

基于区块链的可撤销边缘物联网数据共享方案

白诗童¹, 张嘉伟²

(1. 复旦大学, 上海 200433; 2. 西安电子科技大学网络与信息安全学院, 陕西 西安 710071)

摘要: 边缘物联网通过分布式结构部署在用户附近实现低延迟的数据共享。然而, 数据所有权与控制权的分离导致共享数据面临严重的非法访问和信任问题。密文策略属性基加密 (CP-ABE) 技术能够提供数据机密性和细粒度访问控制, 但难以适用于分布式边缘环境, 且容易受到动态用户身份和属性伪造攻击, 同时也给设备带来沉重负担。此外, 现有大多数方案在时间敏感数据共享和隐私保护方面仍存在不足。同时, 现有方案并未考虑边缘服务器可能破坏数据完整性的问题。本文提出了一种基于区块链的可撤销且可信的数据共享方案 BRTDS。该方案将区块链和零知识证明引入可撤销的 CP-ABE 原语, 面向动态用户实现可信的分布式数据访问控制, 以抵御身份和属性伪造攻击。设计了一种时间释放隐藏策略树支持对时间敏感数据的访问控制和属性隐私保护。此外, 利用智能合约构建了分布式数据审计机制, 实现去中心化的共享数据完整性审计。同时还实现了高效加密与可验证解密以提升整体计算效率。形式化分析和大量仿真实验结果证明了本文方案的安全性、实用性和高效性。

关键词: 边缘计算; 密文策略属性基加密; 区块链; 用户撤销; 数据审计

中图分类号: TN309

文献标志码: A

Enabling revocable data secure sharing based on blockchain for edge-based IoT

BAI Shitong¹, ZHANG Jiawei²

1. College of Computer Science and Artificial Intelligence, Fudan University, Shanghai 200438, China

2. School of Cyber Engineering, Xidian University, Xi'an 710071, China

Abstract: Edge-based Internet of Things (IoT) emerges to facilitate low-latency data sharing in a decentralized way. However, the intrinsic separation of data ownership and control inflicts serious illegal access and loss of trust on the data sharing. Although Ciphertext Policy Attribute-Based Encryption (CP-ABE) enables confidentiality and fine-grained access control, it is susceptible to impersonation attack from distributed and dynamic users, and its high overhead brings heavy burden to devices, which poses a huge challenge to its application in IoT data sharing. Moreover, most of existing schemes are insufficient to exert privacy-preserving access control on time-sensitive data. Furthermore, semi-trusted edge servers may compromise data integrity, which is seldomly considered. In this work, a Blockchain-assisted Revocable and Trustworthy Data Sharing (BRTDS) scheme is proposed for time sensitive data. It combines blockchain and zero knowledge proof with revocable CP-ABE to ensure distributed data access control across dynamic users and resist malicious users with impersonated identities and attributes. By designing a time-sensitive and privacy-preserving access policy tree, privacy-preserving access control is achieved. Furthermore, a distributed data auditing mechanism is devised through smart contracts to guarantee data integrity in access control process without a centralized auditor. Efficient encryption and verifiable data decryption is achieved with a better performance. Finally, the security of our proposal and its practicability and efficiency is demonstrated through extensive performance evaluation with simulations.

收稿日期: XXXX-XX-XX; 修回日期: XXXX-XX-XX

基金项目: 国家自然科学基金资助项目 (No. 62171356)

Foundation Items: The National Natural Foundation of China (No. 62171356)

Key words: edge-computing, CP-ABE, blockchain, user-revocable, data auditing

1 引言

随着人工智能和5G通信技术的快速发展,物联网^[1]被广泛应用在各个领域,包括工业物联网^[2]、卫星物联网^[3]、医疗物联网^[4]、车联网^[5]和无人机网络^[6]等,并成为数字经济发展的关键引擎^[7]。然而,随着以物联网为基础的各种新型应用的推广,物联网设备数量激增,其产生的数据规模也随之爆炸式增长^[8]。根据 Statista 统计,到2034年,全球物联网设备预计将达到约406亿台,每日数据产生规模将大幅跃升^[9]。庞大的数据量严重超出资源受限的物联网设备的存储和处理能力。因此,如何高效地共享和处理规模庞大的物联网数据,已成为物联网面临的一个重要问题。

基于云计算的物联网范式^[10]极大地缓解了海量物联网数据与资源受限的物联网设备之间的矛盾。该范式利用云计算实现物联网数据的外包存储和共享流通。企业和个人可将数据外包至云端,以减轻设备负担并实现与参与方的数据共享。随着很多即时决策和预测应用的出现,低延迟物联网数据外包和共享需求日益增加,推动了基于边缘计算的物联网^[11]发展。然而,分布式的边缘服务器可能因故障或非法牟利而导致数据破坏或隐私泄露,从而造成行业机密外泄或事故发生^[12]。因此,如何均衡边缘物联网数据共享和数据安全至关重要^[13]。

一种可行的应对措施是密文策略属性基加密(CP-ABE)^[14-15]技术。作为属性基加密(ABE)^[16]的一种,CP-ABE通过指定的访问策略对数据密文实施细粒度访问控制。然而,将现有CP-ABE方案直接用于边缘物联网数据安全共享仍存在诸多挑战。由于外包存储在边缘服务器的物联网数据密文与其访问策略同时进行共享,而访问策略中的属性包含大量敏感信息,这样很容易被泄露^[17]。此外,传统CP-ABE方案在加密与解密过程中计算开销较高,给资源受限的物联网设备和用户智能终端带来了沉重负担^[18]。因此,属性隐私泄露和高计算开销为边缘物联网数据安全共享带来了巨大挑战。

在实际应用中,外包存储至边缘服务器的物联网数据通常需要在动态的数据用户之间共享,并且往往需要根据时间顺序进行访问控制^[19-20]。例如,

医院的医疗数据向医生和研究机构进行共享,其中要求研究机构在稍晚的时间点才能够访问共享医疗数据以进行科学研究。同时,当医院禁止将数据向医疗机构共享时,必须立即撤销该医疗机构的访问权限。针对基于时间点进行共享数据访问控制的需求,Hong等人^[21]提出了一种结合时间因素与属性的CP-ABE方案,但仍未解决动态用户撤销的问题。一些方案^[22-23]将直接用户撤销机制引入CP-ABE,但引入较高的计算开销,且未能实现基于时间的访问控制。因此,时间因素和用户动态性的需求给边缘物联网数据安全共享带来新的挑战。

目前,许多已有CP-ABE方案在应用于边缘物联网数据共享时,忽视了访问控制的信任问题以及边缘服务器的恶意行为。一方面,当与密文同时存储在分布式边缘服务器中的访问策略遭到篡改时,会导致用户权限提升^[24]。此外,恶意用户也可能通过伪装攻击实现权限提升。另一方面,边缘服务器可能为了维护声誉而隐瞒因操作失误或外部攻击导致的数据损坏,甚至为节省成本删除不常用的数据^[25],从而造成数据完整性破坏^[26-28]。上述实际应用中的问题在当前的CP-ABE方案中很少被考虑,很容易造成在实际应用中用户权限提升和共享数据破坏。因此,确保共享数据的可信任性和可审计性是边缘物联网数据访问控制是数据安全共享面临的第三大挑战。

根据上述讨论,已有的CP-ABE方案很少能同时解决边缘物联网数据安全共享中的上述挑战。为此,本文提出了一种基于区块链的可撤销且可信的数据共享方案BRTDS。该方案针对边缘物联网的分布式架构、信任性缺失、无法对时间敏感数据向动态用户进行访问控制等问题,实现了基于时间的分布式可信细粒度数据访问控制。同时,BRTDS还支持访问策略隐私保护以及高效的数据加密解密和用户直接撤销。

本文的主要贡献总结如下。

1) 提出了一种适用于边缘物联网数据安全共享的分布式可撤销数据访问控制方法。该方法利用零知识证明实现灵活的用户身份与属性验证,并将区块链和智能合约与可撤销CP-ABE相结合,从而实现分布式的数据访问控制及其可信任性和可撤销

性。该方法还支持去中心化的分布式完整性审计，在解决边缘节点恶意破坏共享数据问题的同时，确保审计结果可信任性。

2) 设计了一种隐私保护且支持时间释放的访问策略树, 以满足物联网时间敏感数据的共享对属性隐私保护和基于时间访问控制的需求。该访问结构通过隐藏访问控制树所有属性值来实现其隐私保护, 并将时间陷门与访问策略树中指定节点相关联来支持基于时间的访问控制。通过将该访问结构与所设计的 CP-ABE 方法相结合, 支持数据用户只有在其属性和访问时间同时满足访问结构时才能访问共享物联网数据的安全功能。

3) 利用在线/离线加密以及可验证的外包解密技术实现较低的计算和存储开销。通过形式化的安全分析证明本文方案在面对选择明文攻击 (CPA) 时的语义安全性, 并通过大量实验和性能评估表明本文方案在计算和存储开销方面具有更高的效率, 适用于实际的物联网数据共享场景。

2 问题描述

2.1 系统模型

本文方案如图1所示，BRTDS共包含五个实体，即边缘服务器（ES）、区块链（BC）、属性授权机构（AA）、数据提供方（DP）和数据用户（DU）。

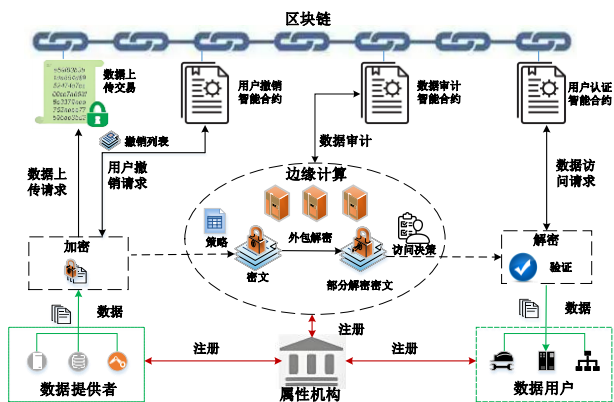


图1 方案BRTDS系统模型

具体描述如下：

1) ES: ES具有可扩展存储和计算资源,可为用户提供数据存储、计算和共享服务,例如对数据提供方的密文进行外包存储、解密以及时间陷门暴露。此外,ES在用户撤销后也承担密文更新的

责任。

2) AA: AA 负责系统初始化, 为整个系统生成系统公共参数和主密钥。同时, 在用户注册成功通过身份认证后, 根据用户的标识和属性集为其生成私钥和解密密钥。此外, 它会在每个时间点 t 定期公开发放时间令牌。

3) BC: BC 负责用于访问控制的不可变元数据存储, 并代表数据用户与 ES 执行数据审计协议, 部署相应的智能合约以完成数据审计以及用户身份和属性验证。BC 还用于作为数据共享、数据访问和用户撤销的辅助。

4) DP: DP生成大量物联网数据,用于监控、决策和处理。为减轻物联网设备的本地负担,DP在对敏感数据进行隐私保护并采用时间释放隐藏策略树实现细粒度访问控制的前提下,将数据外包给ES进行共享。

5) DU: 一个有效且未被撤销的 DU 可在有效时间点根据自身需求访问 ES 中的共享数据。DU 在收到 BC 的数据审计结果以及 ES 返回的部分解密密文并验证正确性后, 可恢复相应的明文。任何存在恶意行为的 DU 将被直接撤销。

2.2 威胁模型

在方案系统模型中, AA 和 DP 被视为完全可信的实体, 而 ES 则被认为是半可信的, 其会诚实执行指定协议, 但可能故意泄露或篡改敏感数据的内容以及外包解密的结果。此外, ES 还可能因硬件或软件错误, 有意或无意地破坏共享数据的完整性。BC 可被视为可信的分布式账本。另外, 一些未经授权或被撤销的 DU 可能会非法访问共享数据, 导致敏感和私密信息泄露。此外, 部分 DU 也可能进行恶意活动以获取额外利益。

2.3 安全模型

假设敌手能够充当不具备足够的权限访问数据的恶意外部实体（类型1）或意图从外包数据中收集敏感信息的已腐蚀ES（类型2）。本文通过描述在挑战者C与敌手A之间的安全游戏，来构建本文方案BRTDS在选择性访问策略和选择明文攻击（CPA）下的不可区分安全性模型。具体如下：

初始化。敌手 A 向挑战者 C 发出一个挑战访问策略树 T 和一个撤销列表 RL' ，并指定一个时间点 t' 。

设置。C 执行系统公共参数 PP 的 $\text{Setup}_{\text{sys}}$ 算法,

然后将PP发送给A。至于ES的设置,则遵循以下两种情况:

● 对于类型1的对手, C运行 $\text{Setup}_{\text{cloud}}$ 并将ES的公钥 PK_c 发送给C。

● 对于类型2的对手, A运行 $\text{Setup}_{\text{cloud}}$ 并将ES的公钥 PK_c 发送给A。

查询1。A发出一系列解密密钥查询,其属性集为 $S_1, S_2, \dots, S_q$,每个属性集均不满足包含所有层级节点 $\ln_i \in T'$ 的挑战访问策略 T' 。

挑战。对手A提交两个等长的消息 M_0, M_1 作为挑战。挑战者C随机选择 $b \in \{0, 1\}$,并根据挑战访问策略 T' 对 M_b 进行加密,得到其密文CT'。然后,挑战者C将CT'返回给A。

查询2。对手A提交的查询与查询1类似。

猜测。敌手A猜测 b 的值 b' 。如果 $b'=b$,则A获胜,其优势定义为: $\text{Adv}_1 = |\Pr[b'=b] - 1/2|$ 。

定义1: 本文方案在上述安全游戏中,任何具有非可忽略优势的敌手都无法破解的前提下,是CPA安全的。

3 方案构造

3.1 时间释放隐藏策略树

时间释放隐藏策略树是方案BRTDS为实现对基于时间和细粒度数据访问控制以及属性隐私保护而设计的一种访问结构。它是对方案TAFC^[21]中提出的时间释放访问策略树的扩展,通过引入属性隐藏技术来保护属性的隐私。用T表示时间释放隐藏策略树。T包含一个叶节点集合X以及一个非叶节点集合。其中,每个叶节点对应一个属性,非叶节点包括不同阈值的门节点。与方案TAFC的访问策略树不同,T包含一个陷阱门节点集合 T_R ,其中每个节点 x 对应一个时间陷阱门 TD_x ,并通过MurHash3隐藏叶节点的值。具体说明如下:

1) x : T中的一个节点。如果 x 是非叶节点,则它是一个阈值门,包含AND、OR和 $n\text{-of-}m$ ($n>m$)门;如果 x 是叶节点,则表示一个特定属性。

2) index_x : T中分配给每个节点 x 的唯一值。

3) num_x : 如果 x 是T的非叶节点,则 num_x 表示 x 的子节点数量。

4) thr_x : 如果 x 是T的非叶节点,则 thr_x 表示 x 的阈值,其中 $\text{thr}_x \in [1, \text{num}_x]$ 。当 $\text{thr}_x = 1$ 时,称该 x 为OR门;当 $\text{thr}_x = \text{num}_x$ 时,称为AND门。若 x

是叶节点,则定义 $\text{thr}_x = 1$ 。

5) $\text{parent}(x)$: 节点 x 的父节点,其中 $x \neq R$ 。

6) att_x : 与叶节点 x 相关联的特定属性。

7) TR: T中的陷阱门节点集合。每个节点 $x \in \text{TR}$ 与一个有状态的时间陷阱 TD_x 相关联,该陷阱具有两种状态,即暴露和未暴露。当指定时间释放隐藏策略树T时,每个时间陷阱 TD_x 都会与某个时间点 t 结合,其原始状态为未暴露。在时间代理于时间点 t 发放时间令牌 Tok_t 后, TD_x 的状态可被转移到暴露状态。(注意,在本文方案中,AA起着时间令牌发放的时间代理作用,而ES则负责各陷阱门节点的状态转移)。

8) T_x : 以节点 x 为根的树T的子树。因此, $T_x(S)=1$ 表示属性集合S满足子树 T_x 。 $T_x(S)$ 通过递归方式计算:如果 x 是叶节点,则 $T_x(S)=1$ 当且仅当 $\text{att}_x \in S$,并且若 TD_x 存在则被暴露;如果 x 是非叶节点,则 $T_x(S)=1$ 当且仅当至少有 thr_x 个子节点 z 满足 $T_z(S)=1$,且相关的陷阱同时被暴露。

3.2 具体构造

(一) 系统初始化阶段

● $\text{Setup}(\kappa) \rightarrow \{\text{PP}, \text{MSK}\}, \{\text{PK}_c, \text{SK}_c\}$: 输入安全参数 κ , AA按如下方式初始化系统:

1) AA生成一个双线性群 (G_1, G_2, e, p) ,其中 p 是一个大素数, G_1 和 G_2 是两个阶为 p 的乘法群,映射 $e: G_1 \times G_1 \rightarrow G_2$ 为双线性映射。

2) AA选择一个生成元 $g \in G_1$,以及 $a, b, \alpha, \beta, \gamma \in {}_R Z_p$ 和 $\theta', \theta_1, \theta_2, \dots, \theta_{l-1}, g_0 \in {}_R G_1$,其中 $l-1$ 是DP公钥的长度。AA选择哈希函数 $H_0: Z_p \rightarrow G_1, H_1: G_2 \rightarrow Z_p^*, H_2: \{0, 1\} \rightarrow G_1$,并选择签名算法Sign和对称加密方案(Enc, Dec)。

3) AA初始化一棵二叉树BT用于管理动态系统用户。每个叶节点 $\eta_i \in \text{BT}$ 对应用户集合 U_p 中的一个唯一DU,并为其分配一个随机值 $t_i \in {}_R Z_p$ 。AA计算 $\{\delta_i = g^{t_i} : i \in [2|U_p| - 2]\}$ 。

4) AA安全地保存系统主密钥 $\text{MSK} = \{g_0^b, a, g^\alpha, \beta, \gamma, \{t_i\}_{i \in [2|U_p| - 2]}\}$,并通过区块链交易将系统公共参数 $\text{PP} = \{G_1, G_2, e, g, g_0, h = g^\gamma, \omega = g^\beta, g^* = g^b, \theta, \theta', \theta_1, \theta_2, \dots, \theta_{l-1}, e(g, g)^\alpha, \text{Sign}, (\text{Enc}, \text{Dec}), \{\delta_i\}_{i \in [2|U_p| - 2]}\}$ 发布至BC。

5) ES和DU完成初始化如下:

a) ES选择一个随机数 $\chi_c \in Z_p$ 作为其密钥 $\text{SK}_c =$

χ_c , 并计算公钥为 $PK_c = g^{\chi_c}$ 。然后, ES 通过一笔交易将公钥 PK_c 发送给 BC。

b) 每个 DU 选择一个随机数 $\chi_u \in Z_p$ 作为其验证密钥 $SK_u = \chi_u$, 并计算验证密钥为 $PK_u = g^{\chi_u}$ 。然后, DU 发布 PK_u 。

(二) 用户注册阶段

● **SKeyGen(PP, MSK, ID_u) → {PK_u, SK_u}**: 用户需按以下方式在系统中注册。

1) AA 随机选择 $x_u, r_u \in Z_p$, 长度为 l 的身份标识 $ID_u = \{d_1, \dots, d_l\}$, 设置用户私钥为 $SK_u = \{SSK_u = \{D_0 = x_u, D_1 = g_0^b(\theta' \prod_{j=1}^{l_1} \theta_j^{d_j})^{x_u}\}\}$, 用户的公钥为 $PK_u = g^{x_u}$ 。

2) AA 发布公钥 PK_u , 并通过安全信道将私钥 SK_u 发送至用户 ID_u 。如果满足方程 $e(D_1, g) = e(g^*, g_0) \cdot e(\theta' \prod_{j=1}^{l_1} \theta_j^{d_j}, PK_u)$, 则用户 ID_u 接受 SK_u 。

3) 系统的每个用户在注册阶段参与 BC, 并被分配一个 EOA, 其验证密钥为 spk_u , 签名密钥为 ssk_u 。

4) AA 在区块链上部署了一个用户认证智能合约。它向用户分发一个验证密钥和一个秘密函数 $f(x)$, 并根据 zkSNARK 算法将相应的验证密钥发送给用户认证智能合约。

● **DKeyGen(PP, MSK, S, ID_u) → DK_u**: 在为 DU 属性集合 S 赋值后, AA 为 DU 选择一个随机数 $\varepsilon, r \in Z_p$, 并为每个属性 $att_i \in S$ 选择一个随机数 $r_i \in_R Z_p$ 。假设 DU 关联到唯一的叶节点 $\eta_j \in BT$, 则 $path(\eta_j) = \{\vartheta_0, \vartheta_1, \dots, \eta_j\}$, 其中 ϑ_0 是 BT 的根节点。算法按如下方式计算解密密钥 DK_u :

$$K_0 = PK_u^a PK_c^e g^{a+r}, K_1 = g^e, K_2 = g^{a/l_j},$$

$$\forall att_i \in S: K_i = g^{r_i} H_0(att_i)^{r_i}, K'_i = g^{r_i}$$

然后 AA 随机选择 $t^* \in Z_p$ 并计算 $\bar{t}_k = t_k \cdot t^*$, 其中, $\vartheta_k \in path(\eta_j)$ 。最后 AA 通过交易将解密密钥 $DK_u = \{S, K_0, K_1, K_2, \{K_i, K'_i\}_{att_i \in S}, \{\bar{t}_k\}_{\vartheta_k \in path(\eta_j)}\}$ 发布给与 DU 账户 EOA_u 相关联的 BC, 且 S 中每个属性均通过 MurmurHash3 算法进行隐藏。

注意, 通过 PK_u, PK_c 随机盲化, K_0 可以防止外包解密过程数据泄露, 因 SK_u, SK_c 为秘密存储信息。

(三) 数据共享阶段

● **Encrypt_{on}(PP) → IT**: 给定系统公参数 PP, DP 随机选择 $s \in_R Z_p, K \in G_2$, 并计算 $\bar{C} = K \cdot e(g, g)^{as}$, $\bar{C}' = g^s, \{V_k = \delta_k^s\}_{k \in [2|U_p| - 2]}$, 以初始化一个池 $IT_{main} = \{(s, K, \bar{C}, \bar{C}', \{V_k\}_{k \in [2|U_p| - 2]})\}$ 。此外, DP 随机选择 $r \in Z_p$ 并计算 $A = g^r, A' = h^r$, 以构造另一个池 $IT_{trap} = \{r, A, A'\}$ 。最后, DP 创建一个中间密文池 $IT = \{IT_{main}, IT_{trap}\}$ 。

● **Encrypt_{on}(PP, T, f, RL, IT, SK_u) → CT**: 使用时间释放隐藏策略树 T、文件 f、撤销列表 RL 以及中间密文池 IT, DP 从 IT_{main} 中随机选择一个元组 $(s, K, \bar{C}, \bar{C}', \{V_k\}_{k \in [2|U_p| - 2]})$ 。对于文件 f, DP 按以下步骤执行在线加密:

1) DP 使用对称密钥 K 对文件 $f = \{m_1, m_2, \dots, m_n\}$ 进行加密, 将 n 个块加密为 $C_s = \{C_s^1, C_s^2, \dots, C_s^n\}$, 其中, $C_s^i = \text{Enc}(K, m_i)$ 。接着, DP 随机选择 $d_u \in Z_p$, 计算 C_s 的签名 $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$, 其中, $\sigma_i = D_1 \cdot (H_2(ID_{f||i})) \cdot \theta^{C_s^i d_u}$ 。然后, DP 为文件 f 生成标签 $\varsigma_f = \varsigma || \text{Sign}_{ssk_u}(\varsigma)$, 其中, $\varsigma = ID_{f||ID_u} || g^{d_u} || n$, ID_f 是文件标识。对于撤销列表 RL, DP 计算 $\{\delta_k^s\}_{\vartheta_k \in \text{cover}(RL)^c}$ 。

2) DP 将秘密值 s 根据各节点 $x \in T$ 分割如下:

$$\begin{cases} \bar{s}_x \cdot \tilde{s}_x = s_x, & x \text{ 绑定一个时间陷门} \\ \bar{s}_x = s_x, \tilde{s}_x = 1, & \text{其他} \end{cases}$$

关于与根节点相关的秘密值 s , 作为 T 的根秘密值, DP 从上到下为 T 中每个节点计算其共享密钥, 具体如下:

a) 如果节点 $x \in X$ 是叶节点, 其中 X 是 T 中的叶节点集合, 则该节点与属性 att_x 相关联, 并被赋予一个秘密分片值 $\bar{s}_x$ 。然后, DP 计算 $C_x = g^{\bar{s}_x}, C'_x = H_0(att_x)^{\bar{s}_x}$ 。

b) 如果节点 x 是非叶节点, 则 DP 构建一个 d_x 次多项式 q_x , 其中 $d_x = \text{thr}_x - 1$, 且 thr_x 是节点 x 的阈值, 并设置 $q_x(0) = \bar{s}_{xx}$ 。接着, 对于节点 x 的每个子节点 y , 其索引值为 ind_y , DP 计算 $s_y = q_x(\text{ind}_y)$ 。

3) 对于每个与陷门节点 $x \in TR$ 相关的时间陷门 TD_x , 结合时间点 $t \in F_T$ 和一个秘密参数 $\tilde{s}_x$, DP 从 IT_{trap} 中随机选择一个元组 $(r_x, A_x = g^{r_x}, A'_x = h^{r_x})$, 并生成节点 x 相关联的 TD_x 为:

$$TD_x = (A_x, B_x = \tilde{s}_x + H_1(e(H_0(t), A'_x)))$$

4) DP 输出文件 ID_f 的密文为 $CT = \{T, \zeta_p RL, C_s, \sigma,$

$$\bar{C}, \bar{C}', \{V_k\}_{k \in \text{cover}(RL)}, \{C_x, C'_x\}_{x \in X}, \{A_y, B_y\}_{y \in TR}\}.$$

5) DP 计算 $V_{ID_f} = \theta^n \prod_{k=1}^{l_1} \theta_k^{d_k}$, 并将其与密文 CT 通过交易上传至 BC。BC 将 V_{ID_f} 和密文 CT 中除 C_s 外的其他部分进行存储, 用于数据访问控制和完整性审计, 需由不可篡改的 BC 进行保护。

6) BC 将密文组件 $\{C_s, \sigma, \zeta_f\}$ 链下上传至 ES 以进行存储和共享。ES 执行如下公式进行验证:

$$e\left(\prod_{i=1}^n \sigma_i, g\right) \stackrel{?}{=} e(g_0, g^*)^n \cdot e(V_{ID_f}, PK_u)^n \cdot e\left(\prod_{i=1}^n (H_2(ID_f || i) \cdot \theta^{C_s^i}), g^{d_u}\right)$$

若该等式成立, ES 存储密文组件并返回地址给 BC 进行记录。

● **TimeRel(PP, t) $\rightarrow$ TD $'_x$:** 系统以均匀的时间运行, 时间通过时间点数量来计数。基于时间机制, AA 以如下方式发布用于暴露时间陷门的时间令牌。

1) **TokenGen:** 每当时间点 $t \in F_T$ 到来时, AA 发布一个时间令牌 $Tok_t = H_0(t)^r$, 该令牌可被系统中的每个实体接收。

2) **Trap:** 当 ES 在时间点 t 收到 AA 发布的 Tok_t 后, 查找存储的所有相关陷门。对于每个陷门 $TD_x = (A_x, B_x)$, ES 计算 $TD'_x = B_x - H_1(e(Tok_t, A_x)) = \tilde{s}_{xx}$, 并用 TD'_x 替换 TD_x 。因此, 如果上述公式正确执行, 则相关陷门将被暴露。

(四) 数据访问阶段

● **Decrypt $_0$ (PP, CT, DK $_u$) $\rightarrow$ CT' 或 null:** 当访问 ES 中的共享数据时, DU 首先执行以下步骤以生成用户身份认证证明。

1) DU 计算 Pederson 承诺 $c_a = g^{h(ID_u)} h^{r_a}$, 其中, $r_a \in {}_R Z_p$, h 表示 MurmurHash3 算法。

2) DU 利用 zkSNARK 计算 $\pi = \text{ProofGen}(SK_u, h(ID_u), r_a, \text{timestamp})$, 其中, DU 根据私有输入 $(h(ID_u), r_a)$ 和公开输入 timestamp , 计算出秘密函数 $f(x)$ 的一个证据。证明由证据与验证钥组合而成。

3) DU 生成一个用户身份认证证明 $\Gamma = \{\pi, c_a, S'\}$, 其中 S' 是 DU 的隐藏属性集。

4) DU 创建一个包含文件标签、其账户 EOA $_u$ 以及用户身份认证证明 Γ 的数据访问请求, 并将其

封装在交易中发送至 BC, 以调用用于验证用户身份和属性的智能合约, 如算法 1 所示。

算法 1 身份与属性验证智能合约

给定系统公共参数 PP, DU 的身份 ID 和撤销列表 RL。

1) 构造 Pederson 承诺 c_a , 并执行 zkSNARK 的验证算法 $\text{VerifyProof}(PK_u, c'_a, \pi, \text{timestamp})$ 。如果 VerifyProof 返回 true, 则多方认证成功, 流程继续。

2) 根据 DU 的 EOA $_u$ 获取解密密钥 DK_u , 通过验证 S' 中的所有属性是否均为 DK_u 隐藏属性集中的属性, 来验证存储在 BC 上的隐藏属性集。

3) 获取访问控制所需的元数据 $\widetilde{CT}$, 将其与解密密钥 DK_u 和文件标签一并发送至 ES。

在接收到元数据 $\widetilde{CT}$ 后, ES 根据文件标签查找相应的密文 CT, 并执行以下外包解密步骤。

a) 对于叶节点 $x \in X$, 若其与属性 att_x 关联, 没有未暴露的后门且 $\text{att}_x \in S$, 则计算 $F_x = \text{DecryptNode}(DK_u, CT, x)$ 如下:

$$F_x = \left(\frac{e(K_x, C_x)}{e(K'_x, C'_x)} \right)^{\tilde{s}_x} = e(g, g)^{r_{s_x} \tilde{s}_x} = e(g, g)^{r_{s_x}}$$

否则, 输出 $F_x = \text{null}$ 。

b) 对于非叶节点 $x \in T$, 计算 $F_x = \text{DecryptNode}(DK_u, CT, x)$ 如下:

$$F_x = \left(\prod_{z \in Q_x} F_z^{\Delta_{\text{ind}_z, \ell'_x}(0)} \right)^{\tilde{s}_x} = \left(\prod_{z \in Q_x} (e(g, g)^{r_{s_z}})^{\Delta_{\text{ind}_z, \ell'_x}(0)} \right)^{\tilde{s}_x} = e(g, g)^{r_{s_x} \tilde{s}_x} = e(g, g)^{r_{s_x}}$$

其中, 集合 Q_x 由任意 thr_x 个子节点组成, 每个节点 z 都满足 $F_z \neq \text{null}$ 。如果 Q_x 存在且其中没有任何未暴露的陷门, 则可得 $Q'_x = \{\text{ind}_z : z \in Q_x\}$ 。

如果属性集 S 满足以根节点 $R \in T$ 为根的子树, 即属性集 S 满足 T 的整棵树, 则 DU 可以递归地计算 $\text{DecryptNode}(DK_u, CT, R)$ 得到 $e(g, g)^{r_s}$ 。

c) 如果 DU 与叶节点 $\eta_i \notin RL$ 关联, 则可得 $\text{path}(\eta_i) = \{\vartheta_0, \dots, \vartheta_k, \dots, \eta_i\}$, 并得到节点 $\vartheta_k = \text{cover}(RL) \cup \text{path}(\eta_i)$, 计算 $v = \bar{t}_j / \bar{t}_k = t_j / t_k$, 其中, $\bar{t}_j$ 和 $\bar{t}_k$ 分别是 DK_u 对应于 η_i 和 ϑ_k 的关键组成部分。然后, DU 计算以下公式:

$$V = e(K_2, V_k)^v = e(g, g)^{as}$$

因此, 对于根节点 R , DU 可得以下结果:

$$C_p = \frac{e(\bar{C}', K_0)}{\text{DecryptNode}(\text{DK}_u, \text{CT}, R) \cdot V \cdot e(\bar{C}', (K_1)^{\text{SK}_c})} \\ = e(\text{PK}_u, g)^{\alpha_s}$$

最后, ES 输出部分解密密文 $\text{CT}' = \{\zeta_p, C_p, C_s, \sigma\}$, 并通过交易发送至 BC 进行数据完整性审计。

● **Audit(PP, CT') → True/False:** 在从 ES 接收包含部分解密密文 CT' 的数据完整性审计交易后, BC 会触发一个预定义的数据完整性审计智能合约, 该合约将执行以下交互步骤以验证 CT' 的完整性。

1) 通过签名验证算法检查文件标签 ζ_f 是否有效。如果无效, 则流程终止; 否则, 解析 ζ_f 并获取标识符 ID_f , ID_u , g^{d_u} 以及分块号 n 。

2) 对于每个 $C_s^j \in C_s$, 其中 $j \in I, I \in [n]$, 随机选择 $\mu_j \in Z_p$ 。然后将证明请求 $\text{req}_{\text{proof}} = \{j, \mu_j\}_{j \in I}$ 发送至 ES。

3) ES 计算数据完整性审计的证据为 $\text{rsp}_{\text{proof}} = \{\psi, \sigma\}$, 其中,

$$\psi = \sum_{j \in I} C_s^j \mu_j, \sigma = \prod_{j \in I} \sigma_j^{\mu_j}$$

ES 将 $\text{rsp}_{\text{proof}}$ 返回给智能合约。

4) 智能合约通过执行以下公式来验证 $\text{rsp}_{\text{proof}}$ 的正确性:

$$e(\sigma, g) = e(g^*, g_0)^{\sum_{j \in I} \mu_j} \cdot e(V_{\text{ID}_f}, \text{PK}_u)^{\sum_{j \in I} \mu_j} \cdot$$

$$e(\prod_{j \in I} H_2(\text{ID}_f || j)^{\mu_j} \cdot \theta^{\psi}, g^{d_u})$$

如果上述等式成立, 则 BC 输出 True, 并将 CT' 发送给 DU 进行本地解密; 否则, 输出 False。

● **Decrypt_i(PP, MSK, PK_u, CT') → f 或 null:** 在接收到部分解密后的密文 CT' 后, DU 将其中 $\{\bar{C}', C_p\}$ 以及公钥 PK_u 发送给 AA, 由 AA 使用系统主密钥 MSK 进行如下解密验证:

$$e(\text{PK}_u^a, \bar{C}') = C_p$$

如果上式成立, 则验证成功, 算法继续执行; 否则输出 null。

解密验证成功后, DU 计算 $C_i = (C_p)^{1/\text{SK}_u} = e(g, g)^{\alpha_s}$, 并进一步计算 $K = \bar{C}' / C_i$ 得到密文解密密钥。然后, DU 计算 $m_i = \text{Dec}(C_s^i, K)$ 恢复明文, 并输出文件 $f = \{m_1, m_2, \dots, m_n\}$ 。

(五) 用户撤销阶段

● **Revoke(RL, ID_u) → {CT'}:** 在收到通过交易由 DP 发送的包含被撤销用户身份 ID_u 的用户撤销请

求后, 系统会调用区块链中预定义的用户撤销智能合约, 并执行以下步骤。

1) 获取当前的撤销列表 RL, 并获得与用户 ID_u 相关联的叶节点 $\eta_i \in \text{BT}$ 。

2) 通过 $\text{RL}' = \text{RL} \cup \eta_i$ 更新撤销列表, 并将其发送至 AA 以生成更新密钥。

3) 使用最新的撤销列表 RL' , AA 生成并返回更新密钥 $\text{UK} = \{t'_j = \rho \cdot t_j\}_{j=0}^{2|U_p|-2}$, 其中 $\rho \in_R Z_p$ 。

4) 收到更新密钥 UK 后, 将其和 RL' 以及相关的文件标签 ζ_f 发送到 ES 进行密文更新。

5) ES 获取最小覆盖集 $\text{cover}(\text{RL}')$ 。对于每个 $\mathcal{G}_{k'} \in \text{cover}(\text{RL}')$, 它执行以下步骤:

a) 如果节点 $\mathcal{G}_{k'} \in \text{cover}(\text{RL}')$ 满足 $\mathcal{G}_k = \mathcal{G}_{k'}$, 则 ES 设置 $V_{k'} = V_k$ 。

b) 如果节点 $\mathcal{G}_{k'} \in \text{cover}(\text{RL}')$ 满足 $\mathcal{G}_k$ 是 BT 中 $\mathcal{G}_{k'}$ 的祖先, 并且假设 $\text{path}(\mathcal{G}_{k'}) = \text{path}(\mathcal{G}_k) \cup \{\mathcal{G}_{\text{depth}(\mathcal{G}_k)+1}, \dots, \mathcal{G}_{\text{depth}(\mathcal{G}_{k'})}\}$, 其中, $\mathcal{G}_{\text{depth}(\mathcal{G}_k)} = \mathcal{G}_k$, $\mathcal{G}_{\text{depth}(\mathcal{G}_{k'})} = \mathcal{G}_{k'}$, 则 ES 设置 $A_k = V_k$ 并依次计算以下公式:

$$A_{j+1} = (A_j)^{\frac{t'_{j+1}}{t'_j}} = \delta_{j+1}^s$$

其中, $j \in [\text{depth}(\mathcal{G}_k), \dots, \text{depth}(\mathcal{G}_{k'}) - 1]$ 。

6) 最后, ES 设置 $V_{k'} = A_{k'}$ 并输出与 RL' 相关的更新密文 $\bar{\text{CT}} = \{\text{RL}', C_s, \bar{C}', \{V_{k'}\}_{\mathcal{G}_{k'} \in \text{cover}(\text{RL}')} \}$ 。

4 安全分析

定理 1 若 DBDH 假设成立, 则任何 PPT 敌对者都不能以非微小优势攻破本方案的 CPA 安全性。

证明 详见附录 1。

5 性能分析

5.1 理论分析

通过对比方案 FE-ABSE-RV^[23] 和方案 TR-AP-CPABE^[22] 等优秀的相关方案, 对所提方案 BRTDS 的计算复杂性和存储复杂性进行理论分析总结, 结果分别列于表 1 和表 2 中。在理论分析中, 本文考虑了 G_1, G_2 中的指数运算 E_1, E_2 、乘法运算 M_1, M_2 、双线性运算 P 以及 G_1, G_2, Z_p 中元素的长度分别记为 $|G_1|, |G_2|$ 和 $|Z_p|$ 。

表 1 展示了从密钥生成、加密、解密和撤销算法的角度 BRTDS 与方案 TR-AP-CPABE 和 FE-

表1 BRTDS与已有方案计算复杂度对比

方案	密钥生成	数据加密	数据解密	用户撤销
TR-AP-CPABE	$(2 S + 5)E_1$ $+ (S + 1)M_1$	$(4 X + C + 2)E_1$ $+ E_2 + M_2$	$(2 X + 1)E_1 + (2 X + 1)M_1$ $+ (X + 1)E_2 + (3 X + 3)M_2$ $+ (3 X + 2)P$	$ \bar{D} E_1$
FE-ABSE-RV	$(2 S + 5)E_1$ $+ M_1$	$ X E_1 + X M_1$ $+ E_2 + M_2 + X P$	$(S + 3)E_1 + M_1$ $+ 2E_2 + 2M_2 + P$	$3 S E_1$
BRTDS	$(2 S + 8)E_1$ $+ (S + 2)M_1$	$(2 X + N)E_1$ $+ D P$	$E_1 + E_2 + M_2 + P$	$ \bar{D} E_1$

ABSE-RV的计算复杂度比较。在密钥生成算法中,由于引入了用户撤销机制以及带数据完整性审计的可验证外包解密,本文方案BRTDS的用户密钥生成计算开销比其他两个方案约为 $3E_1$ 。然而,在加密算法中,由于采用了在线/离线加密方式,方案BRTDS的计算开销约为 $(2|X|+N)E_1$,远低于TR-AP-CPABE方案。而FE-ABSE-RV方案的计算复杂度与BRTDS类似,因为前者虽然有较少的 E_1 运算,但增加了耗时的双线性运算。在解密算法中,本文方案BRTDS的计算复杂度最低,原因在于将大部分计算任务卸载到ES,仅保留对本地设备的计算量。尽管FE-ABSE-RV也通过外包解密来降低时间复杂度,但相比本文方案仍较高,因其每次外包解密时都需要临时生成转换密钥。在撤销算法中,本文方案BRTDS中的计算复杂度与TR-AP-CPABE的相似。然而,FE-ABSE-RV方案中的撤销操作在更新密钥生成时需要更多 G_1 群上指数运算。

另一方面,表2比较了本文方案BRTDS与方案TR-AP-CPABE和FE-ABSE-RV的在公共参数大小、密钥大小、密文大小方面的存储复杂度和通信开销。在公共参数大小的存储复杂度对比中,由于引入了基于身份的数据审计机制,本文方案BRTDS比方案TR-AP-CPABE多消耗 $l_1|G_1|$ 的存储空间,而方案FE-ABSE-RV的存储开销高于其他两个方案,因为它不支持大规模属性集合。在用户密

钥大小的存储复杂度比较方面,本文方案BRTDS略低于方案FE-ABSE-RV,但高于方案TR-AP-CPABE,因为方案BRTDS中每个属性需要更多的组件。然而,在实际应用中,本文方案中的用户仅需保留本地解密所需的私钥,而非保存全部解密密钥,因此对用户设备的实际存储开销最小。在密文大小的存储复杂度对比中,本文方案BRTDS的开销要略大于方案FE-ABSE-RV,这是由于BRTDS在访问策略中集成了用户撤销和时间陷门。在通信开销对比方面,本文方案BRTDS与方案FE-ABSE-RV在解密过程中ES与DU之间的通信开销小于方案TR-AP-CPABE,其原因在于方案TR-AP-CPABE需要将密文的所有组件传输给DU,而本文方案和方案FE-ABSE-RV只需传输外包解密所需的密钥以及部分解密密文即可。

5.2 实验评估

为了评估本文方案BRTDS的实际性能,开展了一系列仿真实验,以验证本文方案的实际效率。通过使用Java编程语言结合JPBC库^[30]来实现这些方案的原型,其中,JDBC库支持双线性群的配对、指数、乘法等运算。方案原型在一台配备了Intel Core i5处理器2.3 GHz和8.00 GB内存的Windows Server 2013系统上运行。为评估BC中运算的性能,采用Ganache工具模拟以太坊平台,并设置5个工作进程进行交易测试和智能合约部署。在这

表2 BRTDS与已有方案存储复杂度对比

方案	公共参数大小	密钥大小	密文大小	通信开销
TR-AP-CPABE	$(2 U_p + 2) G_1 + G_2 $	$(S + 4) G_1 $ $+ PL Z_p $	$(3 X + C + 2) G_1 $ $+ G_2 $	$(3 X + C + 2) G_1 + G_2 $
FE-ABSE-RV	$(U_p + 3) G_1 + G_2 $	$(2 S + 4) G_1 $ $+ Z_p $	$(2 X + 1) G_1 + G_2 $	$(2 S + 2) G_1 + G_2 $
BRTDS	$(2 U_p + l_1 + 5) G_1 $ $+ G_2 $	$(2 S + 3) G_1 $ $+ PL Z_p $	$(2 X + D + C + 1) G_1 $ $+ G_2 $	$(2 S + N + 3) G_1 $ $+ G_2 + PL Z_p $

些实验中,选择在160位椭圆曲线群 $E(F_q): y^2=x^3+x$ 和512位有限域 F_q 上构造阶为素数 p 的群 G_a 和 G_b ,因此元素长度设定为 $|Z_p|=160$, $|G_1|=|G_2|=1024$ 。此外,设置用户身份长度参数 $l_1=6$,密钥长度 $|C|=5$,文件块数量 $N=15$,用于评估数据共享的性能。

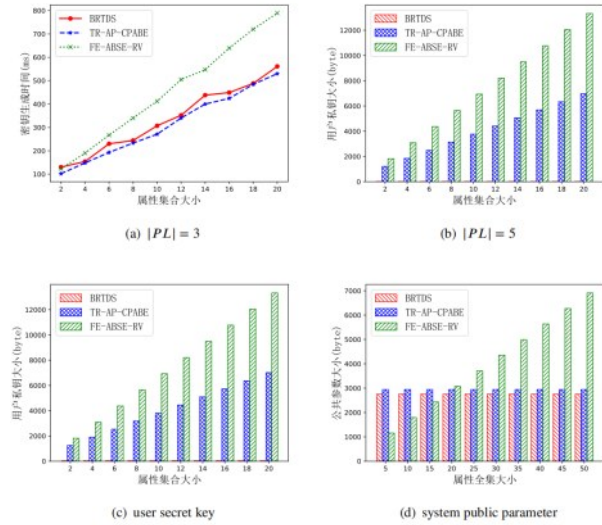


图2 密钥生成时间和存储开销对比

在图2中展示了用户密钥生成的时间开销,以及系统公共参数和用户实际秘密保存的密钥的存储开销。从图2可以看出,三种方案的密钥生成时间成本均随用户属性集大小的增加而上升,其中,方案FE-ABSE-RV的增长率更高,原因在于其需要对每个属性计算哈希结果后再进行指数运算。为了具体评估用户密钥生成的时间成本,该实验将BT的路径长度 $|PL|$ 分别设为3和5,结果分别如图2(a)和图2(b)所示。显然,方案TR-AP-CPABE和方案FE-ABSE-RV中的用户密钥大小均随着用户属性集规模的增大而增长。需要注意的是,本文方案BRTDS中用户密钥的存储成本不受用户属性集规模增加的影响。尽管在理论分析中用户解密密钥的大小会受到用户属性集的影响,但其实际大小是恒定的,因为用户仅需保留其私钥 SK_u ,而解密密钥 DK_u 则在区块链BC上公开存储用于外包解密,从而大大降低了用户设备的密钥存储成本。图2(c)展示了三种方案中系统公共参数的存储开销。显然,在本文方案BRTDS和方案TR-AP-CPABE中,系统公共参数的大小保持不变,而方案FE-ABSE-RV的存储开销则随着属性空间规模的扩大而增加。因此,方案FE-ABSE-RV无法支持大规模属性空间,

也不适用于基于边缘计算的物联网数据共享场景。

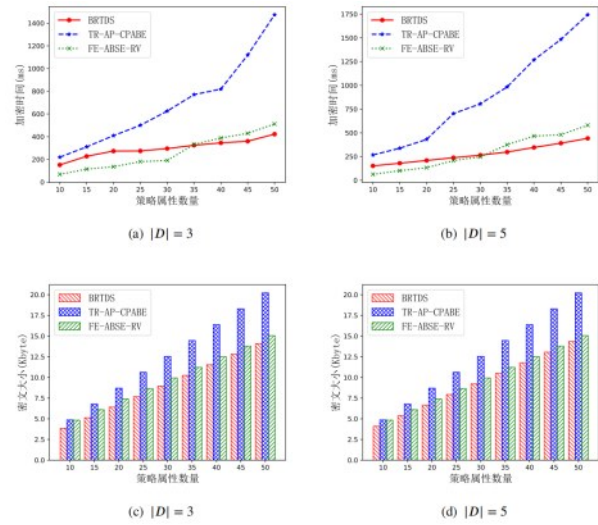


图3 加密时间开销和密文存储开销对比

图3描述了上述三种方案中,数据加密的时间开销以及密文存储开销对比。在该实验中,设置访问策略中属性集的大小 $|X|$ 在 $[10, 50]$ 范围内变化。为了展示时间陷门数量对加密时间和密文存储的影响,该实验将 $|D|=\{3,5\}$ 设定为常数。从图3(a)和3(b)可以明显看出,三个方案的加密时间成本与访问策略中满足的属性数量呈线性关系。由于在加密算法中引入了在线/离线技术,本文方案BRTDS和方案FE-ABSE-RV的加密时间成本均低于方案TR-AP-CPABE。需要注意的是,本文方案BRTDS与方案FE-ABSE-RV的变化曲线存在一个交点,这是由于方案FE-ABSE-RV中的配对操作耗时更多所致。从图3(c)和3(d)中可以看到,三个方案的存储成本随着满足的策略属性数量增加而上升。值得注意的是,方案TR-AP-CPABE中CP-ABE密文的存储开销的增长速度超过了另外两个方案。其原因在于,方案TR-AP-CPABE在每个访问策略属性上增加了一个额外的群 G_1 元素。由于引入了时间陷门和二叉树,本文方案BRTDS中的密文存储成本高于方案FE-ABSE-RV。同时,当访问策略中包含更多的时间陷门时,本文方案BRTDS的密文存储开销也随之增加。

图4展示了三个方案在密文解密与用户撤销算法的时间开销对比以及在不同路径长度 $|PL|$ 下解密通信开销的对比。从图4(a)可看出,方案TR-AP-CPABE的解密算法时间开销与策略属性数量成正

比, 而其他两个方案的解密算法时间开销保持不变。原因在于, 本文方案BRTDS和方案FE-ABSE-RV均采用外包解密方式以降低解密算法的运行时间。由于本文方案BRTDS省去了每次外包解密过程中复杂的变换密钥生成流程, 因此, 其解密时间开销最低。图4(b)给出了三个方案中用户撤销算法的时间开销。设定平均覆盖节点数为 $|D|=5$ 。显然, 本文方案BRTDS以及方案TR-AP-CPABE的时间开销是恒定的, 而方案FE-ABSE-RV的时间开销则与用户属性集大小成正比, 这是因为方案FE-ABSE-RV在每个被撤销用户的属性上需要中额外添加一个 G_1 群元素, 从而增加了撤销操作的负担。在通信开销性能评估中, 设置路径长度 $|PL|$ 为3和5, 其结果对比分别如图4(c)和图4(d)所示。可以观察到, 随着策略属性数量的增加, 本文方案BRTDS和方案FE-ABSE-RV的通信开销保持不变, 因为它们仅涉及固定大小的解密密钥和部分解密密文。而方案TR-AP-CPABE的通信开销会随策略属性数量的增加而上升, 因其需要返回大小变化的CP-ABE密文。

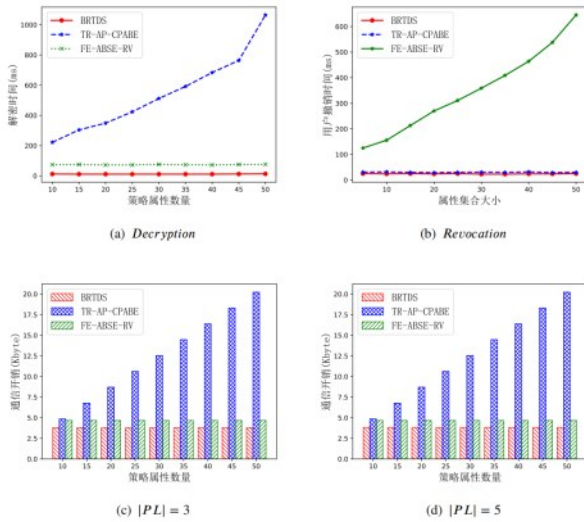


图4 解密和撤销时间开销对比以及通信开销对比

图5描述了在BC上执行数据完整性审计算法的性能评估, 通过将本文方案BRTDS与一种先进的数据完整性审计方案^[26] (简称IBDIA) 进行对比。可以注意到, 两种方案的数据完整性审计时间成本均与文件块数量成正比。由于采用了在线/离线技术相结合的方法, 本文方案BRTDS在数据审计方面耗时显著减少, 且随着文件块数量的增加,

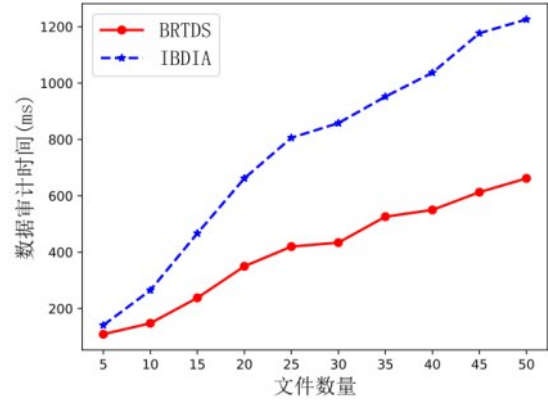


图5 数据审计的时间开销对比

这一优势更加明显。

在评估以太坊平台上的BC操作时, 该实验采用Gas作为衡量工作负载的通用指标。在对BC操作的实验中汇总了所有结果, 显示写入操作的Gas为500856, 查询操作的Gas为40897。此外, 还评估了与写入操作和查询操作相对应的智能合约调用所消耗的时间成本。统计结果显示, 相应智能合约调用的延迟时间略低, 具体而言, 每次写入操作调用耗时约为0.06秒, 每次查询操作调用耗时约为0.04秒。

综上所述, 针对本文方案BRTDS的性能评估与表1和表2中列出的理论分析一致。可以看出, BRTDS是唯一不仅支持在时间敏感数据上实现隐私保护的细粒度访问控制方案, 并具备有效的可信用户撤销和去中心化数据完整性审计功能, 而且还能在分布式边缘计算环境中建立物联网数据安全共享的信任机制。此外, 本文方案BRTDS减少了用户设备本地存储的用户密钥大小以及解密过程中的通信开销。因此, 本文方案BRTDS在基于边缘计算的物联网数据共享应用中具有高效性和实用性。

本文同时对引入zkSNARK实现用户身份和属性验证的过程进行性能评估。在部署的身份与属性验证智能合约中, 在同样的实验设置下使用轻量化电路进行实现。此时, 对用户身份验证需要大约5ms的时间, 这意味着该智能合约为了实现用户身份的验证引入了5ms左右的时延。在DU侧, 生成针对用户身份的zkSNARK证据大约需要消耗60ms-70ms的时间完整证据的生成, 该过程对用户侧的数据访问过程带来了一些计算开销。整体而言, 用户身份验证过程对于DU和智能合约的开销是可以接受的。该过程时延的进一步提升也是后续

工作的重点之一。

6 结束语

为应对边缘物联网中数据共享的安全挑战,本文提出了一种区块链辅助的可信可撤销数据共享方案 BRTDS,用于支持在边缘物联网环境中时间敏感数据的安全共享。BRTDS 设计了一种 CP-ABE 中的时间释放隐藏策略树以实现物联网中时间敏感数据的细粒度且隐私保护的访问控制。结合区块链与零知识证明技术, BRTDS 能够实现面向动态用户的分布式、可信的数据访问控制,并有效抵御身份和属性假冒攻击。此外,通过将基于身份据审计机制与智能合约结合,设计了一种分布式数据完整性审计机制,以防止外包数据被篡改。同时,得益于在线/离线技术以及可验证外包解密机制,本文方案 BRTDS 还实现了较高的计算和存储效率。形式化安全分析及大量实验仿真和性能评估表明,本文方案具备较好的安全性、高效性和实用性。

附录 1 安全证明假设存在一个 PPT 敌手 A 能够以优势 ϵ 攻破 BRTDS 的安全游戏,则可以构造一个模拟器 B,以相同的优势 ϵ 选择性攻破方案^[21]。设 G_1, G_2 均为素数阶 $\mathbb{G}$ 的乘法循环群,对于任意的 $g \in G_1, e: G_1 \times G_1 \rightarrow G_2$ 为双线性映射。

初始化: B 接收 A 发来的挑战访问策略树 T' 、挑战撤销列表 RL'_i 以及挑战时间点 $\square'$, 并将 T' 发送给 C。

设置: C 发布系统公共参数为 $\square\square' = \{G_1, G_2, e, g, g_0, \omega, e(g, g)^{\alpha}\}$ 。然后, B 选取 $t_i \in_R Z_p$ 并计算 $\{\delta_i = g^{t_i}\}_{i \in [2|U_p| - 2]}$ 。对于 ES 设置分为以下两种情况:

情况 1: B 负责 ES 的设置, 它选取随机值 $\chi_c \in Z_p$ 作为 $\square\square_\square$, 并计算 $\square\square_\square = g^{\chi_c}$ 。然后, 它选取 $h \in_R G_1$ 和 $t_i \in_R Z_p$ 计算 $\{\delta_i = g^{t_i}\}_{i \in [2|U_p| - 2]}$ 。最后, B 将 $PP = \{PP', h, \{\delta_i\}_{i \in [2|U_p| - 2]}\}$ 发送给 A。

情况 2: 该情况与情况 1 的区别在于, A 可以腐蚀 ES 以获取外包隐私, 并负责 ES 的设置。它运行 $\square\square\square\square\square\square\square\square$ 算法, 并将 $\square\square_\square$ 发送给 B。

查询 1。 A 向 B 提交一系列形如 $(\square\square_\square, S_i)$ 的私钥和转换密钥请求, B 将其转发给 C。在收到 C 返回的相应秘密属性密钥 $DK_u = \{\bar{K}_0, \{\bar{K}_{i,0}, \bar{K}_{i,1}\}_{att_i \in S}\}$ 后, 若所有属性集 $\square\square_\square$ 均不满足 T' 且 $ID_i \notin RL'_i$, 则 B 选取 $\bar{e}, \bar{a}, \bar{x}, \hat{\chi}_u \in_R Z_p$, 设置用户私钥 $SK_u = \hat{\chi}_u$, 并计算公钥 $PK_u = g^{\hat{\chi}_u}$ 。然后, B 可获得秘密属性密钥为 $DK_u = \{\square_\square = (\bar{K}_0)^{\hat{\chi}_u} PK_c^{\bar{e}} g^{\bar{a}}, \square_1 = g^{\hat{\chi}_u}, \square_2 = g^{\hat{a}\hat{\chi}_u}, \{\square_\square = (\hat{K}_{j,0})^{\hat{\chi}_u}, K_j = (\hat{K}_{j,1})^{\hat{\chi}_u}\}_{att_j \in S}, \{t_j\}_{j \in path(ID_i)}\}$ 。接下来, B 输出 DK_u 并将 (SK_u, PK_u, DK_u) 发送给 A。

挑战。 敌手 A 向模拟器 B 提交两个等长消息 $\square_\square, \square_\square$, B 将其发送给挑战者 C 并获得响应密文 $\square\square = \{T', \square_\square, C_i'\}$, 该密文是在挑战访问策略 T' 下生成的。然后, B 生成 $\{\square_\square, \square_\square =$

$(C_i')^t_k\}_{k \in cover(RL_i)}$, 并随机选取 $b \in_R \{0, 1\}$ 。对于时间点 $\square < \square'$ 的每个陷门 $\square\square_\square$, B 计算 $\square_\square$ 以将其公开; 而对于时间点 $\square > \square'$ 相关的陷门, B 通过随机选取 $r_x \in Z_p$ 计算 $\square\square_\square = \{\square_\square, \square_\square\}$ 。最后, B 将新密文 $CT' = \{\square_\square, \{\square_\square, \square_\square\}_{k \in cover(RL_i)}, \square\square_\square\}$ 返回给敌手 A。

查询 2 敌手 A 提交查询 1 类似的查询。

猜测 敌手 A 输出对 $\square$ 的猜测 $\square_\square$ 。若 $\square_\square = 1$, 则 A 猜测 CT' 中的 $\square_\square$ 是随机密钥, B 输出其对 $\square\square$ 的猜测为 $\square_\square = 1 - \square_\square$ 。若 $\square_\square = 0$, 则 A 猜测 $\square_\square$ 是确定的密文, B 输出 $\square_\square = \square_\square$ 。

因此, 若 A 能够以不可忽略的优势选择性攻破本文方案 BRTDS, 则 B 能够以相同的优势选择性攻破方案^[21]。基于 DBDH 假设, 本文方案 BRTDS 在选择性访问策略和选择明文攻击下实现了相同的安全性。

证毕。

参考文献:

- [1] SALMAN M, SAEED N, KHURSHID K. Transformers for Internet of things security and communications: A comprehensive survey[J]. IEEE Internet of Things Journal, 2026, 1 - 1.
- [2] NASIRAEI H, LIU X. A robust multifactor authentication and key agreement protocol in industrial Internet of things[J]. IEEE Internet of Things Journal, 2026, 1 - 1.
- [3] ZUO Y, YUE M, YANG H, et al. Integrating communication, sensing and computing in satellite Internet of things: Challenges and opportunities[J]. IEEE Wireless Communications, 2024, 31(3): 332-338.
- [4] RAHMAN M A, JAMIL S U, HOSSAIN M S, et al. An LLM-enabled multimodal agentic AI framework for the medical Internet of things (MIoT)[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2026, 22(8): 1 - 35.
- [5] SIDDIQI S J, ALOBAIDI A H, JAN M A, et al. Securing vehicle-to-digital twin communications in the Internet of vehicles[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2026, 22(1): 1-19.
- [6] CHAUDHRY S A, IRSHAD A, AKHTAR Z, et al. Sebac-iod: A secure and efficient batch access control technique for Internet of drones[J]. IEEE Transactions on Vehicular Technology, 2026, 1 - 1.
- [7] 刘妮妮,殷涌泉,张建辉,等. 基于区块链的物联网无证书多重身份认证方案[J]. 计算机学报, 2026, 49(7): 1670-1686. DOI:10.11897/SP.J.1016.2026.01670.
- LIU Y N, YIN Y Q, ZHANG J H, et al. A blockchain-based certificate-less multi-identity authentication scheme for IoT[J]. CHINESE JOURNAL OF COMPUTERS, 2026, 49(7): 1670-1686.
- [8] BOUAFIA K, SLIMANI H, NACER H, et al. Big data analytics and Internet of things combined with artificial intelligence techniques for effective emergency management: A Comprehensive survey[J]. ACM Computing Surveys, 2026, 58(12): 1-43.
- [9] TAYLOR P. Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2034[J]. statista. Accessed: Dec, 2025, 1 - 1.
- [10] YANG Y, ZHAO J, CHEN F, et al. Accurate and efficient privacy-preserving TPE-image retrieval in cloud-assisted Internet of things[J]. IEEE Internet of Things Journal, 2026, 1 - 1.
- [11] SHARMA M, TOMAR A, HAZRA A. Edge computing for industry

- 5.0: Fundamental, applications, and research challenges[J]. IEEE Internet of Things Journal, 2024, 11(11): 19070-19093.
- [12] WANG F, CUI J, ZHANG Q, et al. Blockchain-based secure cross-domain data sharing for edge-assisted industrial Internet of things[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 3892-3905.
- [13] REGAN R, JOSPHINELEELA R, KHAMRUDDIN M, et al. Balancing data privacy and sharing in IIoT: Introducing the GFL-LFF aggregation algorithm[J]. Computer Networks, 2024, 247: 110401.
- [14] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proceedings of the 13th ACM conference on Computer and communications security. Virginia: ACM, 2006: 89-98.
- [15] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]//2007 IEEE symposium on security and privacy (SP'07). Berkeley: IEEE Press, 2007: 321-334.
- [16] SAHAI A, WATERS B. Fuzzy identity-based encryption[C]//Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2005: 457-473.
- [17] HUR J. Attribute-based secure data sharing with hidden policies in smart grid[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(11): 2171-2180.
- [18] ZHANG L, MA Y, WU Z, et al. LMDA: Lightweight multi-authority CP-ABE scheme based on dynamic attributes for system wide information management[J]. IEEE Internet of Things Journal, 2026.
- [19] ANDROULAKI E, SORIENTE C, MALISA L, et al. Enforcing location and time-based access control on cloud-stored data[C]//2014 IEEE 34th International Conference on Distributed Computing Systems. Madrid: IEEE Press, 2014: 637-648.
- [20] 余增文, 张嘉伟, 姜奇, 等. 数据空间中可比较属性的数据访问控制方案[J]. 计算机研究与发展, 2026, 63(3): 735-751. DOI: 10.7544/issn1000-1239.202550411.
- YU Z W, ZHANG J W, JIANG Q, et al. Data access control scheme with comparable attribute for dataspace[J]. Journal of Computer Research and Development, 2026, 63(3): 735-751.
- [21] HONG J, XUE K, XUE Y, et al. TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud[J]. IEEE Transactions on Services Computing, 2017, 13(1): 158-171.
- [22] HAN D, PAN N, LI K C. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 19(1): 316-327.
- [23] CHEN L, XU S, ZHANG H, et al. Fair and exculpable attribute-based searchable encryption with revocation and verifiable outsourced decryption using smart contract[J]. IEEE Internet of Things Journal, 2024, 1 - 1.
- [24] WANG F, CUI J, ZHANG Q, et al. Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial Internet of things[J]. IEEE transactions on dependable and secure computing, 2023, 21(4): 1587-1604.
- [25] ZHANG J, WANG B, WANG X A, et al. New group user based privacy preserving cloud auditing protocol[J]. Future Generation Computer Systems, 2020, 106: 585-594.
- [26] DING R, ZHONG H, MA J, et al. Lightweight privacy-preserving identity-based verifiable IoT-based health storage system[J]. IEEE Internet of Things Journal, 2019, 6(5): 8393-8405.
- [27] SHEN W, QIN J, YU J, et al. Cite auditing and data sharing with sensitive information hiding for secure cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2018, 14(2): 331-346.
- [28] DU Y, DUAN H, ZHOU A, et al. Enabling secure and efficient decentralized storage auditing with blockchain[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 19(5): 3038-3054.
- [29] DE C A, IOVINO V. jpbcc: Java pairing based cryptography[C]//Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC). Kerkira: IEEE Press, 2011, 850 - 855.