计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(4):69−81 [doi: 10.15888/j.cnki.csa.009464]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

边缘计算中基于区块链的轻量级密文访问控制方案①

郑嘉诚[1,2], 何　亨[1,2], 陈月佳[1,2], 肖天哲[3]

[1](武汉科技大学 计算机科学与技术学院, 武汉 430065)
[2](武汉科技大学 智能信息处理与实时工业系统湖北省重点实验室, 武汉 430065)
[3](华中科技大学 计算机科学与技术学院, 武汉 430074)
通信作者: 何　亨, E-mail: heheng@wust.edu.cn

摘　要: 密文策略属性基加密 (ciphertext-policy attribute-based encryption, CP-ABE) 技术可以在保证数据隐私性的同时提供细粒度访问控制. 针对现有的基于 CP-ABE 的访问控制方案不能有效解决边缘计算环境中的关键数据安全问题, 提出一种边缘计算环境中基于区块链的轻量级密文访问控制方案 (blockchain-based lightweight access control scheme over ciphertext in edge computing, BLAC). 在 BLAC 中, 设计了一种基于椭圆曲线密码的轻量级 CP-ABE 算法, 使用快速的椭圆曲线标量乘法实现算法加解密功能, 并将大部分加解密操作安全地转移, 使得计算能力受限的用户设备在边缘服务器的协助下能够高效地完成密文数据的细粒度访问控制; 同时, 设计了一种基于区块链的分布式密钥管理方法, 通过区块链使得多个边缘服务器能够协同地为用户分发私钥. 安全性分析和性能评估表明 BLAC 能够保障数据机密性, 抵抗共谋攻击, 支持前向安全性, 具有较高的用户端计算效率, 以及较低的服务器端解密开销和存储开销.

关键词: 边缘计算; 区块链; 访问控制; 密文策略属性基加密; 椭圆曲线

引用格式: 郑嘉诚,何亨,陈月佳,肖天哲.边缘计算中基于区块链的轻量级密文访问控制方案.计算机系统应用,2024,33(4):69–81. http://www.c-s-a.org.cn/1003-3254/9464.html

Blockchain-based Lightweight Access Control Scheme over Ciphertext in Edge Computing

ZHENG Jia-Cheng[1,2], HE Heng[1,2], CHEN Yue-Jia[1,2], XIAO Tian-Zhe[3]

[1](College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China)
[2](Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China)
[3](School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Ciphertext-policy attribute-based encryption (CP-ABE) can provide fine-grained access control while guaranteeing data privacy. Considering that the existing CP-ABE-based access control schemes can not effectively address critical data security in edge computing, this study proposes a blockchain-based lightweight access control scheme over ciphertext (BLAC) in edge computing. In BLAC, a lightweight CP-ABE algorithm based on elliptic curve cryptography is designed, and fast elliptic curve scalar multiplication is adopted to realize algorithm encryption and decryption. Additionally, most of the encryption and decryption operations are securely transferred to make user devices with limited computing power efficiently complete the fine-grained access control process of ciphertext data with the assistance of edge servers. Meanwhile, a distributed key management method based on blockchain is designed, which enables multiple edge servers to collaboratively distribute private keys for users by blockchain. Security analysis and

① 基金项目: 国家自然科学基金 (62372343, 61602351)
收稿时间: 2023-10-15; 修改时间: 2023-11-15; 采用时间: 2023-12-05; csa 在线出版时间: 2024-01-30
CNKI 网络首发时间: 2024-02-01

© 中国科学院软件研究所 http://www.c-s-a.org.cn

performance evaluation show that BLAC can guarantee data confidentiality, resist conspiracy attacks, and support forward security. Additionally, it has high user-side computational efficiency and low server-side decryption overhead and storage overhead.

Key words: edge computing; blockchain; access control; ciphertext-policy attribute-based encryption (CP-ABE); elliptic curve

云计算通过基础设施搭建服务平台, 为用户提供了可扩展的计算与存储服务. 但其数据传输时具有较高的时延, 无法满足应用服务对于低时延的需求. 边缘计算作为云计算的扩展, 能够在网络边缘为用户提供计算服务, 极大提升了时延敏感型应用交互体验, 有效缓解了轻量级设备资源受限的问题[1−3]. 近年来, 边缘计算技术的快速进步推动了诸如智能家居, 智慧医疗等新型服务模式的发展与应用[4−6]. 在这些应用服务中, 海量数据存储在云服务器上, 并通过边缘计算提供的分布式计算资源进行处理, 但是边缘服务器与云服务器均不完全可信, 它们有可能窥探用户的隐私数据[7−9], 并且用户通常要实现数据的安全共享. 因此, 需要访问控制机制以避免其他用户对数据的非法访问. 传统的访问控制机制使用基于公钥的加密体制来限制用户的访问权限: 用户使用公钥对数据加密后进行共享, 持有对应私钥的用户才能进行解密. 在这种访问控制机制中, 需要可信的授权机构对所有用户的公钥进行管理. 然而边缘计算环境中的用户是海量的, 用户变更也较为频繁, 存储和维护用户的公钥将产生较大的开销, 并且也无法实现一对多的数据共享[10], 即传统的访问控制机制难以有效应用于边缘计算环境中. 密文策略属性基加密[11]是一种一对多的加密算法, 能够在保障数据机密性的同时实现细粒度访问控制. CP-ABE 为数据制定由属性集合组成的访问策略, 只有当用户的属性集合成功匹配访问策略时, 用户才能解密密文. 通过 CP-ABE 可以实现一对多的数据安全共享, 适用于边缘计算环境中.

现有的基于 CP-ABE 的访问控制方案[12]通常会假定一个完全可信的授权机构进行密钥生成与管理, 授权机构可以为自身及所有用户生成密钥, 导致系统中的密文有泄露的风险, 即系统存在密钥托管问题. 部分方案[13−15]采用多授权机构来解决密钥托管问题, 其中每个授权机构所管理的属性密钥各不相同, 然而当系统中属性数量过多时, 若通过单个授权机构所管理的属性密钥就可以解密某份密文, 那么密钥托管问题仍然存在. 与此同时, 用户需要同时与多个授权机构通信, 存在密钥分发困难, 通信成本较高等问题. 其次, 由于边缘计算环境中的终端设备是轻量级的, 如智能家居场景中所使用的嵌入式设备, 可穿戴设备, 传感器设备等, 计算能力低, 电池容量有限, 无法执行复杂度较高的计算任务, 现有方案[16−18]采用的双线性配对计算复杂度较高, 不适用于轻量级设备. 相关研究[19−21]通过将部分计算任务转移到服务器中以减少用户计算开销, 但受困于双线性配对运算的复杂性, 用户计算效率有待进一步提升. 再次, 边缘计算环境中用户对数据隐私保护的需求是多样的. 用户需要系统提供对数据权限的实时灵活的撤销, 如在智慧医疗中, 用户在就医时会授权医生对病历数据的访问权限, 就医结束后需要及时撤销医生的访问权限以保护用户的隐私数据. 相关研究[22−24]提出通过属性更新密钥来解决属性撤销问题, 即授权机构生成属性更新密钥, 将涉及到的密钥和密文同时更新, 但这带来了额外的计算负担, 并且不够灵活, 难以有效满足用户对数据访问权限更改的需求. 同时用户也需要系统提供对任意非法用户共谋攻击的抵抗能力. 如为某份数据文件定义访问控制策略{“Doctor” AND “Surgery”}, 用户 A 具有属性{“Doctor”}, 用户 B 具有属性{“Surgery”}, 两者都不满足访问策略, 但当 A 和 B 通过组合各自属性的方式进行共谋时则可能满足访问策略并能对文件进行非法访问.

针对上述边缘计算环境中存在的关键数据安全问题和现有方案存在的不足, 本文提出了一种边缘计算环境中基于区块链的轻量级密文访问控制方案 (BLAC). 本文的主要工作如下.

(1) 设计了一种基于椭圆曲线密码的轻量级 CP-ABE (BLAC-CP-ABE) 算法. 使用快速的椭圆曲线标量乘法实现 CP-ABE 算法加解密功能, 无需使用计算开销较大的双线性配对运算, 提升整体计算效率. 引入虚拟属性和转换密钥, 可以在不泄露隐私的情况下将大

© 中国科学院软件研究所 http://www.c-s-a.org.cn

部分计算开销转移到边缘服务器, 从而显著降低轻量级设备的计算开销. 通过 BLAC-CP-ABE 可以使得计算能力受限的用户设备在边缘服务器的协助下能高效地完成数据的细粒度访问控制过程.

(2) 结合 BLAC-CP-ABE, 实现了一种基于区块链和秘密共享技术的分布式密钥管理方法. 在边缘服务器之间部署区块链构建可信执行环境, 通过区块链使得多个边缘服务器能够协同安全地为用户分发私钥, 解决了已有方案中存在的密钥托管问题, 并且减少了用户与授权机构的通信开销. 同时, 利用区块链账本存储转换密钥, 通过区块链去中心化和防篡改的特性可以实现实时灵活的属性撤销.

(3) 对 BLAC 进行安全性分析和性能评估, 结果表明 BLAC 能够有效保障数据机密性, 支持前向安全性, 并能抵抗任意多个非法用户的共谋攻击, 同时用户端整体计算效率较高, 并且具有较低的服务器端解密开销, 以及较低的私钥和数据存储开销. 因此 BLAC 能够有效满足边缘计算环境中实际应用对数据安全性和性能的高要求.

1 相关工作

属性基加密由 Sahai 等[25]于 2005 年首次提出, 该算法提取用户的特征信息作为属性来对文件进行加密, 实现了灵活的访问控制机制. Bethencourt 等[11]在此基础上于 2007 年提出 CP-ABE 算法, 将访问策略嵌入密文, 密钥则与用户的属性集合相绑定, 当用户的属性集合与访问策略匹配才可以成功解密密文. Chase 等[26]提出多授权机构 CP-ABE 算法, 每个授权机构分别管理不同的属性集合, 单个授权机构将无法独立破解密文, 同时用户与多个授权机构通信也将产生额外的开销. 随后, 大量方案[27]利用 CP-ABE 算法实现访问控制机制, 但是双线性配对计算效率较低, 阻碍了 CP-ABE 的广泛应用. 研究人员为提升 CP-ABE 计算效率展开相关工作. Das 等[13]提出物联网环境中基于椭圆曲线的多授权机构 CP-ABE 方案, 利用快速的椭圆曲线标量乘法来提升 CP-ABE 的计算效率. Huang[14]提出了一种云环境中支持在线/离线加密的多授权机构 CP-ABE 方案, 通过增加可重用密文池, 减少离线/在线计算量, 用户计算开销较低. Sammy 等[15]提出云环境中层次化 CP-ABE 方案, 对访问控制结构进行优化, 降低了多层次化密文的加密计算开销, 并有效节省了存储空间. 以上方案虽然提升了 CP-ABE 计算效率, 但是方案均基于多授权机构实现, 无法有效解决密钥托管问题, 同时存在密钥分发困难, 通信开销较大等问题.

Pu 等[16]提出了边缘计算环境中基于区块链的属性撤销方案, 通过在区块链网络中维护属性撤销链实现了 CP-ABE 中灵活的属性撤销机制. Jiang 等[17]结合区块链实现边缘计算环境中对电子病历的访问控制机制, 病历数据以交易记录的形式存储在区块链中, 实现了病历数据的可追溯, 保证了数据的完整性. 上述方案均使用区块链实现了灵活的访问控制机制, 但其中 CP-ABE 加解密过程中均涉及到大量双线性配对运算, 计算效率较低. Zheng 等[18]实现了边缘计算环境中支持云边协同计算的 CP-ABE 方案, 该方案支持灵活动态的访问控制策略, 实现了分级访问控制. Yang 等[19]提出边缘计算环境中细粒度 CP-ABE 方案, 结合区块链实现多重访问控制策略保证用户对数据的合法访问, 同时将部分计算任务转移到边缘服务器以降低用户端的计算开销, 并设计重加密算法实现属性撤销. Xie 等[20]为了提升计算效率, 构建混合云环境来转移用户计算开销, 用户计算开销显著降低, 但混合云环境需要搭建私有服务器, 对用户具有较大限制. Qin 等[21]利用区块链实现云环境中属性跨域管理, 通过调用智能合约为用户生成密钥, 减少用户与多个授权机构之间的通信开销, 同时通过云服务器为用户分担部分计算开销. 上述方案均通过将部分计算任务转移到服务器中提升用户计算效率, 但双线性配对计算复杂度较高, 用户计算效率有待进一步提升. Tu 等[22]提出了雾计算环境中可撤销的 CP-ABE 方案, 通过引入属性组密钥的方式实现属性撤销. Li 等[24]提出边缘计算环境中恒定密文大小的可撤销 CP-ABE 方案, 该方案具有恒定的密文大小并使用边缘服务器转移用户计算开销. 上述方案均通过重加密的方式实现了属性撤销, 具有额外的计算开销.

2 预备知识

2.1 椭圆曲线密码学

椭圆曲线密码学 (elliptic curve cryptography, ECC) 是一种基于公钥的加密体制, 相较于其他加密算法, 能够维持相同安全性的同时提供更小的密钥长度和更快的计算时间[28].

定义在有限域 $GF(q)$ 中的椭圆曲线 E 可以由式 (1)

© 中国科学院软件研究所 http://www.c-s-a.org.cn

描述:

$$y^2 = x^3 + ax + b \tag{1}$$

椭圆曲线 E 是由该方程所有的解 (x, y) 与无穷远点 O 组成的集合, 其中 a, b 表示有限域 $GF(q)$ 中的两个元素, 并且满足 $4a^3 + 27b^2 \neq 0$.

2.2 Shamir 秘密共享方案

Shamir 秘密共享方案能够将一个秘密值 s 分成 n 份子秘密, 系统中的每位用户持有一份. 系统指定阈值 t ($1 \leqslant t \leqslant n$), 当且仅当系统中 t 个及以上的用户共享手中的子秘密值时, 才能恢复出秘密值 s, 否则无法得到关于 s 的任何信息.

Shamir(t, n) 秘密共享方案具体流程如下.

随机生成一个阶为 q 的有限域 $GF(q)$, 其中 $q>n$. 从有限域 $GF(q)$ 中选取 n 个互不相同的正整数 a_0, $a_1, \cdots, a_{n-1}$, 构造多项式 $g(x)$, 并令 $a_0 = s$, 如式 (2) 所示.

$$g(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1} \tag{2}$$

从有限域 $GF(q)$ 随机选取 n 个非零元素 x_i, 计算子秘密 $s_i = g(x_i) \bmod q$ ($1 \leqslant i \leqslant n$), 并将子秘密值分发给系统中的用户.

当系统中 t 个及以上的用户共享子秘密值 s_i 时即可恢复秘密值 s. 首先计算拉格朗日多项式 $f(x)$, 如式 (3):

$$f(x) = \prod_{i=1, i\neq j}^{t} (x - x_i)/(x_j - x_i) \tag{3}$$

然后计算重构多项式 $g(x)$, 如式 (4):

$$g(x) = \sum_{i=1}^{t} s_i f(x_i) \tag{4}$$

则秘密值 s=g(0).

2.3 区块链

区块链是基于分布式网络的共享账本. 在区块链网络中, 数据以区块的形式存储起来, 通过哈希计算, 生成每一个区块的散列值. 最终所有的区块通过散列值前后链接起来, 形成了区块链. 区块链具有去中心化, 防篡改, 可追溯等特性, 有利于保护数据完整性[29–31].

(1) 对等节点. 对等节点是区块链网络的基本组成单位, 每一个节点都维护相同的数据账本.

(2) 智能合约. 智能合约是区块链网络中预定义的一段代码, 当满足预定的条件时, 智能合约将自行验证并执行. 区块链中的对等节点都可以部署智能合约, 并以交易事务的形式广播给区块链中的所有节点, 当交易事务得到足够的背书验证后才会写入数据账本中存储, 并基于共识机制, 使所有节点的数据账本保持一致.

2.4 决策性 Diffie-Hellamn 假设

决策性 DDH (decisional Diffie-Hellamn) 假设的定义如下: 设 G 是以大素数 r 为阶的循环群中的生成元, 从域 Z_p 中随机选取 a, b, R. 对于给定元组 (G, aG, bG), 敌手想在多项式时间内区分 abG 和随机元素 R 是困难的. 若存在多项式时间算法 β 满足:

$$|\Pr[\beta(G, aG, bG, Z = abG) = 0]| - |\Pr[\beta(G, aG, bG, Z = R) = 0]| \geqslant \varepsilon$$

则算法 β 能以不可忽略的优势 $\varepsilon > 0$ 解决该假设, 否则该假设成立.

3 方案实现

3.1 方案框架

BLAC 系统模型图如图 1 所示. BLAC 中包含 5 个实体, 分别为授权机构 (trusted authority, TA), 数据拥有者 (data owner, DO), 数据使用者 (data user, DU), 边缘服务器 (edge server, ES) 和云服务器 (cloud server, CS).

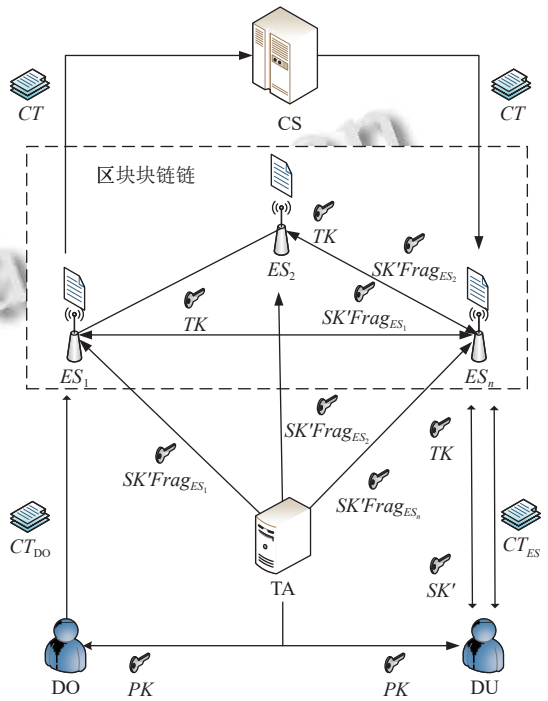

图 1 系统模型图

TA 负责系统初始化, 生成部分属性私钥并分割成切片. TA 与 ES 之间存在安全通道, 用于部分属性密钥切片的分发. TA 是一个诚实的实体.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

DO 是有数据共享需求的用户, 定义访问控制结构对数据文件进行部分加密并上传至 ES, ES 完全加密之后存储至 CS. DO 是一个诚实的实体.

DU 是想要访问数据文件的用户, 将数据访问请求发送给 CS, 并委托 ES 进行部分解密, 最后 DU 进行完全解密. DU 可能是不诚实的实体, 不诚实的 DU 可能通过共谋的方式解密未授权的数据文件.

CS 提供数据存储服务, 为 DO 存储生成的密文. CS 是一个诚实但好奇的实体, 可能会窥探用户的重要隐私数据, 但不会与不诚实 DU 串通并向被撤销的 DU 提供密文.

ES 提供低时延的计算和存储服务, ES 对数据文件进行完全加密后上传至 CS, 同时它也可以为 DU 执行部分解密操作. 多个 ES (ES_1, ES_2, ⋯, ES_n) 形成联盟链网络, 为用户提供区块链服务. ES 可能会窥探用户的重要隐私数据, 是一个诚实但好奇的实体.

首先 TA 执行系统初始化函数生成系统公钥 *PK* 和系统主密钥 *MSK*, *PK* 将公开给 DO 与 DU, 进行数据文件的加解密, *MSK* 将用于生成部分属性私钥 *SK′*.

然后 TA 执行私钥生成函数生成属性全集 *U* 的部分属性私钥 *SK′*并通过秘密共享函数生成 *SK′*的切片 *SK′Frag*, 并将各 ES 对应的切片$SK'Frag_{ES_j}$ ($1 \leqslant j \leqslant n$) 由安全通道分发给各 ES 以实现密钥的分布式管理.

当 DU 向某 ES 请求生成属性私钥时, 该 ES 将在区块链网络中广播密钥生成请求, 各个节点验证密钥生成请求后将$SK'Frag_{ES_j}$发送给请求方. 当得到足够数量的切片后, 该 ES 为用户生成部分属性私钥*SK′*; 用户收到*SK′*后计算生成转换密钥 *TK* 并发送给该 ES, 该 ES 通过交易事务的形式将 *TK* 广播给其他 ES, 各 ES 对交易事务进行验证, 验证成功之后将 *TK* 上传至自己的区块链账本中.

DO 通过 BLAC-CP-ABE 算法加密其需要共享的数据集 *PT* 得到部分加密密文 CT_{DO}, 经过 ES 加密得到完整密文 *CT*, *CT* 由 ES 上传至 CS 进行存储; 当 DU 想要访问 CS 中的数据时, ES 从区块链账本中获取 DU 的转换密钥 *TK*, 并从 CS 处下载密文集 *CT* 进行部分解密计算得到部分解密密文 CT_{ES}; DU 收到 CT_{ES} 后执行完整解密计算得到数据 *PT*.

3.2 方案设计

BLAC 由以下 9 个函数组成, 函数 1 实现了 BLAC-CP-ABE 算法的初始化, 函数 2–5 通过秘密共享技术实现了分布式密钥管理方法, 函数 6–9 则实现了 BLAC-CP-ABE 算法对数据的加密和解密. 表 1 展示了 BLAC 所涉及的主要参数.

表 1 主要参数

参数	含义
k	系统安全参数
PP	系统公共参数
UID	用户标识符
PK	系统公钥
MSK	系统主密钥
U	属性全集
SK′	部分属性私钥
SK	属性私钥
SK'_iFrag	部分属性密钥SK'_i切片
$SK'Frag_{ES_j}$	ES_j所持有的*SK′*切片
TK	转换密钥
USK	用户解密密钥
PT	数据明文
CT_{DO}	用户部分加密密文
CT_{ES}	服务器部分解密密文
CT	完整密文

(1) Setup(k, U)→(PP, UID, PK, MSK)

如函数 1 所示, 系统初始化函数输入安全参数 *k* 和属性全集 *U* 得到系统的全局参数 *PP*, 用户标识符 *UID*, 系统公钥 *PK* 和系统主私钥 *MSK*.

函数 1. 系统初始化

输入: 安全参数 *k*, 属性全集 *U*.
输出: 全局参数 *PP*, 系统公钥 *PK*, 系统主私钥 *MSK*, 用户标识符 *UID*.

1) 生成一个 *q* 阶有限域 $GF(q)$ 和哈希函数H:$\{0,1\}^* \to Z_p$;
2) 从有限域 $GF(q)$ 随机选取生成元为 *G* 的椭圆曲线 *E*;
3) 从有限域 $GF(q)$ 随机选择$|U|$对随机数$\{y_1,k_1\},\cdots,\{y_{|U|-1},k_{|U|-1}\}$, $\{y_{vir},k_{vir}\} \in Z_p$, 与属性全集 *U* 中的属性相对应, 其中$\{y_{vir},k_{vir}\}$对应于用户的虚拟属性;
4) 为系统中的每一位用户定义标识符 *UID*;
5) 输出$PP=\{GF(q),G,U,E,H\}$, $PK=\{y_iG,k_iG\}$, $MSK=\{y_i,k_i\}$, *UID*.

(2) AttrKeyGen(UID, MSK)→SK'

如函数 2 所示, 私钥生成函数通过输入用户标识符 *UID* 和主密钥 *MSK*, 输出用户部分属性私钥*SK′*.

函数 2. 私钥生成

输入: 属性全集 *U*, 用户标识符 *UID*, 主密钥 *MSK*.
输出: 部分属性私钥*SK′*.

1) 对于属性全集 *U* 中的每一个属性, 计算$SK'_i=y_i+k_iH(UID)$;
2) 输出*SK′*.

(3) ShamirS(s, t, n)→$sFrag$

如函数 3 所示, 秘密共享函数通过输入秘密值

© 中国科学院软件研究所 http://www.c-s-a.org.cn

s 和门限值 (t, n), 输出 n 份子秘密值$sFrag$.

函数 3. 秘密共享

输入: 秘密值 s, 门限值 (t, n).
输出: 子秘密值$sFrag$.

1) 从有限域 $GF(q)$ 中随机选择 n 个非零元素 x_i, 其中$1\leqslant i\leqslant n$;
2) 构造多项式: $g(x)=s+a_1x+a_2x^2+\cdots+a_{t-1}x^{t-1}$;
3) 计算子秘密值$sfrag_i=g(x_i)$;
4) 输出$sFrag=\{sfrag_1, sfrag_2,\cdots, sfrag_n\}$.

(4) ShamirR($sFrag$)→s

如函数 4 所示, 秘密重构函数通过输入子秘密值 $sFrag=\{sfrag_1,sfrag_2,\cdots,sfrag_n\}$输出秘密值 s.

函数 4. 秘密重构

输入: 子秘密值$sFrag$.
输出: 秘密值 s.

1) 计算拉格朗日多项式:

$$f(x)=\prod_{i=1,i\neq j}^{t}(x-x_i)/(x_j-x_i);$$

2) 计算重构多项式$g(x)=\sum_{i=1}^{t} sfrag_i f(x_i)$;
3) 计算秘密值 $s=g(0)$;
4) 输出 s.

(5) TransKeyGen(SK')→(TK, USK)

如函数 5 所示, 转换密钥生成函数通过输入用户部分属性私钥进行计算, 输出用户转换密钥 TK 和解密密钥 USK, TK 将上传至区块链进行存储.

函数 5. 转换密钥生成

输入: 部分属性私钥SK'.
输出: 用户转换密钥 TK, 解密密钥 USK.

1) 随机选择$p\in Z_p$;
2) 计算$SK_i=SK'+p=y_i+k_iH(UID)+p$;
3) 输出$TK=\{SK_1,SK_2,\cdots,SK_i\}$, $USK=p$.

(6) Enc.DO(PT, PK, (M,ρ))→CT_{DO}

如函数 6 所示, DO 加密函数通过输入明文 PT, 系统公钥 PK 和用户制定的访问控制策略 (M,ρ) 输出部分加密密文 CT_{DO}.

函数 6. DO 加密

输入: 明文 PT, 系统公钥 PK, 访问控制策略 (M,ρ).
输出: 部分加密密文 CT_{DO}.

1) 随机选取$s\in Z_p$, $ck\in Z_p$;
2) 随机选取$\vec{v}=(s,v_2,\cdots,v_n)^{\mathrm{T}}\in Z_p$;
3) 随机选取$\vec{u}=(0,u_2,\cdots,u_n)^{\mathrm{T}}\in Z_p$;
4) 随机选取$r_{\mathrm{vir}}\in Z_p$, $R=(r_1,r_2,\cdots,r_n)\in Z_p^{l-1}$;
5) 计算 $CT_0=Enc_{ck}(m)$, $CT_1=ck+sG$;
6) 计算 $CT_{\mathrm{H}}=H(CT_0)$;
7) 计算$\lambda_i=M_i\vec{v}_i$, $\omega_i=M_i\vec{u}_i$, $i\in(1,l-1)$;
8) 计算$C_{\mathrm{vir}}=\lambda_{\mathrm{vir}}G+y_{\mathrm{vir}}r_{\mathrm{vir}}G$;
9) 计算$C'_{\mathrm{vir}}=\omega_{\mathrm{vir}}G+k_{\mathrm{vir}}r_{\mathrm{vir}}G$, $R_{\mathrm{vir}}=r_{\mathrm{vir}}G$;
10) 输出: $CT_{\mathrm{DO}}=\{(M,\rho),CT_0,CT_1,CT_H,C_{\mathrm{vir}},C'_{\mathrm{vir}},R_{\mathrm{vir}},R,\lambda,\omega\}$.

(7) Enc.ES(CT_{DO}, PK)→CT

如函数 7 所示, ES 加密函数通过输入系统公钥 PK 对部分密文 CT_{DO} 进行加密, 输出最终密文 CT.

函数 7. ES 加密

输入: 系统公钥 PK, 部分密文 CT_{DO}.
输出: 最终密文 CT.

1) 计算$C_x=\lambda_xG+y_xr_xG$;
2) 计算$C'_x=\omega_xG+k_xr_xG$, $R_x=r_xG$;
3) 输出: $CT=\{(M,\rho),CT_0,CT_1,CT_H,C_x,C'_x,R_x,C_{\mathrm{vir}},C'_{\mathrm{vir}},R_{\mathrm{vir}}\}$.

(8) Dec.ES(PK, TK, CT)→CT_{ES}

如函数 8 所示, ES 解密函数通过输入系统公钥 PK, 转换密钥 TK 和密文 CT 输出部分解密密文 CT_{ES}. 如果 DU 的属性集合满足密文制定的访问策略, 则存在集合$\{c_i\in Z_p\}_{i\in I}$, 使得$\sum_{i\in I}c_iM_i=(1,0,0,\cdots,0)$.

函数 8. ES 解密

输入: 系统公钥 PK, 转换密钥 TK, 密文 CT.
输出: 部分解密密文 CT_{ES}.

1) 计算$\{c_i\in Z_p\}_{i\in I}$使得$\sum_{i\in I}c_iM_i=(1,0,0,\cdots,0)$;
2) 若$\{c_i\in Z_p\}_{i\in I}$能成功求出, 计算:

$$\begin{aligned}D_x&=C_x-SK_{\rho(i),UID}R_x+H(UID)C'_x\\&=\lambda_xG+y_xr_xG-y_xr_xG-k_xH(UID)r_xG\\&\quad-pr_xG+H(UID)\omega_xG+H(UID)k_xr_xG\\&=\lambda_xG+H(UID)\omega_xG-pr_xG\end{aligned}$$

3) 计算:

$$N_1=\sum_{x\in I}c_xD_x=sG-p\sum_{x\in I}c_xr_xG$$

$$N_2=\sum_{x\in I}c_xR_x=\sum_{x\in I}c_xr_xG$$

4) 若$\{c_i\in Z_p\}_{i\in I}$计算失败, 令 $N_1=N_2=\varnothing$;
5) 若 $N_1=N_2=\varnothing$, 输出 $CT_{ES}=\varnothing$,
否则输出 $CT_{ES}=\{CT_0, CT_1, CT_H, N_1, N_2\}$.

(9) Dec.DU(USK, CT_{ES})→PT

如函数 9 所示, DU 解密函数通过输入用户解密密钥 USK 和部分解密密文 CT_{ES} 输出数据明文 PT.

函数 9. DU 解密

输入: 用户解密密钥 USK, 部分解密密文 CT_{ES}.
输出: 数据明文 PT.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

1) 计算:

$$ck=CT_1-N_1+pN_2$$

$$=ck+sG-sG-p\sum_{x\in I}C_xr_xG+p\sum_{x\in I}C_xr_xG=ck$$

2) 计算$m'=Dec_{ck}(CT_0)$;

3) 输出$PT=m'$.

3.3 具体实现

BLAC 包括 7 个阶段: 初始化阶段, 秘密共享阶段, 用户加密阶段, 边缘服务器加密阶段, 密钥生成阶段, 边缘服务器解密阶段, 用户解密阶段. 具体实现过程如图 2 所示.

(1) 初始化阶段. TA 通过 *Setup*() 函数初始化整个系统, 生成系统公钥 *PK*, 系统主密钥 *MSK*, 公共参数 *PP* 和用户标识符 *UID*. 之后, TA 通过 AttrKeyGen() 函数生成属性全集 *U* 的部分属性私钥SK'.

(2) 秘密共享阶段. TA 对SK'进行分割. 首先, TA 通过 ShamirS() 函数得到每一个SK_i'的 n 个切片$SK_i'Frag=\{SK_i'frag_1,SK_i'frag_2,\cdots,SK_i'frag_n\}$并聚合得到属性全集 U 的切片$SK'Frag=\{SK_1'Frag,SK_2'Frag,\cdots,SK_{|U|}'Frag\}$. 之后, TA 通过安全通道向各 ES 分发对应的切片, 如 ES_j 得到$SK'Frag_{ES_j}=\{SK_1'frag_j,SK_2'frag_j,\cdots,SK_{|U|}'frag_j\}$, 同时 TA 自身无需存储$SK'$.

(3) 用户加密阶段. DO 为了限制用户对数据文件 m 的访问权限, 定义访问策略(M,ρ). 利用 Enc.DO() 函数对数据文件 m 进行加密得到部分加密密文 CT_{DO}, 并将 CT_{DO} 上传至 ES.

(4) 边缘服务器加密阶段. ES 收到部分加密密文后, 利用 Enc.ES() 函数生成最终密文 *CT*, 并将 *CT* 上传至 CS.

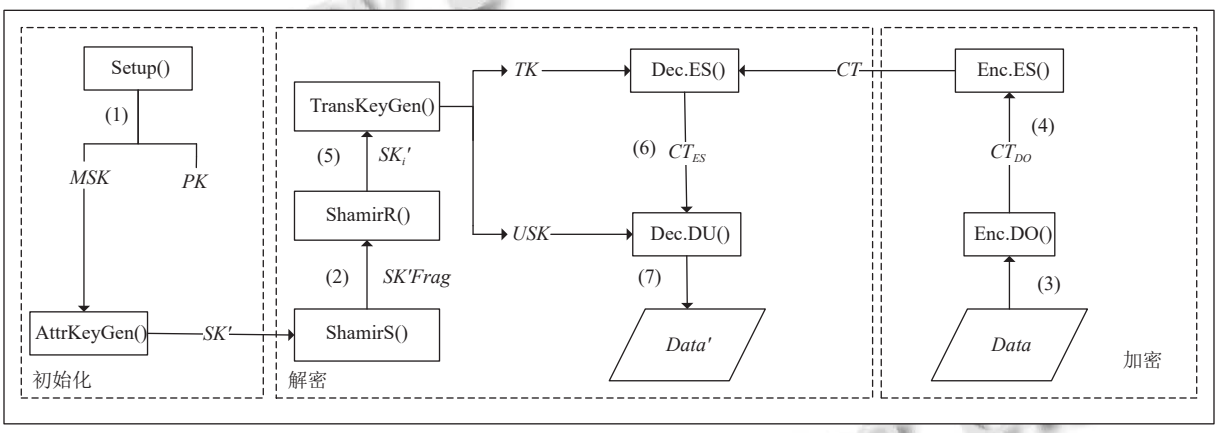

图 2 系统流程图

(5) 密钥生成阶段. DU 将自己的属性集合和标识符 *UID* 发送给 ES 进行密钥生成. ES 收到密钥生成请求后首先向区块链网络中的各个节点发送请求获取$SKFrag_{ES_j}$. 当得到至少 t 个节点验证并且响应后, ES 通过 ShamirR() 函数计算SK_i'. DU 收到SK_i'后执行 TransKeyGen() 函数生成转换密钥 *TK*, 解密密钥 *USK*. *USK* 由 DU 自己保存, *TK* 将发送给 ES, ES 通过算法 1 生成 *TK* 的交易事务Tx_{storage}并提交至区块链网络进行验证, 验证成功之后 *TK* 才能上传到数据账本中存储. 当 DU 向 ES 发送数据请求时, ES 将利用 *TK* 为 DU 进行部分解密. 此外, DU 与单个 ES 交互生成属性密钥的通信开销比多授权机构方案[13]更小.

算法 1. Tx_{storage} 事务生成算法

输入: 事务标识符 *ID*, 转换密钥 *TK*, ES 的签名私钥 BSK_{ES}.
输出: 事务Tx_{storage}.

1) 计算 *TK* 哈希值: $TKcheck=H(TK)$;
2) 计算 Tx_{storage} 哈希值: $MD=H(ID,TKcheck)$;
3) 计算 ES 签名值: $sign=Sign_{BSK_{ES}}(MD)$;
4) 返回$Tx_{\text{storage}}=\{ID,TK,TKcheck,sign\}$.

ID 为事务标识符, 区块链中的每一个事务都具有唯一标识符, BSK_{ES} 为 ES 的签名私钥. Tx_{storage} 事务生成之后将广播给区块链网络中的其他 ES 节点, 其他 ES 节点通过算法 2 对 Tx_{storage} 进行验证, 以避免 *TK* 在传输过程中被篡改.

算法 2. Tx_{storage} 事务验证

输入: 事务Tx_{storage}, ES 的签名公钥 BPK_{ES}.
输出: 事务Tx_{storage}验证结果.

1) 计算Tx_{storage}哈希值: $MD'=H(ID,H(TK))$;
2) 使用公钥计算签名值: $MD=Compute_{BPK_{ES}}(sign)$;

© 中国科学院软件研究所 http://www.c-s-a.org.cn

3) 若 $MD=MD'$, 则计算 $TKcheck'=H(TK)$, 若 $TKcheck'=TKcheck$, 返回验证成功; 否则返回验证失败;
4) 否则返回验证失败.

当足够数量的节点验证 Tx_{storage} 成功后, TK 才能上传至区块链数据账本中存储. 若 DU 某个属性被撤销或者退出系统, 通过算法 3 生成交易事务 Tx_{revoke} 并上传至区块链网络中进行验证, 验证成功之后将 TK 标记为撤销状态, ES 不会利用被撤销的 TK 为用户进行解密操作, 因此实现了实时灵活的属性撤销.

算法 3. Tx_{revoke} 事务生成

输入: 事务标识符 ID, 被撤销的转换密钥 TK_{revoke}, DU 的签名私钥 BSK_{DU}.
输出: 事务 Tx_{revoke}.

1) 计算 TK_{revoke} 哈希值: $TKcheck_{\text{revoke}}=H(TK_{\text{revoke}})$;
2) 计算 TK_{revoke} 哈希值: $MD=H(ID,TKcheck_{\text{revoke}})$;
3) 计算 DU 签名值: $sign=Sign_{BSK_{\text{DU}}}(MD)$;
4) 返回 $Tx_{\text{revoke}}=\{ID,TK,TKcheck_{\text{revoke}},sign\}$.

BSK_{DU} 为 DU 的签名私钥. Tx_{revoke} 事务生成之后将广播到区块链网络, 通过算法 4 对 Tx_{revoke} 进行验证, 同样当足够数量的节点验证 Tx_{revoke} 成功后 TK_{revoke} 将被标记为撤销状态.

算法 4. Tx_{revoke} 事务验证

输入: 事务 Tx_{revoke}, DU 的签名公钥 BPK_{DU}.
输出: 事务 Tx_{revoke} 验证结果.

1) 计算 Tx_{revoke} 哈希值: $MD'=H(ID,H(TK_{\text{revoke}}))$;
2) 使用公钥计算签名值: $MD=Compute_{BPK_{\text{DU}}}(sign)$;
3) 若 $MD=MD'$, 则计算 $TKcheck'_{\text{revoke}}=H(TK_{\text{revoke}})$, 若 $TKcheck'_{\text{revoke}}=H(TK_{\text{revoke}})$, 返回验证成功; 否则返回验证失败;
4) 否则返回验证失败.

(6) 边缘服务器解密阶段. 当 DU 有数据访问需求时, 首先向 CS 发送数据访问请求, CS 将相应密文 CT 发送至 ES 进行委托计算. ES 通过 Dec.ES() 函数执行解密操作得到部分解密密文 CT_{ES}.

(7) 用户解密阶段. DU 收到部分解密密文后若 CT_{ES} 不为空集, 则利用 Dec.DU() 函数进行最终解密得到 m'. 为了验证 m' 是否被篡改, DU 计算 $CT'=Enc_{ck}(m')$, 若 $H(CT')\neq CT_H$, 则 CS 中存储的数据 m 被篡改, DU 直接丢弃 PT, 否则表示 DU 得到的数据没有被篡改.

3.4 安全模型

BLAC 方案中的安全模型由挑战者 β 和敌手 $\mathcal{A}$ 之间的选择明文攻击博弈游戏来描述. 挑战者 β 为 TA, 主要流程如下.

初始化: 敌手 $\mathcal{A}$ 选择访问控制结构 (M,ρ), 并发送给挑战者 β.

系统设置: 挑战者 β 运行 Setup() 生成公钥 PK 和主私钥 MSK, 将 PK 发送给敌手 $\mathcal{A}$.

阶段 1: 敌手 $\mathcal{A}$ 指定属性集合 U, 并向挑战者 β 申请获得与 U 相关的密钥 SK, 同时根据限制, 属性集合 U 不满足初始化中制定的访问控制策略. 挑战者 β 计算得到 SK 发送给敌手 $\mathcal{A}$.

挑战: 敌手 $\mathcal{A}$ 选择两个等长的数据明文 m_0 和 m_1 发送给挑战者 β. 挑战者 β 使用随机选择器 $b\in\{0,1\}$ 和访问策略 (M,ρ) 生成加密密文 m_b, 最后将加密密文 CT^* 发送给敌手 $\mathcal{A}$.

阶段 2: 和阶段 1 类似, 敌手 $\mathcal{A}$ 继续指定属性集合 U, 并向挑战者申请获得与 U 相关的属性私钥.

猜测: 敌手 $\mathcal{A}$ 输出对 b 的猜测 $b'\in\{0,1\}$, 如果 $b'=b$, 则敌手 $\mathcal{A}$ 获得了游戏的胜利. 敌手 $\mathcal{A}$ 所具有的优势定义为: $Adv_{\mathcal{A}}=\left|\Pr[b'=b]-\frac{1}{2}\right|$.

4 安全性分析与性能评估

4.1 安全性分析

在第 3.1 节提出的实体安全性假设以及第 3.4 节提出的安全模型的基础上证明 BLAC 在 DDH 假设下是安全性的, 同时 BLAC 能够保障数据机密性, 抵抗共谋攻击, 支持前向安全性.

(1) 安全性证明

定理 1. 如果敌手在多项式时间内以不可忽略的优势 $\varepsilon>0$ 破解 BLAC, 则挑战者 β 可以在多项式时间内打破 DDH 假设.

设 G 是 r 阶循环群中的生成元, 挑战者随机选择 $x,y\in Z_p$ 与 $R\in Z_p$. 定义随机选择器 $b\in\{0,1\}$, 若 b=0 时, $Z=xyG$, 若 b=1, 则 $Z=RG$. 挑战者 β 发送元组 (G, xG, yG) 给敌手 $\mathcal{A}$, 假设敌手 $\mathcal{A}$ 具有不可忽略的优势 $\varepsilon>0$ 来赢得安全游戏, 安全游戏流程如下.

初始化: 敌手 $\mathcal{A}$ 选择访问控制结构 (M,ρ), 并发送给挑战者 β.

系统设置: 挑战者 β 为属性集合 U 中每一个属性选择随机数 $\{y_i,k_i\}\in Z_p$ 并公布系统公钥 $PK=\{y_iG,k_iG\}$.

阶段 1: 敌手 $\mathcal{A}$ 指定属性集合 U 和用户标识符 UID, 并向挑战者 β 申请相应的密钥 SK, 同时根据限制, 属性集合 U 不满足初始化中制定的访问控制策略. 挑战者

© 中国科学院软件研究所 http://www.c-s-a.org.cn

β随机选择$p \in Z_p$计算得到$SK_{i,UID} = y_i + k_i H(UID) + p$发送给敌手$\mathcal{A}$.

挑战: 敌手$\mathcal{A}$选择两个等长的数据明文 m_0 和 m_1 发送给挑战者 β. 挑战者 β随机选取向量$\vec{v} = (s, v_2, \cdots, v_n)^{\mathrm{T}} \in Z_p$, $\vec{u} = (0, u_2, \cdots, u_n)^{\mathrm{T}} \in Z_p$计算$\lambda_i = M_i \vec{v}$和$\omega_i = M_i \vec{u}$, 其中$M_i$表示矩阵$M$的第$i$行. 挑战者$\beta$使用随机选择器 $b \in \{0, 1\}$ 计算$CT_1 = m_b + sG$, 最后计算生成加密密文 $C_x = \lambda_x G + y_x r_x G$, $C'_x = \omega_x G + k_x r_x G$, $R_x = r_x G$并将$CT = \{(M, \rho), CT_1, C_x, C'_x, R_x\}$发送给敌手$\mathcal{A}$.

阶段 2: 和阶段 1 类似, 敌手$\mathcal{A}$继续指定属性集合 U 和用户标识符 UID, 并向挑战者申请获得与 U 相关的属性私钥, 同样敌手$\mathcal{A}$不能违反属性集合的限制条件.

猜测: 敌手$\mathcal{A}$输出对 b 的猜测$b' \in \{0, 1\}$, 如果$b' = b$, 挑战者 β输出 0 表示$Z = xyG$, 敌手获$\mathcal{A}$得了游戏的胜利. 否则, 挑战者 β输出 1 表示 Z=RG. 若敌手$\mathcal{A}$能以不可忽略的优势$\varepsilon > 0$来赢得安全游戏, 则敌手$\mathcal{A}$获胜的概率为$|\Pr[\beta(G, xG, yG, Z = xyG) = 0]| = \frac{1}{2} + \varepsilon$, 敌手$\mathcal{A}$失败的概率为$|\Pr[\beta(G, xG, yG, Z = R) = 0]| = \frac{1}{2}$. 挑战者$\beta$所具有的优势为式 (5):

$$\frac{1}{2}(|\Pr[\beta(G, xG, yG, Z = xyG) = 0]| + |\Pr[\beta(G, xG, yG, Z = R) = 0]|) - \frac{1}{2} = \frac{\varepsilon}{2} \quad (5)$$

在上述过程中, 假定敌手$\mathcal{A}$具有不可忽略的优势$\varepsilon > 0$来赢得安全游戏, 则挑战者 β的优势为$\frac{\varepsilon}{2}$. 但不存在多项式时间算法能够破解 DDH 假设, 则敌手$\mathcal{A}$无法在多项式时间内以不可忽略的优势$\varepsilon > 0$来赢得安全游戏, 即无法在多项式时间内破解 BLAC. 因此, BLAC 在 DDH 假设下是安全的.

(2) 数据机密性

在 BLAC 中, 数据经过部分加密并在 ES 进行最终加密的过程中, 好奇的 ES 可能试图破解用户的加密数据. 在 ES 进行加密的过程中, ES 已知部分密文集 CT_{DO} 和系统公钥 PK, 真实数据由用户通过对称加密算法进行加密, 对称加密算法可以认定是安全的; 与秘密值相关的$\vec{v}$, R, r_{vir} 由用户随机选择, 虚拟密文集 $C_{\mathrm{vir}}, C'_{\mathrm{vir}}$由用户计算得出, r_{vir}, λ_{vir}, ω_{vir}都不会发送给 ES, 因此 ES 无法获取密文的有效信息. 在解密过程中, ES 利用转换密钥 TK 进行部分解密得到 N_1 和 N_2 后, 无法完全解密获取数据明文, 因为解密密钥 USK=$\{p\}$保存在用户手中, ES 无法获取. 与此同时, 只有满足访问控制策略的 DU 才能正确解密密文数据, 不满足策略的 DU 无法解密.

在 BLAC 中, 无论系统中属性数量规模如何, 任意单个 ES 均无法独立生成属性私钥去解密密文, 因此所采用的分布式密钥管理方法可以有效解决密钥托管问题.

(3) 共谋攻击

在 BLAC 中, 单个 DU 的属性集合不满足访问控制策略时, 可能通过与其他用户组合属性集合的方式非法解密数据. 在密钥生成阶段, 每个 DU 生成的部分属性私钥$SK'_{i,UID} = y_i + k_i H(UID)$与用户标识符 UID 绑定. 若存在 DU 相互勾结, 由于每位 DU 的 UID 各不相同, 在解密阶段, 无法正确计算并得到 D_x, 后续解密过程中无法推出 sG, 所以无法解密密文. 因此, BLAC 可以抵抗任意多个用户的共谋攻击. 同样, ES 无法获取其他用户标识符 UID, 故 ES 与 DU 共谋也无法解密密文.

(4) 前向安全性

在 BLAC 中, 用户可以随时加入系统或者退出系统, 退出系统的用户可能继续利用属性私钥来解密系统中的密文. 用户的转换密钥 TK 被保存在区块链当中. 当用户退出系统后, 区块链中与用户标识符 UID 相关的 TK 都将被标记为无效, 此时 ES 将无法为用户执行部分解密操作. 同时, 如果用户任意单个或者多个属性被撤销时, 区块链中的转换密钥 TK 同样会被标记, 在 ES 进行解密时, 被标记为无效的属性私钥将无法使用. 区块链防篡改和可追溯的特性保证了 ES 无法更改区块链中的转换密钥. 这样系统实现了实时灵活的属性撤销并保证了前向安全性.

4.2 性能评估

本节将 BLAC 与密切相关的文献[13,16,20,21]进行比较, 比较方面包括: 方案功能, 计算开销, 存储开销. 实验具体过程为: 使用 Intel Core i5-8300H @2.30 GHz, 16 GB RAM, 搭建环境为 Ubuntu 18.04 操作系统的服务器作为 ES 和 TA, 并在 ES 上部署 Hyper Ledger Fabric V1.4.6 联盟链; 使用 Intel Core i5-4460 @3.20 GHz, 8 GB RAM, 搭建环境为 Ubuntu 18.04 操作系统的设备作为 DO 和 DU; 使用阿里云 S6 云服务器作为 CS. 仿真实验基于 Java Pairing-based Cryptography (JPBC) 库实现, 选择 512 位有限域上 160 阶的 TypeA 类型椭圆曲线,

© 中国科学院软件研究所 http://www.c-s-a.org.cn

实验结果为 30 次实验平均值.

(1) 方案功能

方案功能比较如表 2 所示. 文献[13]主要应用于物联网环境, 使用快速的椭圆曲线标量乘法和外包解密提升计算效率, 文献[16]主要应用于边缘计算环境, 使用区块链并结合秘密共享技术对密文进行切片实现了属性撤销, 文献[20]主要应用于云计算环境, 采用外包解密提升计算效率. 上述 3 篇文献都采用多授权机构进行密钥管理, 具有较大的通信开销, 且当属性数量过多时仍然存在密钥托管问题. 文献[21]主要应用于云计算环境, 通过区块链实现了密钥管理, 并使用外包解密提升计算效率, 但受限于双线性配对计算的复杂度, 计算效率有待进一步提升, 且缺少属性撤销功能. BLAC 基于快速的椭圆曲线标量乘法设计并实现高效的算法功能, 并将大部分加解密计算操作转移到 ES, 降低用户端计算开销. 同时, BLAC 通过区块链结合秘密共享技术对部分属性私钥进行切片, 不仅实现了密钥的分布式管理, 也具备灵活的属性撤销功能, 可以有效解决边缘计算环境中存在的数据安全问题.

表 2 功能比较

方案	应用环境	密钥管理	双线性配对	属性撤销	外包加解密
文献[13]	物联网	多授权机构	No	No	解密
文献[16]	边缘计算	多授权机构	Yes	Yes	无
文献[20]	云计算	多授权机构	Yes	No	解密
文献[21]	云计算	区块链	Yes	No	解密
BLAC	边缘计算	区块链	No	Yes	加解密

(2) 计算开销与存储开销

基础运算开销和基础存储开销分别如表 3 和表 4 所示. 其中 E_1 和 E_T 分别表示群 G_1 和 G_T 中一次标量乘法或幂运算时间, M_1 和 M_T 表示群 G_1 和 G_T 内元素乘法运算时间, P 表示群 G_1 中一次双线性配对运算时间. Div_T 表示 G_T 群内一次除法运算时间. $|A_{\rm pol}|$表示访问策略中定义的属性数量, $|A_{\rm DU}|$表示 DU 的属性总数, $|A_U|$表示系统中的属性总数. 从表 3 中可知, ES 执行双线性配对计算开销是标量乘法的 2.46 倍, DO 或 DU 执行双线性配对计算开销是标量乘法的 3.42 倍, 采用快速的椭圆曲线标量乘法可以有效减少计算开销. 从表 4 中可知Z_p中的元素所需存储空间较小, 为 20 B, 而 G_1 和 G_T 域中元素所需存储空间皆为 128 B.

理论计算开销如表 5 所示. 在用户加密计算过程中, BLAC 通过虚拟属性的方式将大部分计算开销转移到 ES, 用户计算效率有效提升, 只需要 $6E_1$ 计算量. 而其余方案中用户加密计算开销将会随着访问控制策略中所包含的属性数量增加呈现线性增加趋势. 其中, 文献[21]增长率最高. 在解密过程中, 除了文献[16]以外都将部分计算转移到服务器上进行以减少用户计算开销. 在服务器解密过程中, BLAC 与文献[13]计算开销相同, 为$4|A_{\rm DU}|E_1$. 文献[20,21]计算过程中采用了大量双线性配对计算, 计算开销较大. 在用户解密过程中, 文献[20]将大部分解密计算转移到私有 CS 上进行, 用户只需一次除法计算即可解密, 计算开销最低. 但这种方式需要用户自行搭建完全可信的 CS, 不适用于大部分用户. 文献[21]由服务器承担了大部分解密计算开销, 用户解密仅需要 (M_T+E_T) 计算量, 但总解密开销仍然较大. BLAC 和文献[13]均需要 E_1 计算量, 总解密计算开销小, 适用于轻量级设备.

表 3 基础运算开销比较 (ms)

实体	E_1	$E_{\rm T}$	M_1	$M_{\rm T}$	P	$Div_{\rm T}$
ES	1.90	0.35	0.13	0.05	4.69	0.36
DO/DU	4.37	2.36	0.16	0.09	14.93	2.74

表 4 基础存储开销比较 (B)

基础存储	L_{G_1}	L_{G_T}	L_{Z_p}
存储开销	128	128	20

理论存储开销如表 6 所示. 在私钥存储开销方面, BLAC, 文献[13,16,20]的私钥存储开销均与属性数量正相关, 文献[16]中私钥为 G_1 群中元素, 存储开销增长率最大. 文献[21]中私钥存储开销最小且恒定, 为$3L_{G_T}$. 在密文存储开销方面, BLAC 存储开销相较于另外 4 个方案存储开销更大. BLAC 存储开销为$(3|A_{\rm pol}|+2)L_{G_1}$, 所涉及密文项更多, 且与属性相关联, 当属性数量增加时 BLAC 存储开销增长率最大, 然而采用快速的椭圆曲线标量乘法显著提升了 BLAC 加解密效率, 利用少量存储空间换来更快的解密时间对用户是更为有利的.

DO 加密计算时间如图 3 所示. 在文献[13,16,20,21]中, 随着属性数量的增加, 用户加密计算时间也逐步增加. 其中文献[21]的增长率最高, 文献[20]次之, 文献[13]和文献[16]计算开销接近. 而 BLAC 将繁重的加密计算工作转移到给 ES, 用户计算量稳定且最少, 与同样采用椭圆曲线标量乘法的文献[13]相比, 当属性数量为 11 时, BLAC 计算开销只有文献[13]的 26.1%. 与采用双线性配对计算的文献[21]相比, BLAC 计算开销仅有文献[21]的 13%, 随着属性数量增加, BLAC 在计算效率方面的优势将更加显著.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

表 5 理论计算开销比较

方案	用户加密	服务器解密	用户解密
文献[13]	$(2\|A_{\text{pol}}\|+1)E_1$	$4\|A_{\text{DU}}\|E_1$	E_1
文献[16]	$(2\|A_{\text{pol}}\|+1)E_1+2E_T$	—	$(2\|A_{\text{DU}}\|+1)P+(\|A_{\text{DU}}\|+2)Div_T$
文献[20]	$(3\|A_{\text{pol}}\|+1)E_1+\|A_{\text{pol}}\|M_1+E_T+M_T$	$(2\|A_{\text{DU}}\|+1)P+\|A_{\text{DU}}\|E_T+(2\|A_{\text{DU}}\|-1)M_T$	Div_T
文献[21]	$(2\|A_{\text{pol}}\|+1)E_T+(\|A_{\text{pol}}\|+1)M_T+3\|A_{\text{pol}}\|E_1+\|A_{\text{pol}}\|M_1$	$2\|A_{\text{DU}}\|P+2\|A_{\text{DU}}\|E_T+(2\|A_{\text{DU}}\|-2)M_T$	E_T+M_T
BLAC	$6E_1$	$4\|A_{\text{DU}}\|E_1$	E_1

表 6 理论存储开销比较

方案	私钥SK	密文CT
文献[13]	$\|A_{\text{DU}}\|L_{Z_p}$	$(2\|A_{\text{pol}}\|+2)L_{G_1}$
文献[16]	$(2\|A_{\text{DU}}\|+1)L_{G_1}$	$(2\|A_{\text{pol}}\|+1)L_{G_1}+L_{G_T}$
文献[20]	$(\|A_{\text{DU}}\|+2)L_{G_1}$	$(2\|A_{\text{pol}}\|+1)L_{G_1}+L_{G_T}$
文献[21]	$3L_{G_T}$	$2L_{G_1}+(\|A_{\text{pol}}\|+1)L_{G_T}$
BLAC	$\|A_{\text{DU}}\|L_{Z_p}$	$(3\|A_{\text{pol}}\|+2)L_{G_1}$

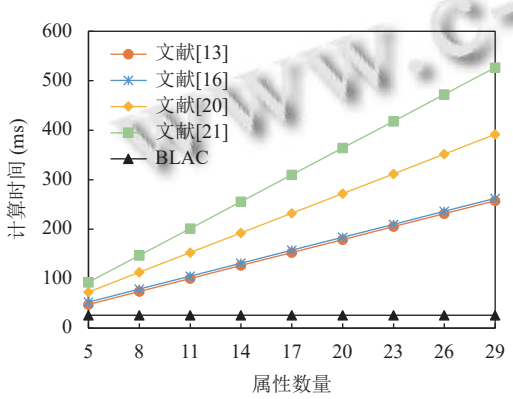

图 3 DO 加密计算时间

服务器解密计算时间如图 4 所示. 文献[13,20,21]中通过 CS 提供部分解密服务, BLAC 则通过 ES 完成部分解密工作. 文献[16]不涉及外包解密计算, 故图 4 中不存在文献[16]数据线段. BLAC 和文献[13]计算开销一致, 数据线段重合. 文献[20]和文献[21]计算开销基本一致, 数据线段接近重合, 同时两个方案都采用了大量双线性配对操作, 计算复杂度更高. 当属性数量为 11 时, BLAC 相较于文献[20], 解密计算效率提升了 34.5%, BLAC 采用的椭圆曲线标量乘法有效降低了计算量.

DU 解密计算时间如图 5 所示. BLAC 与文献[13]计算开销相同, 数据线段重合. 文献[16]由用户完成所有解密计算工作, 计算开销最大. 其余 4 种方案用户解密计算开销均恒定. BLAC 在解密时需要一次标量乘法操作, 其用户解密计算开销为 4.37 ms, 比文献[20]和文献[21]耗时更长, 但这对于 DU 来说是完全可以承担的. 文献[20]中采用私有 CS, 将几乎所有解密操作交给私有 CS, DU 通过一次 Div_T 运算即可解密, 但要求 CS 是完全可信的, 也带来了额外的服务器运营维护开销. 文献[21]由服务器承担大部分解密计算开销, 用户解密计算开销小, 但总解密计算开销相较于 BLAC 更大.

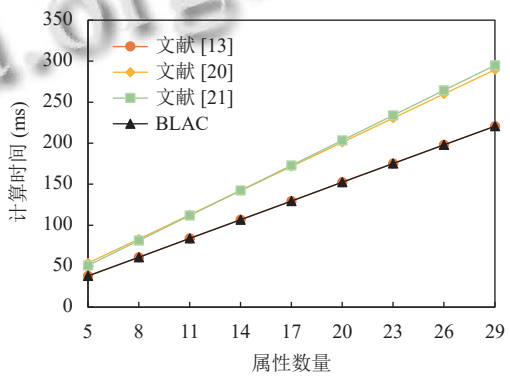

图 4 服务器解密计算时间

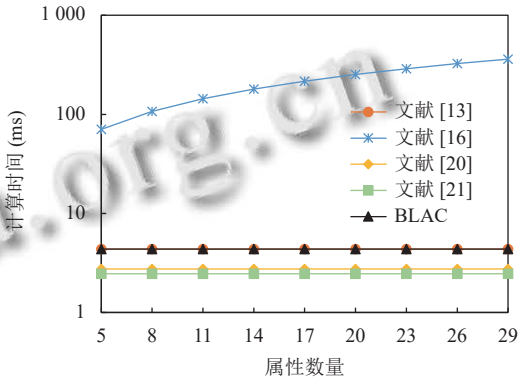

图 5 DU 解密计算时间

私钥存储开销如图 6 所示. BLAC 与文献[13]私钥存储开销相同, 数据线段重合. 文献[16]的私钥存储开销最大, 文献[20]存储开销次之, 当属性数量增加时, 与其余方案的差异也越来越大. 文献[21]的私钥存储开销较小且恒定, BLAC 和文献[13]私钥存储开销较小, 增长也相对缓慢, 当属性数量增加到 29 时, 密钥大小也仅不到 1 KB. 同时, BLAC 通过区块链和秘密共享技术实现分布式密钥管理, 相较于文献[13]具有更好的系统稳定性.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

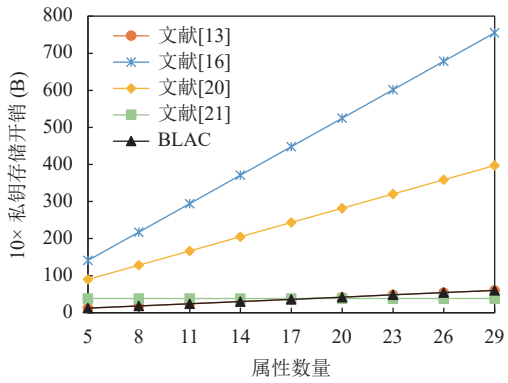

图 6 私钥存储开销

密文存储开销如图 7 所示. 文献[13,16,20]的密文存储开销相同, 数据线段重合. BLAC 在加密时应用椭圆曲线加密, 需要额外密文项$R_x = r_xG$对密文进行随机化处理. 而在同样使用椭圆曲线加密的文献[13]中, 缺失密文项$R_x = r_xG$, 安全性较低. 所有方案的密文开销均随着属性数量增加而增加, 当属性数量增加到 29 时, BLAC 密文存储开销也仅为 11 KB. 密文存储由 CS 承担, 存储能力完全可以满足 BLAC 需求.

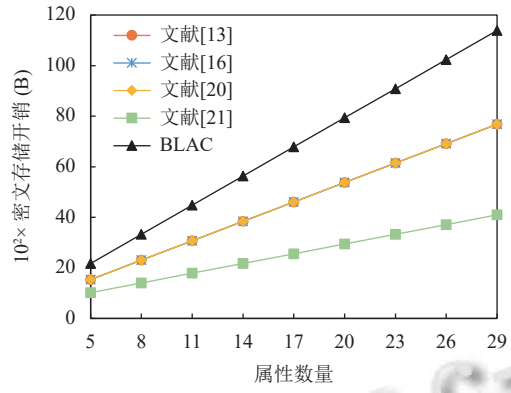

图 7 密文存储开销

5 结论与展望

针对现有的基于 CP-ABE 的访问控制方案不能有效满足边缘计算环境中关键数据安全问题, 本文提出一种基于区块链的轻量级密文访问控制方案 BLAC. 在 BLAC 中, 设计了一种基于椭圆曲线密码的轻量级 CP-ABE 算法, 采用快速的椭圆曲线标量乘法来实现算法加解密功能, 并通过虚拟属性和转换密钥将大部分加解密操作安全地转移到边缘服务器上, 使得用户设备能高效地完成数据的细粒度访问控制过程; 同时, 实现了一种基于区块链和秘密共享技术的分布式密钥管理方法, 使得边缘服务器能够协同地为用户分发私钥, 解决了现有 CP-ABE 方案存在的密钥托管问题并实现了实时灵活的属性撤销. 下一步将结合属性基加密研究边缘计算环境中基于区块链的轻量级海量数据的检索机制.

参考文献

1 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望. 计算机研究与发展, 2019, 56(1): 69–89. [doi: 10.7544/issn1000-1239.2019.20180760]

2 郑逢斌, 朱东伟, 臧文乾, 等. 边缘计算: 新型计算范式综述与应用研究. 计算机科学与探索, 2020, 14(4): 541–553. [doi: 10.3778/j.issn.1673-9418.1911042]

3 Kong XJ, Wu YH, Wang H, *et al*. Edge computing for internet of everything: A survey. IEEE Internet of Things Journal, 2022, 9(23): 23472–23485. [doi: 10.1109/JIOT.2022.3200431]

4 Khan LU, Yaqoob I, Tran NH, *et al*. Edge-computing-enabled smart cities: A comprehensive survey. IEEE Internet of Things Journal, 2020, 7(10): 10200–10232. [doi: 10.1109/JIOT.2020.2987070]

5 Oh SR, Seo YD, Lee E, *et al*. A comprehensive survey on security and privacy for electronic health data. International Journal of Environmental Research and Public Health, 2021, 18(18): 9668. [doi: 10.3390/ijerph18189668]

6 Rasori M, La Manna M, Perazzo P, *et al*. A survey on attribute-based encryption schemes suitable for the Internet of Things. IEEE Internet of Things Journal, 2022, 9(11): 8269–8290. [doi: 10.1109/JIOT.2022.3154039]

7 Kong LH, Tan JL, Huang JQ, *et al*. Edge-computing-driven Internet of Things: A survey. ACM Computing Surveys, 2022, 55(8): 174.

8 Zhang JL, Chen B, Zhao YC, *et al*. Data security and privacy-preserving in edge computing paradigm: Survey and open issues. IEEE Access, 2018, 6: 18209–18237. [doi: 10.1109/ACCESS.2018.2820162]

9 Hartmann M, Hashmi US, Imran A. Edge computing in smart health care systems: Review, challenges, and research directions. Transactions on Emerging Telecommunications Technologies, 2019, 33(3): e3710.

10 李晓伟, 陈本辉, 杨邓奇, 等. 边缘计算环境下安全协议综述. 计算机研究与发展, 2022, 59(4): 765–780. [doi: 10.7544/issn1000-1239.20210644]

11 Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. Proceedings of the 2007 IEEE

© 中国科学院软件研究所 http://www.c-s-a.org.cn

Symposium on Security and Privacy. Berkeley: IEEE, 2007. 321–334.
12 Oberko PSK, Obeng VHKS, Xiong H. A survey on multi-authority and decentralized attribute-based encryption. Journal of Ambient Intelligence and Humanized Computing, 2022, 13(1): 515–533. [doi: 10.1007/s12652-021-02915-5]
13 Das S, Namasudra S. Multiauthority CP-ABE-based access control model for IoT-enabled healthcare infrastructure. IEEE Transactions on Industrial Informatics, 2023, 19(1): 821–829. [doi: 10.1109/TII.2022.3167842]
14 Huang KQ. Multi-authority attribute-based encryption for resource-constrained users in edge computing. Proceedings of the 2019 International Conference on Information Technology and Computer Application (ITCA). Guangzhou: IEEE, 2019. 323–326.
15 Sammy F, Vigila SMC. An efficient blockchain based data access with modified hierarchical attribute access structure with CP-ABE using ECC scheme for patient health record. Security and Communication Networks, 2022, 2022: 8685273. [doi: 10.1155/2022/8685273]
16 Pu YW, Hu CQ, Deng SJ, *et al*. R^2PEDS: A recoverable and revocable privacy-preserving edge data sharing scheme. IEEE Internet of Things Journal, 2020, 7(9): 8077–8089.
17 Jiang Y, Xu XL, Xiao F. Attribute-based encryption with blockchain protection scheme for electronic health records. IEEE Transactions on Network and Service Management, 2022, 19(4): 3884–3895. [doi: 10.1109/TNSM.2022.3193707]
18 Zheng KF, Ding CY, Wang JC. A secure data-sharing scheme for privacy-preserving supporting node-edge-cloud collaborative computation. Electronics, 2023, 12(12): 2737. [doi: 10.3390/electronics12122737]
19 Yang YF, Shi RH, Li KC, *et al*. Multiple access control scheme for EHRs combining edge computing with smart contracts. Future Generation Computer Systems, 2022, 129: 453–463. [doi: 10.1016/j.future.2021.11.002]
20 Xie MD, Ruan YY, Hong HB, *et al*. A CP-ABE scheme based on multi-authority in hybrid clouds for mobile devices. Future Generation Computer Systems, 2021, 121: 114–122. [doi: 10.1016/j.future.2021.03.021]
21 Qin XM, Huang YF, Yang Z, *et al*. A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing. Journal of Systems Architecture, 2021, 112: 101854. [doi: 10.1016/j.sysarc.2020.101854]
22 Tu SS, Waqas M, Huang FM, *et al*. A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing. Computer Networks, 2021, 195: 108196. [doi: 10.1016/j.comnet.2021.108196]
23 Al-Dahhan RR, Shi Q, Lee GM, *et al*. Survey on revocation in ciphertext-policy attribute-based encryption. Sensors, 2019, 19(7): 1695. [doi: 10.3390/s19071695]
24 Li X, Liu T, Chen CY, *et al*. A lightweight and verifiable access control scheme with constant size ciphertext in edge-computing-assisted IoT. IEEE Internet of Things Journal, 2022, 9(19): 19227–19237. [doi: 10.1109/JIOT.2022.3165576]
25 Sahai A, Waters B. Fuzzy identity-based encryption. Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Aarhus: Springer, 2004. 457–473.
26 Chase M, Chow SSM. Improving privacy and security in multi-authority attribute-based encryption. Proceedings of the 16th ACM Conference on Computer and Communications Security. Chicago: ACM, 2009. 121–130.
27 Zhang YH, Deng RH, Xu SM, *et al*. Attribute-based encryption for cloud computing access control: A survey. ACM Computing Surveys, 2020, 53(4): 83.
28 Koblitz N. Elliptic curve cryptosystems. Mathematics of Computation, 1987, 48(177): 203–209. [doi: 10.1090/S0025-5718-1987-0866109-5]
29 Rajasekaran AS, Azees M, Al-Turjman F. A comprehensive survey on blockchain technology. Sustainable Energy Technologies and Assessments, 2022, 52: 102039. [doi: 10.1016/j.seta.2022.102039]
30 王利朋, 关志, 李青山, 等. 区块链数据安全服务综述. 软件学报, 2023, 34(1): 1–32. [doi: 10.13328/j.cnki.jos.006402]
31 佟兴, 张召, 金澈清, 等. 面向端边云协同架构的区块链技术综述. 计算机学报, 2021, 44(12): 2345–2366. [doi: 10.11897/SP.J.1016.2021.02345]

(校对责编: 牛欣悦)

© 中国科学院软件研究所 http://www.c-s-a.org.cn