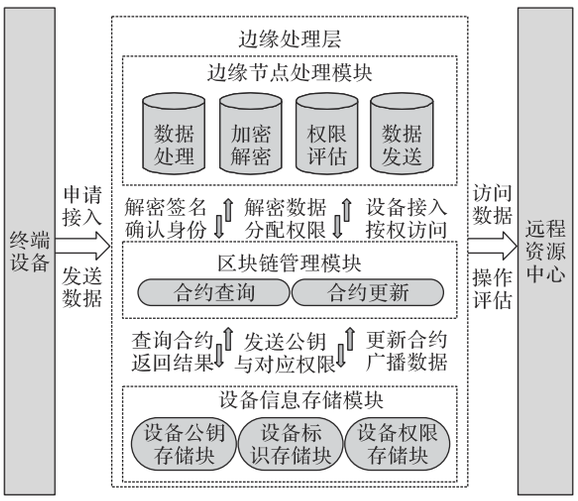


图3　具体设备访问数据的处理控制流程

Fig. 3　Flowchart of processing and controlling of specific device accessing data

### 2.1　系统初始化阶段

默认所处网络边缘的相关节点均为边缘节点,包括边缘网关和边缘服务器。边缘节点(Edge Node, *En*)需要先基于ECC算法计算出自己的公钥和私钥。具体如下:

1)*En*在$p$个元素($p$为素数)中选择两个小于$p$的非负整数$a,b$,得到椭圆曲线$Ep(a,b)$,再选择椭圆曲线上$n$阶的点$P$,使得数乘$n\times P=O$。

2)*En*从小于$n$的正整数中随机选择一个数作为自己的私钥$pKEn$,并根据$pKEn\times p$获取公钥$PkEn$,并由式(1)计算出物理地址(Media Access Control address, *MAC*)的哈希值后公开以下信息:$\{PkEn, Ep(a,b), P, p, Hash\}$。

$$Hash=\mathrm{SHA256}(MAC, P(x), P(y)) \tag{1}$$

3)区块链建立创世块,并存储*En*信息、属性和权限。

### 2.2　注册阶段

1)*En*计算每一个接入设备的ECC密钥对和哈希值,分配设备其具体属性如感知、发送和接受数据、分析、受控等,基于属性分配权限,并将设备的信息{注册*ID*,设备公钥$Pk_{\mathrm{IoT}}$,设备属性,设备权限,设备MAC地址,*Hash*}发送至区块链中进行存储。

2)*En*将设备的私钥$PK_{\mathrm{IoT}}$存储在本地数据中,仅由边缘网关查看。

3)区块链调用智能合约对设备信息进行一一存储。

### 2.3　访问阶段

1)设备生成随机素数$r(r<n)$,通过ECC算法随机生成密钥对$\big(r, R(x,y)\big)$,并基于椭圆曲线secp256k1的数字签名算法(Elliptic Curve Digital Signature Algorithm, ECDSA)对发送数据$M$进行数字签名加密,具体签名过程如算法1所示:

算法1　ECDSASIGN签名算法。

输入　设备数据$M$,设备私钥$PK_{\mathrm{IoT}}$,secp256k1椭圆曲线$T=(p,a,b,P,n,h)$;

输出　$Sign=(Sr,Ss)$。

```
begin
  while true
    do
    r = random_prime(); 1 < r < n
    R(x, y) = rP
    Sr = (R(x) mod n)
    if Sr = 0
      重新选择新的随机素数r再次生成R(x,y)
      continue
    Ss = r^-1 (Hash(M) + pK_IoT) mod n
    if Ss = 0
      重新选择新的随机素数r再次生成R(x,y)
      continue
    done
  return Sign = (Sr, Ss)
end
```

2)*En*收到设备接入时发来的数据$M$,向区块链申请公钥$Pk_{\mathrm{IoT}}$进行签名解密验证,具体过程如算法2所示:

算法2　ECDSASIGN签名验证算法。

输入　设备数据摘要$e=Hash(M)$,签名结果$Sign=(Sr,Ss)$,设备公钥$Pk_{\mathrm{IoT}}$,secp256k1椭圆曲线$T=(p,a,b,P,n,h)$;

输出　签名验证结果true or false。

```
begin
  U1 = (eSign(Ss)^-1) mod n
  U2 = ((Sign(Sr))Sign(Ss)^-1) mod n
  R(x1, y1) = U1P + U2Pk_IoT
  if Sign(Sr) == R(x1) mod n
    签名验证成功
    return true
  else
    return false
end
```

3)*En*解密数据$M$,获得设备的哈希地址值和请求访问的数据,向数据存储中心发送信息获取数据的哈希地址值,发送至区块链。

### 2.4　授权阶段

1)管理合约(MC)通过解密后的Hash值获取对应权限,将其发送至判断合约(JC)。

2)JC比较设备权限和数据权限大小,根据结果确定是否可以读写或非法操作并发送至访问控制合约(ACC)和惩罚合约(PC)。

3)PC根据JC的结果决定对该设备实施惩罚或奖励机制,并将结果发送至ACC。

4)ACC根据JC的结果,向*En*发送访问控制的查询结果,根据PC的结果,由MC对该设备的权限进行提高或降低,或在设备表中删除该设备。

### 2.5　PoW共识算法优化

考虑到传统PoW共识算法是将前一区块*Hash*值、此区块交易数据、时间戳和区块中的随机数*nNonce*组合在一起由SHA256算法计算出对应*Hash*值,通过*nNonce*的自加来得到最终前*nBits*位为0的哈希值。由于每次*Hash*值均差别巨大,得到符合前*nBits*位为0的哈希值可能花太多时间来计算。因此本文考虑对随机数*nNonce*作随机序列化操作,将初始值改为随机数,并在每次计算*Hash*值时进行判断,如果所得到的*Hash*值第一位为0,则*nNonce*自加后得到前*nBits*位为0的可能性较小,因此*nNonce*设为随机数;而如果第1位不为0,则*nNonce*在现有基础上进行自加。具体算法如下:

算法3　PoW共识算法优化。

输入　区块链难度*nBits*，区块数据*DataHash*，随机数*nNonce*，前区块哈希值*prevHash*，最大随机数范围*N*，当前时间戳*Date*；

输出　当前区块最终哈希值*BlockHash*。

```
begin
  while true
    do
    nNonce = random(), 1 ≤ nNonce ≤ N
    Data = (DataHash + nNonce + prevHash)
    BlockHash = SHA256(Data + Date)
    if BlockHash前nBits位≠ 0
      if BlockHash第1位≠ 0
        nNonce = nNonce + 1
      else
        nNonce = random()
    else
      break
    done
  return BlockHash
end
```

## 3　实验验证与分析

### 3.1　实验设置

#### 3.1.1　实验环境构建

本实验是在1台运行Windows 10专业版19042.928版本的i5-8400 2.80 GHz 24 GB运行内存的个人计算机(Personal Computer，PC)和3台拥有温湿度传感器DHT11模块的CC2530(CPU为8051，运行内存8 KB)上进行，区块链由Node.Js Version 10.16.0实现。通过将3台CC2530作为物联网设备，获取的温湿度数据存储至PC中作为物联网数据，PC作为边缘节点基于SC-ABAC实现设备与数据的安全访问。

#### 3.1.2　访问控制策略设计

此次实验中设计了两组访问控制策略，包括设备类型与操作许可。对于设备类型，管理员M0管理云C1、边缘节点E2和终端设备D3，云的子设备为智能网关G4，边缘节点和终端设备的子设备为底层物联网设备d5~d7，物联网设备管理数据D8；操作许可则包括数据的创建、更新、读取和删除4种操作。如图4所示。

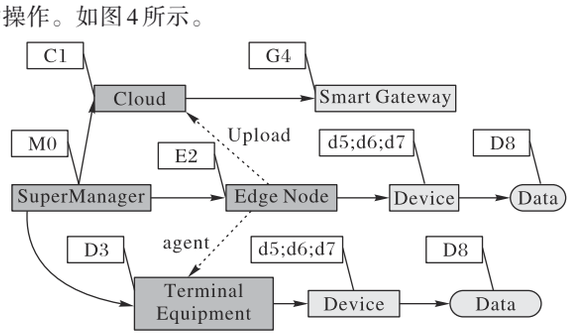


图4　访问控制策略的设计

Fig. 4　Design of access control policy

### 3.2　安全性分析

在针对SC-ABAC模型的安全分析中，本文假设终端设备存在数据泄露风险，而边缘设备*En*中存在恶意用户，存储节点存在不良服务商，网络中存在恶意攻击者。

1)针对终端设备的数据泄露风险。

SC-ABAC将终端设备上传数据和对应权限信息在交易侧进行加密后以*DataHash*方式存储至网络中，原始数据移交给数据中心存储，数据的访问需要*En*拥有对应的*DataHash*来调用智能合约查询，仅当设备获取到正确的*DataHash*且拥有对应权限后才会返回原始数据。而数据的传输为P2P(Point-to-Point)模式，不存在第三方捕获，且原始数据在获取后可进行签名认证，不存在存储节点的恶意修改。

2)针对边缘设备中的恶意用户。

SC-ABAC将所有接入的*En*以MAC生成对应*HashID*进行存储并在每一次添加设备时确保对应ID唯一，而接入*En*基于职能对不同设备有不同的读写权限，每一次的越权均有对应惩罚机制，可有效降低恶意用户在网络中的生存时间。

3)针对存储节点中的不良服务商。

SC-ABAC的数据以智能合约的方式存于存储节点，原始数据的上传、访问、失效都是以带有时间戳的链式结构进行记录，并且全节点同步记录信息，因此存储节点的非法访问可被追溯且不可篡改。

4)针对网络中存在的恶意攻击者。

SC-ABAC将性能相对弱小的终端设备，以*En*代理的方式进行管理，终端设备不接触外部网络，而*En*作为边缘设备，拥有较高的安全防护能力。存储节点则由防护能力最强的*En*担当。

### 3.3　基于SC-ABAC的智能合约实现

图5展示了通过调用智能合约的ACC()将策略加入区块中；图6展示了访问控制过程中设备的温湿度数据上链。

```
调用智能合约进行设备权限注册
策略添加完毕
新区块中添加策略:
Block {
  transaction:
   { parentChildMapping:
     [ [Object],
       [Object],
       [Object],
       [Object],
       [Object],
       [Object],
       [Object],
       [Object],
       [Object] ],
    roleModuleMapping:
     { SuperAdmin: [Object],
       Cloud: [Object],
       EdgeManager: [Object],
       TEquipment: [Object],
       SmartGateway: [Object],
       Device: [Array],
       DataCenter: [Object] } },
  previousHash: '',
  BlockTime: '2021-06-21 16:36:09',
  RandomNum: 1,
  Hash:                          对应策略区块PoW所得Hash值
   '1184089f38faaa16c8b2ce08ebe60e3becdfed67c401fcd5231d232241101db8' }
POW完成00237e1b2de14a77841ac180b290204b7dc401f879773426ad73187ad75f1959
```

图5　ACC()调用访问控制策略并加入区块中

Fig. 5　ACC() calling access control policy and adding it to block

此外，为了体现SC-ABAC的优势，重新设计了一个空的策略文件，通过ManageAdd()加入所需的策略，边缘节点通过QueryData()向数据中心进行数据的查询，如图7所示。查

询时通过 PublishToACC() 调用 JudgeToPC(),基于主体、客体、行为来进行访问权限判断,当非法访问时会导致惩罚措施,即从该策略文件中删除一条主体的访问策略;相反,当访问合法则会增加策略。通过调用 ManageUpdate() 更新了边缘节点的访问策略,使其访问范围扩大,实现了合法数据访问。图 8 表明在访问到数据后,通过验证终端设备的数字签名,证实数据的可靠性。

```
设备数据加入链中
Chain {
  Difficult: 2,
  chain:
  [ Block {
      transaction: 'ancestorBlock',
      previousHash: '',
      BlockTime: '2021-06-21 16:36:09',
      RandomNum: 322,
      Hash:
       '000ef23ddfba7dd0f2e4ce782be9d9df55511176fad42c5293f29737279a5612' },
    Block {
      transaction: [Object],
      previousHash:
       '000ef23ddfba7dd0f2e4ce782be9d9df55511176fad42c5293f29737279a5612',
      BlockTime: '2021-06-21 16:36:09',
      RandomNum: 54,
      Hash:                          策略区块PoW所得Hash值
       '00237e1b2de14a77841ac180b290204b7dc401f879773426ad73187ad75f1959' },
    Block {
      transaction: [TEquipment],
      previousHash:
       '00237e1b2de14a77841ac180b290204b7dc401f879773426ad73187ad75f1959',
      BlockTime: '2021-06-21 16:36:09',
      RandomNum: 226,
      Hash:                          终端设备数据PoW所得Hash值
       '00c3463b872dc16d6bf89d56c7aa352f0b47b29bd8c2a103aa91ddcbd4a55222' } ] }
```

图 6　设备数据的入链

Fig. 6　In-chain of device data

```
访问控制策略的添加
新策略文件内容:  []
添加策略:
[ { SuperAdmin:
    { moduleName: 'M0',
      operations: [ 'C', 'R', 'U', 'D', [length]: 4 ] } },
  { Clode:
    { moduleName: 'C1', operations: [ 'R', 'U', 'D', [length]: 3 ] } },
  { EdgeManager: { moduleName: 'E2', operations: [ 'R', 'U', [length]: 2 ] } },
  { TEquipment: { moduleName: 'D3', operations: [ 'R', 'U', [length]: 2 ] } },
  { Device:
    [ { moduleName: 'd5', operations: [ 'C', 'U', 'D', [length]: 3 ] },
      { moduleName: 'd6', operations: [ 'C', 'U', 'D', [length]: 3 ] },
      { moduleName: 'd7', operations: [ 'C', 'U', 'D', [length]: 3 ] },
      [length]: 3 ] },
  { DataCenter: { moduleName: 'D8', operations: [ 'R', 'U', [length]: 2 ] } },
  { moduleName: 'E2',
    parent: 'M0',
    children: [ 'd5', 'd6', 'd7', [length]: 3 ] },
  { moduleName: 'D3',
    parent: 'M0',
    children: [ 'd5', 'd6', 'd7', [length]: 3 ] },
  { moduleName: 'd5',
    parent: 'E2',
    children: [ 'D8', [length]: 1 ] },
  [length]: 9 ] color:black
--------------------------------  边缘设备非法访问未授权数据
边缘设备访问数据节点D8
无访问权限,仅可访问:  [ 'd5', 'd6', 'd7' ]
--------------------------------
惩罚:删除非法设备访问权限 边缘设备访问策略被删除
[ { SuperAdmin: { moduleName: 'M0', operations: [Array] } },
  { Clode: { moduleName: 'C1', operations: [Array] } },
  { TEquipment: { moduleName: 'D3', operations: [Array] } },
  { Device: [ [Object], [Object], [Object] ] },
  { DataCenter: { moduleName: 'D8', operations: [Array] } },
  { moduleName: 'E2',
    parent: 'M0',
    children: [ 'd5', 'd6', 'd7' ] },
  { moduleName: 'D3',
    parent: 'M0',
    children: [ 'd5', 'd6', 'd7' ] },
  { moduleName: 'd5', parent: 'E2', children: [ 'D8' ] } ]
```

图 7　PublishToACC() 下的惩罚机制

Fig. 7　Punishment mechanism under PublishToACC()

```
再次访问数据:
--------------------------------
奖励:再次添加设备访问权限
[ { SuperAdmin: { moduleName: 'M0', operations: [Array] } },
  { Clode: { moduleName: 'C1', operations: [Array] } },
  { TEquipment: { moduleName: 'D3', operations: [Array] } },
  { Device: [ [Object], [Object], [Object] ] },
  { DataCenter: { moduleName: 'D8', operations: [Array] } },
  { moduleName: 'E2',
    parent: 'M0',
    children: [ 'd5', 'd6', 'd7', 'D8' ] },
  { moduleName: 'D3',
    parent: 'M0',
    children: [ 'd5', 'd6', 'd7' ] },
  { moduleName: 'd5', parent: 'E2', children: [ 'D8' ] },
  { EdgeManager: { moduleName: 'E2', operations: [Array] } },
  { EdgeManager: { moduleName: 'E2', operations: [Array] } } ]

数据查询完成
d5  temp:12.4  hum:8  2021-06-21 16:36:11
验证数据签名:
数据签名:  3045022034d6e9060964d1c7152eb4374686f5a6cf0e248b91d604c456b25e
25e395caf869802210083c000af1f897f4f80c3dfa7b47ce83e8988e02ff13045a21b514d44ce5c99ba
签名验证:  true
--------------------------------
POW完成 006472a8031e279baf0256bad9718d8cdb31f0cc30fe42012753bf4b63bf6307
```

图 8　合法访问的策略保障和基于数字签名的验证机制

Fig. 8　Policy guarantee for legal access and authentication mechanism based on digital signature

### 3.4　区块链系统测试

#### 3.4.1　区块链系统的性能测试

在对模型进行安全性分析后,本文通过设置主体对客体的不同访问次数来检测本区块链系统的性能。通过设置连续的高访问并发来确定性能,并通过资源监视器和任务管理器来查看访问过程的 CPU 利用率。

图 9 是连续的不同主体共 500 次对不同数据的访问耗时,对应数据为每 10 次的平均耗时。可以看出随着访问次数的增加,每次访问的耗时呈线性增长,在第 500 次的每次访问的平均耗时约为 1.3 s。这是因为在每次的访问过程结束后都会将访问记录存入区块链中,使得每次的访问查询数据在不断增加,从而导致访问耗时随访问次数的增加而线性增长。图 10 是在连续访问过程中的 CPU 利用率情况,可以看出在访问过程中 CPU 的利用率稳定在 30% 左右,说明本系统的访问过程对 CPU 的使用较为稳定。

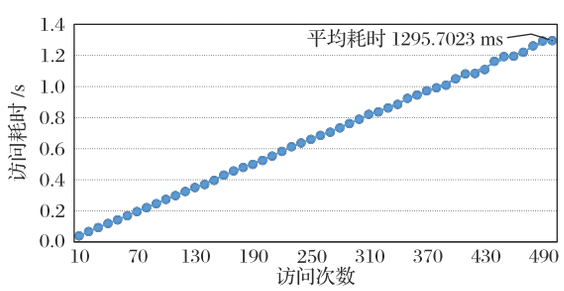


图 9　不同访问次数下的耗时

Fig. 9　Time consumption under different visit times

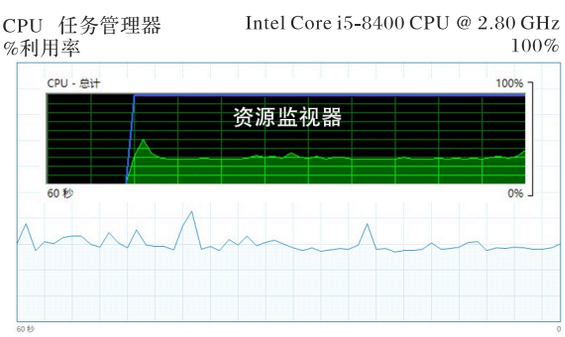


图 10　连续访问过程的 CPU 利用率

Fig. 10　CPU utilization rate during continuous access

#### 3.4.2 系统安全性测试

本文系统采用PoW共识机制实现区块链的生成，因此对区块链的安全性测试即对共识机制的测试。本文采用文献[19-20]中提出的PoW安全性分析案例，即用二项随机过程描述诚实节点和攻击者间的区块创建。只有当攻击者的区块创建速度大于链中所有诚实节点的区块生成速率才可被视为攻击成功。因此假设攻击者创建下一个区块的概率为$q$，诚实节点创建下个区块的概率为$p$，而区块链从$z$块开始被攻击者追上，则攻击者追上真实区块长度的概率如式(2)所示：

$$q^z=\begin{cases}1, & p\leqslant q\\ (q/p)^z, & p>q\end{cases} \tag{2}$$

而对攻击者而言确定成功的区块数目可看成泊松分布，其期望值如式(3)：

$$\lambda=zq/p \tag{3}$$

将其与式(2)相乘即为攻击者在攻击过程中创建区块长度下可超过诚实节点的真实区块长度，即攻击成功的概率如式(4)，其中$k$为攻击者创建的区块数：

$$P=\sum_{k=0}^{\infty}\frac{\lambda^k e^{-\lambda}}{k!}\begin{cases}1, & k>z\\ (q/p)^{(z-k)}, & k\leqslant z\end{cases} \tag{4}$$

转化为式(5)即为攻击者成功概率：

$$P=1-\sum_{k=0}^{z}\frac{\lambda^k e^{-\lambda}}{k!}\left[1-(q/p)^{(z-k)}\right] \tag{5}$$

通过Matlab 2019b仿真结果如图11所示，当攻击者创建下一区块的概率$q$越大(即算力越强)则攻击成功概率越高，当拥有全网50%的算力时则可实现完美攻击；而当自身创建下一区块的概率越低，则越难攻击。因此本文系统控制区块链中区块数量至少为6块，由图中可看出在攻击者自身算力不超过全网30%时均可保证系统的安全性。

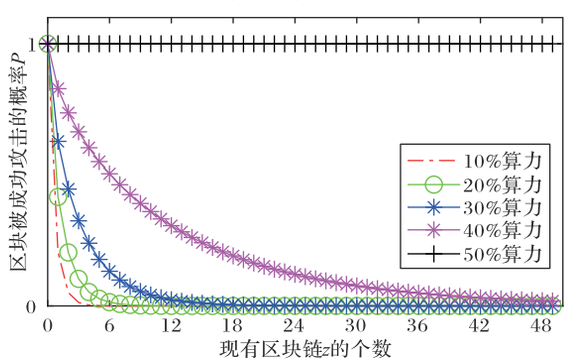


图11 区块链系统被成功攻击概率对比

Fig. 11 Comparison of probability of blockchain system being successfully attacked

### 3.5 SC-ABAC的性能分析

#### 3.5.1 不同智能合约下的时延对比

通过模拟多线程客户的并发请求，测试在PoW的难度为0和难度为2下不同并发请求次数下的ManageAdd()、QueryData()和PublishToACC()的处理时间，如图12所示。可以看到，在不同并发的请求数量下，添加和判断合约几乎不会耗时，证明了边缘节点在物联网场景下的优越性。由于在查询过程中需要在区块链中完成共识、访问记录和同步查询，因此随着请求次数的增加，导致耗时增加。由于难度区分度不大，使得PoW所需时间相较于查询时间可以忽略不计。但相比较而言，本文提出的方案仍具有较高的应用价值。

#### 3.5.2 共识机制的优化

针对PoW共识机制进行优化，并通过比较本文的访问控制方案和PoW共识机制在相同节点数和难度为5下不同区块中构建的成本时间来测试分布式系统数据一致性效率。实验中区块数目设置从1~100，结果如图13所示，加粗线为本文优化的PoW共识机制，虚线为原PoW共识机制，其中突出的峰值为达成共识时的耗时。从平均时间上优化机制较原PoW每100个区块的构建时间减少了18.37%。由此可以看出，本文的优化机制在区块数量增长的情况下要优于PoW。

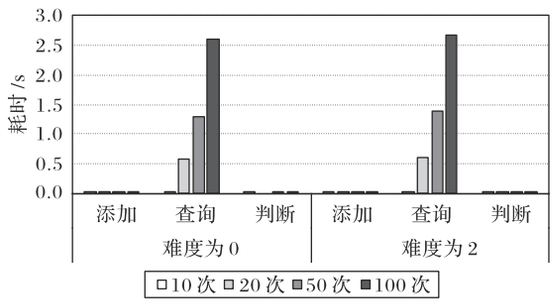


图12 ManageAdd()、QueryData()和PublishToACC()的处理时间

Fig. 12 Processing time for ManageAdd(), QueryData() and PublishToACC()

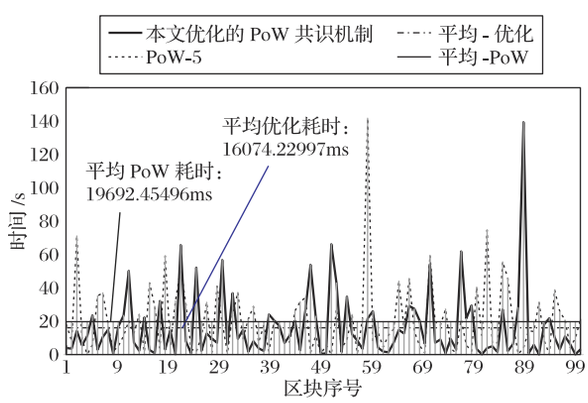


图13 难度5下的优化后共识算法和PoW对比

Fig. 13 Comparison of optimized consensus algorithm and PoW under difficulty 5

## 4 相关工作讨论

文献[9]虽集成边缘计算和区块链技术来解决物联网架构下的数据安全问题，但仅给出解决方案的概念设计并无实验进行支撑。

文献[10]同样集成边缘计算和区块链技术，采用区块链技术和布隆过滤器[21]解决工业物联网中设备的身份认证和数据的访问控制，但同样未给出实验结果。

文献[11-13]在对物联网访问控制的研究中仅考虑到使用区块链技术，包括提出访问控制框架、采用访问令牌或设计更优的访问控制策略等方式，但未考虑到边缘节点在当前物联网规模下的不可或缺性，因此在物联网的访问控制研究中未考虑到边缘节点的存在以及边缘计算与区块链集成下对现有物联网访问控制研究的影响。

国内与本文类似的研究为文献[14]，都是面向物联网架构，将边缘计算与区块链中的智能合约相结合，但解决的问题不同。本文的重点是物联网的访问控制模型，而文献[14]

是物联网的数据管理。在访问控制上,文献[14]采用访问控制列表方式,存在访问控制力度低等问题;本文则设计SC-ABAC模型、两组访问控制策略以及奖励与惩罚机制等,实现对设备和数据的细颗粒度动态访问控制。在数据管理上,文献[14]通过对称加密、签名算法和分布式存储实现数据的安全存储,由智能合约通过附录文件进行数据的查询与获取;但文献[14]对附录文件未做安全性处理,使得任意节点均可获取附录文件从而获取到加密数据和数据所有者信息,安全性不足。本文将非对称加密后的数据存入数据中心,将DataHash索引与签名存入区块链中,由智能合约通过DataHash和权限许可两方面进行数据获取,且每次数据的访问记录均加入区块,确保数据的安全访问和可追溯。

## 5　结语

本文结合区块链技术设计了一种通过物联网边缘节点对终端设备和资源的访问控制模型,旨在为目前处理边缘计算和区块链在物联网安全体系下访问控制研究的不足提供新的解决思路。基于原有的集成边缘计算的物联网体系,设计出集成边缘计算的访问控制架构,将区块链中的智能合约与基于属性的访问控制相结合,提出了SC-ABAC模型。考虑到安全性能,基于ECC和ECDSA实现了基于区块链的访问控制管理机制,并通过实验来验证模型与架构的安全可靠性。对于未来的工作,考虑到本研究只针对边缘物联网下数据的访问控制进行研究,对于数据本身存在的环境及个人隐私信息未作相应的处理,下一步将试图结合基于属性的加密来构建高效且保护隐私的边缘物联网环境。

**参考文献** (References)


[1] PULIAFITO C, MINGOZZI E, ANASTASI G. Fog computing for the Internet of mobile things: issues and challenges [C]// Proceedings of the 2017 IEEE International Conference on Smart Computing. Piscataway: IEEE, 2017: 1-6.

[2] SHI W S, CAO J, ZHANG Q, et al. Edge computing: vision and challenges [J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.

[3] YANG Q, LU R X, RONG C M, et al. Guest editorial the convergence of blockchain and IoT: opportunities, challenges and solutions[J]. IEEE Internet of Things Journal, 2019, 6(3): 4556-4560.

[4] DI FRANCESCO MAESA D, MORI P. Blockchain 3.0 applications survey [J]. Journal of Parallel and Distributed Computing, 2020, 138: 99-114.

[5] DWIVEDI A D, SRIVASTAVA G, DHAR S, et al. A decentralized privacy-preserving healthcare blockchain for IoT[J]. Sensors, 2019, 19(2): No. 326.

[6] LIN C, HE D B, KUMAR N, et al. HomeChain: a blockchain-based secure mutual authentication system for smart homes [J]. IEEE Internet of Things Journal, 2020, 7(2): 818-829.

[7] TULI S, MAHMUD R, TULI S, et al. FogBus: a blockchain-based lightweight framework for edge and fog computing [J]. Journal of Systems and Software, 2019, 154: 22-36.

[8] REN Y J, LENG Y, CHENG Y P, et al. Secure data storage based on blockchain and coding in edge computing [J]. Mathematical Biosciences and Engineering, 2019, 16(4): 1874-1892.

[9] NYAMTIGA B W, SICATO J C S, RATHORE S, et al. Blockchain-based secure storage management with edge computing for IoT[J]. Electronics, 2019, 8(8): No. 828.

[10] REN Y J, ZHU F J, QI J, et al. Identity management and access control based on blockchain under edge computing for the industrial Internet of things[J]. Applied Sciences, 2019, 9(10): No. 2058.

[11] 史锦山,李茹,松婷婷. 基于区块链的物联网访问控制框架[J]. 计算机应用, 2020, 40(4): 931-941. (SHI J S, LI R, SONG T T. Blockchain-based access control framework for Internet of things[J]. Journal of Computer Applications, 2020, 40(4): 931-941.)

[12] 王思源,邹仕洪. 多域物联网中基于区块链和权能的访问控制机制[J]. 应用科学学报, 2021, 39(1): 55-69. (WANG S Y, ZOU S H. Blockchain and capability based access control mechanism in multi-domain IoT [J]. Journal of Applied Sciences — Electronics and Information Engineering, 2021, 39(1): 55-69.)

[13] 张建国,胡晓辉. 基于以太坊的改进物联网设备访问控制机制研究[J]. 计算机工程, 2021, 47(4): 32-39, 47. (ZHANG J G, HU X H. Research on improved access control mechanism of internet of things devices based on Ethereum [J]. Computer Engineering, 2021, 47(4): 32-39, 47.)

[14] 程冠杰,黄诤杰,邓水光. 基于区块链与边缘计算的物联网数据管理[J]. 物联网学报, 2020, 4(2): 1-9. (CHENG G J, HUANG Z J, DENG S G. Data management based on blockchain and edge computing for Internet of things[J]. Chinese Journal on Internet of Things, 2020, 4(2): 1-9.)

[15] YANG R Z, YU F R, SI P B, et al. Integrated blockchain and edge computing systems: a survey, some research issues and challenges [J]. IEEE Communications Surveys and Tutorials, 2019, 21(2): 1508-1532.

[16] HU V C, KUHN D R, FERRAIOLO D F, et al. Attribute-based access control[J]. Computer, 2015, 48(2): 85-88.

[17] NYAME G, QIN Z G, AGYEKUM K O-B O, et al. An ECDSA approach to access control in knowledge management systems using blockchain[J]. Information, 2020, 11(2): No. 111.

[18] ZHANG Y Y, KASAHARA S, SHEN Y L, et al. Smart contract-based access control for the internet of things[J]. IEEE Internet of Things Journal, 2019, 6(2): 1594-1605.

[19] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system [EB/OL]. [2021-01-20]. https://bitcoin.org/bitcoin.pdf.

[20] JAVAID U, JAMEEL F, JAVAID U, et al. Rogue device mitigation in the Internet of Things: a blockchain-based access control approach[J]. Mobile Information Systems, 2020, 2020: No. 8831976.

[21] HIEB J, SCHREIVER J, GRAHAM J. Using Bloom filters to ensure access control and authentication requirements for SCADA field devices [C]// Proceedings of the 2012 International Conference on Critical Infrastructure Protection, IFIPAICT 390. Berlin: Springer, 2012: 85-97.


This work is partially supported by National Natural Science Foundation of China (61561055), Yunnan Fundamental Research Program (202101AT070098), Young Talent Program of Yunnan Ten Thousand People Project, Graduate Innovation Fund of Yunnan Normal University (ysdyjs2020148).

**ZHANG Jie**, born in 1997, M. S. candidate. His research interests include internet of things security, blockchain, access control, edge computing.

**XU Shanshan**, born in 1994, M. S. candidate. Her research interests include lake surface water temperature, sensor.

**YUAN Lingyun**, born in 1980, Ph. D., professor. Her research interests include internet of things, sensor network.