研究与开发

面向物联网的基于智能合约与CP-ABE的访问控制方案

孙昌霞[1,2]，张传虎[1]，刘冰杰[1,2]，Yingjie Yang[3]，Fernando BAÇÃO[2,4]，刘倩[1,2]
（1. 河南农业大学信息与管理科学学院，河南 郑州 450046；
2. 河南省农业大数据与人工智能国际联合实验室，河南 郑州 450000；
3. 德蒙福特大学计算机、工程和媒体学院，英国 莱斯特 LE1 9BH；
4. 新里斯本大学NODV信息管理学院，葡萄牙 里斯本 1070-312）

摘　要：随着物联网设备数量激增，传统的集中式访问控制方案在面对当前大规模物联网环境时显得力不从心，现有的分布式访问控制方案存在高货币成本和处理访问请求的低吞吐量等问题。针对这些问题提出一种区块链智能合约结合密文策略属性基加密（ciphertext policy attribute based encryption，CP-ABE）实现对物联网资源的访问控制方案。以超级账本（Hyperledger Fabric）为底层网络，对功能令牌执行属性基加密，利用星际文件系统（interplanetary file system，IPFS）保存令牌密文，通过智能合约公开令牌获取地址实现一对多授权。进一步设计合约部署到区块链实现对令牌请求的去中心化权限评估，维护主体在特定资源对象上允许的操作，实现更为细粒度的属性访问控制。仿真实验及性能分析表明，所提方案与现有方案相比能够使数据所有者在更短的时间内完成对大量请求主体的安全访问授权，压力测试表明链码具有较好性能。

关键词：物联网；访问控制；区块链；智能合约；密文策略属性基加密
中图分类号：TP393.0
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2024227

An access control scheme for IoT based on smart contracts and CP-ABE

SUN Changxia[1,2], ZHANG Chuanhu[1], LIU Bingjie[1,2], YANG Yingjie[3], BAÇÃO Fernando[2,4], LIU Qian[1,2]
1. College of Information and Management Science, Henan Agricultural University, Zhengzhou 450046, China
2. Henan International Joint Laboratory of Agricultural Big Data and Artificial Intelligence, Zhengzhou 450000, China
3. Faculty of Computing, Engineering and Media, De Montfort University, Leicester LE1 9BH, UK
4. NOVA Information Management School, New University of Lisbon, Lisbon 1070-312, Portugal

收稿日期：2024−05−10；**修回日期：**2024−10−13
通信作者：刘倩，kuzikurt@163.com
基金项目：河南省科技攻关项目（No.232102211087）；河南省杰出外籍科学家工作室项目（No.GZS2024006）；河南省中央引导地方科技发展资金资助项目（No.Z20231811005）；河南省联合基金资助项目（No.232103810020）；河南省高等学校重点科研项目（No.23A520005）
Foundation Items: The Key Scientific and Technological Project of Henan Province Department of China (No.232102211087), Henan Province Outstanding Foreign Scientists Workstation (No.GZS2024006), Henan Province Central Government Guidance Fund for Local Science and Technology Development (No.Z20231811005), Henan Province Science and Technology Research and Development Plan Joint Fund Project (No.232103810020), Key Research Projects of Higher Education Institutions in Henan Province (No.23A520005)

Abstract: As the number of Internet of things (IoT) devices increases, traditional centralized access control solutions are inadequate for the current large-scale IoT environment. Existing distributed access control schemes suffer from high monetary costs and low throughput in processing access requests. To address these issues, a blockchain smart contract combined with ciphertext policy attribute based encryption (CP-ABE) was proposed to implement access control for IoT resources. Using Hyperledger Fabric as the underlying network, attribute-based encryption was applied to functional tokens, and token ciphertexts were stored using the interplanetary file system (IPFS). Through smart contracts, token retrieval addresses were publicly exposed to achieve 1-to-*N* authorization. Furthermore, contracts were designed to be deployed on the blockchain for decentralized permission evaluation of token requests, maintaining the allowed operations for subjects on specific resource objects, realizing more fine-grained attribute-based access control. Simulation experiments and performance analysis demonstrate that compared to existing solutions, this approach enables data owners to securely authorize access for a large number of requesting subjects in a shorter time frame. Stress tests show that the chaincode performs well.

Key words: IoT, access control, blockchain, smart contract, ciphertext-policy attribute-based encryption

0　引言

据预测，截至2025年物联网将连接超300亿台设备，这些设备涉及从传感器到智能家居、智能出行，再到农业、工业等各个生产领域，物联网数据也将呈现指数级增长[1]。物联网设备容易受到未经授权的安全攻击，尤其是一些计算能力有限的传感器和执行器，它们通常缺乏强大的安全保护机制。当设备涉及个人信息处理时，未经授权的访问会导致信息泄露和系统故障等严重后果，威胁人们的安全和隐私。另外，实际生产中缺乏一套统一的设备认证和授权标准，导致设备制造商采用各自的方法来提供这些功能，从而引发大量针对资源受限设备的攻击[2]。

传统访问控制模式中，为便于管理通常采用中心化授权和存储访问权限信息，但物联网设备可能属于不同域组织，性能与灵活性均存在限制，导致集中的访问控制难以满足物联网环境下的访问控制要求。传统访问控制机制在应对物联网的复杂性和规模方面显得力不从心，区块链技术[3]的崛起为构建更安全、更透明和去中心化的物联网访问控制提供了新的可能性。区块链以其难以篡改、支持智能合约、点对点（peer-to-peer，P2P）和去中心化等特点，成为改善物联网访问控制的理想选择[4]。从现有的访问控制方案可以看出，区块链的使用主要可以分为两类，一是实现基于分布式账本的去中心化存储，二是支持基于智能合约的分布式服务[5]。文献[6]提出了一种基于许可区块链和边缘计算的物联网访问控制方案，使区块链与基于属性的访问控制（attribute based access control，ABAC）模型相结合，将访问控制策略写入智能合约，部署在区块链上进行调用，并引入Kraft共识算法提升共识效率。文献[7]提出了一种面向物联网基于区块链的设备身份认证框架，使用区块链存储设备身份信息，并提出了用于记录身份验证交易的物联网区块链网关。在文献[8]提出的系统中，通过在以太坊区块链平台上部署智能合约构建了一个基于角色的访问控制（role-based access control，RBAC）框架。所有用户角色识别和分配都由智能合约完成，然后部署在区块链上。该系统提供了一种安全有效的方法来定义用户角色分配，并验证用户的角色。文献[9]设计了3种智能合约，实现了一种基于超级账本（Hyperledger Fabric）区块链框架和ABAC的物联网访问控制系统Fabric-IoT，以解决服务器单点故障。

但是，这些方法通常使用高级加密标准（advanced encryption standard，AES）等机制对数据

进行加密，需要数据所有者和用户建立一对一的密钥共享通道，特别是在物联网环境中就不可避免地增加了数据所有者的通信成本。为解决该问题，有研究引入基于属性的加密方案，实现一对多的细粒度访问控制。文献[10]提出了一种基于许可区块链技术的安全可靠的农产品追溯系统，探讨了密文策略属性基加密（ciphertext-policy attribute-based encryption，CP-ABE）算法下的细粒度授权访问和农产品质量安全追溯机制。文献[11]提出了利用基于属性的加密（attribute-based encryption，ABE）进行细粒度访问控制和安全共享的数据访问控制和共享模型，达到细粒度访问控制和安全共享的目的。文献[12]结合区块链提出一种高效且保护隐私的可追溯的ABE方案来保证数据的完整性和不可否认性。文献[13]提出一种在物联网环境下高性能的ABE与区块链的轻量级解密访问控制方案，在云不受信任的假设下，确保外包安全解密。文献[14]将分布式账本技术物联网应用（Internet of things application，IOTA）与CP-ABE技术相结合提出了一种新的物联网访问控制方案，实现了对大量主体的分布式授权，方案采用IOTA Tangle[15]（IOTA的底层数据库）保存密文数据导致其存储效率低，IOTA智能合约的不完善使其并未实现去中心化权限验证。

为解决上述问题，本文提出一种细粒度、无费用且可扩展的物联网环境下设备资源访问控制方案，并引入简单的令牌管理。首先通过使用CP-ABE技术对访问功能令牌进行加密，并通过智能合约公开其密文获取地址，实现一对多的授权。CP-ABE中设置访问策略对访问主体进行静态属性限制，令牌中的约束则实现动态属性限制，从而达到更加灵活和细粒度的属性访问控制。其次，利用星际文件系统（interplanetary file system，IPFS）服务中的星际命名系统（interplanetary name system，IPNS）功能保存令牌密文，确保在分布式存储环境下，令牌更新时其密文获取地址始终不变。最后，基于Hyperledger Fabric联盟链，通过智能合约进行去中心化权限评估，完成权限验证，实现对物联网设备资源安全高效地管理。

1 关键技术

1.1 双线性对

G、G_{T}为大素数阶q的两个循环乘法群，从G分别取G_1和G_2，使得$e:(G_1\times G_2)\to G_{\mathrm{T}}$表示双线性映射[16]，该映射具有以下性质。

（1）双线性：给定$\forall_x\epsilon G_1,\forall_y\epsilon G_2$和$a,b\epsilon Z_{\mathrm{q}}$,有$e(x^a,y^b)=e(x,y)^{ab}$。

（2）非退化性：$\forall_x\epsilon G_1,\forall_y\epsilon G_2$，$e(x,y)\neq 1$。

（3）可计算性：如果$\forall_x\epsilon G_1,\forall_y\epsilon G_2$，即可计算$e(x,y)$。

如果$G_1=G_2$，配对为对称，否则为非对称。

1.2 访问结构

系统中定义一个属性集合$\{a_1,a_2,\cdots,a_n\}$，对于单调集合$A=2^{\{a_1,a_2,\cdots,a_n\}}$，如果存在$A_1\subseteq 2^{\{a_1,a_2,\cdots,a_n\}}$，集合$A_1$称为授权集。对于$\forall B,C$，如果$B\epsilon A$且$B\epsilon C$，得到$C\epsilon A$，即$A\in 2^{\{a_1,a_2,\cdots,a_n\}}$，称$A$为单调访问结构，且本文只采用单调访问结构。

本文采用单调访问结构，访问结构可以直观地表示为带有AND和OR门的单调布尔公式，使用单调跨度程序转换为矩阵$\boldsymbol{M}$和映射ρ，其中$\boldsymbol{M}$的第i行标记为属性$\rho(i)$。设S是一组属性，如果存在系数$\{\omega_i\}_{\rho(i)\epsilon S}$如$\sum_i\omega_i M_i=(1,0,\cdots,0)$则表示$S$满足访问结构（$\boldsymbol{M}$，$\rho$）。

1.3 Hyperledger Fabric区块链

本文选择Hyperledger Fabric[17]联盟区块链项目构建所提方案区块链网络。Fabric是一个许可的区块链系统，区块及区块链数据结构如图1所示，它只允许已识别的节点参与网络。与公共区

块链相比，联盟区块链执行更高效、更便宜的共识算法，如Kafka和Raft[18]。这些算法每秒处理数千笔交易，保证联盟区块链更有效地验证交易和形成新区块，联盟链的特征适合本文方案所需的效率[19-20]。

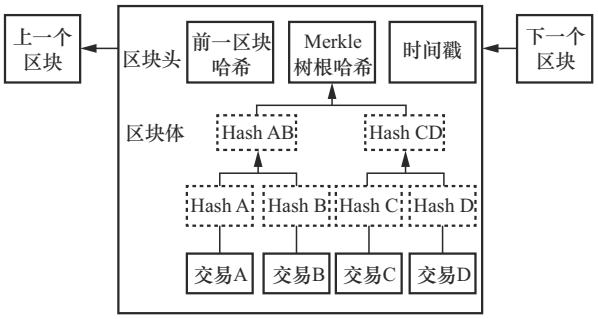


图1　区块及区块链数据结构

Hyperledger Fabric 应用程序涉及以下两个主要组件。

- 智能合约：智能合约在Fabric中又称为链码，从链上实现应用程序逻辑，用图灵完备的编程语言编写，例如GoLang、Java、NodeJS等。
- 背书策略：背书策略定义哪些Peer节点可以执行交易并给予背书。指定的管理员在集合上使用单调逻辑表达式（如“3 out of 5”）定义执行和背书给定事务的背书对等方的子集[13]。

1.4　IPFS

IPFS[21]是一种对等分布式文件系统，其核心特性是内容可寻址，IPFS自动将大的数据文件拆分为多个小数据块文件，分布存储在各个存储器中，然后再依据拆分后的文件内容生成哈希值和存储位置信息。IPFS为每个存储的文件分配一个唯一的标识符，使用文件内容的哈希值作为标识符，而不再依赖传统的文件路径或统一资源定位器（uniform resource locator，URL）。为了增加文件的动态性，IPFS引入了IPNS，IPNS允许节点的PeerID限定的命名空间提供一个指向具体IPFS文件（目录）哈希的指针，用户通过更改指针来随时指向最新的文件内容，而其他人始终可以访问最新的数据。

2　系统模型

2.1　系统架构

方案包括8类实体，分别是数据所有方、数据请求方、分布式存储IPFS服务、可信的属性授权中心（attribute authority，AA）、证书颁发机构（certificate authority，CA）、区块链、智能合约和物联网智能网关。系统架构如图2所示。

（1）数据所有方（data owner，DO）：负责数据访问控制策略的制定。调用合约生成令牌，属性基加密后上传IPFS服务，将令牌获取地址公开保存到区块链状态库。

（2）数据请求方（data user，DU）：获取物联网资源令牌，发送资源请求。

（3）属性授权中心：对系统中与其有交互的数据所有方和数据请求方完全可信，负责CP-ABE算法系统参数的初始化生成与分配，生成密钥和相应参数返回给方案参与方。

（4）IPFS服务：提供分布式存储服务，保存经CP-ABE加密后的令牌密文。

（5）区块链网络：DO、DU和智能网关作为链用户节点参与到区块链网络。

（6）智能合约：负责令牌地址与属性的获取，以及对DU所请求的正确令牌执行去中心化权限评估。

（7）CA服务器：为受信任实体，负责管理链用户数字证书。CA证书扩展域中注册数据请求方属性信息，如角色、姓名、年龄、单位、部门等。

（8）智能网关：负责将物联网设备与云连接，访问和分析传感器数据并作出决策。接收DU发送的资源请求，调用合约完成权限评估，验证DU请求根据RESTful应用程序接口（appli-

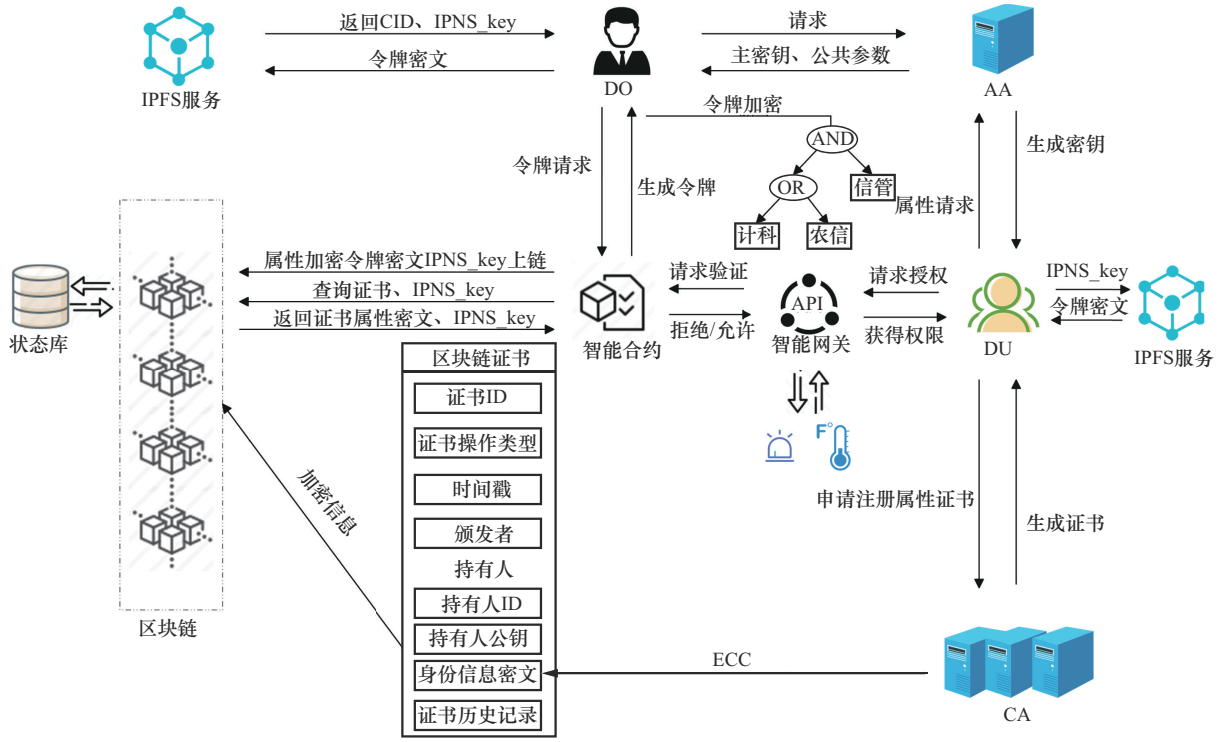


图2　系统架构

cation programming interface，API）将数据返回。

2.2　系统流程

方案系统流程简述如下。

（1）用户属性注册：DU连接Fabric-CA服务器，生成数字证书与公私钥，将用户属性集经椭圆曲线加密（elliptic curve cryptography，ECC）处理后注册到CA证书扩展域。

（2）DO功能令牌颁发：DO调用合约生成JavaScript对象表示法网络令牌（JavaScript object notation Web token，JWT），对明文JWT执行CP-ABE加密，生成密文文件。将该密文文件上传至分布式存储服务IPFS/IPNS，并将返回的密文文件哈希值公开保存到Fabric区块链中，以记录其状态。

（3）DU功能令牌获取：DU调用智能合约获取证书扩展域中保存的属性集密文，本地解密ECC密文得到用户属性集并发送给AA，AA生成属性密钥后返回给DU。DU调用智能合约获取CP-ABE密文文件哈希值，通过该值从IPNS下载密文文件，使用属性密钥执行CP-ABE解密，最终得到明文JWT。

（4）网关权限验证：DU将物联网资源请求与JWT发送给智能网关，智能网关接入区块链网络，并调用智能合约对DU请求执行去中心化权限评估。

（5）访问权限更新：DO执行访问控制策略更新，调用智能合约更新Fabric区块链上所保存的访问控制策略状态，完成新JWT的颁发。更新后的密文文件可按照不变的名称地址IPNS_key重新上传至IPFS。

3　方案详细设计

方案访问控制流程如图3所示，主要包括用户属性注册、DO功能令牌颁发、DU功能令牌获取、网关权限验证和访问权限更新。

3.1　用户属性注册

DU通过Fabric-SDK连接Fabric-CA服务器，输入用户属性集Attrs（key1:“attr1”，key2:

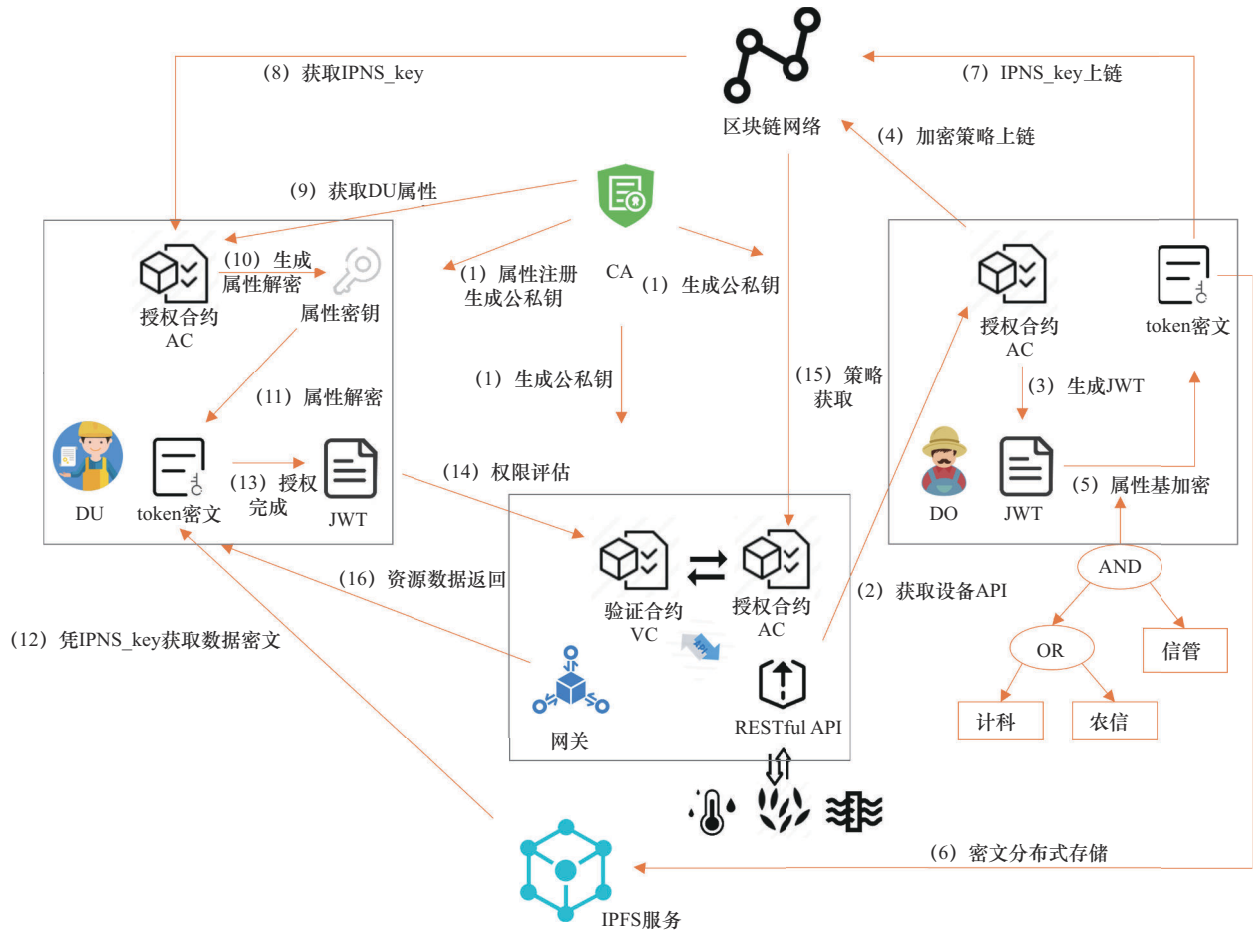


图3　方案访问控制流程

“attr2”, key3:“attr3”）执行身份注册。因公钥基础设施（public key infrastructure，PKI）证书体系是公开透明的，为保证用户数据隐私安全，本地对 DU 所注册属性集 S_{DU} 执行非对称加密，使得 CA 证书扩展域所呈现的为密文状态数据。这里采用 ECC 非对称加密方案，本地为 DU 生成公钥 PK 以及私钥 SK。注册用户算法如算法 1 所示。

算法 1　注册用户算法(uname, password, attrs, PK)

输入　uname, password, attrs, PK

输出　err or success

begin:

// 创建 Fabric CA 客户端实例

HFCAClient caClient = new HFCAClient ("http://ca.example.com", ca 证书)

// 构建注册请求

RegistrationRequest reg = new RegistrationRequest(username)

reg.setSecret(password)

ct_attrs←ECC. Encrypt(PK, attrs)//对属性集 ECC 加密

reg.addAttribute(new Attribute("attrs", ct_attrs))

// 发送注册请求并获取注册响应

enrollmentSecret = caClient.register(reg, admin)

if enrollmentSecret != null

写入区块链钱包

return success

else

return err

end

3.2 功能令牌颁发

AA初始化属性基加密生成公共参数GP，并构造双线性映射$e:(G_1\times G_2)\to G_T$，其中G_1、G_T、G_2均为阶为p的乘法循环群，g为G_1的生成元，Z_p为有限域。生成随机数$a\epsilon Z_p$，并计算$Y=e(g,g)^{\alpha}$，生成随机数$\beta\epsilon Z_p$，计算g^{β}，得到系统主密钥MSK$=g^{\alpha}$，系统公钥MPK$=\{Y,g^{\beta}\}$，将随机的G_T元素导出并执行SHA256哈希运算后的值设为对称的随机密钥（secret key random，SKR）。制定访问控制策略P，并根据访问控制结构构建访问控制树（access control tree，ACT）。调用授权合约（authorization contract，AC）生成JWT，并将访问控制策略$P=$“(A AND B) OR C”以字节数组的形式上链。

使用随机生成的对称密钥对生成的JWT进行AES加密得到密文CT，再将其写入密文属性文件（ciphertext-properties，CTP）。对随机对称密钥加密处理，生成随机数$s\epsilon Z_p$，计算密文组件$C=e(g,g)^{\alpha(\text{alpha}\,s)}$，$C_0=g^s$，将$s$作为秘密沿着ACT进行拆分，使得与叶子节点的属性$i$对应的秘密分片为$\gamma_i$。对于每个叶子节点的属性$i$进行哈希运算得到$H(i)$，以将属性映射到乘法循环群$G_1$上，随机生成$p_i\epsilon Z_p$，计算$C_i^1=g^{\beta\gamma_i}H(i)^{-p_i}$，$C_i^2=g^{p_i}$，最终得到SKR加密密文SKC$=\left\langle C,C_0,\{C_i^1,C_i^2\}_{i\in\text{leafNodes}}\right\rangle$，leafNodes为访问控制树叶子节点属性集合，再次将密文SKC写入CTP文件。

DO上传CTP到IPFS后返回cid，创建IPNS_key并将cid追加到IPNS目录中，调用智能合约链上保存IPNS_key状态并对任何主体公开其获取权限，实现1对多细粒度授权。块数据加密授权算法如算法2所示。

算法2 块数据加密授权算法

输入 privateKey$_{DO}$, JSON$_{token}$, MPK

输出 err or null

begin:

token ← AC. chaincodeQuery (JSON$_{token}$. subject)//查询设备ID是否已在链上注册

if (token == null)

token ← GenerateToken(privateKey$_{DO}$, JSON$_{token}$)//生成JWT，使用DO链用户私钥签名

ct_token = CPABE. Enc (token, MPK, JSON$_{token}$.policy)//执行属性基加密

cid _ipfs = IPFS_Add(ct_token)//密文上传IPFS

cid _ipns = Generate_IPNSkey(type, size)//生成IPNS目录key

cid _ipns = IPFS_Publish(cid _ipns , cid _ipfs)//追加到key目录下

err←AC. chaincodePutstate(SHA(JSON$_{token}$. subject), JSON$_{token}$.policy)

ID ← SHA256(SHA(JSONtoken. subject))// 使用SHA-3哈希算法生成key进行存储

err ←AC. chaincodePutstate(ID, JSONtoken. policy)

err ←AC. chaincodePutstate(JSON$_{token}$. subject, cid _ipns)

if err != null then

return Error (err.Text)

return null

else

return null

end

3.3 功能令牌获取

功能令牌获取分为以下两个阶段。

（1）DU属性密钥的生成。DU凭身份证书连接区块链网络，调用智能合约AC获取证书扩展域中保存的属性集密文CS$_{DU}$，使用本地生成的非对称私钥SK执行解密得到用户明文属性集S_{DU}，将属性集S_{DU}发送给AA生成属性密钥。AA执行密钥生成，选取随机数$t\epsilon Z_p$，计算$D=g^{\alpha}g^{\beta t}$，$D_0=g^t$，对DU所传递的用户属性集S_{DU}生

成属性列表 attList，对属性列表中的每一个属性 i，计算 $D_i=H(i)^t$，得到 $\{D_i\}_{i\epsilon S_{\text{attList}}}$，并将其与上述生成的部分属性私钥结构 D, D_0 进行重组，生成用户属性密钥 $\text{ABESK}=\left\langle D,D_0,\{D_i\}_{i\epsilon S_{\text{attList}}}\right\rangle$，完成属性密钥 ABESK 的生成后返回给 DU。

（2）CTP 文件的解密。CTP 文件的获取地址公开，DU 调用智能合约 AC 在链上获取 IPNS_key，凭借该 key 值从 IPFS 下载 CTP 密文文件。DU 执行密文解密，输入属性密钥 ABESK 与密文 CTP，对于密钥属性集合 S_{DU} 和 ACT 的叶子节点属性集合中重合的属性 i，计算 $P_i=e(C_i^1,D_0)e(C_i^2,D_i)=e(g,g)^{\beta ty_i}$，将所有重合的 P_i 作为秘密分片进行拉格朗日差值因子指数运算处理，以 $e(g,g)^{\beta ts}$ 的形式最终恢复根节点秘密值，计算 $e(C_0,D)=e(g,g)^{\alpha s}e(g,g)^{\beta ts}$，得到 $e(g,g)^{\alpha s}$，最后计算得到明文 $C/e(g,g)^{\alpha s}=\boldsymbol{M}$。

块数据解密授权算法如算法 3 所示。

算法 3　块数据解密授权算法

输入　SK, req_subject, MPK, MSK

输出　token or err

begin:

cid _ipns ←AC.getCid(req_subject) //获取存储地址

ct_token = IPNS_Get(cid _ipns)//获取密文

err, ct_attrs ←AC.getCAttributevalue(stub_{DU}) //获取证书属性

if err != null then

return Error (err.Text)

end if

err,attrs = ECC.Decrypt (ct_attrs, SK)

if err != null then

return Error (err.Text)

end if

abe_sk = CPABE.Keygen(MPK, MSK , attrs)

err,token = CPABE .Dec(ct_token, abe_sk)

if err != null then

return Error (err.Text)

end if

return token

end

3.4　网关权限验证

智能网关在收到 DU 发送来的资源请求和 JWT 后，对 JWT 执行去中心化权限验证。调用验证合约（verification contract，VC）解析令牌（token）JavaScript 对象表示法（JavaScript object notation，JSON）对象，输入 DO 链用户公钥 $\text{publicKey}_{\text{DO}}$ 验证 JSON Web 签名（JWS），确保令牌无篡改且有效。当策略变更，为避免资源请求与令牌内容不一致，需从令牌中获取请求策略 req_policy，判断且仅当 req_policy 与 AC 账本上记录的策略一致时即表示数据请求方可使用该令牌访问这一设备资源。智能合约 VC 对 DO 定义的动态条件约束，如时间、地点、IP 地址等执行权限评估，流程如图 4 所示，经合约循环判断令牌字段如设备、资源、权限和条件约束均满足 DO 所设定的策略后，授权 DU 访问指定的设备资源。块数据验证算法如算法 4 所示。

算法 4　块数据验证算法

输入　$\text{publicKey}_{\text{DO}}$,req_subject,token,req_right

输出　permit or deny

begin:

err = ParseToken(token, $\text{publicKey}_{\text{DO}}$)//执行令牌解析

if err ! = null then

return deny

end if

policy←VC←AC.chaincodeQuery(req_subject)

req_policy = token.getPolicy()

if (policy == req_policy)

condition = token.getCondition()

if (condition is valid && token is not expired)

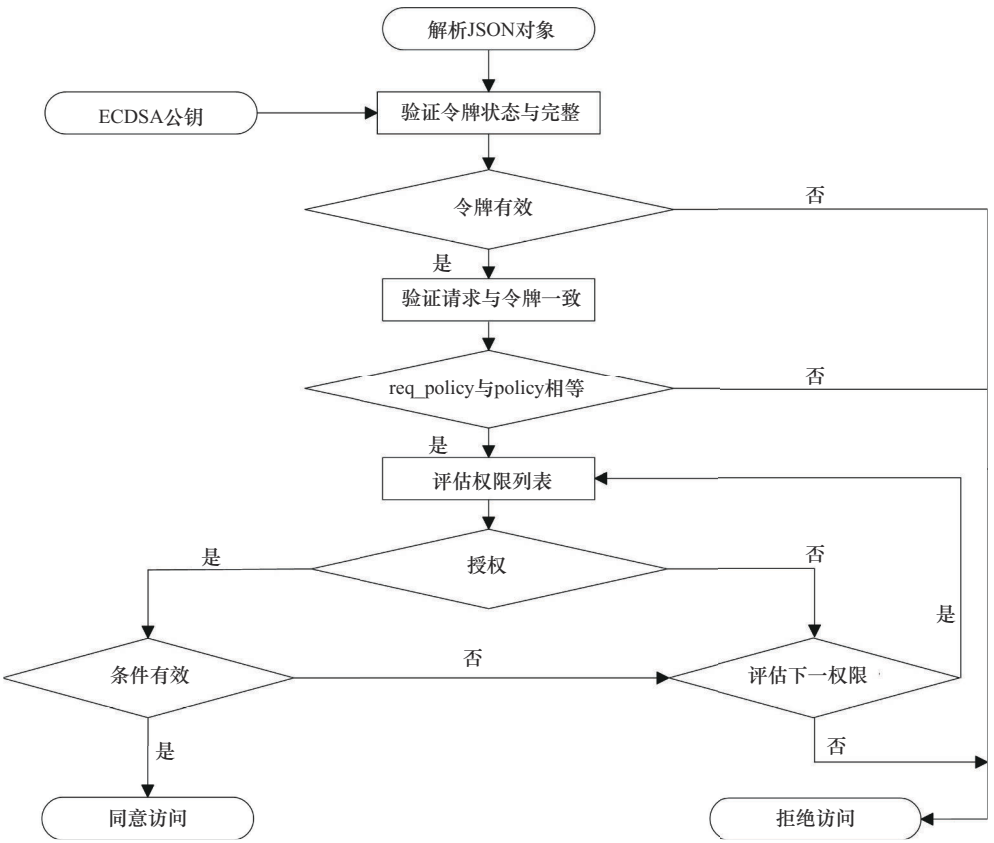


图4　权限评估流程

rights = token.getRights()

for (rights : right) do //循环验证

if (right. resource == req_right. resource && right.action == req_right.action && right.conditions is valid)//检查动态属性

return permit

end if

return deny

end if

return deny

end if

end

3.5　访问权限更新

执行访问权限变更，DO设置新的访问控制策略P，并根据本地访问策略，重新生成令牌JSON对象，DO可以为请求方制定访问控制时需要满足的其他约束。根据更新后的策略P重新构建ACT，DO重新执行属性基加密，随机生成AES对称加密密钥SKR，对明文$\boldsymbol{M}$、对称密钥SKR重新加密得到新的属性基加密文件CTP。将密文数据CTP上传至IPFS，返回的cid追加到IPNS目录，由于旧的令牌在发生更新时不再具有最新的访问权限，因此在从IPFS检索访问权限时，数据所有者始终引用IPNS文件目录中的最后一条记录。最后调用智能合约AC更新区块链状态库上保存的策略状态。

4　实验仿真与分析

4.1　安全性分析

本文方案实现了基于属性的访问控制，而方案的安全性主要由CP-ABE方案的安全性与Fabric-CA的安全性保证，所以本节从属性角度出发对CA安全性以及CP-ABE方案进行安全性分析。

4.1.1　CA 安全性

Fabric-CA 通过传输层安全（transport layer security，TLS）/安全套接层（secure sockets layer，SSL）协议保护传输中的用户属性集，确保注册过程安全，并通过私钥保护和证书生命周期管理保障证书签发和管理的安全性。此外，Fabric-CA 作为 Fabric 区块链的内置组件，采用角色和权限管理、多因素认证增强访问控制，记录操作日志并进行异常检测和响应，确保操作透明并且可追踪。

4.1.2　CP-ABE 安全性

基于挑战者和攻击者之间的博弈过程，证明方案在选择明文攻击（chosen plaintext attack，CPA）下的安全性。

定义 1　如果在多项式时间内，任何攻击者都不能以显著的概率赢得游戏，则该方案是安全的。

证明　定义挑战者与攻击者的博弈过程，挑战者根据规则响应攻击者的查询。

（1）系统初始化阶段：挑战者 B 运行初始化算法 Setup，生成公钥 PK 并保存系统主密钥 MK。攻击者 A 选择一个访问控制策略 A^*，并将其发送给挑战者 B。

（2）密钥查询阶段 1：攻击者 A 选择一系列不满足访问控制策略 A^* 的属性集合 $T_1, T_2, \cdots, T_q$。挑战者 B 计算并返回 $T_1, T_2, \cdots, T_q$ 对应的属性私钥给攻击者 A。

（3）挑战阶段：攻击者 A 选择两条等长消息 ME_0 和 ME_1，并提交给挑战者 B。挑战者 B 随机选择一个比特 $\beta \epsilon \{0,1\}$，使用公钥 PK 加密 ME_β，生成挑战密文 C^* 并发送给攻击者 A。

（4）密钥查询阶段 2：攻击者 A 再次选择一系列不满足访问控制策略 A^* 的属性集合 $T_{q+1}, T_{q+2}, \cdots, T_r$，并向挑战者 B 请求这些属性的私钥。挑战者 B 计算并返回 $T_{q+1}, T_{q+2}, \cdots, T_r$ 对应的属性私钥给攻击者 A。

（5）猜测阶段：攻击者 A 输出对 β 的猜测值 β^*，其中 $\beta^* \epsilon \{0,1\}$。如果 $\beta^* = \beta$，则攻击者 A 赢得游戏。

（6）安全性定义：CP-ABE 方案在选择明文攻击下的安全性可以通过攻击者 A 在猜测阶段的成功率来衡量，安全性优势 Adv_A 定义为 $Adv_A = \left| Pr_{[\beta^* = \beta]} - \frac{1}{2} \right|$，其中，$Pr_{[\beta^* = \beta]}$ 表示攻击者 A 正确猜测 β 的概率。CP-ABE 方案在选择明文攻击下的安全性要求 Adv_A 必须是一个非常小的值，通常是小于一个可以忽略的常数，以此证明方案对于攻击者的能力是具有挑战性的。

4.2　实验仿真

基于 Docker 集成区块链实验网络，由 2 个组织（Org1、Org2）、4 个 Peer 节点、2 个 CA 节点、4 个 couchDB 节点、1 个 Orderer 节点、4 个 AC 链码节点、4 个 VC 链码节点和 3 个 ipfs_node 存储节点组成，智能合约的开发采用 Golang 语言，并在配置 2 GB RAM 的 Ubuntu 18.04 虚拟机上完成构建。为了评估在 IPFS 中存储与下载令牌密文的时间成本，构建一个由 3 个服务器节点组成的 IPFS 私有网络。用户设备使用 Fabric-SDK 与区块链进行交互，实验通过 Fabric-SDK-Node 调用区块链底层网络完成访问控制，模拟发送请求从服务器访问并获取设备传感器温度、湿度和灯光亮度具体数据。

令牌结构如图 5 所示，定义 JSONObject 各属性字段，“sub”为物联网设备 ID，“iss”定义令牌颁发者，“iat”表示签发时间，“nbf”和“exp”指定功能令牌的生存期，“address”为存储在 IPNS 服务中的功能令牌哈希标识，“policy”为令牌 CP-ABE 加密策略，“rights”定义设备资源详情以及对应的权限操作，其中“resource”为设备资源应用程序接口，“right”为权限操作，“conditions”为限制数据请求方的条件约束，包含动态属性，如时间、地点、IP 地址等。根据本地的访问策略，数据所有者可为请求对象制定访问控制时需要满足的其他约束。

```
"sub": "89458",
"iss" : "张三",
"iat" : "1696900384",
"nbf": "1696903984",
"exp": "1696904984",
"address": "k2k4r8p8mw07c7pixw59p20915p9ignf86tnbidk4gjqByus 5peukeue",
"policy": "院系 ： 信管 AND (专业 ： 农艺 OR 专业：农信)",
"rights":[{
  "resource": "89458/sensors/temp1",
  "right": "GET"},
  {
  "resource": "89458/sensors/lamp1",
  "right": "POST",
  "conditions":
    {"type":"Timespan",
     "value":{"startTime":1703211266,
              "endTime":1703211266}}
  },
  {
  "resource": "89458/sensors/hum1",
  "right": "GET"}]
```


图5　令牌结构

4.2.1　功能令牌颁发

实验设计采用开源CP-ABE的快速基于属性的消息加密（fast attribute-based message encryption，FAME）[22]，使用Golang进行程序设计。为了更好地分析和证明本方案具有的优势，方案与文献[14]设置相同策略“Division：IS AND Role：Student”执行令牌加密完成授权，发布令牌如图6所示。

4.2.2　功能令牌获取

DU访问设备资源，调用智能合约AC通过设备ID获取在链上保存的令牌IPNS地址，解密后获取令牌，如图7所示。

4.2.3　权限验证与评估

当DU所发送的请求满足DO所定义的设备、资源、权限和约束条件时，权限评估的返回结果会作为物联网网关调用DU所请求设备资源的RESTful API，成功授权后物联网网关会将通过RESTful API获取的资源数据返回给DU，详细的链码执行日志如图8所示。

授权成功如图9所示，经合约评估授权成功后区块链底层网络将RESTful API响应{"resource":"89458/sensors/hum1","right":"GET"}返回物联网网关，物联网网关凭借该RESTful API获取湿度传感器实时数据并返回DU。


```
zhangchuanhu@zhangchuanhu-virtual-machine:~/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript$ node invoke.js
Wallet path: /home/zhangchuanhu/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript/wallet
Transaction has been submitted. Response: eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZGRyZXNzIjoiUW1YTVBuVlo1eDczUDdTTkNQeEFIVkwyU202NEpFaG9ZODVZMnhYZEF2YlJ2YSIsImV
4cCI6MTcwNDAyMzU0NywiaWF0IjoiMTcwNDAxNjQwNyIsImlzcyI6IkFsaWNlIiwibmJmIjoxNzA0MDE2MzQ3LCJwb2xpY3kiOiJJUyBBTkQgU3RlZGVudCIsInJpZ2h0cyI6Ilt7XCJyZXNvdXJjZVwiOlwiODk0N
Tgvc2Vuc29ycy9odW0xXCIsXCJyaWdodFwiOlwiR0VUXCJ9LHtcInJlc291cmNlXCI6XCI4OTQ10C9zZW5zb3JzL2xhbTFcIixcInJpZ2h0XCI6XCJQT1NUXCJ9XSIsInN1YiI6Ijg5NDU4In0.b7Lw55XroAEBCd6
EOnkeAnbkjGwqIDTioHaKYGmJsbzRmdRk0-9-zlPj55HChX5dG5CEKtuEbO30MCRwsPFRCA
zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ ./cpabe_encrypt -token eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZGRyZXNzIjoiUW1YTVBuVlo1eDczUDdTTkNQeEFIV
kwyU202NEpFaG9ZODVZMnhYZEF2YlJ2YSIsImV4cCI6MTcwNDAyMzU0NywiaWF0IjoiMTcwNDAxNjQwNyIsImlzcyI6IkFsaWNlIiwibmJmIjoxNzA0MDE2MzQ3LCJwb2xpY3kiOiJJUyBBTkQgU3RlZGVudCIsInJ
pZ2h0cyI6Ilt7XCJyZXNvdXJjZVwiOlwiODk0NTgvc2Vuc29ycy9odW0xXCIsXCJyaWdodFwiOlwiR0VUXCJ9LHtcInJlc291cmNlXCI6XCI4OTQ10C9zZW5zb3JzL2xhbTFcIixcInJpZ2h0XCI6XCJQT1NUXCJ9X
SIsInN1YiI6Ijg5NDU4In0.b7Lw55XroAEBCd6EOnkeAnbkjGwqIDTioHaKYGmJsbzRmdRk0-9-zlPj55HChX5dG5CEKtuEbO30MCRwsPFRCA -policy ./policy.txt -ct_token ./ct_token.txt
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZGRyZXNzIjoiUW1YTVBuVlo1eDczUDdTTkNQeEFIVkwyU202NEpFaG9ZODVZMnhYZEF2YlJ2YSIsImV4cCI6MTcwNDAyMzU0NywiaWF0IjoiMTcwNDAxNjQwN
yIsImlzcyI6IkFsaWNlIiwibmJmIjoxNzA0MDE2MzQ3LCJwb2xpY3kiOiJJUyBBTkQgU3RlZGVudCIsInJpZ2h0cyI6Ilt7XCJyZXNvdXJjZVwiOlwiODk0NTgvc2Vuc29ycy9odW0xXCIsXCJyaWdodFwiOlwiR0V
UXCJ9LHtcInJlc291cmNlXCI6XCI4OTQ10C9zZW5zb3JzL2xhbTFcIixcInJpZ2h0XCI6XCJQT1NUXCJ9XSIsInN1YiI6Ijg5NDU4In0.b7Lw55XroAEBCd6EOnkeAnbkjGwqIDTioHaKYGmJsbzRmdRk0-9-zlPj5
5HChX5dG5CEKtuEbO30MCRwsPFRCA
加密成功!
加密完成：./ct_token.txt
Method execution time: 9.231046ms
zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ ipfs add ct_token.txt
added QmYHhSMeoKJmFCrtz7hBZyDSXGXTPNgx2giNpc7g6WHvYM ct_token.txt
 7.03 KiB / 7.03 KiB [=============================================================================================================================================] 100.00%
zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ ipfs key gen --type=rsa --size=2048 iot89458
k2k4r8p8mw07c7pixw59p2o9i5p9ignf86tnbidk4gjq8yus5peukeue
zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ time ipfs name publish -key=iot89458 QmYHhSMeoKJmFCrtz7hBZyDSXGXTPNgx2giNpc7g6WHvYM
Published to k2k4r8p8mw07c7pixw59p2o9i5p9ignf86tnbidk4gjq8yus5peukeue: /ipfs/QmYHhSMeoKJmFCrtz7hBZyDSXGXTPNgx2giNpc7g6WHvYM

zhangchuanhu@zhangchuanhu-virtual-machine:~/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript$ time node invoke.js AddToken
Wallet path: /home/zhangchuanhu/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript/wallet
Transaction has been submitted. Response: 令牌信息已发布上链!
```

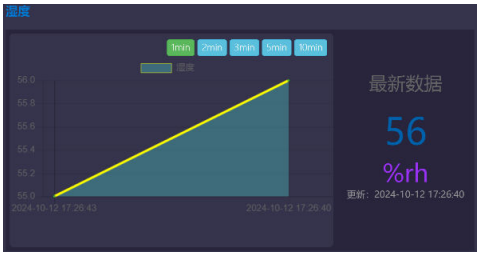

图6　发布令牌

```
zhangchuanhu@zhangchuanhu-virtual-machine:~/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript$ time node query.js GetCid 89458
Wallet path: /home/zhangchuanhu/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is: k2k4r8p8mw07c7pixw59p2o9i5p9ignf86tnbidk4gjq8yus5peukeue

zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ ipfs get /ipns/k2k4r8p8mw07c7pixw59p2o9i5p9ignf86tnbidk4gjq8yus5peukeue
Saving file(s) to k2k4r8p8mw07c7pixw59p2o9i5p9ignf86tnbidk4gjq8yus5peukeue
 7.03 KiB / 7.03 KiB [=============================================================================================] 100.00% 0s
zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ cp k2k4r8p8mw07c7pixw59p2o9i5p9ignf86tnbidk4gjq8yus5peukeue ct_token.txt

zhangchuanhu@zhangchuanhu-virtual-machine:~/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript$ time node query.js GetAttrs
Wallet path: /home/zhangchuanhu/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript/wallet
Transaction has been evaluated, result is: eyJkaXZpc2lvbiI6IklTIiwicm9sZSI6IlN0dWRlbnQifQ==

zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ ./cpabe_keygen -attrs eyJkaXZpc2lvbiI6IklTIiwicm9sZSI6IlN0dWRlbnQifQ== -seckey ./SecKey.pem
密钥文件已生成：./SecKey.pem
Method execution time: 5.852261ms

zhangchuanhu@zhangchuanhu-virtual-machine:~/桌面/CP-ABE$ ./cpabe_decrypt -ct token ./ct_token.txt -seckey ./SecKey.pem
成功获得令牌：eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZGRyZXNzIjoiUW1YTVBuVlo1eDczUDdTTkNQeEFIVkwyU202NEpFaG9ZODVZMnhYZEF2YlJ2YSIsImV4cCI6MTcwNDAyMzU0NywiaWF0IjoiMTcwNDAxNjQwNyIsImlzcyI6IkFsaWNlIiwibmJmIjoxNzA0MDE2MzQ3LCJwb2xpY3kiOiJJUyBBTkQgU3R1ZGVudCIsInJpZ2h0cyI6Ilt7XCJyZXNvdXJjZVwiOlwiODk0NTgvc2Vuc29ycy9odW0xXCIsXCJyaWdodFwiOlwiR0VUXCJ9LHtcInJlc291cmNlXCI6XCI4OTQ10C9zZW5zb3JzL2xhbTFcIixcInJpZ2h0XCI6XCJQT1NUXCJ9XSIsInN1YiI6Ijg5NDU4In0.b7Lw55XroAEBCd6E0nkeAnbkjGwqIDTioHaKYGmJsbzRmdRk0-9-zlPj55HChX5dG5CEKtuEb030MCRwsPFRCA
```

图 7　获取令牌

```
zhangchuanhu@zhangchuanhu-virtual-machine:~/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript$ node invoke.js Evaluate
Wallet path: /home/zhangchuanhu/go/src/github.com/hyperledger/fabric-samples/fabcar/javascript/wallet
Transaction has been submitted. Response: {"resource":"89458/sensors/hum1","right":"GET"}

获取属性交易编号Txid:254d522862915bb359ea16e33a3c2b46c3fef3d15a5a7609959c10864e168a90
操作链用户id:
0f6ad74c6f7b8c02c2ce95dac0a1e4177056c849b01ace08381a1a12247689bb
解析令牌交易编号Txid:2120bed79c9645bcedb765698625e50c34e37298af5debded37d769198ffaa11
签名验证通过!
Token is valid
Claims: map[address:QmXMPnVZ5x73P7SNCPxAHVL2Sm64JEhoY85Y2xXdAvbRva exp:1.704023547e+09 iat:1704016407 iss:Alice nbf:1.704016347e+09 policy:IS AND Student rights:[
{"resource":"89458/sensors/hum1","right":"GET"},{"resource":"89458/sensors/lam1","right":"POST"}] sub:89458]
IS AND Student
IS AND Student
req policy与policy验证通过!
资源89458/sensors/hum1已授权!
解析以及权限评估时间 2.941418ms
```

图 8　链码执行日志




```
{"ResultObj":{"Unit":"%rh","ApiTag":"hum1","Groups":1,"Protocol":1,"Name":"湿度","CreateDate":"2024-01-22 14:25:11","TransType":0,
 "Coordinate":null,"Value":56,"RecordTime":"2024-10-12 17:29:38"},"Status":0,"StatusCode":0,"Msg":null,"ErrorObj":null}
------数据获取成功：56%rh
```

图 9　授权成功

4.3　性能分析

本文方案授权效率由 3 个方面的因素决定。CP-ABE 加密和解密过程的效率取决于访问策略的复杂性和设备的计算能力。加密过程对比解密耗时相对较长，尤其是对于复杂的访问策略。解密涉及静态属性和动态约束的匹配，需要确保操作的实时性和效率。区块链与 IPNS 的结合确保了令牌密文获取地址在令牌更新时的稳定性，从而避免了分布式存储环境下的地址变化问题，提高系统的可靠性和稳定性。Hyperledger Fabric 智能合约的性能决定着本文方案最终的去中心化权限验证和评估的效率。

4.3.1　属性基加密

DO 授权时间消耗如图 10 所示，属性基加密

阶段设置策略属性数量分别为3、6、9、12，按照属性数量的不同进行耗时对比。可以看出IPFS存储效率非常稳定，令牌存储时间并没有太大变化，而链码软件开发工具包调用占用了绝大部分时间，由于Fabric默认出块时间为2 s，通过Node-SDK调用AddToken()合约函数会在链上产生交易消耗较多时间，该阶段中随着属性数量增加，CP-ABE加密运行时间增长较慢。

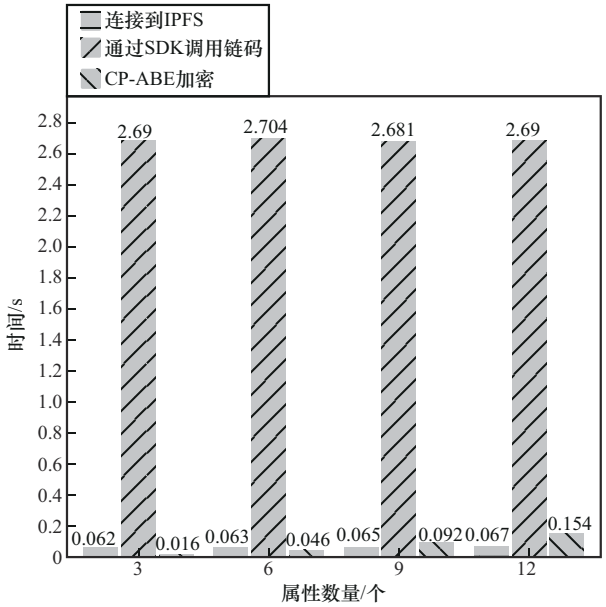


图10　DO授权时间消耗

授权时间对比如图11所示，可以看出，同属授权阶段在时间耗费方面本文方案优于文献[14]方案。原因在于文献[14]方案选择在分布式账本存储令牌密文信息，导致令牌分布式存储产生过长的时间消耗。

为体现本文方案算法优势性，分别对文献[16]、文献[23]所提BSW-CP-ABE、LLW-CP-ABE属性基加密算法进行程序仿真并应用于本文访问控制方案，属性基加密对比如图12所示，可以看出FAME方案随着属性数量增加其在加密效率以及稳定性上优于其他方案，因此更适用于访问控制场景。

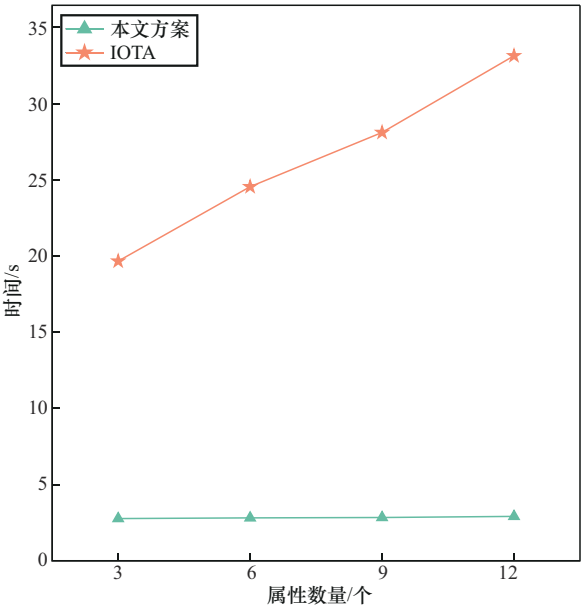


图11　授权时间对比

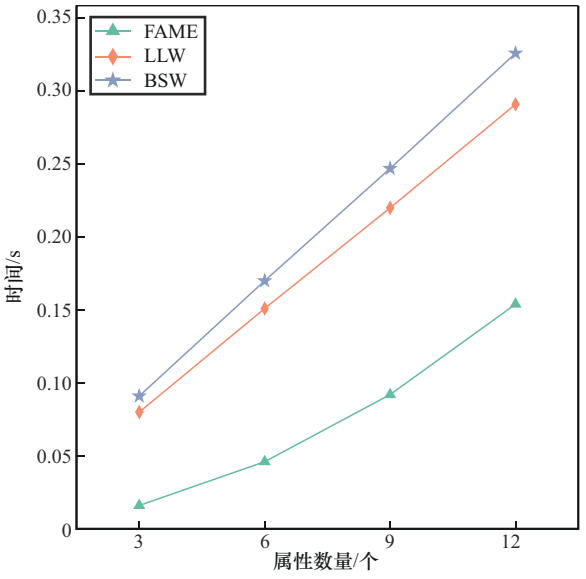


图12　属性基加密对比

4.3.2　属性基解密

DU授权时间消耗如图13所示，属性基解密阶段时间消耗在CP-ABE解密、IPNS令牌下载和链码调用，此过程中Fabric从链上读取数据不会产生交易，所以此次链码调用不会有太多时间消耗。随着属性数量增加，CP-ABE解密运行时间增长缓慢，整体较为稳定。

4.3.3　链码性能

在区块链网络中对链码性能进行评估和测试，选择开源的基准测试工具 Hyperledger Caliper 通过压力测试来测量区块链网络的性能。理论上，Hyperledger Fabric 的吞吐量可以达到每秒 3 500 笔交易[13]。但由于测试环境的限制，仅在原型中引入了一个 Orderer 节点，且使用 Solo 共识算法进行执行。因此，在测试过程中，系统的吞吐量最高仅达到每秒 350 笔交易。

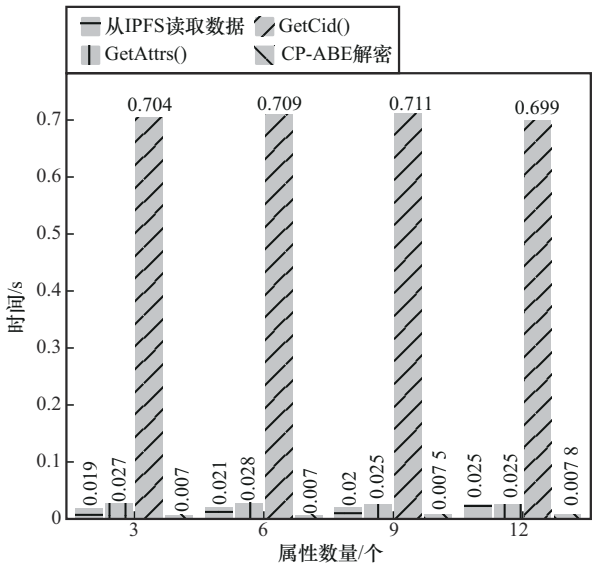
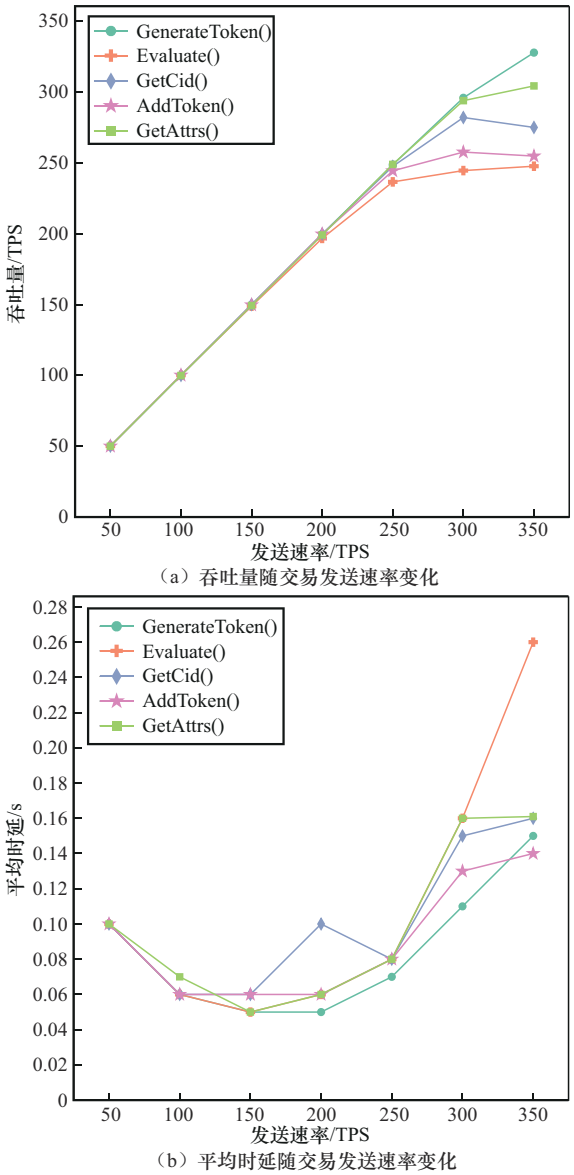


图 13　DU 授权时间消耗

为了测试实验中所设计链码的性能，设置可变的交易发送速率，范围为每秒 50～350 笔交易，固定区块大小为 10，并且发送 1 000 个事务来测试主要合约函数。链码吞吐量随交易发送速率变化如图 14（a）所示，所测试的链码函数中评估函数 Evaluate 需执行 JWS 验签以及较复杂的权限判断逻辑，所以当实际发送速率达到每秒 300 笔交易时，其吞吐量达到最高每秒 247 笔交易，而其他设计包括令牌生成 GenerateToken 函数、密文地址获取 GetCid 函数、令牌发布 AddToken 函数以及 CA 属性获取 GetAttrs 函数在访问过程中只是执行简单的合约查询和调用，具体为状态的读出与写入，吞吐量最高可达到每秒 327 笔交易。系统的吞吐量随着请求数量的增加而增加，当吞吐量达到一定值时趋于稳定。

平均时延随交易发送速率变化如图 14（b）所示。从时间角度来看，随着请求并发数量的增加，链码查询和调用操作的平均时延会逐渐增长，最终会降低至趋于稳定，相较于其他合约函数执行简单的状态读出与写入，权限评估 Evaluate 函数因其功能性需要平均时延最高达到 0.26 s，但其整体性能表现满足权限验证需求。



（a）吞吐量随交易发送速率变化

（b）平均时延随交易发送速率变化

图 14　链码性能

5 结束语

本文针对物联网资源的授权问题，提出了一种基于智能合约与属性基加密的访问控制方案。方案通过Fabric智能合约将CP-ABE与权能访问控制[24]模型结合实现对多主体的资源授权，有效缓解了中心化授权压力。链上解析功能令牌并对请求方所发送请求进行分布式权限验证，该过程实现了去中心化的基于属性的访问控制，对动态属性的约束使其达到细粒度访问控制的效果。因此，本文为物联网环境下在区块链网络中实现ABAC模型提供了一种安全、可行的方案。仿真实验与性能分析表明，本方案的授权效率优于现有方案且链码具有良好性能，可以满足应用需求。

参考文献：

[1] GÜRFIDAN R, ERSOY M. A new approach with blockchain based for safe communication in IoT ecosystem[J]. Journal of Data, Information and Management, 2022, 4(1): 49-56.

[2] CHELLAPPAN V, SIVALINGAM K M. Security and privacy in the Internet of things[M]//Internet of Things. Amsterdam: Elsevier, 2016: 183-200.

[3] NAKAMOTO S.Bitcoin: A peer-to-peer electronic cash system [EB]. 2008.

[4] WANG Q, ZHU X Q, NI Y Y, et al. Blockchain for the IoT and industrial IoT: a review[J]. Internet of Things, 2020, 10: 100081.

[5] GUO S Y, HU X, GUO S, et al. Blockchain meets edge computing: a distributed and trusted authentication system[J]. IEEE Transactions on Industrial Informatics, 2020, 16(3): 1972-1983.

[6] ZHANG L H, LI B P, FANG H D, et al. An Internet of things access control scheme based on permissioned blockchain and edge computing[J]. Applied Sciences, 2023, 13(7): 4167.

[7] GONG L Q, ALGHAZZAWI D M, CHENG L. BCoT sentry: a blockchain-based identity authentication framework for IoT devices[J]. Information, 2021, 12(5): 203.

[8] CRUZ J P, KAJI Y, YANAI N. RBAC-SC: role-based access control using smart contract[J]. IEEE Access, 2018(3): 12240-12251.

[9] LIU H, HAN D Z, LI D. Fabric-IoT: a blockchain-based access control system in IoT[J]. IEEE Access, 2020(8): 18207-18218.

[10] ZHANG G F, CHEN X, FENG B, et al. Research on a safe and reliable agricultural product traceability system driven by permissioned BlockChain technology[M]//Lecture Notes in Electrical Engineering. Singapore: Springer Nature Singapore, 2022: 955-966.

[11] 王秀利, 江晓舟, 李洋. 应用区块链的数据访问控制与共享模型[J]. 软件学报, 2019, 30(6): 1661-1669.
WANG X L, JIANG X Z, LI Y. Model for data access control and sharing based on blockchain[J]. Journal of Software, 2019, 30(6): 1661-1669.

[12] WU A X, ZHANG Y H, ZHENG X K, et al. Efficient and privacy-preserving traceable attribute-based encryption in blockchain[J]. Annals of Telecommunications, 2019, 74(7): 401-411.

[13] QIN X M, HUANG Y F, YANG Z, et al. LBAC: a lightweight blockchain-based access control scheme for the Internet of things[J]. Information Sciences, 2021(554): 222-235.

[14] ZHANG Y Y, NAKANISHI R, SASABE M, et al. Combining IOTA and attribute-based encryption for access control in the Internet of things[J]. Sensors, 2021, 21(15): 5053.

[15] SILVANO W F, MARCELINO R. Iota Tangle: a cryptocurrency to communicate Internet-of-things data[J]. Future Generation Computer Systems, 2020(112): 307-319.

[16] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]//Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07). Piscataway: IEEE Press, 2007: 321-334.

[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. New York: ACM Press, 2018: 1-15.

[18] ONGARO D, OUSTERHOUT J. In search of an understandable consensus algorithm[J]. Proceedings of the 2014 USENIX Annual Technical Conference, USENIX ATC 2014, 2014: 305-319.

[19] LI Z T, KANG J W, YU R, et al. Consortium blockchain for secure energy trading in industrial Internet of things[J]. IEEE Transactions on Industrial Informatics, 2018, 14(8): 3690-3700.

[20] DING S, CAO J, LI C, et al. A novel attribute-based access control scheme using blockchain for IoT[J]. IEEE Access, 2019(7): 38431-38441.

[21] ZHENG Q H, LI Y, CHEN P, et al. An innovative IPFS-based storage model for blockchain[C]//Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI). Piscataway: IEEE Press, 2018: 704-708.

[22] AGRAWAL S, CHASE M. FAME: fast attribute-based message encryption[C]//Proceedings of the 2017 ACM SIGSAC

Conference on Computer and Communications Security. New York: ACM Press, 2017: 665-682.

[23] LI D W, CHEN J, LIU J W, et al. Efficient CCA2 secure revocable multi-authority large-universe attribute-based encryption [M]//WEN S, WU W, CASTIGLIONE A, eds. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017: 103-118.

[24] XU R, CHEN Y, BLASCH E, et al. Blendcac: A blockchain-enabled decentralized capability-based access control for IoTs[C]// 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). Piscataway: IEEE Press, 2018: 1027-1034.

[作者简介]

孙昌霞（1976– ），女，博士，河南农业大学教授、硕士生导师，主要研究方向为区块链、信息安全、农业大数据等。

张传虎（1998– ），男，河南农业大学硕士生，主要研究方向为区块链、农业信息化。

刘冰杰（1991– ），女，博士，河南农业大学讲师，主要研究方向为边缘计算、资源分配和博弈论。

Yingjie Yang（1965– ），男，博士，德蒙福特大学教授，主要研究方向为灰色系统、模糊集、粗糙集、神经网络及其在土木工程、交通运输、环境工程和管理科学中的应用。

Fernando Bação（1968– ），男，博士，新里斯本大学教授，主要研究方向为决策支持系统、机器学习和数据科学。

刘倩（1978– ），女，河南农业大学讲师，主要研究方向为数据库应用及农业大数据应用。