自动化学报
ACTA AUTOMATICA SINICA

基于区块链的策略隐藏大数据访问控制方法

林莉 储振兴 刘子萌 郭馥宾 解晓宇 张建标

A Policy-hidden Big Data Access Control Method Based on Blockchain

LIN Li, CHU Zhen-Xing, LIU Zi-Meng, GUO Fu-Bin, XIE Xiao-Yu, ZHANG Jian-Biao

在线阅读 View online: https://doi.org/10.16383/j.aas.c211178

基于区块链的策略隐藏大数据访问控制方法

林 莉 [1, 2] 储振兴 [1, 2] 刘子萌 [1, 2] 郭馥宾 [1, 2] 解晓宇 [1, 2] 张建标 [1, 2]

摘 要 针对大数据应用中用户共享数据的访问控制由半可信云服务商实施所带来的隐私泄露、策略和访问日志易被篡改等问题, 提出一种基于区块链的策略隐藏大数据访问控制方法 (A policy-hidden big data access control method based on blockchain, PHAC). 该方法采用区块链技术实施访问控制以减少对服务商的信任依赖, 引入属性基加密 (Attribute-based encryption, ABE) 以及双线性映射技术, 实现在不泄露访问控制策略的前提下, 通过智能合约正确执行访问控制策略. 同时, 解耦访问控制策略, 简化用户策略的发布、更新和执行. 并应用链上和链下存储相结合方式, 解决智能合约和访问控制策略占用区块链节点资源不断增大的问题. 最后, 对该方法进行了理论分析和 HyperLedger Fabric 环境下的实验评估, 结果表明该方法能在策略隐藏情况下有效实现访问控制, 但不会给数据拥有者、区块链节点增加过多额外计算和存储开销.

关键词 数据共享, 访问控制, 区块链, 策略隐藏, 智能合约

引用格式 林莉, 储振兴, 刘子萌, 郭馥宾, 解晓宇, 张建标. 基于区块链的策略隐藏大数据访问控制方法. 自动化学报, 2023, **49**(5): 1031−1049

DOI 10.16383/j.aas.c211178

A Policy-hidden Big Data Access Control Method Based on Blockchain

LIN Li[1, 2] CHU Zhen-Xing[1, 2] LIU Zi-Meng[1, 2] GUO Fu-Bin[1, 2] XIE Xiao-Yu[1, 2] ZHANG Jian-Biao[1, 2]

Abstract In the current big data application, the access control of user shared data is implemented by the incomplete trusted cloud service provider, which brings problems such as privacy disclosure, policy and access log easy to be tampered. To solve this problem, this paper presents a policy-hidden big data access control method based on blockchain (PHAC), which exploits blockchain technology to implement access control to reduce the reliance of data owners on cloud servers. Attribute-based encryption (ABE) and bilinear mapping are introduced to implement access control policies correctly through smart contracts without disclosing access control policies. Meanwhile, access control policies are decoupled to simplify their release, update and execution. The combination of on-chain and off-chain storage is applied to solve the problem that smart contracts and access control policies occupy too much blockchain node resources. Finally, theoretical analysis and comprehensive experiments in the HyperLedger Fabric environment have been conducted, which demonstrate the effectiveness of the proposed method. It can effectively implement access control while supporting access control policies hidden, however it does not impose too much extra computing and storage overhead on data owners and blockchain nodes.

Key words Data sharing, access control, blockchain, policy-hidden, smart contract

Citation Lin Li, Chu Zhen-Xing, Liu Zi-Meng, Guo Fu-Bin, Xie Xiao-Yu, Zhang Jian-Biao. A policy-hidden big data access control method based on blockchain. *Acta Automatica Sinica*, 2023, **49**(5): 1031−1049

社会信息化和网络化的飞速发展, 导致用户数据爆炸式增长, 云计算、大数据技术的深度应用催生了数据资源开放共享中的数据安全和隐私保护问题. 访问控制是确保数据不被未授权者访问的一种重要手段, 在大数据外包存储模式下, 用户数据的访问控制策略通常由云服务商执行. 然而, 云服务商经常是非完全可信的, 例如 2018 年 Facebook 有超过 500 万用户信息外泄, 2019 年 Gnosticplayers 出售用户数据, 2020 年中信银行信息泄露等安全事件, 都表明存在服务商非法执行访问控制策略、随意篡改策略或访问日志等安全风险. 由于区块链具有去中心化、透明性、不可篡改性等特点, 将其与访问控制技术相结合, 既可减少对云服务商的信任依赖, 确保访问控制策略和访问日志不被随意篡改, 又可利用智能合约及分布式协商机制, 实现链中访问控制策略的正确自动执行. 目前已有研究者提出利用区块链实现访问控制[1−3], 以确保云计算、大数

收稿日期 2021-12-09 录用日期 2022-07-06
Manuscript received December 9, 2021; accepted July 6, 2022
国家自然科学基金 (61502017), 北京市自然科学基金 (M21039) 资助
Supported by National Natural Science Foundation of China (61502017) and Natural Science Foundation of Beijing Municipality (M21039)
本文责任编委 张俊
Recommended by Associate Editor ZHANG Jun
1. 北京工业大学信息学部 北京 100124 2. 北京工业大学可信计算北京市重点实验室 北京 100124
1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124 2. Beijing Key Laboratory of Trusted Computing, Beijing University of Technology, Beijing 100124

据等场景下的数据安全共享.

现有基于区块链的访问控制研究主要包括基于交易进行策略/权限管理和基于智能合约进行访问控制两个方面[3]. 主流解决方案对应策略存储区块[4−6]和策略直接写入智能合约[7−9]两类. 策略存储区块的工作把区块链当成访问控制中策略管理的数据库[4], 同时引入一个可信的策略执行点和借助智能合约实现属性权威 (Attribute authority, AA)、策略管理点和策略决策点功能, 使得用户可随时查看策略, 并利用智能合约保证策略决策被自动执行, 可避免任何一方通过篡改策略实现未授权数据访问等欺诈行为. 策略直接写入智能合约工作, 借助智能合约自动执行的特性, 使得合约中的访问控制策略自动实施, 同时利用区块链的不可篡改性、透明性和协商一致性原则, 保证智能合约下策略决策的正确性、透明性和可审计性.

然而, 一方面由于访问控制策略是实现合法用户获取有效访问权限的约束条件, 策略中往往包含合法用户身份、属性等敏感信息, 故无论是策略存储区块, 还是把策略直接写进智能合约的方式, 任何人均可随意查看策略, 自然带来数据拥有者隐私泄露风险[10−11], 例如某患者医疗数据的访问控制策略规定只允许神经科医生访问该数据, 攻击者易从该策略推断该患者患有神经方面的疾病. 另一方面, 由于区块链只可增加和查看, 不可修改和删除, 所以随着用户及其数据量的增加, 策略规模不断增大, 区块链中部署的策略和智能合约越来越多, 导致区块链节点的存储和计算资源开销不断增大. 因此, 如何避免策略执行中数据拥有者的隐私泄露, 同时降低智能合约和策略部署给区块链节点引入的资源开销, 成为基于区块链访问控制技术亟待解决的重要问题之一.

为此, 本文提出一种基于区块链的策略隐藏大数据访问控制方法 (A policy-hidden big data access control method based on blockchain, PHAC). PHAC 借鉴同态加密思想, 引入属性基加密 (Attribute-based encryption, ABE)[12] 以及双线性映射技术, 实现在不泄露访问控制策略的前提下, 通过智能合约正确执行访问控制策略. 同时, 在 PHAC 中解耦访问控制策略, 简化用户访问控制策略的发布、更新和执行, 此外, 采用链上和链下存储相结合方式, 解决大数据场景下智能合约和访问控制策略占用区块链节点资源不断增大的问题. 与现有工作相比, 本文提出方法具有如下优点:

1) 利用属性基加密及双线性映射技术, 实现访问控制策略隐藏, 保证区块链节点能在策略隐藏的情况下实现访问控制, 可避免恶意攻击者通过对区块链节点均公开的访问控制策略分析挖掘数据拥有者的隐私.

2) 为避免对权威授权中心的依赖, 提出访问密钥由数据访问者自己构造、密钥验证由区块链节点执行的智能合约. 当访问者发起请求时, 密钥验证合约自动验证访问者密钥的正确性和可用性, 以保证访问者为合法用户.

3) 为减低区块链节点的资源开销, 首先解耦访问控制策略和访问控制判决逻辑; 其次访问控制策略再次进行解耦, 并以事务的形式采用链上和链下相结合的方式存储, 访问控制判决逻辑以智能合约实现. 用户在自主更新访问控制策略时, 仅需更新解耦后访问控制策略的变动部分即可, 无需重新编写整个策略, 同时无需同步更新访问控制判决合约.

4) 对提出方法进行了理论分析和 HyperLedger Fabric 环境下实验评估. 结果表明, 本文方法能在策略隐藏情况下实现访问控制, 但不会给区块链节点增加过多额外计算和存储开销.

1 相关工作

目前, 在云计算、大数据等应用场景实现数据安全共享主要有两类访问控制方法: 1) 引入可信授权中心, 采用属性集加密技术; 2) 利用区块链技术实现去中心化的访问控制.

在云计算、大数据应用中, 由于第三方服务提供商半可信, 故一些学者提出基于密码学的访问控制技术. Boneh 等[13] 提出身份基加密 (Identity-based encryption, IBE), Sahai 等[14] 在 IBE 的基础上, 提出了属性基加密即访问者属性集满足访问结构时, 即可解密加密数据. 早期的属性基加密仅支持简单的门限访问控制策略, 为更灵活地表达访问控制策略, 后续有学者相继提出密钥策略属性基加密和密文策略属性基加密 (Ciphertext-policy ABE, CP-ABE)[15]. 之后, 有众多学者[16−18] 对属性基加密关于访问结构、合谋攻击属性撤销等展开研究, 但大多假设存在单授权中心. 由于单授权中心具有单点故障、安全可信、性能瓶颈等问题, 文献 [19−21] 提出采用属性分组或者全局唯一标识的多授权中心方案. Lin 等[22] 基于双线性 Diffie-Hellman 假设提出一种多授权中心方案, 采用 (t, n) 门限结构, 要求 $t+1$ 个授权中心完全可信. Jung 等[20] 提出一种多权威中心的云数据访问控制方案, 方案中每一个授权中心基于属性为用户颁发密钥组件, 存在一个超级授权中心合并所有的密钥组件生成用户密钥. 由于该方案采用全局唯一标识, 本质上仍需解决超级

单授权中心的安全问题.

因区块链技术具有去中心化、透明性和不可篡改性等特点, 目前已有不少工作提出把区块链作为云计算、大数据访问控制的基础设施来使用[3]. 例如, 刘敖迪等[6] 提出一种基于区块链的大数据访问控制机制, 其借助区块链的可追溯、不可篡改等特点, 通过区块链事务存储管理访问控制策略及属性. Ding 等[23] 结合基于角色的访问控制技术和基于属性的访问控制技术提出一种安全访问控制方法, 其采用链下加密存储数据, 策略存储在区块链中, 利用智能合约实现访问控制策略的自动化判决. Ba 等[24] 结合区块链和 CP-ABE 提出一种隐藏访问控制结构的数据共享方案, 但仅隐藏部分访问控制结构. Wang 等[25] 提出基于区块链和 ABE 的访问控制框架, 为了保证安全性及密钥的实时发放, 其需要数据拥有者实时在线参与计算分发密钥. 张建标等[26] 提出利用区块链实现域间的可信细粒度访问控制. Makhdoom 等[27] 提出一种用于在智能城市环境中保护隐私和安全的物联网数据共享方案. Gao 等[28] 提出一种基于 CP-ABE 的访问控制系统, 利用区块链保证去中心化, 同时实现了属性的隐藏, 但其需数据访问者在本地匹配策略, 并生成一个证明令牌, 区块链节点验证令牌, 判决访问者是否可以访问数据, 区块链无法验证令牌中属性的真实性, 故存在伪造风险. Zhang 等[29] 分析当前公有云数据共享现状提出一种结合线性秘密共享 (Linear secret sharing scheme, LSSS) 和区块链的访问控制方案, 但由于公有云数据量大、用户多样以及 LSSS 表达复杂等, 该数据共享方案存在一定的局限性, 如访问控制判决时间长等. 在上述工作中, 文献 [25] 为了保证密钥的实时发放, 额外增加了数据拥有者的计算开销. 文献 [24, 27, 29] 需要引入完全可信授权中心属性权威 , 易引入单点失效问题, 违背分布式访问控制的初衷. 文献 [6, 23, 25, 28] 侧重关注访问策略的不可篡改、访问控制判决自动执行及访问控制判决结果可追溯, 并未考虑访问控制策略经常包含用户敏感信息, 易造成隐私泄露的问题. 此外, 由于区块链只可增加和查看, 因此随着区块链中部署策略和智能合约的增多, 会给区块链节点带来巨大的存储及计算开销. 虽然文献 [23] 提出采用链上链下相结合的方式, 但在数据拥有者更新访问控制策略时, 需要更新整个智能合约, 仍会给数据拥有者带来巨大计算开销.

综上所述, 目前仍需要一种基于区块链的大数据访问控制方法以解决策略明文存储带来的隐私泄露和智能合约、策略更新部署等所引入的区块节点存储、计算开销过大的问题.

2　系统模型

本文方法针对当前典型的基于区块链数据共享通用场景[23−25], 如图 1 所示, 主要涉及数据拥有者端、数据访问者端、区块链平台和云存储服务端四个部分的交互.

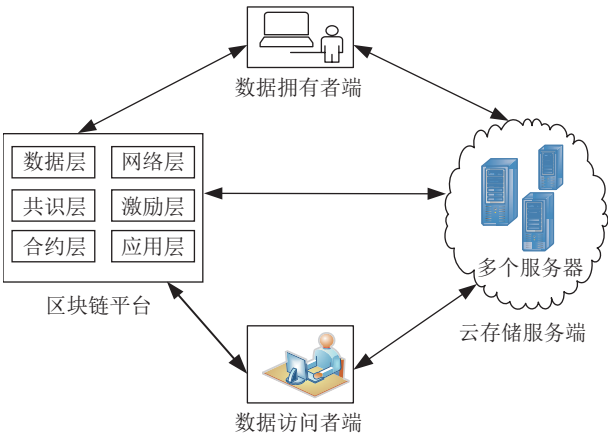

图 1　基于区块链的数据安全共享通用场景

Fig. 1　General scenarios of secure data sharing based on blockchain

1) 数据拥有者端把其共享数据上传至云存储服务, 具有对该数据的绝对管理权即为共享数据制定访问控制策略, 并通过区块链客户端发起交易申请, 将共享数据的访问控制策略上传至区块链中.

2) 数据访问者端是共享数据的请求者, 通过客户端向区块链发起访问请求, 类似发起一个交易请求, 区块链中智能合约触发后, 会依据事先约定的策略进行访问控制判决. 数据访问者通常需要携带相关权威颁发的属性证书或者密钥.

3) 区块链平台是整个数据安全共享架构的基础设施, 主要包括数据层、网络层、共识层、激励层、合约层和应用层: a) 数据层以基于哈希算法的 Merkle 树结构进行存储, 存储内容包括共享数据相应的策略信息、智能合约、访问控制执行结果等信息; b) 网络层实现各个网络节点的通信, 维护协同协议版本号、通信节点等信息; c) 共识层通过工作证明算法、股权证明算法、实用拜占庭容错算法等共识机制, 实现协商一致, 确保交易请求信息、访问控制策略、策略判决结果等数据信息的正确性; d) 激励层负责发放奖励给区块链节点, 以促使每个节点积极并诚实维护区块链, 并对发起交易请求的用户收取相应手续费, 以避免用户发送恶意请求信息; e) 合约层是智能合约集合, 是一种在区块链中存储, 并能够在区块链网络各节点上自动执行的计算机脚本[30−31], 在基于区块链的数据共享架构中, 智能合约

用于执行共享数据的访问控制策略; f) 应用层是区块链平台和数据拥有者端、数据访问者端、云存储服务端交互的接口.

4) 云存储服务端由云服务商提供, 负责数据拥有者共享数据的存储, 在基于区块链数据共享的通用场景中, 其会与区块链平台、数据拥有者端、数据访问者端交互.

然而, 现有的区块链访问控制技术, 主要将共享数据的访问控制策略直接存储于区块链平台的区块中或编写到智能合约中, 区块链的透明性使得策略对所有用户均透明可见, 而为了实现对共享数据的细粒度访问控制, 共享数据策略经常包含数据拥有者的敏感信息, 因而存在针对策略的隐私保护问题. 此外, 随着共享数据的用户规模增加, 其对共享数据更新频繁, 共享数据策略更新迭代速度快, 自然会给区块链平台各节点引入巨大的存储和计算资源开销. 为此, 遵循上述基于区块链的数据共享通用场景, 本文提出一种基于区块链的策略隐藏大数据访问控制方法 PHAC.

3 PHAC 方法设计

3.1 工作原理

本文提出的 PHAC 方法, 在数据拥有者端增加了访问控制树、加密参数构造模块及数据加密发送模块; 在区块链平台增加了密钥验证合约、访问控制合约、属性等事务模块; 在数据访问者端增加密钥构造模块和数据接收解密模块.

访问控制树构造模块、加密参数构造模块和数据加密发送模块部署在数据拥有者端. 访问控制树构造模块负责把数据拥有者对共享数据的访问控制策略转换成访问控制树; 加密参数构造模块用于构造访问控制树的加密参数, 并通过客户端发布至区块链中; 数据加密发送模块用于把待共享数据加密并上传至云存储服务, 同时把数据解密密钥用数据访问者的公钥加密并发送至数据访问者端.

密钥构造模块和数据接收解密模块部署在数据中. 密钥构造模块根据区块链平台发布的公共参数、数据拥有者端发布的加密参数以及自身属性信息生成自己的密钥, 在向区块链发起访问请求时, 携带该密钥; 数据接收解密模块用于接收待访问数据的密钥, 并用于解密从云端接收到的数据. 特别地, 为了实现策略隐私保护以及解耦访问控制策略减少区块链平台节点开销, 本文 PHAC 方法除了在区块链平台上部署图 1 所述的数据层、共识层等区块链基础功能模块外, 还把原来策略写进合约的存储方式, 解耦成以事务和合约的方式分别存储, 由此区块链平台中增加了属性事务、访问策略事务、访问树事务、密文事务、公共参数事务、用户参数事务、密钥验证合约、访问控制合约. 其中, 属性事务是用于对区块链平台中属性信息的管理; 访问策略事务用于管理用户的访问策略, 包括发布、更新和撤销; 访问树事务是用于管理用户制定的访问树, 包括发布、更新和撤销; 密文事务用于管理用户的密文, 包括发布、更新和撤销; 公共参数事务是用于管理区块链平台发布的公共参数; 用户参数事务用于管理数据拥有者发布的相关参数; 密钥验证合约用于核实验证用户密钥的真实性, 当数据访问者携带密钥发起访问请求时, 其先执行; 访问控制合约用于根据访问控制策略对访问请求进行判决.

PHAC 方法架构如图 2 所示.

为便于后续阐述, 下面先定义本文方法用到的术语.

定义 1. 记区块链平台待服务用户具有的所有属性集合为 $ATT = \{A_1, A_2, \cdots, A_n\}$, 则称 A_i $(1 \leq i \leq n)$ 为区块链平台属性, 其中 n 代表区块链平台属性总数. 若属性 A_i 的值域为 $\{v_{i,1}, v_{i,2}, \cdots, none, v_{i,t}\}$, 其中 $v_{i,t}$ $(1 \leq t \leq n_i)$ 表示属性 A_i 的属性值, 若用户的 A_i 属性取值为 $none$, 代表该用户无 A_i 属性.

需要说明的是, 在本文针对场景中, 访问主体属性的值域大多是离散且有限的, 比如角色、安全级别、性别、年龄等属性类型. 对于考虑时间这类连续型环境属性的访问控制应用, 通常会先将时间按照区间划分后根据时间段进行访问控制, 毕竟基于某一个时间段进行访问控制, 比基于某一时刻进行访问控制更具实际意义. 由此本文沿用 ABE 思想[14] 在定义 1 中, 对属性 A_i 的值域采用了有限离散集表示. 对于连续型属性, 需要对其进行离散化预处理后, 再采用 PHAC 方法.

定义 2. 记用户具有的属性为 $Atts = \{A_1 = a_1, A_2 = a_2, \cdots, A_i = a_i\}$, 其中 $A_i \in ATT$, $a_i \in \{v_{i,1}, v_{i,2}, \cdots, none, v_{i,t}\}$.

为了支持细粒度访问控制, PHAC 采用 ABE 提出的访问控制树描述共享数据的访问控制策略.

定义 3. 用树结构 T 表示基于属性的访问控制策略, 称其为访问控制树, 其中树的内部节点表示与 (AND) 运算、或 (OR) 运算和门限 (Threshold) 运算, 叶子节点为属性表达式, 叶子节点的父节点称为末端内部节点.

需要指出的是, 由于树结构下任何末端内部节点都能转成与 (AND) 运算的形式[10], 故本文方法采用访问控制树的末端内部节点一定表示与 (AND)

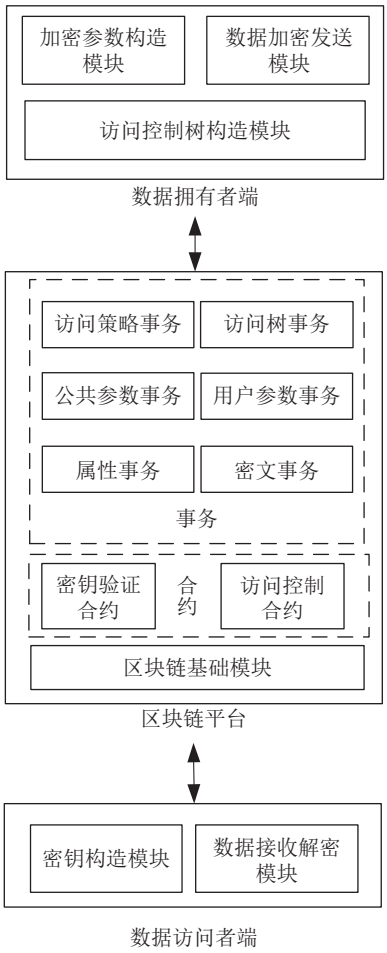

图 2　PHAC 方法架构

Fig. 2　Architecture of PHAC

运算, 具体示例如图 3 所示, 该策略表示属性满足 $\{A=1, B=2, C=3, D=4, E=5, F=6\}$ 或 $\{C=3, F=6\}$ 或 $\{A=1, B=2, F=6\}$ 或 $\{C=3, D=4, E=5, F=6\}$ 或 $\{A=1, B=2, D=4, E=5\}$ 或 $\{A=1, B=2, C=3, F=6\}$ 或 $\{A=1, B=2, C=3\}$ 或 $\{A=1, B=2, D=4, E=5, F=6\}$ 或 $\{D=4, E=5, F=6\}$ 或 $\{A=1, B=2, C=3, D=4, E=5\}$ 的主体能访问加密存储的共享数据.

同时, 为简化数据拥有者在区块链中对访问控制树的更新操作, PHAC 方法解耦访问控制树, 引入访问树.

定义 4. 若 T 是一个访问控制树, 则称去除 T 中叶子节点的树 T' 为访问树, 而被去除的叶子节点称为访问策略.

例如, 图 4 为图 3 示例对应的访问树, 去掉了访问策略.

PHAC 实现策略隐藏区块链访问控制的流程

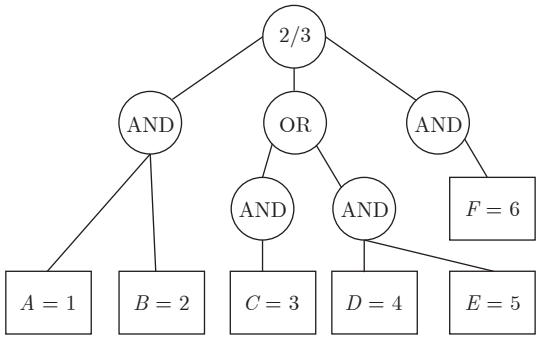

图 3　访问控制树示例

Fig. 3　Example of access control tree

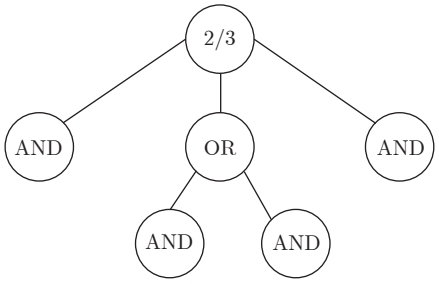

图 4　图 3 示例对应的访问树

Fig. 4　Corresponding access tree in Fig. 3 example

如图 5 所示, 分为两个阶段: 1) 参数准备阶段. 区块链平台执行群生成算法生成公共参数; 数据拥有者阶段根据平台发布的公共参数, 运用加密参数构造模块, 构造属于自己的加密参数, 同时运用访问控制树构造模块, 构造访问控制树, 并发布至区块链中; 数据访问者根据区块链平台发布的公共参数、数据拥有者发布的加密参数及自身属性信息运用密钥构造模块构造自己的密钥. 2) 策略执行阶段. 数据访问者携带密钥向区块链发起访问请求; 区块链平台运用密钥验证合约验证访问者密钥的合法性及正确性, 同时验证数据的完整性和可用性; 然后运用访问控制合约, 计算并验证访问控制树 T 的末端内部节点 α 的秘密值; 通过末端内部节点秘密值计算访问控制树根节点的秘密值; 根据根节点的秘密值计算数据拥有者签名的数据拥有凭证 M; 把访问请求转至云存储服务和数据拥有者端; 云存储服务返回加密数据、数据拥有者端通过数据加密发送模块发送数据密钥至数据的数据接收解密模块.

PHAC 利用了循环群的乘同态性, 为阐述方便, 给出符号说明: 在乘法循环群生成算法 M 中输入安全参数 λ, 输出元组 (p, G, G_T, e), 其中 p 为给定安全常数 λ 相关的大质数; G 和 G_T 为 p 阶的循环群, g 和 g_T 分别为 G 和 G_T 的生成元; $e: G \times G_T \to G_T$ 为线性映射.

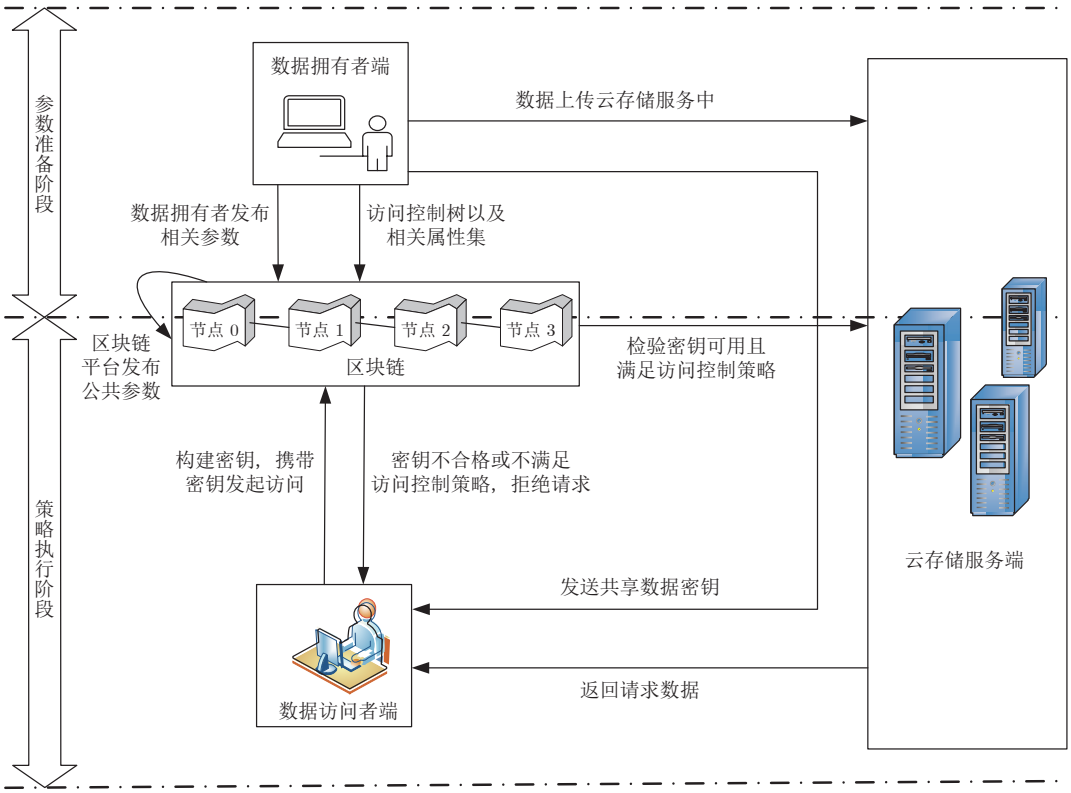

图 5 PHAC 的访问控制流程

Fig. 5 Access control process of PHAC

双线性映射是基于椭圆曲线, 是众多密码方案的数学基础. 双线性映射 e 满足如下性质:

1) 双线性. 对于任意的 $a, b \in Z_p$, $h \in G_T$, 满足 $e(g^a, h^b) = e(g, h)^{ab}$.

2) 非退化性. $e(g, g) \neq 1$.

3) 可计算性. 对于任意 $g, h \in G$, 存在有效多项式时间内计算 $e(g, h)$ 值的算法.

3.1.1 参数准备阶段

1) 区块链平台参数准备

区块链平台发布的参数如下:

$$PK = \{G, G_T, g, e, \{A_{i,t}, \overline{A}_{i,t}\}_{1\leq i\leq n, 1\leq t\leq n_i}\} \quad (1)$$

式中, 区块链平台调用参数准备合约运行群生成算法 $M(\lambda)$ 获得 (p, G, G_T, e), 同时对区块链平台中属性 A_i 的每一个属性值 $v_{i,t}$ $(1 \leq i \leq n, 1 \leq t \leq n_i)$, 随机选择 $a_{i,t} \in Z_p^*$, 计算属性值分量 $A_{i,t} = g^{a_{i,t}}$, $\overline{A}_{i,t} = g^{1/a_{i,t}}$. PK 中的 G、G_T、g、e 在区块链中以公共参数事务存储, $\{A_{i,t}, \overline{A}_{i,t}\}_{1\leq i\leq n, 1\leq t\leq n_i}$ 以属性事务存储.

式 (1) 是区块链平台对平台中所有属性的初始化计算, 在密文策略基础上, 用乘法循环群计算公钥参数. 式中 $A_{i,t}$, $\overline{A}_{i,t}$ 属性值分量的计算目的是基于循环群的特点为策略隐藏考虑, 同时为了后续数据拥有者构造用户参数、访问者构造访问密钥及访问控制判决做准备.

2) 数据拥有者参数准备

数据拥有者在区块链平台中发布的参数如下:

$$PK_u = \{X, \overline{X}, Y, \overline{Y}, Z_1, \overline{Z}_1, Z_2, \overline{Z}_2\} \quad (2)$$

式中, 数据拥有者随机选择 ω、$\overline{\omega}$、β, $\overline{\beta} \in Z_p^*$, 然后根据区块链平台发布的公共参数 PK, 计算 $X = g^{\beta}$, $\overline{X} = g^{\overline{\beta}}$, $Y = e(g, g)^{\omega}$, $\overline{Y} = e(g, g)^{\overline{\omega}}$, $Z_1 = g^{\omega/\beta}$, $Z_2 = g^{1/\beta}$, $\overline{Z}_1 = g^{\overline{\omega}/\overline{\beta}}$, $\overline{Z}_2 = g^{1/\overline{\beta}}$. PK_u 在区块链中以用户参数事务存储.

式 (2) 是数据拥有者发布的用户参数, 用于数据访问者构造访问者密钥及区块链平台的访问控制判决. 式 (2) 中参数 X、$\overline{X}$、Y、$\overline{Y}$、Z_1、$\overline{Z}_1$、Z_2、$\overline{Z}_2$ 是基于区块链平台发布的公共参数 PK 及循环群的特点计算, 其中 X、$\overline{X}$、Y、$\overline{Y}$ 是用于数据拥有者构造密文数据, Z_1、$\overline{Z}_1$、Z_2、$\overline{Z}_2$ 用于数据访问者构造访问密钥. 本文提出式 (2) 是为了避免对权威授权中心的依赖, 同时在保证数据访问者密钥安全可用的前提下, 实现策略隐藏下的访问控制判决.

考虑到云存储服务会存在一定中心化问题, PHAC 要求数据拥有者对数据加密后再存到云服务器上, 并在区块链中记录加密数据散列值等相关信息, 提

供访问控制合约去校验加密数据的完整性.

数据拥有者在区块链中发布的信息如下构成:

$$Date_u = \{ADD, H_{\text{value}}, ATT_u, C\} \tag{3}$$

式中, ADD 为加密数据存放在云服务器中的地址; H_{value} 为加密数据的散列值; ATT_u 为数据拥有者选定的访问者属性集, 包含访问控制真实属性及无关属性, 无关属性选取越多, 隐私保护效果越好, 越不容易被猜测属性范围. 另外在 PHAC 中, ATT_u 不必包含所有真实属性, 以此避免因指定访问者属性集, 恶意用户通过属性集范围推测访问控制策略, 而造成的隐私泄漏风险; $C = (C_0, C_1, T', \{C_\alpha\}_{\alpha \in leaf(T')})$ 为加密数据的访问控制策略.

在 C 中, $C_0 = X^q = g^{\beta q}$, $C_1 = M \cdot e(g, g)^{\omega q}$ (M 是数据拥有者签名的数据拥有凭证), 数据拥有者构造访问控制树 T 并去掉叶子节点形成访问树 T', $C_\alpha = \{\overline{C}_{\alpha 1}, \overline{C}_{\alpha 2}, \{C^\alpha_{i,t}\}_{1 \le i \le n, 1 \le t \le n_i}\}$. 为实现策略隐藏, C_α 中 $\overline{C}_{\alpha 1} = \overline{X}^{q_\alpha}$, $\overline{C}_{\alpha 2} = \overline{Y}^{q_\alpha}$, $C^\alpha_{i,t}$ 的计算步骤如下: 首先随机选择 $q \in Z_p^*$ 作为访问控制树 T 根节点的秘密值, 执行秘密共享算法, 使得 T 中的每个末端内部节点 α 得到一个 q 的秘密值 q_α; 然后, 利用混淆的思想, 在 α 下为区块链平台属性的所有属性值 $v_{i,t}$ 计算密文分量. 按照区块链平台属性及取值是否出现在节点 α 下, 可分为以下三种情况计算属性值密文分量:

1) 如果属性 A_i 的取值 $v_{i,t}$ 出现在节点 α 下的叶子节点所代表的表达式中, 则计算方式为 $C^\alpha_{i,t} = A^{q_\alpha}_{i,t}$.

2) 如果属性 A_i 未出现在节点 α 下的叶子节点所代表的表达式中, 则属性 A_i 所有取值 $v_{i,t}$ 计算方式为 $C^\alpha_{i,t} = A^{q_\alpha}_{i,t}$, 与情况 1) 计算方式相同.

3) 如果属性 A_i 出现在节点 α 下叶子节点所代表的表达式中, 但其取值 $v_{i,t}$ 没有出现在叶子节点中, 则随机选择 $q_{i,t} \in Z_p^*$ ($q_{i,t} \ne q_\alpha$), 计算方式为 $C^\alpha_{i,t} = A^{q_{i,t}}_{i,t}$.

需要指出的是, 为了实现策略的混淆隐藏需要对平台中所有属性值计算密文分量. 另外, PHAC 在计算末端内部节点 α 的秘密值时, 需要将访问者密钥中的每一个密钥分量和节点 α 下对应的属性值密文分量做双线性计算, 然后利用循环群的特点计算 α 节点的秘密值. 由于可能存在访问者密钥中出现节点 α 下真实访问策略未要求属性的情况, 故为避免一个合法密钥解密不出节点 α 的正确秘密值, PHAC 中情况 1) 和情况 2) 的计算方式相同. 同时, PHAC 也支持策略无需隐藏的数据共享, 此时只需按照情况 1) 方法进行计算属性值密文分量并发布即可, 不必计算情况 2) 和情况 3).

由此, 有节点 α 下的密文分量计算如下:

$$C^\alpha_{i,t} = \begin{cases} A^{q_{i,t}}_{i,t}, & A_i \in ATT(T_\alpha) \text{ 且 } v_{i,t} \notin T_\alpha \\ A^{q_\alpha}_{i,t}, & \text{否则} \end{cases} \tag{4}$$

式中, $A^{q_\alpha}_{i,t} = g^{a_{i,t} q_\alpha}$, $A^{q_{i,t}}_{i,t} = g^{a_{i,t} q_{i,t}}$, $1 \le i \le n$, $1 \le t \le n_i$.

这样, 共享数据的访问控制树 T 由访问树 T' 和 T 中加密混淆后的叶子节点访问策略组成, 数据拥有者在区块链中发布 $Date_u$, 其中 T' 以访问树事务的形式存储管理, $\{C_\alpha\}_{\alpha \in leaf(T')}$ 叶子节点表示的访问策略以访问策略事务存储管理, ADD、ATT_u、H_{value}、C_0、C_1 用密文事务存储. 为了防止恶意数据拥有者发布不合实际的 $Date_u$ 浪费区块链平台资源, PHAC 方法要求数据拥有者根据发布 $Date_u$ 复杂程度支付相应代币.

3) 数据访问者参数准备

访问者密钥:

$$R_{sk} = \{D_0, \overline{D}_0, \{D_i\}_{1 \le j \le k}\} \tag{5}$$

式中, $D_0 = Z_1 \cdot Z_2^k = g^{(\omega + k)/\beta}$, $\overline{D}_0 = \overline{Z}_1 \cdot \overline{Z}_2^k = g^{(\overline{\omega} + k)/\overline{\beta}}$. 数据访问者根据区块链平台发布的 PK 以及数据拥有者发布的 PK_u、$Date_u$ 构造密钥属性集 $Atts_s$, 其中 $ATT_u \subseteq Atts_s \subseteq Atts$. 由于 ATT_u 可能不包含所有真实属性, 因此访问者在构造访问者密钥时, 可能需要多于 ATT_u 的属性来构造密钥, 同时访问者可根据隐私保护需要, 对自己的某些属性取值为 $none$, 从而避免访问者本身的属性隐私泄漏风险. 对于属性集 $Atts_s$ 中的每个属性 att_i ($1 \le i \le k$), 其中 k 是属性集 $Atts_s$ 的阶: 从 PK 中选择对应属性分量 $D_i = \overline{A}_{j,t} = g^{1/a_{j,t}}$ ($1 \le i \le k$, $1 \le j \le n, 1 \le t \le n_j$) 构成密钥分量, 并对其进行签名, 确保属性真实性.

3.1.2 策略执行阶段

访问者携带密钥向区块链发起访问请求, 首先密钥验证合约验证密钥的正确有效性, 如果无效, 直接拒绝. 本文要求用户发起访问请求时, 需要支付一定的代币, 以用于弥补区块链平台对于密钥验证的开销, 同时防止访问者用密钥恶意试探浪费区块链平台资源. 其次, 根据访问控制树 T 中末端内部节点的秘密值重构出 T 的根节点秘密值, 进而计算得到 M. 最后, 区块链进行验证共识, 并把访问者的访问请求转至云服务器, 云服务器返回请求数据值, 数据拥有者端返回数据密钥至访问者端.

为了达到策略混淆隐藏的目的, 式 (2) 计算属性值密文分量时, 分为三种情况, 同时访问者密钥

根据式 (5) 计算, 故在计算重构 T 的末端内部节点 α 的秘密值时, 需要把访问者密钥中的 $\{D_i\}_{1\le i\le k}$ 和节点 α 下对应的属性值密文分量 $\{C^{\alpha}_{i,t}\}_{1\le i\le n,\,1\le t\le n_i}$ 一一对应进行双线性映射计算, 并根据循环群的特点把所有计算结果相乘, 如此则可保证只有数据访问者拥有正确属性时才能计算出正确的秘密值; 反之, 则是计算出一个乱码值. 因此 T 的末端内部节点 α 的秘密值计算如下:

$$E_{\alpha}=\prod_{i=1}^{k}e(C^{\alpha}_{i,t},\,D_i),\ \ 1\le i\le n,\,1\le t\le n_i \tag{6}$$

式中, $e(C^{\alpha}_{i,t},\,D_i)$ 的计算方式分为两种, 当 $A_i\in ATT(T_{\alpha})$ 并且 $v_{i,t}\notin T_{\alpha}$ 时为 $e(g^{a_{i,t}q_{i,t}},\,g^{1/a_{i,t}})$; 其他情况的计算方式为 $e(g^{a_{i,t}q_{\alpha}},\,g^{1/a_{i,t}})$, k 是访问者密钥属性集的阶.

如果访问者的密钥满足节点 α 下的所有访问策略, 则可计算出 α 的正确秘密值:

$$E_{\alpha}=\prod_{i=1}^{k}e(g,\,g)^{q_{\alpha}}=e(g,\,g)^{kq_{\alpha}} \tag{7}$$

如下检验秘密值是否正确:

$$\phi_{\alpha}=\frac{\overline{C}_{\alpha 2}E_{\alpha}}{e(\overline{C}_{\alpha 1},\,\overline{D}_0)}=\frac{e(g,\,g)^{\overline{\omega}q_{\alpha}}e(g,\,g)^{kq_{\alpha}}}{e(g^{\overline{\beta}q_{\alpha}},\,g^{\frac{\overline{\omega}+k}{\overline{\beta}}})}=1 \tag{8}$$

式中, 1 表示群 G_T 的单位元, 当前密钥满足 α 下的访问策略, 得到正确的秘密值; 否则, 得出 ϕ_{α} 为随机值. 计算并验证完访问控制树 T 中所有的末端内部节点后, 判断是否可重构 T 的根节点秘密值.

访问控制树 T 中内部节点 δ 的秘密值如下计算:

$$E_{\delta}=\begin{cases}\prod_{i=1}^{num(\delta)}E_{child(\delta,i)}, & op(\delta)='\text{AND}'\\ E_{child(\delta,i)}, & op(\delta)='\text{OR}'\\ \prod_{i=1}^{k(\delta)}E_{child(\delta,i)}^{\prod_{j=1,\,j\ne i}^{k(\delta)}\frac{-i}{j-i}}, & op(\delta)='\text{Threshold}'\end{cases} \tag{9}$$

式中, $num(\delta)$ 表示节点 δ 的子节点数量; $op(\delta)$ 表示节点 δ 对应的运算符; $E_{child(\delta,i)}$ 为节点 δ 下第 i 个子节点的秘密值. 根据定义 3 可知, $op(\delta)$ 可为 AND、OR 和 Threshold. 当 $op(\delta)$ 为 AND 时, 节点 δ 的秘密值为子节点的秘密值相乘即用 $\prod_{i=1}^{num(\delta)}E_{child(\delta,i)}$ 计算; 当 $op(\delta)$ 为 OR 时, 计算节点 δ 的秘密值等于任意一个子节点的秘密值即用 $E_{child(\delta,i)}$ 计算; 当 $op(\delta)$ 为 Threshold 时, 计算节点 δ 的秘密值利用插值多项式计算即用 $\prod_{i=1}^{k(\delta)}E_{child(\delta,i)}^{\prod_{j=1,\,j\ne i}^{k(\delta)}-i/(j-i)}$ 计算, $k(\delta)$ 为节点 δ 的门限值.

如果访问者的密钥满足访问控制树 T, 根据式 (9) 计算根节点的秘密值 $E_{root}=e(g,\,g)^{kq}$, 最后计算签名凭证 M 如下:

$$\frac{C_1\cdot E_{root}}{e(C_0,\,D_0)}=\frac{M\cdot Y^{q}\cdot e(g,\,g)^{kq}}{e(g^{\beta q},\,g^{\frac{\omega+k}{\beta}})}=\frac{M\cdot e(g,\,g)^{\omega q}\cdot e(g,\,g)^{kq}}{e(g,\,g)^{\omega q+kq}}=M \tag{10}$$

区块链平台用访问控制合约, 对访问者密钥和访问控制树的解密验证. 如果访问者的访问密钥满足访问控制树, 则数据拥有者端会发送共享数据的密钥至数据, 同时区块链平台和云服务器交互, 返回访问者请求数据; 反之, 拒绝访问.

为了降低区块链节点的资源开销, PHAC 方法将解耦后的访问控制策略以事务形式存储及管理, 访问控制判决逻辑用智能合约实现, 因此, 事务存储及管理、密钥验证合约及访问控制合约是 PHAC 方法的核心技术, 下面详细介绍.

3.2 事务存储及管理

本文方法引入了属性事务、访问策略事务、访问树事务、密文事务、公共参数事务、用户参数事务. 这些事务数据在区块链平台上以 Merkle 树形式进行存储即采用哈希函数将任意长度的事务数据映射成固定长度的字符串作为 Merkle 树的节点, 由此可利用哈希函数的单向性使用 Merkle 树快速验证数据是否被篡改. 事务管理的数据结构如图 6 所示, 包括消息标识号、事务类型、执行的操作、发布者的公钥、属性事务、公共参数事务等具体事务、时间戳和代表发起交易申请用户对该事务的签名.

访问控制树 T 解耦成访问树 T' 和加密混淆后的访问策略后, 分别以访问树事务和访问策略事务存储, 解耦后的访问控制树示例如图 7 所示.

在数据拥有者构造访问控制树的过程中, 访问控制树末端内部节点 α 的属性值密文分量如第 3.1.1 节所述分成三种情况进行计算. 例如, 如图 8 所示, 若属性 A 的取值 v_1 出现在节点 α 下, 那么 v_1 的密文分量按情况 1) 计算, 而未出现的属性 A 取值 v_i $(1\le i\le n,\,i\ne 1)$ 按情况 3) 计算密文分量; 若属性 B 未出现在节点 α 下, 则属性 B 取值的密文分量按情况 2) 计算. 当数据拥有者需要更新访问控制策略, 如将节点 α 下属性 A 的取值更新为 v_2、属性 B 的取值不变时, 那么数据拥有者只需重新计算节点 α 下属性 A、B 取值的密文分量即属性 A 取值 v_2 按情况 1) 计算, 其他未出现取值 v_i $(1\le i\le n,\,i\ne 2)$ 按情况 3) 计算; 属性 B 未变则属性 B 仍按情况 2) 计算. 之后, 只需把节点 α 下重算的密文分量重发布至区块链, 其他非 α 节点下的属性值密文分

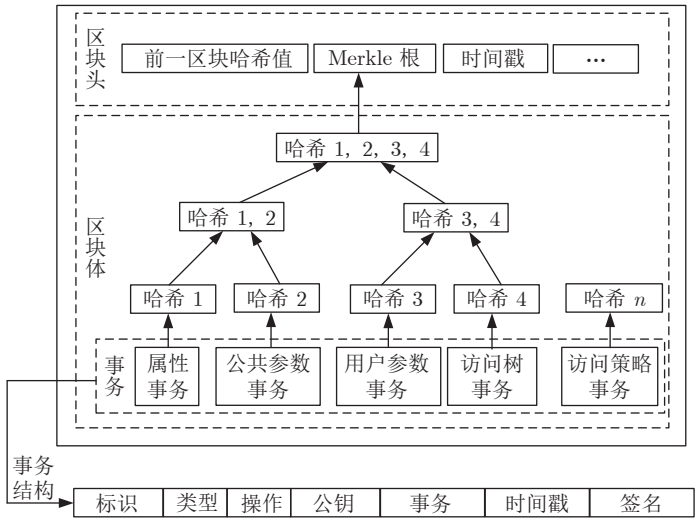

图 6　区块数据结构

Fig. 6　Block data structures

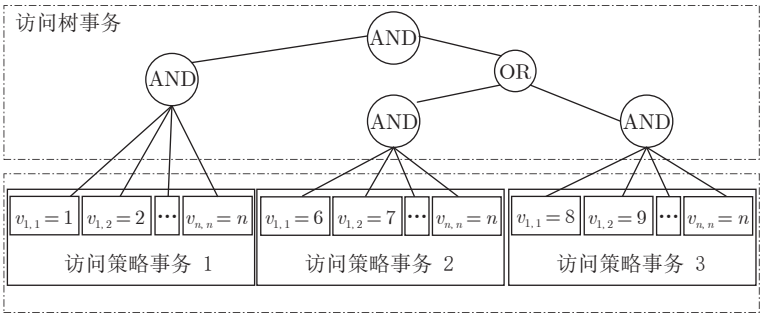

图 7　区块链平台存储的访问控制树示例

Fig. 7　Example of access control tree stored on the blockchain platform

量不用重发布, 同时访问树也不用更新, 这大大降低了共享数据增加、策略更新给区块链平台节点引入的资源开销.

PHAC 中较大事务存储于链下[30], 链下存储结构如图 9 所示. 较小事务直接存储于链上即仅在区块中存储该事务存放地址及事务的散列值, 该值被用来验证策略是否被篡改, 保证策略数据上链之前的完整性. 策略上链后, 由智能合约保证策略被正确自动执行, 即策略上链后的安全性由区块链自身特点保证.

3.3　参数准备合约

参数准备合约为区块链平台生成相关参数、为数据拥有者上传参数、数据等, 参数具体计算见第 3.1.1 节. 另外, 参数准备合约把参数、数据等存储至相应的事务中, 具体事务存储可见第 3.2 节. 参数准备合约可被区块链平台调用生成公共参数 PK, 也可被数据拥有者调用上传用户参数 PK_u、数据 $Date_u$, 具体见算法 1.

算法 1. 参数准备合约

输入. 调用者标识 $flag$.

输出. 结果 TRUE、FALSE.

1) IF($flag = Blockchain$);

2) {;

3) ATT {}= $input$ (); //输入属性集合及属性值域, 字典格式;

4) RETURN $Blockchain\_Attribute\_Calculation(ATT\{\})$;

5) };

6) IF($flag = DateOwner$);

7) {;

8) $PK_u = input$ (); //输入用户参数;

9) $Date_u$ = $input$ (); // 输入数据信息;

10) RETURN $Date\_Owner\_Upload(PK_u, Date_u)$;

11) };

12) RETURN FALSE.

3.4　密钥验证合约

在 PHAC 方法中, 数据访问者密钥 R_{sk} 是由其

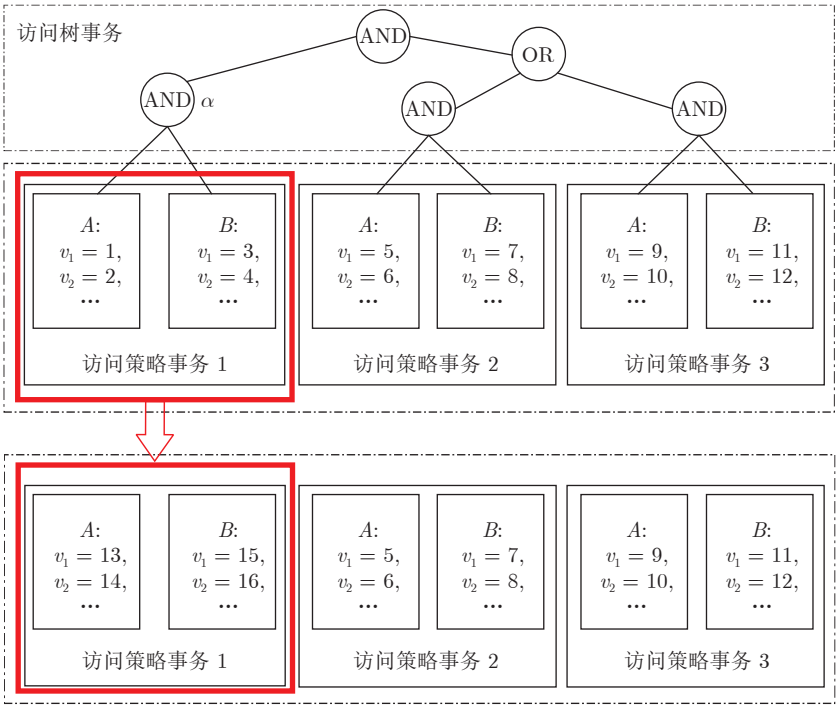

图 8 访问策略事务更新

Fig. 8 Update of access policy transaction

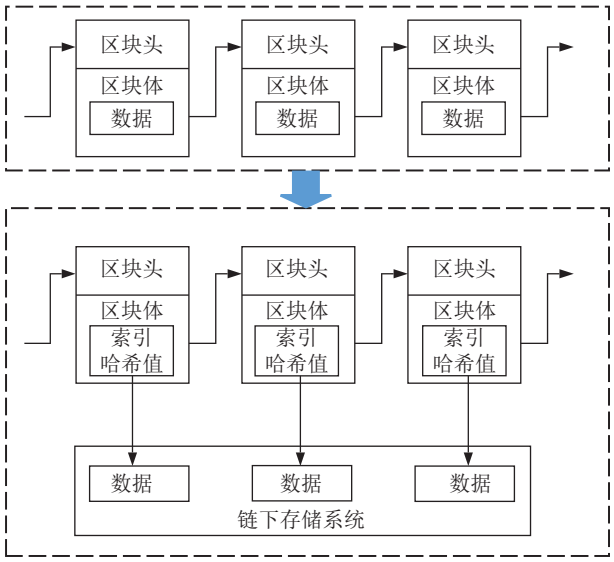

图 9 链下存储结构

Fig. 9 Storage structure under chain

自己根据自身属性、区块链平台发布的公共参数及数据拥有者发布的加密参数构造, 所以存在攻击者非法、越权构造密钥的可能, 故在执行策略前需要验证访问者密钥是否真实可用. 为避免攻击者用违规密钥恶意试探浪费区块链平台资源, 在区块链平台完成密钥验证时, PHAC 要求用户支付一定代币用于弥补区块链平台各节点开展密钥验证的开销.

区块链平台主要从两个方面检验密钥 R_{sk}: 1) 密钥是否真实可用, 检验结果有密钥可用 (AVAILABLE) 和密钥不可用 (UNAVAILABLE) 两种; 2) 在密钥可用的前提下, R_{sk} 是否满足数据拥有者规定的访问者属性集范围 ATT_u, 结果有满足 ATT_u (AVAILABLE_SATISFY) 和不满足 ATT_u (AVAILABLE_UNSATISFY) 两种. 具体见算法 2.

算法 2. 密钥验证合约

输入. 数据访问者密钥 R_{sk}、数据拥有者规定的属性集 ATT_u.

输出. 结果 UNAVAILABLE、AVAILABLE_SATISFY、AVAILABLE_UNSATISFY.

1) $result$ = NULL;
2) FOR i = 1 TO $R_{sk}.length$ DO;
3) {;
4) $result$ = $Authenticity\_Decide$ ($R_{sk}[i].signature$);
5) };
6) IF ($result$ = {AVAILABLE});
7) {;
8) $temp$ = TRUE;
9) FOR j = 1 TO $ATT_u.length$ DO;
10) {;
11) $temp$ = $Satisfy\_Decide$ ($ATT_u[j]$, R_{sk});
12) IF ($temp$ = FALSE);
13) BREAK;

14) };

15) IF ($temp$ = TRUE);

16) $result$ = AVAILABLE_SATISFY;

17) ELSE;

18) $result$ = AVAILABLE_UNSATISFY;

19) };

20) RETURN $result$.

3.5 访问控制合约

PHAC 用访问控制合约验证待访问数据 $Date_u$ 的完整性和可用性, 并基于访问者密钥 R_{sk} 和数据拥有者发布的 $Date_u$ 进行匹配判决, 输出允许 (PERMIT) 或拒绝 (DENY). 该合约首先通过 $Verify\_data$ 验证待访问数据, 其次通过查找算法 $Find\_Policy$ 查找到数据拥有者发布的最新访问树事务和访问策略事务; 再以 $AccessControl\_Decide$ 利用数据访问者密钥及访问树事务和访问策略事务计算访问树 T' 的叶子结点的秘密值, 具体计算见式 (6) ~ (9), 如果计算不出任何一个秘密值分量便直接返回DENY. 反之, 采用式 (9) 重构秘密值, 用式 (10) 计算 M, 并返回相关访问请求给云存储服务, 完成交互. 具体见算法 3.

算法 3. 访问控制合约

输入. 数据访问者密钥 R_{sk}、数据拥有者发布的 $Date_u$、区块链 $Blockchain$.

输出. 策略判决结果 PERMIT、DENY.

1) $result$ = NULL, $policy$ = NULL, $tree$ = NULL;

2) IF (!$Verify\_data$ ($Date_u$));

3) RETURN NULL;

4) FOR $i = 1$ TO $Blockchain.length$ DO;

5) {;

6) FOR $j = 1$ TO $Blockchain[i].Transaction\_data.length$ DO;

7) {;

8) IF($policy$ = NULL);

9) {;

10) $temp = Find\_Policy\,(Date_u.signature, ACPT, Blockchain[i].Transaction\_data)$;

11) IF ($temp$! = NULL);

12) $policy$ + = $temp$;

13) };

14) IF ($tree$ = NULL);

15) {;

16) $temp = Find\_Policy\,(Date_u.signature, ACTT, Blockchain[i].Transaction\_data)$;

17) IF ($temp$! = NULL);

18) $tree$ = $temp$;

19) };

20) IF ($policy$! = NULL && $tree$! = NULL);

21) BREAK;

22) };

23) IF ($policy$! = NULL && $tree$! = NULL);

24) BREAK;

25) };

26) $result = AccessControl\_Decide(\,R_{sk}, policy, tree)$;

27) RETURN $result$.

区块链节点执行访问控制判决的具体步骤如下: 1) 数据访问者通过客户端向区块链发起交易, 携带密钥发起访问请求; 2) 区块链节点接收到访问请求, 并向区块链网络中其他节点广播接收到的访问请求; 3) 区块链中的背书节点模拟执行访问控制, 并签名返回至客户端; 4) 背书通过后, 客户端把访问请求及背书节点的签名发送到排序节点; 5) 排序节点把所有接收到的交易数据进行排序, 每过 2 s 或当交易数据达到 10 个时, 生成新的区块; 6) 排序节点把生成的新区块广播到确认节点; 7) 确认节点对新区块中的交易数据进行校验, 并广播新区块; 8) 区块链节点接收到确认节点的新区块后, 更新本地数据; 9) 客户端接收到访问请求的访问控制判决结果; 10) 返回步骤 1).

4 安全性及性能理论分析

本节对提出 PHAC 方法的策略隐藏能力、密钥安全性、存储开销及计算开销进行理论分析. 为了便于分析, 符号说明如下: Pr 表示概率函数, per 表示个人算力, all 表示区块链平台所有算力.

4.1 策略隐藏能力

命题 1. 对于任意的访问控制树 T, 采用 PHAC 方法, 区块链节点无法获知 T 中的属性表达式即数据拥有者的访问控制策略.

证明. 访问控制树 T 中的叶子节点是加密混淆后的访问控制策略, 具体见式 (2) ~ (4). 数据拥有者把构造好的访问控制树 T 上传至区块链中, T 中末端内部节点包含区块链平台中每个属性的每个属性值的密文分量, 通过混淆的方式达到隐藏真实策略属性的目的. 另外, 区块链中的智能合约在计算末端内部节点的秘密值时, 需将访问者的所有密钥分量代入计算, 并验证秘密值是否正确, 因此只可判断访问者的密钥是否满足该末端内部节点的访问策略, 不可以得出末端内部节点具体的访问策略. 因此 PHAC 支持策略隐藏.

对于式 (2) ~ (4), 随机选择 a、b、c、$d \in Z_p^*$ 和 $T \in G_T$, 则给定元组 $D = (g, g^a, g^b, g^c, T)$ 判断等式 $T = e(g, g)^{abc}$ 是否成立是困难的即为一个 Diffie-Hellman 问题[32−34]. 那么对于给定算法 θ, 其攻破 Diffie-Hellman 问题的优势计算如下: $Ad_\theta = |Pr[\theta(D, e(g, g)^{abc}) = 1]| - |Pr[\theta(D, T) = 1]|$. 因为对于任意多项式时间算法 θ, 其优势 Ad_θ 是可以忽略的, 故命题 1 成立. □

4.2 密钥安全性

PHAC 方法的密钥安全性即访问者密钥在无权限访问数据的情况下, 无法通过非法手段伪造密钥访问到数据.

命题 2. 用户密钥 R_{sk} 在不合法、不真实和不满足访问控制树 T 的情况下, 无法访问数据.

证明. 在 PHAC 方法中, 密钥验证合约会对数据访问者的密钥进行验证, 以确保密钥的合法性及真实性. 用户密钥在无权限的情况下, 解密访问共享数据的优势为 $Ad_\theta = |per \geq all \times 0.5|$. 在区块链平台中任何用户的算力优势 Ad_θ 是可以忽略的, 如果存在用户的算力优势 Ad_θ 比较大, 其进行解密的付出要远大于收益[35], 故命题 2 成立. □

由命题 2 可知, 本文提出 PHAC 中的策略隐藏方法与文献 [10] 不同. 文献 [10] 虽然通过混淆方式实现了策略隐藏, 但访问者密钥是由权威中心统一构造发放的, 当访问者密钥恰好符合第 3.1.1 节中计算密文分量的情况 2) 时, 密钥可以解密访问控制树, 因此文献 [10] 不满足命题 2. 而本文提出的 PHAC 要求访问者根据数据拥有者发布的参数构造密钥, 数据拥有者发布的参数会指定一个访问者属性集 ATT_u (包括真实属性和无关属性), 访问者构造密钥时必须包含 ATT_u 属性, 由此可避免上述访问者密钥能解密访问控制策略的风险. 在 PHAC 方法中, ATT_u 不必包含所有真实属性, 以此避免因指定访问者属性集导致恶意用户通过属性集范围推测访问控制策略而造成的隐私泄露风险, 恶意用户通过属性集范围推测访问控制策略, 而造成的隐私泄漏风险. 因此访问者在构造访问者密钥时, 需要多于 ATT_u 的属性来构造密钥. 同时访问者可根据隐私保护需要, 对自己的某些属性取值为 $none$ 后, 再构造密钥, 从而避免访问者本身的属性隐私泄漏风险.

4.3 存储和计算开销

在 PHAC 方法中, 需要数据拥有者构建用户参数、密文数据, 数据访问者构建密钥、区块链平台存储用户参数和密文数据, 区块链平台进行访问控制策略判决 (策略执行阶段). 下面分析区块链平台的存储开销、数据拥有者构建用户参数和访问控制树、数据访问者构建密钥、区块链平台访问控制策略判决的计算开销. 首先, 假设在区块链平台中属性个数是 n, n_i 表示第 i 个属性的取值个数, k ($k \leq n$) 表示用户密钥属性的个数, k_i 表示第 i 个属性的取值个数, $|a|$ 表示数据拥有制定访问控制策略树中的末端内部节点个数, l 表示访问控制结构的通配符数量 (AND、OR 等), $|G|$ 和 $|G_T|$ 分别表示群 G 和 G_T 中元素的大小, G 和 G_T 表示群 G 和 G_T 中运算开销, E 表示双线性映射运算开销, B 表示区块链共识运算开销.

命题 3. 区块链平台存储开销为: $(7 + |a| + |a|\sum_{i=1}^{n} n_i)|G| + (3 + |a|)|G_T|$.

证明. 由式 (2)、式 (3) 可知, 存储开销主要包括 PK_u、C_0、C_1 和 $\{C_\alpha\}_{\alpha \in leaf(T)}$, $\{C_\alpha\}_{\alpha \in leaf(T)}$ 代表 T 下所有末端内部节点的密文分量集合, 所以区块链平台的存储开销为 $(7 + |a| + |a|\sum_{i=1}^{n} n_i)|G| + (3 + |a|)|G_T|$. □

命题 4. 数据拥有者计算开销为: $(7 + |a| + |a|\sum_{i=1}^{n} n_i)G + (3 + |a|)G_T$.

证明. 由式 (2)、式 (3) 可知, 数据拥有者的计算开销主要包括参数 PK_u 计算和访问控制树构建计算, 其中访问控制树构建的主要计算开销包括计算 C_0、C_1 和 $\{C_\alpha\}_{\alpha \in leaf(T)}$, 所以访问控制树构建计算开销为 $(7 + |a| + |a|\sum_{i=1}^{n} n_i)G + (3 + |a|)G_T$. □

命题 5. 密钥构建计算开销为 $2G$.

证明. 根据式 (5)、式 (7)、式 (9) 可知, 访问者密钥构建主要计算 D_0, $\overline{D}_0$, 故密钥构建计算开销为 $2G$. □

命题 6. 访问控制策略判决计算开销为 $(1 + |a| + k|a|)E + (2 + |a| + k|a|)G_T + B$.

证明. 根据式 (6)、式 (7)、式 (9) 可知, 访问控制策略判决计算开销主要包括对 T' 末端内部节点 α 的密文分量计算、重构访问控制树及区块链共识, 所以访问控制策略判决计算开销为 $(1 + |a| + k|a|)E + (2 + |a| + k|a|)G_T + B$. □

基于上述理论分析, 与现有相关研究进行比较, 结果如表 1 所示. 由表 1 可知, 访问控制树较复杂时, 也就是末端内部节点数 $|a|$ 较大时, 相比于文献 [11, 24, 36−37], PHAC 方法的数据存储开销、策略隐藏时加密开销、访问控制判决计算开销略大, 与文献 [10] 大致持平, 优于文献 [29]. 但当用户密钥属性数 k 远少于区块链平台中属性总数 n 时, PHAC 访问控制判决计算开销优于文献 [11, 29, 36−37]. 另外, PHAC 方法和文献 [10] 采用的是树型访问结构, 相比于门限访问结构, 树型访问结构比较复杂,

表 1　本文方法 PHAC 和其他文献方法的对比

Table 1　Comparison of the proposed PHAC with other literature methods

方案	群阶	访问结构	访问者密钥长度	数据存储开销	策略隐藏时加密开销	访问控制判决计算开销
文献 [10]	合数	树	$(2+k)\|G\|$	$(1+\|a\|+\|a\|\sum_{i=1}^{n} n_i)\|G\|+(1+\|a\|)\|G_T\|$	$(2+2\|a\|+2\|a\|\sum_{i=1}^{n} n_i)G+(1+\|a\|)G_T$	$(1+\|a\|+k\|a\|)E+(2+\|a\|+k\|a\|)G_T$
文献 [11]	素数	门	$8\|G\|$	$(8+\sum_{i=1}^{n} n_i)\|G\|+\|G_T\|$	$(14+\sum_{i=1}^{n} n_i)G+2G_T$	$nG+8E+8G_T$
文献 [24]	素数	门	$k\|G\|$	$3\|G\|+\|G_T\|$	$(l+\sum_{i=1}^{n} n_i)G+3G_T$	$3E+2lG+B$
文献 [29]	素数	LSSS	$(10\sum_{i=1}^{n} k_i)\|G\|$	$(2+\|a\|\sum_{i=1}^{n} n_i)\|G\|+n\|G_T\|$	$(2+\sum_{i=1}^{\|a\|} i+4\sum_{i=1}^{n} i+4\sum_{i=1}^{l} i)G+(1+\sum_{i=1}^{l} i)G_T$	$(3\sum_{i=1}^{n} i+\sum_{i=1}^{l} i)E+(1+\sum_{i=1}^{\|a\|} i)G_T+B$
文献 [36]	素数	门	$(1+2n)\|G\|$	$(1+n+\sum_{i=1}^{n} n_i)\|G\|+\|G_T\|$	$(n+\sum_{i=1}^{n} n_i)G+3G_T$	$(1+3n)E+(1+3n)G_T$
文献 [37]	合数	门	$(1+n)\|G\|$	$(1+\sum_{i=1}^{n} n_i)\|G\|+\|G_T\|$	$2(n+\sum_{i=1}^{n} n_i)G+2G_T$	$(1+n)E+(1+n)G_T$
PHAC	素数	树	$(2+k)\|G\|$	$(7+\|a\|+\|a\|\sum_{i=1}^{n} n_i)\|G\|+(3+\|a\|)\|G_T\|$	$(7+\|a\|+\|a\|\sum_{i=1}^{n} n_i)G+(3+\|a\|)G_T$	$(1+\|a\|+k\|a\|)E+(2+\|a\|+k\|a\|)G_T+B$

其能够表达复杂的访问控制策略, 文献 [29] 采用 LSSS 矩阵表达访问控制策略, 会大大增加策略隐藏的计算开销. 相比于上述工作需要可信授权中心来实现参数构造和密钥分发, PHAC 方法基于区块链实现去中心化的访问控制, 可避免对授权中心的依赖, 使数据拥有者真正掌控数据. 需要指出的是, 考虑不同方法对比的一致性, 表 1 中的数据存储开销表示数据拥有者对共享数据及其访问控制策略加密后的密文大小, 访问控制判决计算开销表示数据访问者发起访问请求到得到是否可以访问数据即访问控制判决反馈的计算开销. 由于 PHAC 和文献 [24, 29] 是基于区块链实现的访问控制, 而在区块链访问控制中, 访问控制判决结果是需要区块链共识后才能反馈的, 因此表 1 中 PHAC 和文献 [24, 29] 的访问控制判决计算开销, 考虑了共识带来的计算开销; 而文献 [10−11, 36−37] 不是基于区块链实现的, 故表 1 中文献 [10−11, 36−37] 的访问控制判决计算开销, 没有共识开销.

本文 PHAC 方法中, 策略隐藏是通过混淆的方式实现, 是为访问控制树 T 下的每个末端内部节点计算所有属性值的密文分量, 因此构造访问控制树的存储开销、计算开销和末端内部节点数 $|a|$ 及属性值总数 $\sum_{i=1}^{n} n_i$ 相关, 故在第 5 节实验评估了内部节点数和属性值总数对访问控制树构造即策略构造开销的影响. 同时, 在访问控制判决即解密访问控制树时, 需要基于访问者密钥中的密钥分量和末端内部节点下的密文分量进行双线性映射计算, 计算出末端内部节点的秘密值, 因此访问控制判决的计算开销、末端内部节点数 $|a|$ 和访问者密钥数 k 有关, 故第 5 节实验评估了末端内部节点和访问者密钥中属性个数对访问控制判决计算开销的影响.

5　实验评估

5.1　实验环境

本文基于 HyperLedger Fabric 搭建实验环境. 实验中使用了 4 台物理机, 其中 1 台作为区块链平台, 1 台充当云服务商部署分布式文件系统 Fastdfs 提供存储服务, 1 台作为数据拥有者端上传数据, 1 台作为数据发起访问请求, 实验拓扑见图 10.

HyperLedger Fabric 是一种模块化、支持可插拔组件的基础联盟链框架结构, 根据 Fabric 架构, 证书节点是可选择的, 甚至可以用其他成熟的第三方颁发证书, 同时每个组织中的网络节点数可以有一个或多个, 每一个网络节点必定是一个记账节点, 除此也可以担任一种或多种角色, 比如担任背书节点角色主要对交易预案进行校验、模拟执行和背书, 担任记账节点角色主要负责检验交易的合法性, 并更新和维护区块链数据和账本状态, 担任主节点角色可与节点角色通信, 担任锚节点角色与其他组织锚节点角色通信. 在本文实验中, 不失一般性, 选用 2 个证书节点来模拟区分 2 个不同组织, 每个组织内部有 2 个网络节点, 具体如图 10 所示. 组织 1 的网络节点 0 担任记账节点、主节点和背书节点, 网络节点 1 担任记账节点和锚节点; 组织 2 的网络节点 0 担任记账节点、主节点和背书节点, 网络节点 1 担任记账节点和锚节点. 证书节点主要提供基于数字证书的身份信息, 排序节点按照一定规则确定交易顺序, 数据库节点是存储数据的数据库.

另外, 实验中使用 JPBC2.0 库中的基本密码学代码和双线性映射代码. JPBC2.0 是常见的密码学代码库, 适用于 Java 编程环境. 实验中用到属性基加密代码是在 Cpabe-0.11 的基础上修改的. 节点配置为 3.30 GHz Core(TM)i5-4590 CPU, 8 GB 内

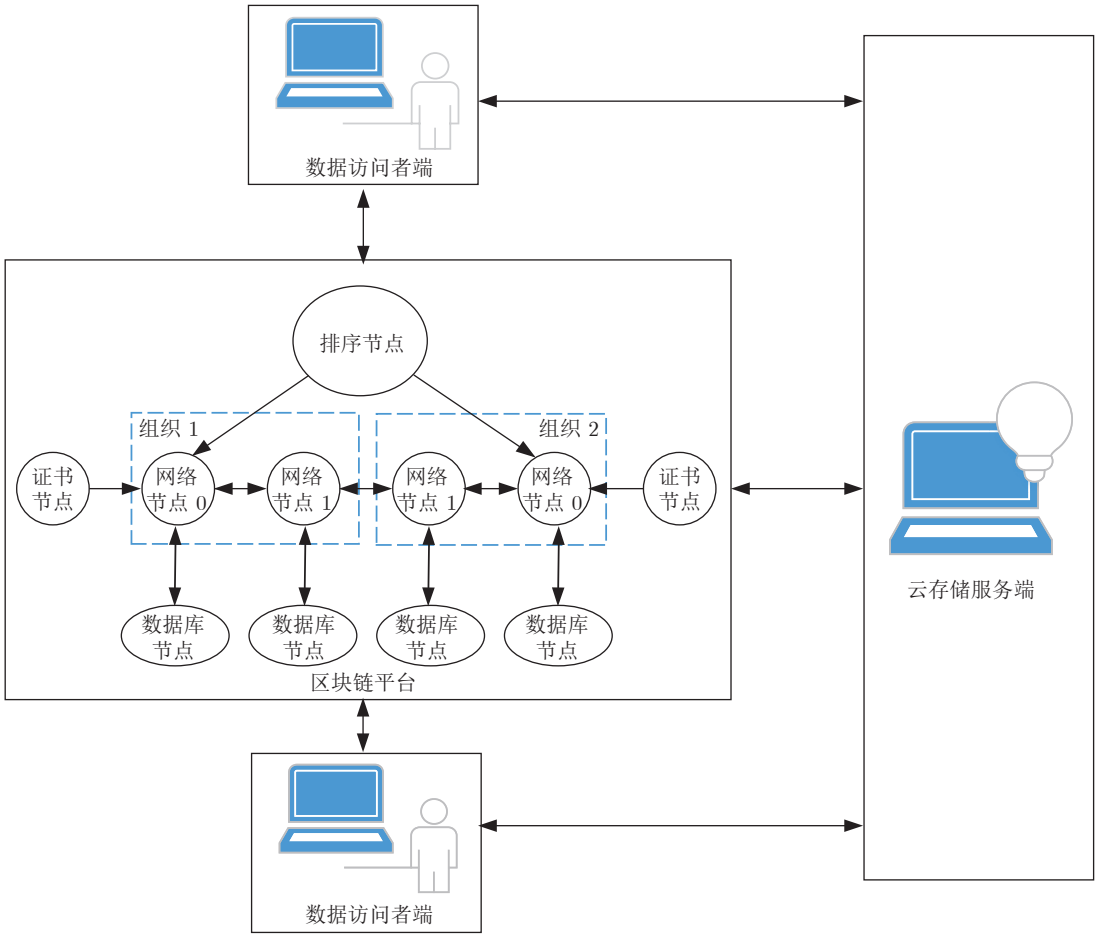

图 10 基于 HyperLedger Fabric 的 PHAC 实验拓扑

Fig. 10 Experimental topology of PHAC based on HyperLedger Fabric

存, 操作系统为 Windows7 和 Ubuntu18.04. 同样, 不失一般性, 实验中设定数据访问者构造密钥所用的属性集为 4 个, 系统属性总数在 4 ~ 14 个之间, 每个属性平均拥有 5 个不同的属性值.

5.2 策略构建时间

本文方法 PHAC 针对第三服务方不可信的情况, 提出访问控制树 T 由数据拥有者来构建, 同时为了实现策略隐藏, 需要为 T 下的每个末端内部节点计算区块链平台属性的所有属性值的密文分量 (见式 (3)). 随着 T 下的末端内部节点的增加, 访问控制树结构变得复杂, 会增加数据拥有者的加密负担, 因此, 本实验通过数据拥有者端构建访问控制树, 测试在不同复杂情况下访问控制树的加密时间, 同时, 由于不同大小的文件作为访问控制树的加密根 (加密根大小和格式是由区块链平台规定, 数据拥有者构建访问控制树的时候使用), 其会对访问控制树的加密效率有所影响, 所以本文选择不同的根加密文件分别为 50 B、80 B、100 B、150 B 和 200 B 进行模式测试.

实验分成两组: 1) 固定 T 下的末端内部节点为 8, 改变区块链平台中属性值总数, 其中区块链平台中属性值总数为区块链平台属性总数和单个属性类型取值数相乘, 不失一般性, 实验中假设不同类型属性的取值数相同. 实验结果如图 11 所示, 其中纵坐标代表加密所需时间, 横坐标代表区块链平台属性值总数. 实验结果表明, 随着属性值总数的增加, 加密时间呈现线性增长. 2) 固定区块链平台属性值总数为 40, 改变 T 下的末端内部节点数. 实验结果如图 12 所示, 其中纵坐标代表加密所需时间, 横坐标代表 T 下末端内部节点数. 实验结果表明, 加密时间会随着 T 下末端内部节点数增加而增加.

由图 11、图 12 可知, 在访问控制树 T 下的末端内部节点固定或是在区块链平台属性值总数固定的情况下, 改变另外一个变量, 访问控制树的加密时间会不断增加, 另外, 加密属性总数即 T 下的末端内部节点总数和区块链平台属性值总数的乘积相近时, 访问控制树加密时间基本相同, 这是因为

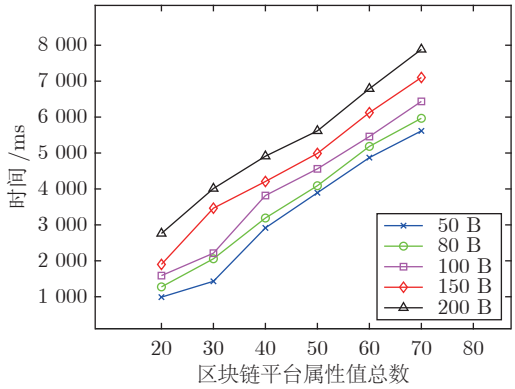

图 11　T 末端内部节点固定时访问控制树的加密时间

Fig. 11　Access control tree encryption time when the internal node of T terminal is fixed

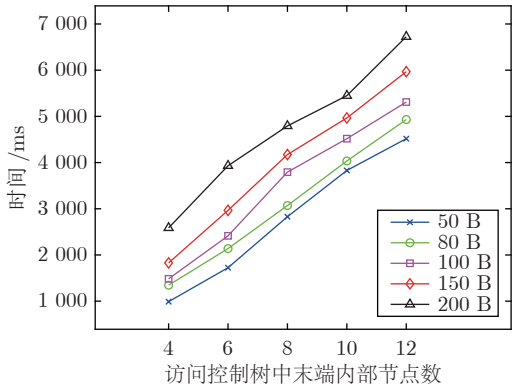

图 12　属性值总数固定时的访问控制树加密时间

Fig. 12　Access control tree encryption time when the total number of attribute values is fixed

PHAC 方法在对访问控制树 T 加密时, 需要为 T 下的每个末端内部节点计算区块链平台属性所有属性值的密文分量, 所以只要加密属性总数增加, 需要计算的密文分量也会增多, 相应访问控制树的加密时间就会增加, 即访问控制树加密时间和加密属性总数呈现正相关. 另外, 根据图 11、图 12 进一步可知, 加密开销随着根加密文件增大而增加, 当加密根为 80 B 以下时, 可忽略不计该时间对用户带来的影响.

5.3　策略执行效率

区块链平台的执行效率通常是指在单位时间内平台执行完成的工作量, 而区块链平台执行完成某项工作的标志是新区块产生并得到共识, 因此共识算法的效率会影响到区块链平台的执行效率, 共识算法是区块链领域的一个重要研究方向. 本文主要关注基于区块链的大数据访问控制中的策略隐藏问题, 因此在策略执行效率实验方面, 重点关注访问控制策略执行效率.

本文方法 PHAC 把策略判决算法写入智能合约, 当访问者发起访问时, 智能合约会先完成访问者密钥验证, 再进行策略判决. PHAC 方法策略判决是通过访问者密钥和访问控制树进行解密计算及区块链达成共识, 由于在解密计算的过程中, 需要首先计算访问控制树 T 的末端内部节点的秘密值, 末端内部节点越多, 访问控制树越复杂意味着区块链节点需要进行的计算就越多. 文献 [6, 9] 均提出基于区块链的访问控制方法, 并采用区块链结合基于属性访问控制的方式. 其中, 文献 [6] 的性能测试实验结果显示, 随着策略的增加, 访问控制判决时间增长较为明显. 文献 [9] 基于以太坊完成性能测试. 然而文献 [6, 9] 都未考虑策略隐藏的问题, 因此, 本文首先进行策略未隐藏下的区块链访问控制实验, 与文献 [6, 9] 比较策略判决时间. 策略判决时间是指数据访问者发起请求到获取数据的时间统计.

由于访问控制策略判决是通过访问者密钥解密访问控制树实现的, 也就是说越复杂的访问控制树末端内部节点的数量越多, 访问控制树越复杂会影响解密计算开销, 所以本文选择复杂程度不同的访问控制树进行测试. 实验中, 本文去除了访问控制树末端内部节点的混淆密文分量, 只留下真实可用的密文分量, 假定每个末端内部节点的真实密文分量为 2 个; 策略判决时间为输入访问者密钥和访问控制树开始到返回判决结果时结束. 实验结果见图 13, 其中纵坐标代表策略判决时间, 横坐标代表访问控制树的末端内部节点数量. 由图 13可知, 访问控制树越复杂即末端内部节点数量越多, 策略判决时间也随之增加, 但本文方法在策略未隐藏情况下, 与文献 [6, 9] 方法相比, 时间性能较优.

然后, 本文进行策略隐藏下的区块链访问控制实验, 测试 PHAC 方法对策略判决时间的影响. 在前面策略隐藏实验中, 增加末端内部节点的混淆密文分量, 并选择针对不同的区块链平台属性值总数和不同复杂程度的访问控制树即访问控制树的末端内部节点数量不同进行测试. 实验中, 策略隐藏下的策略判决时间为从输入访问者密钥和访问控制树开始, 到返回判决结果结束. 实验结果如图 14 所示, 其中纵坐标代表策略判决时间, 横坐标代表区块链平台属性值总数. 由图可以看出, 随着区块链平台属性值总数的增加, 策略判决时间大致是保持不变的, 说明策略判决时间开销与属性值总数无关. 同时, 当选择复杂程度不同的访问控制树进行测试时, 实验结果表明, 访问控制树越复杂即末端内部节点数量越多, 策略判决时间会有所增加.

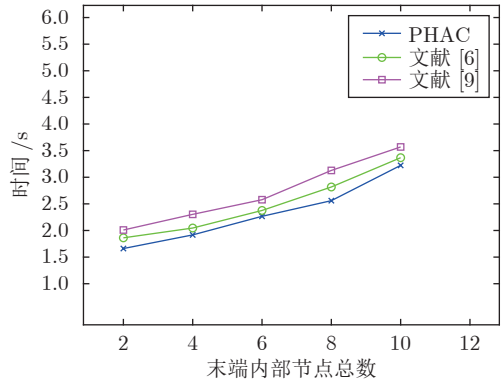

图 13 策略未隐藏下的策略判决时间

Fig. 13 Policy decision time without policy hidden

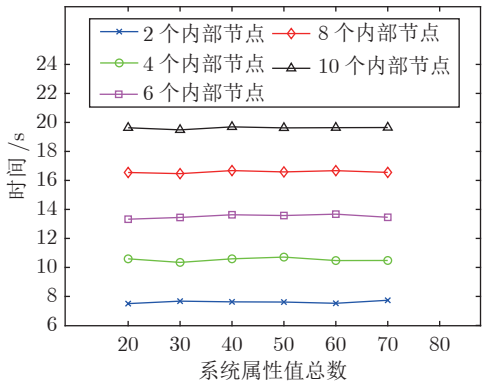

图 14 策略隐藏下的策略判决时间

Fig. 14 Policy decision time under policy hidden

根据上述实验, 策略判决时间开销与属性值总数即访问控制树末端内部节点的密文分量总数无关. 然而, 由于在策略判决时需要首先恢复末端内部节点的秘密值, 秘密值是根据访问者密钥中的密钥分量和末端内部节点的密文分量对应进行双线性计算的, 而访问者密钥中的密钥分量由访问者属性集构造. 因此, 本文通过实验进一步测试访问者属性集对策略判决时间的影响. 不失一般性, 实验中, 设定区块链平台属性总数为 14, 属性值总数为 70, 访问控制树末端内部节点为 4 个, 改变访问者属性集进行实验, 实验结果如图 15 所示. 实验结果表明, 随着访问者属性集规模的增大, 策略判决时间也会随之增加.

此外, 本文还比较了 PHAC 方法在策略隐藏下和策略未隐藏时的策略判决时间开销, 并与现有策略隐藏文献 [24, 29] 进行了对比. 如图 16 所示, 文献 [24, 29] 和 PHAC 在策略隐藏时的策略判决时间开销均高于策略未隐藏时的策略判决时间, 这是由于策略隐藏会导致解密匹配操作复杂化[34−35], 进而导致策略判决时间增加; 而 PHAC 方法的时间开销优于文献 [29], 但是稍微高于文献 [24], 因为文献 [24] 仅对部分策略进行隐藏, 而 PHAC 是对全策略进行混淆隐藏; 另外, 文献 [24, 29] 需要引入可信授权中心 AA, 这会违背分布式访问控制的初衷, PHAC 方法提出访问密钥由数据访问者自己构造、密钥验证由区块链节点执行的智能合约, 以避免对 AA 的信任依赖, 安全性更高.

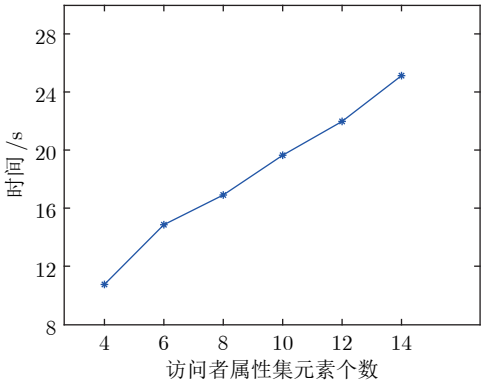

图 15 访问者属性集对策略判决时间的影响

Fig. 15 Influence of visitor attribute sets on policy decision time

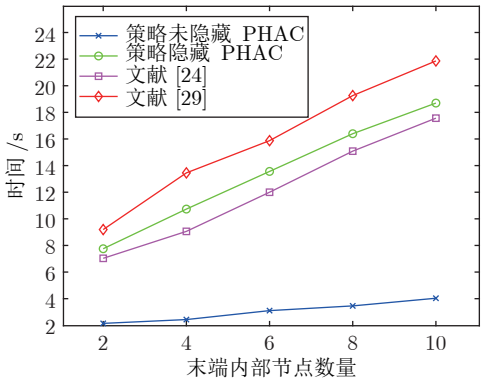

图 16 策略隐藏和策略未隐藏下的策略判决时间对比

Fig. 16 Comparison of policy decision times under policy hidden and without policy hidden

5.4 策略存储开销

由于区块链具有不可篡改性、可追溯性等特点, 存储在区块链平台的访问控制策略不能修改、删除, 每次更新访问控制策略都需把新的访问控制策略发布至区块链平台中, 因此随着策略更新次数的增加, 区块链平台节点的存储开销通常都会增加. 而在本文方法 PHAC 中, 访问控制策略用访问控制树来描述, 访问控制树中的访问树和加密混淆后的访问策略被解耦, 分别以访问树事务和访问策略事务存储, 当用户更新访问控制策略时, 只需更新变化的访问

策略即可, 其他未变的访问策略不需要更新, 同时也不必更新访问树, 因此可降低区块链平台存储开销的增加幅度.

为了验证 PHAC 的存储开销, 本文在实验中构造了拥有 4 个末端内部节点的访问控制树存放到区块链平台中, 每一个末端内部节点下的密文分量大小约在 500 B 左右, 并对区块链平台中同一个访问控制策略进行了多次更新, 统计了未解耦访问控制树和解耦访问控制树后不同内部节点变化情况下, 策略更新次数由小到大时的区块链平台存储开销, 结果如图 17 所示, 其中横坐标代表策略更新次数, 纵坐标代表区块链平台存储开销. 实验结果表明, 随着策略更新次数的增加, 无论哪种情况, 区块链平台存储开销都会线性增加, 但是由于在本文方法 PHAC 中, 当用户更新访问控制策略时, 只需更新变化的访问策略即可, 因此区块链平台的存储开销增加相对平缓. 而文献 [6, 9, 24, 29] 均属于未解耦访问控制树的情况, 其采用每次更新策略都重新编写发布整个访问控制策略至区块链平台, 因而区块链平台存储开销会大幅度增加.

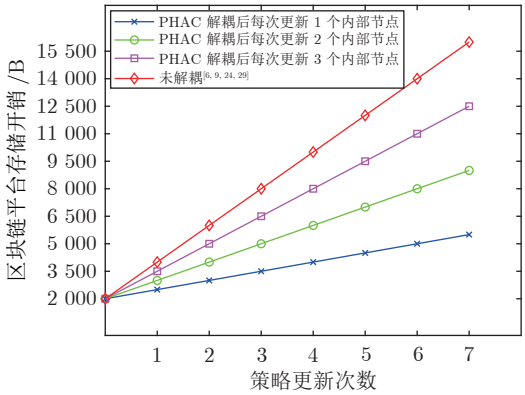

图 17　区块链平台存储开销

Fig. 17　The storage overhead of blockchain platform

此外, 本文方法直接采用了 Fabric 共识算法, 故未专门实验测试共识运算的计算资源开销. 值得注意的是, 共识算法是区块链研究的一个重点方向, 本文后续会关注共识算法的计算开销问题.

6　结束语

本文提出一种基于区块链的策略隐藏大数据访问控制方法, 该方法融合区块链和双线性映射技术, 借助区块链的去中心、不可篡改、透明性、协商一致性等特点, 让用户从委托半可信云服务商来提供访问控制服务的模式下解脱出来, 同时利用双线性映射技术实现在不泄露访问控制策略的前提下, 利用智能合约执行访问控制策略. 理论分析和实验评估结果表明, 本文方法能在策略隐藏情况下实现访问控制, 与现有经典工作相比, 进一步增加了访问控制的安全性和机密性, 并拥有不错的时间开销.

目前, 本文方法利用双线性映射实现策略隐藏, 随着属性数量、属性值的增多和访问控制树的复杂度增加, 双线性映射计算可能会增加访问控制策略的判决时间, 未来打算借鉴代理重加密及外包加解密计算技术解决此问题. 另外, 本文并未对节点、平台收益问题及区块链本身展开过多叙述, 未来将继续对收益问题和区块链本身 (共识算法等) 进行探索研究, 本文当前侧重对数据拥有者访问控制策略进行隐私保护, 尚未考虑数据访问者的隐私保护问题[35, 38], 未来将进一步研究支撑供需双方隐私保护以实现大数据最大安全共享的访问控制方法.

References

1　Berdik D, Otoum S, Schmidt N, Porter D, Jararweh Y. A survey on blockchain for information systems management and security. *Information Processing & Management*, 2021, **58**(1): Article No. 102397

2　Liu Ming-Da, Chen Zuo-Ning, Shi Yi-Juan, Tang Ling-Tao, Cao Dan. Reseach progress of blockchain in data security. *Chinese Journal of Computers*, 2021, **44**(1): 1−27
(刘明达, 陈左宁, 拾以娟, 汤凌韬, 曹丹. 区块链在数据安全领域的研究进展. 计算机学报, 2021, **44**(1): 1−27)

3　Yuan Yong, Wang Fei-Yue. Editable blockchain: Models, techniques and methods. *Acta Automatica Sinica*, 2020, **46**(5): 831−846
(袁勇, 王飞跃. 可编辑区块链: 模型、技术与方法. 自动化学报, 2020, **46**(5): 831−846)

4　Maesa D D F, Mori P, Ricci L. Blockchain based access control. In: Proceedings of the 17th IFIP International Conference on Distributed Applications and Interoperable Systems. Cham, Switzerland: Springer, 2017. 206−220

5　Yang C, Tan L, Shi N, Xu B, Cao Y, Yu K. Authprivacychain: A blockchain-based access control framework with privacy protection in cloud. *IEEE Access*, 2020, **8**: 70604−70615

6　Liu Ao-Di, Du Xue-Hui, Wang Na, Li Shao-Zhuo. A blockchain-based access control mechanism for big data. *Journal of Software*, 2019, **30**(9): 2636−2654
(刘敖迪, 杜学绘, 王娜, 李少卓. 基于区块链的大数据访问控制机制. 软件学报, 2019, **30**(9): 2636−2654)

7　Maesa D D F, Mori P, Ricci L. Blockchain based access control services. In: Proceedings of the IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data. New York, USA: IEEE, 2018. 1379−1386

8　Wang Xiu-Li, Jiang Xiao-Zhou, Li Yang. Model for data access control and sharing based on blockchain. *Journal of Software*, 2019, **30**(6): 1661−1669
(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. 软件学报, 2019, **30**(6): 1661−1669)

9　Maesa D D F, Mori P, Ricci L. A blockchain based approach for

the definition of auditable access control systems. *Computers & Security*, 2019, **84**: 93−119

10 Song Yan, Han Zhen, Liu Feng-Mei, Liu Lei. Attribute-based encryption with hidden policies in the access tree. *Journal on Communications*, 2015, **36**(9): 119−126
(宋衍, 韩臻, 刘凤梅, 刘磊. 基于访问树的策略隐藏属性加密方案. 通信学报, 2015, **36**(9): 119−126)

11 Wang Hai-Bin, Chen Shao-Zhen. Attribute-based encryption with hidden access structures. *Journal of Electronics & Information Technology*, 2012, **34**(2): 457−461
(王海斌, 陈少真. 隐藏访问结构的基于属性加密方案. 电子与信息学报, 2012, **34**(2): 457−461)

12 Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security. New York, USA: 2006. 89−98

13 Boneh D, Franklin M. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 2003, **32**(3): 586−615

14 Sahai A, Waters B. Fuzzy identity-based encryption. In: Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Germany: Springer, 2005. 457−473

15 Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proceedings of the 28th IEEE Symposium on Security and Privacy. Oakland, USA: IEEE, 2007. 321−334

16 Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 14th ACM Conference on Computer and Communications Security. New York, USA: 2007. 195−203

17 Zhou Z, Huang D, Wang Z. Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption. *IEEE Transactions on Computers*, 2015, **64**(1): 126−128

18 Hong Cheng, Zhang Min, Feng Deng-Guo. AB-ACCS: A cryptographic access control scheme for cloud storage. *Journal of Computer Research and Development*, 2010, **47**(Suppl.): 259−265
(洪澄, 张敏, 冯登国. AB-ACCS: 一种云存储密文访问控制方法. 计算机研究与发展, 2010, **47**(增刊): 259−265)

19 Wang Y, Li F, Xiong J, Niu B, Shan F. Achieving lightweight and secure access control in multi-authority cloud. In: Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Helsinki, Finland: IEEE, 2015. 459−466

20 Jung T, Li X Y, Wan Z, Wan M. Privacy preserving cloud data access with multi-authorities. In: Proceedings of the 32nd IEEE Conference on Computer Communications. Turin, Italy: IEEE, 2013. 2625−2633

21 Guan Zhi-Tao, Yang Ting-Ting, Xu Ru-Zhi, Wang Zhu-Xiao. Multi-authority attribute-based encryption access control model for cloud storage. *Journal on Communications*, 2015, **36**(6): 120−130
(关志涛, 杨亭亭, 徐茹枝, 王竹晓. 面向云存储的基于属性加密的多授权中心访问控制方案. 通信学报, 2015, **36**(6): 120−130)

22 Lin H, Cao Z, Liang X, Shao J. Secure threshold multi authority attribute based encryption without a central authority. *Information Sciences*, 2010, **180**(13): 2618−2632

23 Ding X, Yang J. An access control model and its application in blockchain. In: Proceedings of the International Conference on Communications, Information System and Computer Engineering. Haikou, China: IEEE, 2019. 163−167

24 Ba Y, Hu X, Chen Y, Hao Z, Li X, Yan X. A blockchain-based CP-ABE scheme with partially hidden access structures. *Security and Communication Networks*, 2021, **2021**: Article No. 4132597

25 Wang S, Zhang Y, Zhang Y. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access*, 2018, **6**: 38437−38450

26 Zhang Jian-Biao, Zhang Zhao-Qian, Xu Wan-Shan, Wu Na. Inter-domain access control model based on blockchain. *Journal of Software*, 2021, **32**(5): 1547−1564
(张建标, 张兆乾, 徐万山, 吴娜. 一种基于区块链的域间访问控制模型. 软件学报, 2021, **32**(5): 1547−1564)

27 Makhdoom I, Zhou I, Abolhasan M, Lipman J, Ni W. PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities. *Computers and Security*, 2020, **88**: Article No. 101653

28 Gao S, Piao G, Zhu J, Ma X, Ma J. Trustaccess: A trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain. *IEEE Transactions on Vehicular Technology*, 2020, **69**(6): 5784−5798

29 Zhang Z, Zhang J, Yuan Y, Li Z. An expressive fully policy-hidden ciphertext policy attribute-based encryption scheme with credible verification based on blockchain. *IEEE Internet of Things Journal*, 2022, **9**(11): 8681−8692

30 Xia Qing, Dou Wen-Sheng, Guo Kai-Wen, Liang Geng, Zuo Chun, Zhang Feng-Jun. Survey of blockchain consensus protocols. *Journal of Software*, 2021, **32**(2): 277−299
(夏清, 窦文生, 郭凯文, 梁赓, 左春, 张凤军. 区块链共识协议综述. 软件学报, 2021, **32**(2): 277−299)

31 Zhang Y, Kasahara S, Shen Y, Jiang X, Wan J. Smart contract-based access control for the internet of things. *IEEE Internet of Things Journal*, 2018, **6**(2): 1594−1605

32 Shparlinski I E. Communication complexity and Fourier coefficients of the Diffie-Hellman key. In: Proceedings of the 4th Latin American Symposium on Theoretical Informatics. Berlin, Germany: Springer, 2000. 259−268

33 Boneh D, Boyen X. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 2011, **24**: 659−693

34 Yang Hao-Miao, Sun Shi-Xin, Li Hong-Wei. Research on bilinear Diffie-Hellman problem. *Journal of Sichuan University (Engineering Science Edition)*, 2006, **38**(2): 137−140
(杨浩淼, 孙世新, 李洪伟. 双线性 Diffie-Hellman 问题研究. 四川大学学报 (工程科学版), 2006, **38**(2): 137−140)

35 Ghayvat H, Pandya S, Bhattacharya P, Zuhair M, Rashid M, Hakak S, et al. CP-BDHCA: Blockchain-based confidentiality-privacy preserving big data scheme for healthcare clouds and applications. *IEEE Journal of Biomedical and Health Informatics*, 2021, **26**(5): 1937−1948

36 Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures. In: Proceedings of the 6th International Conference on Applied Cryptography and Network Security. Berlin, Germany: Springer, 2008. 111−129

37 Lai J, Deng R H, Li Y. Fully secure cipertext-policy hiding CP-ABE. In: Proceedings of the 7th International Conference on Information Security Practice and Experience. Berlin, Germany: Springer, 2011. 24−39

38 Xu Ke, Ling Si-Tong, Li Qi, Wu Bo, Shen Meng, Zhang Zhi-Chao, et al. Research progress of network security architecture and key technologies based on blockchain. *Chinese Journal of Computers*, 2021, **44**(1): 55−83
(徐恪, 凌思通, 李琦, 吴波, 沈蒙, 张智超, 等. 基于区块链的网络安全体系结构与关键技术研究进展. 计算机学报, 2021, **44**(1): 55−83)

林 莉 北京工业大学信息学部副教授. 主要研究方向为大数据安全与隐私保护, 访问控制和区块链应用. 本文通信作者.
E-mail: linli_2009@bjut.edu.cn
(**LIN Li** Associate professor at the Faculty of Information Technology, Beijing University of Technology. Her research interest covers big data security and privacy protection, access control, and blockchain application. Corresponding author of this paper.)

储振兴 北京工业大学信息学部硕士研究生. 主要研究方向为区块链与访问控制.
E-mail: tianzhenxingchu@163.com
(**CHU Zhen-Xing** Master student at the Faculty of Information Technology, Beijing University of Technology. His research interest covers blockchain and access control.)

刘子萌 北京工业大学信息学部硕士研究生. 主要研究方向为区块链与云安全. E-mail: zimeng_liuu@163.com
(**LIU Zi-Meng** Master student at the Faculty of Information Technology, Beijing University of Technology. Her research interest covers blockchain and cloud security.)

郭馥宾 北京工业大学信息学部硕士研究生. 主要研究方向为网络安全与区块链.
E-mail: gfb18438607915@163.com
(**GUO Fu-Bin** Master student at the Faculty of Information Technology, Beijing University of Technology. His research interest covers network security and blockchain.)

解晓宇 北京工业大学信息学部硕士研究生. 主要研究方向为区块链与云计算. E-mail: 18733655212@163.com
(**XIE Xiao-Yu** Master student at the Faculty of Information Technology, Beijing University of Technology. Her research interest covers blockchain and cloud computing.)

张建标 北京工业大学信息学部教授. 主要研究方向为信息安全与云计算. E-mail: zjb@bjut.edu.cn
(**ZHANG Jian-Biao** Professor at the Faculty of Information Technology, Beijing University of Technology. His research interest covers information security and cloud computing.)