doi: 10.7690/bgzdh.2026.06.016

面向物联网应用场景的 CP-ABE 加密数据访问控制策略

张家华，张明飞，李国辉，杨绍禹
（华北水利水电大学信息工程学院，郑州 450045）

摘要：针对物联网环境中设备规模庞大、数据共享频繁以及访问控制复杂多变，传统访问控制与隐私保护机制难以满足实际需求的问题，提出一种面向物联网的数据权限管理与安全共享方案。该方案融合 CP-ABE、Fabric 与 IPFS 分布式存储技术，构建链上链下协同的数据共享模型，并通过引入树形访问控制结构，将访问策略与属性层次进行统一建模，实现细粒度权限控制与动态授权管理。同时，设计智能合约以支持用户属性管理、策略更新及数据访问验证。为评估方案的性能，在 Fabric 环境下针对不同属性规模和并发请求场景进行实验。结果表明：随着属性规模和请求数量的增加，系统响应时间呈稳定增长趋势，整体吞吐性能良好，能够有效支撑复杂访问控制需求和大规模数据访问场景。所提方案在保障数据安全与隐私的同时，提高了权限管理的灵活性与系统的可扩展性，为物联网环境中的数据安全共享提供有效支撑。

关键词：物联网；数据共享；Fabric；智能合约；数据安全
中图分类号：TP309.2　　**文献标志码：**A

CP-ABE Encrypted Data Access Control Strategy for Internet of Things

Zhang Jiahua, Zhang Mingfei, Li Guohui, Yang Shaoyu
(*School of Information Engineering*, *North China University of Water Resources and Electric Power*, *Zhengzhou* 450045, *China*)

Abstract: Aiming at the problem that in the Internet of Things (IoT) environment, traditional access control and privacy protection mechanisms are difficult to meet the actual needs due to the large scale of devices, frequent data sharing and complex and changeable access control, a data rights management and secure sharing scheme for IoT is proposed. The scheme integrates CP-ABE, Fabric and IPFS distributed storage technologies, and constructs a data sharing model of on-chain and off-chain cooperation. Through the introduction of tree access control structure, the access policy and attribute hierarchy are modeled uniformly, and fine-grained access control and dynamic authorization management are realized. At the same time, smart contracts are designed to support user attribute management, policy updates, and data access validation. In order to evaluate the performance of the scheme, experiments are carried out in Fabric environment for different attribute sizes and concurrent request scenarios. The results show that with the increase of attribute size and the number of requests, the system response time shows a steady growth trend, and the overall throughput performance is good, which can effectively support complex access control requirements and large-scale data access scenarios. The proposed scheme not only ensures data security and privacy, but also improves the flexibility of authority management and the scalability of the system, providing effective support for secure data sharing in the Internet of Things environment.

Keywords: Internet of Things; data share; Fabric; chaincode; data security

0 引言

随着物联网智能化规模逐渐扩大，人们管理大量物联网设备并共享其中数据信息的需求也不断增大。怎样管理这些数据以及如何保障安全和隐私等问题严重妨碍了其发展[1-2]。物联网终端设备采集的海量数据给有限计算力、有限存储能力的物联网设备带来了挑战，需要能够提供安全保障机制和分布式的数据文件存储[3-5]。且在物联网环境下，设备动态扩展性差、拥有者自主性差，访问控制过程灵活，组合多样，不宜采用集中式的传统访问控制方案。

区块链和智能合约作为核心技术已经被广泛应用于物联网访问控制中，以实现更安全、高效和可扩展的分布式访问控制[6]。这些方法可以通过智能合约或在区块链上记录访问策略（权限凭证）来做出分布式访问决策，可以通过查询区块链数据来证明用户已授予访问权限[7-8]。

在基于区块链的数据访问控制模型中，于金刚等[9]研究并设计了基于区块链的访问控制方案，使用智能合约对访问控制请求进行处理。Mcfarlane 等[10-12]针对物联网设备数据碎片化严重、传输过程

收稿日期：2024-12-08；**修回日期：**2025-01-25
基金项目：河南省住房城乡建设科学技术计划项目（HNJS-2024-K34）；河南省学术学位研究生核心课程项目（YJS2026XSKC03）
第一作者：张家华（2005—），男，陕西人。

不安全等问题，提出基于以太坊的智能合约实现对数据信息的访问控制。Zhou[13]等提出一种基于角色的加密(RBE)方案，该方案将 RBAC 与 CP-ABE 相结合，用于安全的云存储，使用角色的公共参数对数据进行加密，分配给该角色的用户能够解密密文，从而简化了整体校验性及隐私信息保护不足、共享效率低的问题。张晓东等[14-15]基于 CP-ABE 提出了 PKEM-CPABE 的方案，并与区块链进行结合，实现高效率的安全共享机制。Kumar 等[16]提出了基于区块链的增强 Bell-LaPadula 模型，在不同的许可和安全级别对对等点和事务进行了分类，并在基于区块链的网络上提供动态访问控制功能，将数据存储到 IPFS 中，并且实现了对数据在存储之前进行本地加密。

在这种背景下，有学者提出了基于属性的访问控制(attribute-based access control，ABAC)，基于属性的访问控制通过添加线性访问策略的方式限制不合法用户的访问。这种线性的规则不仅难以设计和维护，而且遇到复杂的规则将会导致授权不够灵活；因此，笔者提出了使用树形的结构并结合基于属性的加密方案(key policy attribute-based encryption，KP-ABE)替代线性规则去解决上述问题，做到了对属性的层次化管理，实现对属性的灵活管理，适用于更加复杂的场景。

1　准备知识

1.1　参数定义

基于访问控制树与 CP-ABE 算法所涉及的主要参数如表 1 所示。

表 1　本方案主要参数

参数	含义	参数	含义
r	根节点	sk	系统私钥
v	普通节点	s	加密中的随机数
$\rho(v)$	与节点相关的参与方	M	共享明文
A_v	节点相关联的授权集合	CT	共享密文
e	配对映射	S	秘钥属性集合
msk	系统主密钥	U	用户属性列表
pk	系统公钥		

1.2　双线性映射

双线性映射是一个函数 $e:G_1\times G_2\rightarrow G_T$，将椭圆曲线上 2 个点映射到一个有限域 G_T 上的元素，其中，G_1 和 G_2 是 2 个椭圆曲线上的群，G_T 是一个有限域。

双线性映射满足以下性质：

1) 双线性性：对于任意 P，$Q\in G$ 和 $a, b\in Z$，有：

$$e(aP, bQ)=e(P, Q)^{ab}。\tag{1}$$

2) 非退化性：存在一个生成元 $P\in G$，使得 $e(P, P)\neq 1$。

3) 可计算性：双线性映射的计算可以进行有效计算。

1.3　访问控制树

访问控制树是一种用于描述访问控制策略的数据结构，基于线性秘密共享方案来定义。

1.3.1　访问控制树节点

1) 根节点：根节点表示被保护的秘密或资源，将其表示为根节点 r。2) 内部节点：对于每个内部节点 $v\in V$，表示一个授权集合或访问权限，用 A_v 表示与节点 v 相关联的授权集合。3) 叶子节点：对于每个叶子节点 $v\in V$，表示一个参与方，用 $\rho(v)$表示与节点 v 相关联的参与方。

1.3.2　访问控制树的构建和恢复

1) 构建：对每个内部节点 v，选择一个权重常数 $\omega_v\in Z_p$，表示授权集合的权重 A_v。2) 恢复：假设秘密 s 被共享为 l 个有效份额，λ_i 表示第 i 个参与方的有效份额。可以表示授权集合 A 为 $A=v|\rho(v)\in A$，其中 A 是访问授权集合。如果 A 中的参与方可以重构秘密 s，则存在常数 ω_v 使得 $\sum_{v\in A}\omega_v\lambda_v=s$ 成立，其中 λ_v 是参与方 $\rho(v)$的有效份额。

2　树形结构树形加密算法

2.1　树形属性结构

树形属性结构用于组织和表示属性的层次结构，其中叶子节点代表具体属性值，而非叶子节点则表示相应属性组的门限值。在该结构中，节点间的层次关系清晰表达了属性间层次结构。图 1 为具有 3 层次结构的树形属性访问控制结构。

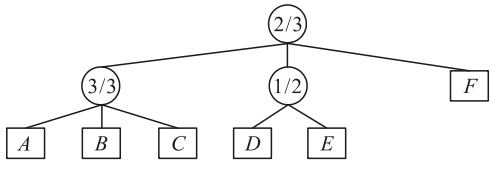

图 1　树形属性访问控制结构

树中的方形表示属性值，圆形表示门限值。门限值表示方式“1/3、2/3、3/3、1/2”，其中分子表示需要满足的子节点个数，分母为该非叶子所拥有的子节点数。在物联网场景下，由于用户身份的多样性，双线性映射被广泛应用于实现复杂物联网的

身份认证和秘钥协商，而访问控制树则被用于策略表达和动态权限调整，以适应物联网复杂用户场景，使用两者，可确保数据安全和系统可靠性。

当 Alice 拥有属性集合{*A*, *D*, *E*, *F*}，Bob 拥有属性集合{*A*, *B*, *D*, *E*}时，Alice 可以根据“*D*”或“*E*”属性恢复出“1/2”节点，将“1/2”节点加上“*F*”属性节点可以恢复出“2/3”节点，故 Alice 可以得到秘密值。反观 Bob 无法解密出根节点秘密值。

2.2 共享方案

在加密阶段使用树形结构替代传统的线性秘密共享方案，数据的上传方事先定义树形属性访问控制策略，进行数据共享策略管理。

2.2.1 初始化阶段

1) 生成双线性映射相关公共参数 $<e,g,G_1, G_T, Z_r>$；

2) 选取随机数 $\alpha\in Z_r$，并计算 $Y=e(g, g)^{\alpha}$；

3) 选取随机数 $\beta\in Z_r$，并计算 g^{β}；

4) 系统主密钥 $msk=g^{\alpha}$，公钥 $pk=<Y, g^{\beta}>$；

5) 每个属性对应一个 G_1 群元素，使用 hash 算法对属性进行 hash 计算。

2.2.2 秘钥生成

1) 选取一个随机数 t，计算 $D=g^{\alpha}g^{\beta t}$，$D_0=g^t$。

2) 对于用户属性列表 ATTList={attr_1, attr_2, attr_3, …, attr_i,…, attr_n}$_{\text{attr}_i(i=1,\cdots, n)}$的每个属性 i，$D_{\text{attr}_i}=H(\text{attr}_i)^t$。

3) 用户私钥 $sk=<D, D_0, \{D_{\text{attr}_i}\}_{\text{attr}_i\in \text{ATTList}}>$。

2.2.3 加密

1) 选取随机数 $h\in Z_r$，针对明文消息 $M\in G_T$，$C=Me(g, g)^{\alpha s}$，$C_0=g^s$。

2) 将 h 作为秘密 $h=<\lambda_1, \lambda_2, \lambda_3,\cdots, \lambda_j,\cdots, \lambda_w>$，$w$ 为门限值，采用树形结构进行拆分，构造树形属性结构，其中产生的叶子节点为 x，叶子节点 x 的属性为 $j=\text{attr}(x)$。对每个叶子节点属性，$C_j^1=g^{\beta\lambda j}H(j)^{-r_j}$，$C_j^2=g^{r_j}$。

3) 最后，完整的密文为 $ct=<C,C_0,\{C_j^1, C_j^2\}_{j\in\{1\ldots w\}}>$。

2.2.4 解密

1) 秘钥的属性集合 S 需要满足密文访问树 T 的情况才能进行解密。

2) 对于秘钥属性集合 S 和密文访问 T 的叶子节点属性集合中重合的属性 i。$P_i=e\left(C_i^1, D_0\right)e\left(C_i^2, D_{\text{attr}_i}\right)=e(g, g)^{-\beta t\lambda}$。

3) 从根节点开始，做递归计算，在这个过程中，将秘密嵌入在指数中，将整个 P_i 作为秘密分片来处理。

4) 计算 $e(C_0,D)=e(g, g)^{\alpha h}e(g, g)^{\beta th}$，进一步求得 $e(g, g)^{\alpha s}$。

5) 可以得到 $M=C/e(g, g)^{\alpha h}$，恢复出初始的明文消息。

2.3 安全模型

本文中的安全模型选择明文攻击下的不可区分性(indistinguish ability under hosen-plaintext attack，IND-CPA)，分为挑战者和敌手 A 2 个角色：

1) 初始化。挑战者将系统算法进行安装，并初始化系统，保存系统的主密钥 msk，公钥 pk，以及属性 ATTList 对应在 G_1 群元素，将输出的系统公钥公布给敌手 A。

2) 阶段 1。敌手在本地构建属性集 S={S_1, S_2, S_3,…, S_n}发送给挑战者，挑战者存储这些数据，并告知敌手存储的所处编号。

3) 挑战阶段。敌手随机选择 2 个等长的明文 M_0 和 M_1 发送给挑战者，挑战者随机选择一条明文 M_x，$x\in$ {0, 1}运行加密算法，输出密文 CT_x，将密文发送给敌手。

4) 阶段 2。重复阶段 1。

5) 猜测。敌手对 x 猜测，结果记为 x'。如果 $x=x'$，则敌手胜利。敌手在这个游戏中的优势如下：$[Adv]_A=|pr[x=x']-\frac{1}{2}|$可以忽略，本文中方案安全。

3 数据共享系统模型

3.1 系统模型

针对现有物联网数据存储和隐私保护以及共享的问题，笔者结合 CP-ABE 算法设计了针对区块链访问控制的模型，如图 2 所示。该系统由 5 部分组成：物联网设备(internet of things，IoT)，边缘设备(edge device，ED)，区块链网络(blockchain，BC)，普通用户(user)，管理员(admin)，星际文件系统(inter planetary file system，IPFS)。

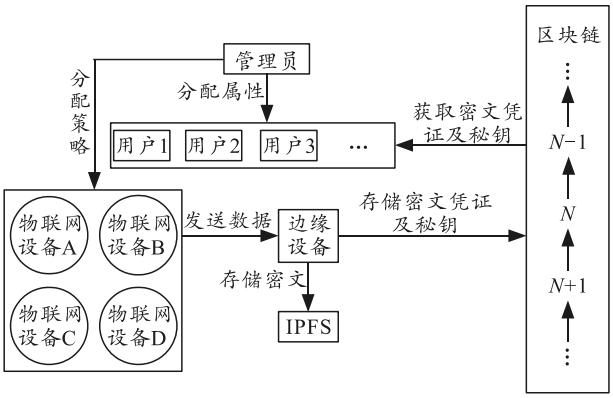

图 2　物联网系统模型架构

图 2 描述了物联网设备产生的数据由边缘设备进行对称加密，存储到 IPFS 网络中，并将代表文件地址的内容标识符(content identifier，CId)和对称秘钥隐藏到访问控制策略写入到区块链中。用户通过访问区块链，智能合约根据用户凭证查找用户拥有的属性，并将其作为算法参数执行智能合约中的 CP-ABE 算法，解密出密文凭证和秘钥。管理员用于对设备制定访问策略和对用户分配属性。

3.2　系统交互流程

物联网数据交互流程如图 3 所示。

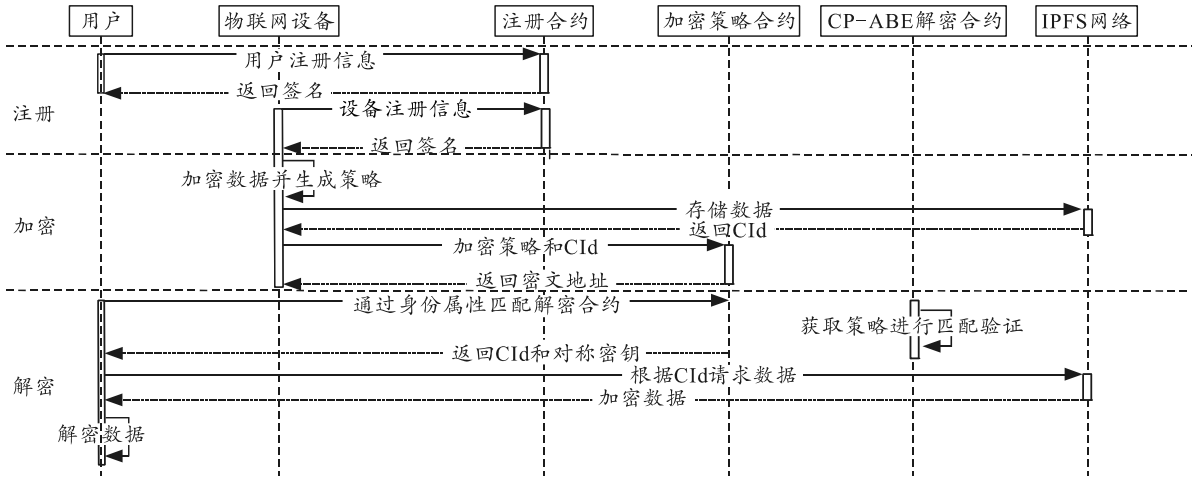

图 3　物联网数据交互流程

1) 注册：将合约在区块链上进行初始化，注册管理员以及若干用户身份和物联网设备身份。

2) 加密：物联网设备在本地产生的数据，发送到就近边缘设备执行 AES 对称加密，并记录加密密钥。将加密的数据发送到 IPFS 网络中，记录返回的 CId 值。

物联网设备向区块链平台发送自己的身份信息以及请求信息。区块链通过智能合约验证物联网设备的身份信息，检测其权限，并执行区块链智能合约将 CId 与策略进行加密存储，返回响应。

3) 解密：用户与区块链平台进行交互，发送自己的身份属性以及需要请求的信息。区块链通过智能合约去检测用户具有的属性信息，验证其权限。若通过则执行区块链智能合约取出用户的所需信息，返回用户 CId 和解密密钥。

用户根据返回的 CId，向 IPFS 网络请求数据。存储下载加密的文件。用户在本地根据解密密钥解密文件。

3.3　合约设计

1) 注册合约：注册合约主要是在开始时将用户具有的属性存储到区块链中，用于后期用户申请资源时进行匹配。接收管理员 ID 和用户属性列表 2 个参数，若条件符合，会将属性列表进行格式化存储，并返回用户 ID，否则返回错误信息。

```
Algorithm 1User Registration Attributes
Input: <AdminIdentity,AttributesList>
Output: UserIdentity or true or err
//检查是否是管理员，否则权限不足，添加失败
If APIStub.
GetState(AdminIdentity)!=Admin
Return Error('Bad info')
//格式化属性列表
NewAttributesList=Format(AttributesList)
//将用户访问控制属性存储到区块链中
err←APIStub.
PutState(UserIdentity, NewAttributesList)
If err!=nil
Return Error('ADD Failure')
Return UserIdentity
```

添加设备访问策略：将设备 ID 和 JSON 格式访问控制树策略字符串以及秘密值作为输入参数。查询设备是否存在以及查询设备的策略信息是否存

在。如果设备和策略信息存在，那么程序将使用输入的 JSON 字符串更新访问控制策略信息，并将新的策略信息存储到账本中。最后，该函数将返回成功信息以表示操作成功。如果在执行过程中出现错误，程序将返回错误消息。

Algorithm 1Update Device Info IntoBlockChain
Input: <DeviceIdentity, DevicePolicy, Val>
Output: true or err
If APIStub. GetState(DeviceIdentity). isExist
Return Error('Bad Device')
//获取设备信息
PolicyKey, err←APIStub. GetState(Device Identity)
//判空
If err!=nil
Return Error('Mistake DeviceIdentity')
//格式化策略
If Format(NewDevicePolicy)==false
Return Error('Bad DeviceInfo')
//将 Val 和格式化之后的策略进行加密
EncPolicyVal←CP-ABE. Encrypt(Val, NewDevicePolicy)
//将访问控制策略绑定到设备中
err←APIStub. PutState(PolicyValKey, EncPolicyVal)
If err!=nil
Return Error('ADD Failure')
Return true

解密合约：程序在用户请求获取文件 CId 及其密钥值时，首先从区块链中获取该用户的信息。如果用户不存在，程序返回错误信息。若成功程序会进一步从区块链中检索设备的信息，若设备不存在，同样响应错误信息。接下来，程序会从设备信息中获取访问规则，将规则和用户属性进行 CP-ABE 算法判断是否成功解密。若成功，便返回文件的 CId 和对称密钥，否则返回错误信息。

Algorithm 2GetDecryptCId ()
Get CId from BlockChain
Input: <UserIdentity, DeviceIdentity>
Output: CId and Secrete or err
//通过用户密钥获取用户属性
UserAttr, err←APIStub. GetState(UserIdentity)
If err!=nil//判空
Return Error('Mistake UserIdentity')
//获取该设备的信息
DeviceAttr,err←APIStub.GetState(DeviceIdentity)
If err!=nil//判空
Return Error('Mistake DeviceIdentity')
//获取该设备的策略
EncPolicyVal,err←APIStub.GetState(DeviceAttr.policy)
//进行 CP-ABE 解密，检测该用户能否恢复树的根节点
Cid,Secrete,err←CP-ABEDecrypt(UserAttr,EncPolicyVal)
If err!=nil//判断是否出现错误
Return Error('Mistake by CP-ABE')
Return Cid,Secrete

4 方案分析

4.1 安全性分析

1) 机密性：网关设备使用 AES 算法对数据进行加密，并将密文存储在 IPFS 上。将数据密文的哈希值存储在区块链上，确保了数据在存储过程中的保密性。只有拥有正确的密钥，才能解密并获取原始数据。智能合约使用 CP-ABE 方案对 AES 算法的密钥和数据密文的哈希值进行加密，在加密过程中，随机选择一个参数 s，并使用椭圆曲线离散对数困难问题计算相关的值，并将其用于加密 AES 密钥和数据密文的哈希值。这样的做法确保了加密密钥的安全性，只有满足访问控制策略的合法用户才能解密获取 AES 密钥和数据密文的哈希值，并进一步解密数据。该方案依赖于 AES 算法和 CP-ABE 方案的安全性。AES 算法是一种广泛使用的对称加密算法，目前没有已知的有效攻击方法。CP-ABE 方案基于椭圆曲线离散对数困难问题，已被广泛研究和证明其安全性；因此，只有具备解密属性的合法用户才能获得数据的有效信息。

2) 完整性：IPFS 使用内容寻址的方式来标识和检索数据。每个文件在 IPFS 中都有唯一的 CId，该 CId 是通过对文件内容进行哈希计算得到的；因此，只要文件内容不发生变化，其 CId 就会保持不变。由于区块链上存储的 CId 是文件内容的哈希值，因此一旦 CId 被存储到区块链上，就可以确保文件内容的完整性。

3) 抗单点故障：区块链和 IPFS 皆是使用了分布式的架构，当某一个节点发生了故障，不会影响到整个系统的正常运行，可以有效解决传统访问控制中单点故障问题。

4.2 功能性分析

笔者提出的模型实现了去中心化的访问控制，

利用 CP-ABE 和访问控制树对数据进行授权管理，实现了对数据的细粒度访问控制。传统管理模型与本文中方案对比情况如表 2 所示。

表 2　传统管理模型与本文中方案对比

方案	隐私保护	细粒度访问控制	层级属性结构管理
文献[3]	×	×	×
文献[7]	×	√	×
文献[11-12]	√	√	×
本文中方案	√	√	√

由表 2 可知：本文中方案相对于文献[3]，实现了数据的隐私保护，相对于文献[7]实现了更加细粒度的访问控制，相对于文献[11-12]增加了结合访问控制树去进行策略管理，从而实现属性层级结构管理和属性的继承性。

4.3　实验仿真分析

在实验中，使用 fabric-caliper 测试工具对该模型进行检测。

环境准备：搭建 fabric 网络，网络结构中有 $orderer_0$、$orderer_1$、$orderer_2$ 3 个排序节点，org_1 和 org_2 2 个组织，在每个组织下各有 $peer_0$ 和 $peer_1$ 2 个节点，其中每个组织下的 $peer_0$ 为锚节点。使用 ipfs-desktop 安装搭建 IPFS，进行数据上传和下载实验测试，网络拓扑管理如图 4 所示。

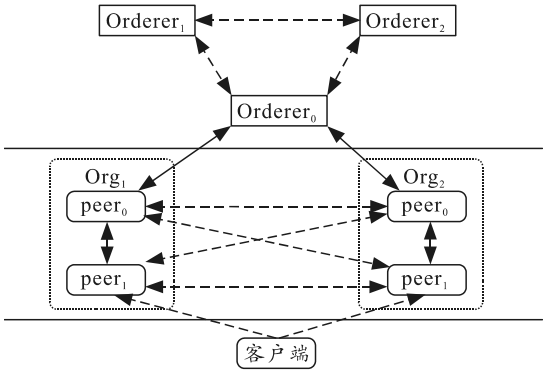

图 4　网络拓扑管理

区块链的客户端使用 Java 语言编写，智能合约使用 Go 语言完成，分别在版本环境中运行测试，如表 3 所示。

表 3　环境工具版本列表

Name	Version
Ubuntu	20.04
HyperLedger Fabric	2.4.4
Docker	24.0.4
Docker-compose	1.29.2
Golang	1.18.3
JDK	8
Maven	3.5.4
Node	16.16.0
Npm	8.11.0
fabric-caliper	0.5.0

初始阶段：在初始阶段把合约部署在已经搭建成功的区块链网络中，部署完成之后执行其中的初始化合约，生成之后需要使用的管理员信息和若干用户信息以及设备信息，以供后续的测试使用。

实验验证：为了检验方案的性能，笔者在实验中使用 fabric-caliper 去测试网络的吞吐量。在实验中模拟用户上传数据的场景，设置用户拥有属性的个数对响应时间的影响，设置用户属性个数从 1～18，分别进行测试其响应时间，得到如图 5 和 6 所示的结果。

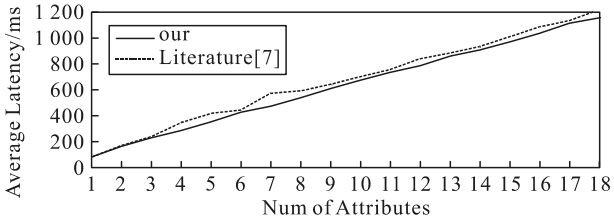

图 5　GetDecryptCId 不同属性数量下的节点响应时间对比

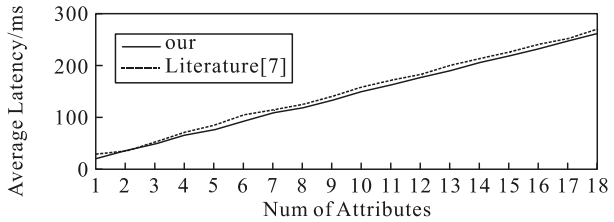

图 6　UploadEncryptPolicy 不同属性数量下的节点响应时间对比

从图 5 和 6 中可以看出：随着用户解密所需属性的增加，解密所依赖的时间也会随之增长。主要过程包括将数据加密为一段密文，以及生成对应访问策略的访问密钥。增加属性的数量，随之增加了生成访问策略的复杂性，因为需要计算属性间的逻辑关系和密钥的组合规则。随着属性数量的增加，加密过程所需的时间也会增加。同时，随着属性的增加，响应时间也线性增长。然而，获取数据和加密操作的时间消耗波动幅度相对较小，总体时间增量在可接受的范围内。

笔者进行了另一组试验，目的是在保持属性个数为 3 的情况下，测试系统的并发处理能力。试验中，使用了数量为 200、400、600、800、1 000 的请求进行测试。实验结果如图 7、8 所示。

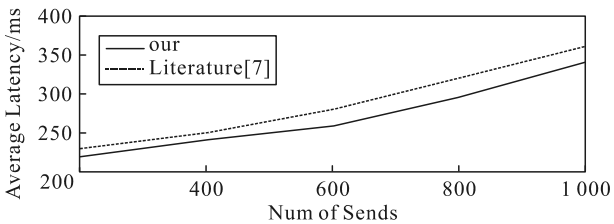

图 7　GetDecryptCId 不同并发数量下的节点响应时间对比

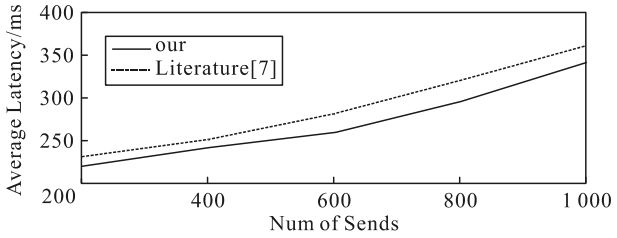

图 8 UploadEncryptPolicy 不同并发数量下的节点响应时间对比

从图 7 和 8 中可以明显看出：随着请求数量的增加，系统的平均延迟也在逐步增大。这可能是因为多个请求或进程在同时竞争系统的计算资源，比如 CPU、内存、磁盘和网络带宽等。通过数据分析发现：当并发请求量达到 600 时，系统开始出现资源短缺的情况，资源竞争变得激烈，导致每个请求的处理时间明显增长。整体的趋势表明：该方案的实施能够充分地利用物理设备资源，并在各种环境中保持高效的运行。

5 结束语

笔者设计了一个基于 CP-ABE 物联网数据存储模型，将树形属性访问控制结构的灵活性与区块链的不可篡改性和可验证性相结合，并根据此设计智能合约，以实现可信决策，为物联网数据的安全存储提供了新思路。目前，系统的访问控制决策效率和响应时间还有待提升，特别是在快速验证 CP-ABE 中的属性方面，存在改进空间；因此，笔者把提高访问控制决策效率和响应时间作为后续研究的重点，以进一步完善系统的性能和安全性。

参考文献：

[1] ZHANG P, LIU H Y, LI W J, et al. Industrial intelligent network: deepening and upgrading of industrial Internet[J]. Communications, 2018, 39(12): 134-140.

[2] QIU C, YU F R, YAO H P, et al. Blockchain-based software-defined industrial Internet of Things: A dueling deep Q-Learning approach[J]. IEEE Internet of Things, 2018, 6(3): 4627-4639.

[3] NOVO O. Blockchain meets IoT: An architecture for scalable access management in IoT[J]. IEEE internet of things, 2018, 5(2): 1184-1195.

[4] HAMMI M T, HAMMI B, BELLOT P, et al. Bubbles of Trust: A decentralized blockchain-based authentication system for IoT[J]. Computers & Security, 2018, 78: 126-142.

[5] EL-AZZAOUI A, CHEN H, KIM S H, et al. Blockchain-based distributed information hiding framework for data privacy preserving in medical supply chain systems[J]. Sensors, 2022, 22(4): 1371-1388.

[6] 李永强, 刘兆伟. 基于区块链的车联网安全信息共享机制设计[J]. 郑州大学学报(工学版), 2022, 43(1): 103-110.

[7] ALKHATEEB A, CATAL C, KAR G, et al. Hybrid blockchain platforms for the internet of things (IoT): A systematic literature review[J]. Sensors, 2022, 22(4): 1304-1323.

[8] ATTKAN A, RANGA V. Cyber-physical security for IoT networks: a comprehensive review on traditional, blockchain and artificial intelligence based key-security[J]. Complex & Intelligent Systems, 2022, 8(4): 3559-3591.

[9] 于金刚, 张弘, 李姝, 等. 基于区块链的物联网数据共享模型[J]. 小型微型计算机系统, 2019, 40(11): 2324-2329.

[10] MCFARLANE C T, BEER M, BROWN J, et al. Patientory: A healthcare peer-to-peer EMR storage network v1.1[R]. Atlanta:Patientory Inc, 2017.

[11] SUN S, DU R, CHEN S, et al. Blockchain-based IoT access control system: towards security, lightweight, and cross-domain[J]. IEEE Access, 2021, 9: 36868-36878.

[12] AZARIA A, EKBLAW A, VIEIRAT, et al. Medrec: Using blockchain for medical data access and permission management[C]//2016 2nd international conference on open and big data (OBD). IEEE, 2016: 25-30.

[13] ZHOU L, VARADHARAJAN V, HITCHENS M. Enforcing role-based access control for secure data storage in the cloud[J]. Computer, 2011, 54(10): 1675-1687.

[14] 张晓东, 陈韬伟, 余益民, 等. 基于属性基加密的区块链数据共享模型[J]. 计算机应用研究, 2021, 38(8): 2278-2283.

[15] 巴阳, 陈越, 胡学先, 等. 基于区块链与属性基加密的数据共享方案[J]. 信息工程大学学报, 2022, 23(4): 443-451.

[16] KUMAR R, TRIPATHI R. Scalable and secure access control policy for healthcare system using blockchain and enhanced Bell–LaPadula model[J]. Ambient Intelligence and Humanized Computing, 2021, 12(2): 2321-2338.