计算机科学
COMPUTER SCIENCE

支持分层访问控制的弱中心化敏感数据共享方案

郑开发, 孙炜, 周俊旭, 吴云坤, 徐振, 刘志全, 何强

引用本文

郑开发, 孙炜, 周俊旭, 吴云坤, 徐振, 刘志全, 何强. 支持分层访问控制的弱中心化敏感数据共享方案[J]. 计算机科学, 2026, 53(2): 431-441.

ZHENG Kaifa, SUN Wei, ZHOU Junxu, WU Yunkun, XU Zhen, LIU Zhiquan , HE Qiang. Weakly-decentralized Scheme for Sensitive Data Sharing with Hierarchical Access Control [J]. Computer Science, 2026, 53(2): 431-441.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.250900047

支持分层访问控制的弱中心化敏感数据共享方案

郑开发[1] 孙 炜[2] 周俊旭[2] 吴云坤[5] 徐 振[3] 刘志全[2] 何 强[4]
1 北京航空航天大学网络空间安全学院 北京 100191
2 暨南大学网络空间安全学院 广州 510632
3 杭州芯云半导体集团有限公司 杭州 310052
4 东北大学计算科学与工程学院 沈阳 110169
5 奇安信科技集团股份有限公司 北京 100044
(zhengkaifa@buaa.edu.cn)

摘 要 在云边协同等分布式应用场景下,实现高效性、可检索性与弱中心化的细粒度访问控制是保障敏感数据安全共享的核心挑战。然而,传统方案存在高昂的计算开销、缺少密文检索功能和中心化架构固有的安全风险等问题。对此,提出一种支持分层访问控制的弱中心化敏感数据共享方案(HAC-SDS)。首先,通过云边端协同计算方式,将终端侧开销转移到云边侧,降低计算和存储开销。其次,通过构建加密的倒排索引,支持对云端文件进行快速、细粒度的检索,并结合属性撤销与动态更新机制,显著提升效率。最后,将区块链技术应用于密钥管理,通过其弱中心化的特性,从根本上消除传统中心化方案的单点瓶颈与信任风险。安全分析表明,密文的不可区分性有效保障了数据的机密性。实验结果表明,所提密文检索方案在实际应用中是高效可行的。

关键词:分层访问控制;敏感数据共享;云边端协同计算;密文检索;弱中心化
中图分类号 TP309

Weakly-decentralized Scheme for Sensitive Data Sharing with Hierarchical Access Control

ZHENG Kaifa[1],SUN Wei[2],ZHOU Junxu[2],WU Yunkun[5],XU Zhen[3],LIU Zhiquan[2] and HE Qiang[4]
1 School of Cyber Science and Technology,Beihang University,Beijing 100191,China
2 School of Cyber Security,Jinan University,Guangzhou 510632,China
3 Hangzhou Xinyun Semiconductor Group Co.,Ltd.,Hangzhou 310052,China
4 School of Computer Science and Engineering,Northeastern University,Shenyang 110169,China
5 Qi Anxin Technology Group Co.,Ltd.,Beijing 100044,China

Abstract In distributed application scenarios such as cloud-edge collaboration,achieving efficient,searchable,and decentralized fine-grained access control for sensitive data sharing presents a core challenge. Traditional schemes are often hindered by high computational overhead,a lack of ciphertext retrieval functionality,and the inherent security risks of centralized architectures. Therefore,this paper proposes a hierarchical access control scheme for sensitive data sharing in a semi-decentralized manner (HAC-SDS). Firstly,by employing a cloud-edge-device collaborative computing model,the scheme offloads significant computational and storage burdens from the client-side to cloud and edge servers,effectively reducing overhead. Secondly,an encrypted inverted index is constructed to support fast and fine-grained ciphertext retrieval,which is integrated with an attribute revocation and dynamic update mechanism to significantly enhance efficiency. Finally,blockchain technology is applied to key management,its decentralized nature fundamentally eliminates the single-point bottleneck and trust risks inherent in traditional centralized solutions. Security analysis demonstrates that the ciphertext achieves indistinguishability,thereby effectively guaranteeing data confidentiality. Experimental results confirm that the proposed ciphertext retrieval scheme is both efficient and practical for real-

到稿日期:2025-09-07 返修日期:2025-12-02
基金项目:国家重点研发计划(2022YFB3104900);国家自然科学基金面上项目(62272195);北京市高层次创新创业人才支持计划科技新星计划(20250484975);山东省自然科学基金(ZR2024MF084)
This work was supported by the National Key Research and Development Program of China(2022YFB3104900),General Program of the National Natural Science Foundation of China(62272195), Sponsored by Beijing Nova Program(20250484975) and Natural Science Foundation of Shandong Province(ZR2024MF084).
通信作者:何强(heqiang@bmie.neu.edu.cn)

world applications.

Keywords Hierarchical access control, Sensitive data sharing, Cloud-edge-end collaborative computing, Searchable encryption, Decentralization

1 引言

大数据与云计算时代[1]推动了数据向云平台迁移，但云和边缘节点的不可信性[2-3]带来了严峻的数据安全与隐私挑战[4]。密文策略属性基加密（Ciphertext-Policy Attribute-Based Encryption，CP-ABE）[5]通过将访问策略嵌入密文，成为实现敏感数据精细化访问控制的关键技术。然而，CP-ABE 的实际部署面临三大核心挑战。

首先是高昂的计算开销。CP-ABE 的加/解密，尤其是涉及大量配对运算的解密过程，对算力有限的终端设备构成严峻挑战，甚至催生了计算外包的研究方向[6]。其次是功能与效率的短板。传统方案的属性撤销机制效率低下[7]，且加密破坏了数据可检索性，而现有的可搜索 ABE 方案[8]尚不成熟。集成这两种功能会带来巨大的计算开销[9]，难以满足动态应用需求。最后是中心化架构固有的安全风险。传统方案依赖单一可信权威，存在单点故障和密钥托管问题。现有弱中心化方案[10]也未能完全消除单个权威不可信的风险。

为应对上述计算开销、密文检索和中心化信任挑战，本文提出一种支持文件分层访问控制的弱中心化敏感数据共享方案（HAC-SDS）。该方案构建了集区块链、云、边、端于一体的协同架构，通过任务卸载、加密索引和弱中心化密钥管理，旨在实现性能、功能与安全的全面提升。

本文的主要贡献和创新点如下：

1）采用云边端协同计算，降低计算负载。具体地，数据所有者保留完整密钥并上传加密后的数据至云端。在边缘节点承担部分计算，减轻数据所有者、数据使用者和云服务的计算负担。

2）支持文件分层访问控制的密文检索方案。数据所有者首先提取文件的关键词及文件标号构建倒排索引，并基于对称加密对文件主体加密。随后采用分层属性基加密对对称密钥和索引文件加密，支持动态策略更新。

3）弱中心化可消除单点故障和信任风险。通过基于区块链的密钥管理方案消除对中心化的依赖，将密钥的控制权交到用户手中，从而保证密钥的生成与管理过程的透明性和安全性。

为了实现以上目标，本文构建了一个包含区块链（Block Chain System，BCS）、数据所有者（Data Owner，DO）和数据使用者（Data User，DU）、边缘节点（Edge Node，EN）、云服务提供商（Content Security Policy，CSP）等组件的完整的系统架构。在此架构下，不仅实现了数据的精细化密文检索，还降低了整体的计算负载、消除可信权威机构的安全性被破坏而导致敏感数据存在安全隐患的风险。

2 相关工作

保障半可信云环境下的数据安全需兼顾机密性、检索与访问控制。针对传统属性基加密（Attribute-Based Encryption，ABE）存在的计算开销大与单点故障风险，本章从 ABE 演进、功能融合及弱中心化 3 个维度回顾相关工作。

在属性加密方面，从 Sahai 等[11]的 KP-ABE 到 Bethencourt 等[5]的 CP-ABE，奠定了该领域的基础。Wang 等[12]提出的 FH-CP-ABE 通过分层结构简化了策略，但存在权限继承的越权风险。为解决计算瓶颈问题，近期研究[13-14]转向云边协同外包架构，Xia 等[15]也在此背景下提出了面向信任的边缘存储机制以优化物联网数据的安全性与效率。这些工作共同显著减轻了终端负担。尽管如此，这些方案并未降低 ABE 算法的核心复杂度，且在追求复杂功能时可能引入新的安全隐患。

在密文检索与功能融合方面，相关技术从 Song 等[16]的基础 SE 演进到 Zheng 等[17]结合 ABE 实现细粒度搜索权限的 ABKS。近期，Huang 等[18]通过改进认证加密提升了搜索效率。随后，Fan 等[19]解决了云边端场景下的动态隐私保护问题。为进一步满足动态需求，高效的属性撤销[20]成为另一关键技术，并催生出从效率到后量子安全的广泛研究[21]。前沿方案，如 Liu 等[22]通过云边协同实现了高效的多关键词属性可搜索加密，Li 等[23]引入了隐藏访问策略与可验证机制以提升发布订阅服务的隐私性与可靠性，其他工作[24-25]趋向于集成外包撤销、多关键词搜索等多种功能。但此类融合方案普遍继承了底层高计算开销，且依赖“权威完全可信”这一过强假设，限制了方案的现实应用。

在弱中心化方面，为解决单点故障，研究从 Chase[26]的多权威概念演进到 Lewko 等[27]的完全去中心化方案。近年来，区块链被广泛用于增强系统鲁棒性[28-29]，实现了去中心化访问控制与高效撤销，如 Sasikumar 等[30]在 2024 年提出的块链辅助的层级属性加密方案。尽管如此，现有设计多侧重于防御权威间合谋，普遍假设单个权威诚实可靠，难以有效应对权威机构自身作恶或被攻破的安全威胁。

3 系统模型

3.1 数据共享模型

本方案构建了一个包含数据所有者（DO）、数据使用者（DU）、边缘节点（EN）、云服务提供商（CSP）和区块链系统（BCS）5 个实体的协同数据共享模型。该模型的核心思想是通过云、边、端协同计算，将计算密集型任务从资源受限的终端设备（DO 和 DU）卸载至边缘节点，从而在保障数据安全的前提下，实现高效、可扩展的数据共享。模型的运作流程可分为以下 3 个核心阶段。

1）系统初始化与密钥生成阶段

在该阶段，区块链系统（BCS）作为去中心化的信任根，负责生成并分发全局公共参数 pp。它取代了传统的中心化权威机构，通过链上合约为各属性节点和用户生成并管理各自的私钥组件。这种去中心化的密钥管理方式从根本上消除了单点故障和密钥托管风险，为整个数据共享体系奠定了透明、

安全且可信的基础。

2）数据加密与上传阶段

此阶段是一个由数据所有者（DO）与边缘节点（EN）协同完成的高效计算卸载过程。首先，DO 执行轻量级的本地加密，包括使用对称密钥加密文件、构建加密索引，并生成一个不含复杂访问策略的中间密文 CT。随后，DO 将此中间密文上传至 EN，由 EN 作为计算代理，完成将复杂分层访问树（τ）嵌入密文的重加密计算，生成最终密文 CT' 并将其存储至云服务提供商（CSP），从而将策略绑定的计算开销从资源受限的终端剥离。

3）数据检索与解密阶段

该阶段始于数据使用者（DU）生成一个针对目标关键词的、一次性的搜索陷门 $T\omega'$ 并提交给 CSP。CSP 执行密文检索后，将匹配到的加密文件 CT' 直接发送至边缘节点（EN）而非用户。EN 利用其强大的计算能力，代表 DU 执行与复杂访问树相关的、计算密集型的部分解密操作，并将一个轻量级的中间密文安全地返回给 DU。最终，DU 仅需进行简单的本地计算即可恢复明文，实现了高效且保护隐私的解密体验。

3.2 分层访问控制模型

为了实现对敏感数据的精细化和结构化访问控制，本方案提出了一种基于访问控制树（τ）的分层访问控制模型，将现实权限映射为密码学结构。其中，叶子节点代表如“角色＝董事长”的原子属性，而非叶子节点则定义了与特定数据类别（如“机密数据”）相关联的门限访问策略。访问权限的判定基于属性满足度，用户只有在持有的属性集合满足某节点的门限策略时，才能通过秘密共享机制重构出解密该节点关联数据所需的秘密值。该模型的设计天然地赋予了两个核心特性：一是权限继承，即对父节点的访问权限自动延伸至其所有子节点，极大地简化了策略管理；二是数据隔离，确保树的不同分支在没有独立授权的情况下相互不可见，从而将复杂的访问规则转换为一个直观、安全且高效的加密控制体系。

3.3 安全模型

本方案根据实体的能力和环境，假设敌手 Adv 和挑战者 Chal 游戏，建立安全模型。

选择明文攻击（CPA）：敌手可选择任意明文并获取其对应的密文，目标是破解加密系统。

定义 1 若无多项式时间的敌手 Adv 能以显著优势赢得以下游戏，则系统满足 CPA 安全（基于 DBDH 假设）。

准备：挑战者（Chal）运行初始化算法，生成主密钥 msk 和公共参数 pp，并将 pp 给予敌手。

阶段 1 Adv 向 Chal 查询任意内容密钥 ck，并获得其对应密文 CT。

挑战：Adv 提交两个等长的内容密钥（ck，ck），Chal 随机选择一个比特 $b\in\{0,1\}$，加密 ck 生成挑战密文 CT^*，并返回给 Adv。

查询阶段 2 敌手重复阶段 1 的查询，但敌手获取的私钥 SK 不满足挑战访问结构 $\mathbb{T}^*$。

猜测 Adv 输出对 b 的猜测 b'。若 $b'=b$，则 Adv 获胜。

攻击者 A 获胜的优势被定义为：

$$Adv(A)=\left|Pr[b'=b]-\frac{1}{2}\right| \tag{3}$$

3.4 设计目标

本文的设计目标主要为以下几点：

1）功能性目标：方案应在实现密文策略属性基加密的基础上，集成高效的密文检索功能，支持用户根据层级访问策略对加密数据进行细粒度访问和关键词搜索。此外，系统需支持属性撤销与动态更新机制，确保当用户权限发生变更时，系统能及时更新密钥与密文，保障数据共享的时效性与灵活性。

2）安全性目标：方案需满足选择明文攻击（CPA）下的不可区分性，确保数据在云端和边缘端的机密性。并且需保证索引与搜索陷门的隐私性，防止不可信的云服务商推断出关键词信息。更重要的是，利用区块链技术构建弱中心化的密钥管理机制，消除传统方案中完全依赖单一可信授权中心带来的单点故障与密钥托管风险。

3）性能目标：针对资源受限的终端设备，方案应通过云边端协同架构，将复杂的密码学计算从终端卸载至边缘节点，显著降低终端的计算与存储开销。系统设计应追求高效性，使得加密、生成陷门等操作的计算复杂度与属性数量或关键词数量尽可能趋向于常数级开销，以适应大规模数据共享场景。

4 算法设计

4.1 系统架构

本方案的系统模型如图 1 所示。系统模型由 5 个实体组成：区块链（BCS）、数据所有者（DO）和数据使用者（DU）、边缘节点（EN）、云服务提供商（CSP）。

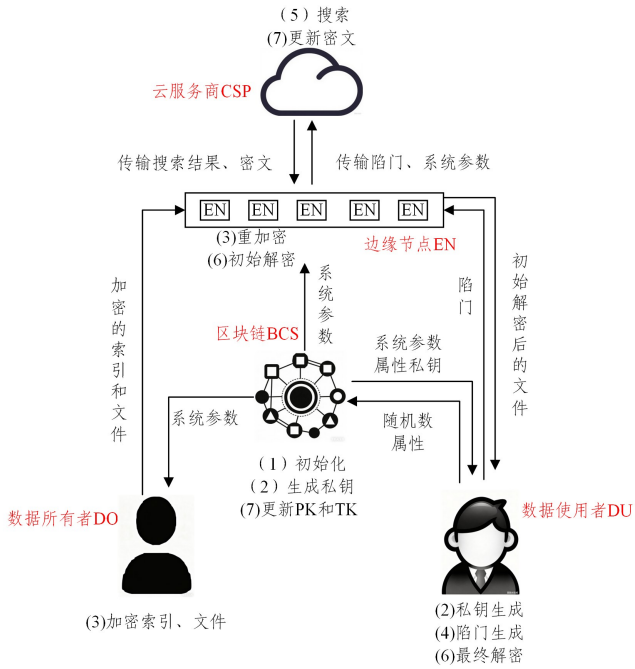

图 1 系统架构

Fig. 1 System architecture

1）区块链（BCS）：负责系统初始化与弱中心化密钥管理。创世节点负责初始化密钥管理，生成参数，通过密钥分片技术将主密钥托管至各属性节点，随后离线以消除单点信任风险。属性节点生成用户部分私钥及自身私钥。

2)数据拥有者(DO):数据的创建者与策略定义者。其负责加密文件、定义访问策略,并生成密文和索引上传至 EN。

3)数据使用者(DU):合法的数据请求者。其负责向 CSP 请求数据,接收满足其属性的密文,并用私钥解密恢复文件。

4)边缘节点(EN):不可信的分布式辅助节点。负责卸载部分计算任务并缓存加密数据以降低系统负载,无法访问明文。

5)云服务提供者(CSP):半可信的存储与检索实体。其负责大规模存储加密数据,并根据用户陷门执行加密搜索、返回密文。

4.2 符号说明

表 1 列出了本方案中用到的符号并给出了定义描述。

表 1 相关符号说明

Table 1 Explanation of related symbols

符号	定义
MSK_c	系统主密钥
PSK_i	属性节点私钥
SK	用户主密钥
S	用户属性集
CT_{record}	文件密文
CT_{index}	索引密文
CT_{key}	密钥密文
CT	总密文
CT'	重加密密文
$T_{\omega'}$	关键词ω'对应的搜索陷门

4.3 算法形式化定义

本方案由 7 个部分组成,分别为系统建立、私钥生成、加密阶段、陷门生成、搜索阶段、解密阶段、属性更新阶段。其流程如图 2 所示。

1)系统建立:$Setup(\lambda)\rightarrow(pp,PSK_i)$。该算法由区块链中的创世节点执行。输入安全参数$\lambda$,输出系统公共参数 pp、各属性节点的私钥 PSK_i。

2)私钥生成:$KeyGen(pp,PSK_i,S,z_u)\rightarrow SK_u$。该算法由属性节点与数据使用者(DU)协同执行。输入系统公共参数 PK、用户属性集 S 和数据使用者选择的随机数z_u,输出 DU 的私钥 SK_u。

3)加密阶段:$Encrypt(pp,M,W,\tau)\rightarrow CT$。该算法由数据拥有者(DO)与边缘节点(EN)共同完成。输入系统公共参数 pp、文件集合 M、关键词集合 W 以及访问控制树τ,输出加密后的数据和索引 。

4)陷门生成:$Trap(pp,\omega')\rightarrow T_{\omega'}$。该算法由数据使用者(DU)执行。输入系统公共参数 PP 和查询关键词ω',输出对应关键词ω'的陷门$T_{\omega'}$。

5)搜索阶段:$Search(pp,CT,SK_u,S,T_{\omega'})\rightarrow(C)$。该算法由 CSP 执行。输入属性集 S 和陷门$T_{\omega'}$,检索是否匹配陷门,如果匹配则返回对应的加密文件。

6)解密阶段。$Decrypt(pp,SK_u,CT,S)\rightarrow m_{\omega'}$。该算法由边缘节点(EN)与数据使用者(DU)协同执行。输入系统公共参数 pp、用户私钥 SK_u、密文 CT 以及用户属性集 S,输出明文文件$m_{\omega'}$;若属性集不满足访问策略,则输出$\perp$。

7)属性更新阶段:$Revoke(pp,SK,\chi)\rightarrow(PK^*,SK^*)$。该算法由区块链(BCS)和云服务(CSP)共同执行。当属性需要更新时,创世节点更新 PK 和计算 TK。之后,云服务(CSP)更新密文 CT'。

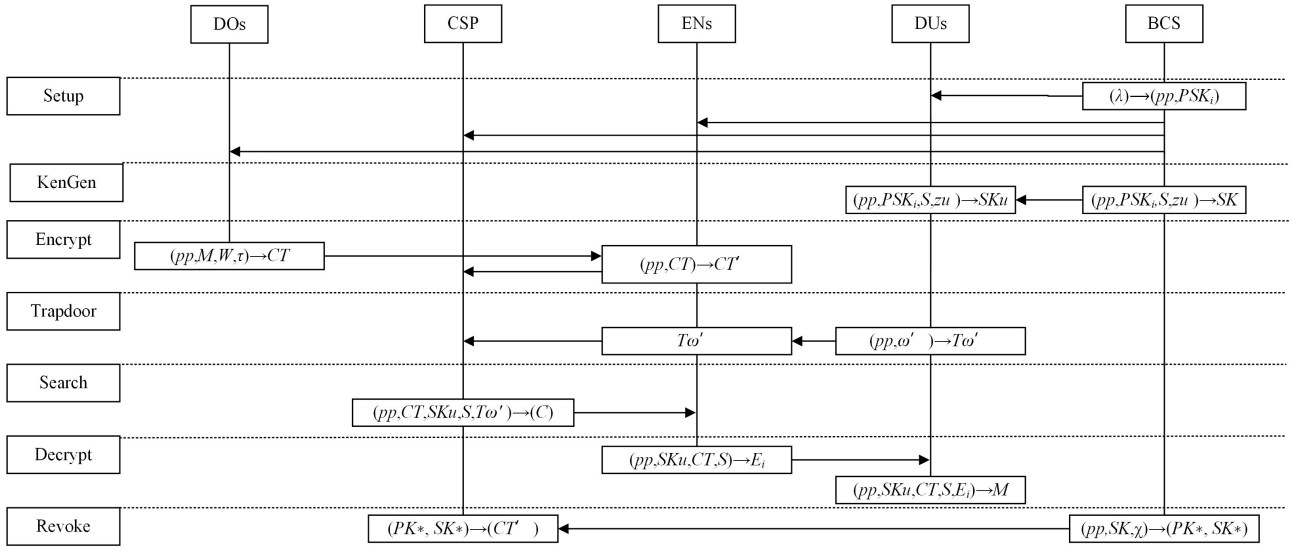

图 2 HAC-SDS 总体交互图

Fig. 2 Overall interaction diagram of HAC-SDS

5 方案构造

本文方案由以下 7 个算法构成,分别为 $Setup$,$KeyGen$,$Encrypt$,$Trap$,$Search$,$Decrypt$,$Revoke$。详细描述如下。

5.1 初始化阶段 $Setup(\lambda)\rightarrow(pp,PSK_i)$

该阶段由区块链的创世节点执行,生成系统参数和主密钥。初始化完成后,创世节点将离线。具体过程如下。

1)首先,创世节点挑选一个阶为 p 的乘法循环群 G,其中 g 为群 G 的生成元,$e(g,g)$表示双线性映射。然后,选择两个哈希函数$H_0:\{0,1\}^*\rightarrow\mathbb{Z}_p^*$,$H_1:\{0,1\}^*\rightarrow G$,接着选择 6 个随机元素$(\alpha,\beta,a,b,c,d)\in\mathbb{R}_p^*$,其中$\{\alpha,\beta\}$作为系统的主密钥。拉格朗日系数为$\Delta_{j,S_x'}(x)=\prod_{m\in S_x',m\neq j}\frac{x-m}{j-m}$。令 N 个属性节点表示为 $P=\{P_1,P_2,\cdots,P_N\}$。最后执行算法,输出公共参数,

如式(4)所示：

$$pp=\{G,g,g^a,e(g,g),g^a,g^bg^c,g^d,H_0,H_1,h_1\cdots,h_u\} \tag{4}$$

2)为每个属性节点生成私钥。首先，将主密钥的片段 α 分发给参与者 P_i，使用 $Shamir(m,n)$——门限秘密共享方案。创世节点随机选择一个度为 $m-1$ 的多项式 $f(x)=t+t_1x+t_2x^2+\cdots+t_{m-1}x^{m-1}(\mathrm{mod}\ p)$，使得 $\alpha=t$ 和 $t_1,t_2,\cdots,t_{m-1}$ 为某个有限域中的随机元素。创世节点计算并分配 α_i 给属性节点 P_i。然后，对于每个属性 j，算法随机选择 $r_j\in Z_p^*$，并为属性生成随机数集 $R=\{r_1,r_2,\cdots,r_N\}$。

3)创世节点为属性节点 P 构造私钥：计算 k，并利用 $f(i)$ 和随机数集 R 构造属性节点的私钥。

$$PSK_i=\{k,f(i),R\} \tag{5}$$

5.2 私钥生成 $KeyGen(pp,PSK_i,S,z_u)\rightarrow SK_u$

该阶段由属性节点执行该算法。将数据使用者 u 的属性集 S 以及数据使用者 u 选取的随机数 z_u 作为参数输入系统中。以下为私钥的生成过程。

1)首先数据使用者 DU 选择随机数 $z_u\in Z_p$，之后 DU 选择一个度为 $q-1$ 的多项式 $h(x)=u+u_1x+u_2x^2+\cdots+u_{q-1}x^{q-1}(\mathrm{mod}\ p)$ 使得 $u=z_u$ 和 $u_1,u_2,\cdots,u_{q-1}$ 为某个有限域中的随机元素。然后 DU 计算 $z_1=h(1),z_2=h(2),\cdots,z_n=h(N)$，之后数据使用者秘密地分发 z_u 和 g^{z_u} 给属性节点 P_i。

2)属性节点生成部分 DU 的私钥。属性节点 P_i 使用私钥 PSK_i 计算 DU 的私钥的片段信息。

$$\begin{cases}D_{u,i1}=k^{f(i)}\\ D_{u,i2}=k^{h(i)}\end{cases} \tag{6}$$

对于属性 $att_v^u\in S$，计算属性子密钥组件：

$$\begin{cases}D_{j,h_{x_j}}=g^{z_ur_jH_1(h_i)}\\ D'_{j,h_{x_j}}=g^{r_j}\end{cases} \tag{7}$$

最后，计算：

$$T=\prod_{i=1}^{n}e(g,g)^{f(i)} \tag{8}$$

3)数据使用者 DU 生成最终密钥。该部分由数据使用者运行，当数据使用者 DU 构造他们自己的密钥时，使用拉格朗日插值方法的参数，按照式(9)进行密钥构建：

$$\begin{cases}D_{u1}=\prod_{i=1}^{N}D_{u,i1}^{\Delta_{i,S(0)}}\\ D_{u2}=\prod_{i=1}^{N}D_{u,i2}^{\Delta_{i,S(0)}}\end{cases} \tag{9}$$

并且选取随机数 $\upsilon\in\mathbb{Z}_p^*$，构成 $D_w=g^{\upsilon(a-z_u)}$。

最后，完整的数据使用者 DU 的密钥如式(10)所示：

$$SK_u=\begin{Bmatrix}\{D_{u1},D_{u2}\},D_w,\upsilon,\\ (D_{j,att^u})^{b\upsilon}=(g^{z_ur_jH_1(h_i)})^{b\upsilon}\\ D'_{j,att^u}=g^{\upsilon r_j}\end{Bmatrix} \tag{10}$$

5.3 加密阶段 $Encrypt(pp,M,W,\tau)\rightarrow CT$

本方案使用分层访问策略。根据一棵访问控制树 τ，每个叶子节点对应一个属性，树中的非叶子节点与某个敏感文件相关联，每棵子树都表示一个访问控制策略。

1)数据所有者 DO 方的加密

(1)首先，对文件进行加密。根据给定的文件集 $M=\{M_1,\cdots,M_k\}$(每个文件都具有不同的访问层级)，DO 选定对称密钥集合 $ck=\{ck_1,\cdots,ck_k\}$。这里采用对称加密的加密形式生成密文集合。

$$CT_{\mathrm{record}}=\{Enc_{ck_1}(M_1),Enc_{ck_2}(M_2),\cdots,Enc_{ck_k}(M_k)\} \tag{11}$$

(2)其次，对索引进行加密。DO 根据文件集 $M=\{M_1,\cdots,M_k\}$ 以及关键词集 W 构建如下形式的倒排索引。对某个确定的关键词 $\omega\in W$，可以构建如式(12)所示的公式：

$$index(\text{索引})=\{(\omega_1,\{M_1\}),(\omega_2,\{M_1,M_2\}),\cdots|\omega\in W\} \tag{12}$$

DO 对某个确定的关键词 $\omega\in W$，计算 $\varphi=g^{ab}$。选取秘密值 $s\in\mathbb{Z}_p$ 得到秘密值集合 $s_i=\{s_1,s_2,\cdots,s_k\}$，之后遍历文件。如果文档 M_i 包含关键词 ω，则计算 $t_i=g^{bH_0(w)+ts_i}$，$t_i'=g^{s_i}$，否则设置 $t_i=1,t_i'=1$。故索引密文如式(13)所示：

$$CT_{\mathrm{index}}=\{(t_1,t_1'),\cdots,(t_k,t_k')\} \tag{13}$$

(3)最后，数据拥有者 DO 在树 τ 中，数据所有者从根节点向下设定 k 个层次节点，分别对应 $\{ck_1;\cdots;ck_k\}$。之后使用 $s_i=\{s_1,s_2,\cdots,s_k\}$ 对所有的层次节点计算 $\widetilde{C}_i=ck_iT^{s_i}$ 和 $C_i'=g^{s_i}$，得到密钥密文公式，如式(14)所示：

$$CT_{\mathrm{key}}=\{\tau,\widetilde{C}_i,C_i'\} \tag{14}$$

DO 将 3 部分密文合并为 CT 后，将预密文 CT 发送给边缘节点。

$$CT=\{CT_{\mathrm{record}},CT_{\mathrm{index}},CT_{\mathrm{key}}\} \tag{15}$$

2)边缘节点 EN 方加密

EN 节点随机选取一个秘密值 s_i'。从根节点开始，为访问控制树 τ 上的每个节点以自上而下的方式选取一个多项式。对树中的每个节点 x，多项式的最高次数 $l=n-1$，其中 n 为门限值。从根节点开始，数据所有者设置 $q_1(0)=s_1'$，并选择 l_1 个剩余节点对其进行完整定义。为了定义根节点的多项式 q_1，DO 首先设置 $q_1(0)=s_1'$，然后选择 l_1 个随机元素实现完整定义。

检查是否为访问树的叶子节点。如果 x 是非叶子节点，则设置 $q_x(0)=s_x'=q_{\mathrm{parent}(x)}(index(x))$，同时选择 l_x 个随机元素进行完整定义：$q(x)=s_x'+a_1x+\cdots+a_{d_x}x_{d_x}$。

若 X 是访问树的非叶子节点集，令传输节点 x 的集合为 X，传输节点 x 的子节点的门限集合为 $TN-CT(x)=\{ch_1,\cdots,ch_i,\cdots\}$。则对每个节点进行计算，如式(16)所示：

$$\widehat{C_{x,j}}=g^{\frac{cd}{q_{ch_j}(0)}},\widehat{C_{x,j}}'=g^{ds_i'} \tag{16}$$

假设 Y 是访问树的叶子节点集，则对于每个 $y\in Y$，计算 $C_y=g^{dq_y(0)},C_y'=g^{dH_1(h_i)}$。

最终，对密钥加密后的密文如式(17)所示：

$$CT'=\begin{Bmatrix}CT_{\mathrm{record}},CT_{\mathrm{index}},CT_{\mathrm{key}},C_j'=g^{\frac{s_j'}{s_j}}\\ \{\widehat{C_{x,j}}=g^{\frac{cd}{q_{ch_j}(0)}},\widehat{C_{x,j}}'=g^{\frac{s_i'}{cd}}\}_{\forall x\in X}\\ \{C_y=g^{dq_y(0)},C_y'=g^{dH_1(h_i)}\}_{\forall y\in Y}\end{Bmatrix} \tag{17}$$

最后，边缘节点 EN 将密文 CT' 发送给云服务提供者 CSP。

5.4 陷门生成 $Trap(pp,SK_u,\omega')\rightarrow T_{\omega'}$

当 DU 想访问包含其想要查询的关键词 ω' 的加密文件时，首先需要根据关键词 ω' 和自身属性集生成陷门 $T_{\omega'}$，之后

将他们发送给云服务器 CSP。具体步骤为，DU 首先生成一个随机元素 $t\in\mathbb{Z}_p^*$，之后分别计算 $T_1=D_w$，$T_2=g^{bH_0(w')}$，$T_3=g^{ab}$，陷门构成如式(18)所示：

$$T_{w'}=\left\{\begin{matrix}T_1,T_2,T_3,\forall j\in S,\\ \overset{\wedge}{D_j}=((D_{j,h_{x_j}})^{bv})^t,\overset{\wedge}{D_j}'=(D'_{j,h_{x_j}})^t\end{matrix}\right\} \tag{18}$$

5.5 搜索阶段 $Search(pp,CT,SK_u,S,T_{w'})\rightarrow(C)$

当收到属性集 S 和陷门 $T_{w'}$ 后，CSP 首先会检查此属性集最高可以匹配到哪一层的访问结构，如果有匹配的层的访问结构，则会执行递归算法 $DecryptNode(CT,SK_u,x)$。然后，CSP 检查索引是否匹配陷门，如果匹配则返回对应的加密文件。具体方案如下。

首先，定义递归算法 $DecryptNode(CT,SK_u,x)$。

如果 x 是树 τ 中的叶子节点，令 i 为 x 的属性，并定义 $DecryptNode(CT,SK_u,x)$ 如下：

如果 $i\notin S$，则 $DecryptNode(CT,SK_u,x)=\perp$；

如果 $i\in S$，则

$$\begin{aligned}DecryptNode(CT,SK_u,x)&=\frac{e(\overset{\wedge}{D_i},C_x)}{e(\overset{\wedge}{D_i}',C_x')}\\&=\frac{e((g^{r_jbvz_uH_1(h_i)})^t,g^{dq_x(0)})}{e((g^{vr_j})^t,g^{dH_1(h_i)})}\\&=e(g,g)^{bz_uq_x(0)}\end{aligned} \tag{19}$$

如果 x 是树 τ 的非叶子节点，那么对于每一个子节点 n，都调用递归算法 $Decrypt(CT,SK,n)$。当 x 的子节点不在集合 S 中时，输出 $F_n=\perp$；否则，反复执行递归算法，并且设置 $i=index(n)$，$S_x'=\{index(n):n\in S_x\}$。

$$\begin{aligned}F_x&=\prod_{n\in S_x}F_n^{\Delta_{i,S_x}(0)}\\&=\prod_{n\in S_X}(e(g,g)^{bz_uq_y(0)})^{\Delta_{i,S_x}(0)}\\&=e(g,g)^{bz_uq_x(0)}\end{aligned} \tag{20}$$

最后，通过式(21)查找陷门与索引是否匹配，如果匹配则返回对应密文，否则输出 $\perp$。

$$e(t_i,T_3)=e(\varphi,T_2)\frac{F_{(x_i)}}{e(C_i',g)}e(T_1,t_i') \tag{21}$$

5.6 解密阶段 $Decrypt(pp,SK_u,CT,S)\rightarrow m_{\omega'}$

1)边缘节点 EN 侧的解密

如果 DU 的属性集合满足 τ 的第 i 层级访问控制权限，则可以继续往下恢复出 $e(g,g)^{bz_uq_i(0)}$ $(i\subseteq[1;k])$，具体如式(22)所示：

$$\begin{aligned}A_i&=DecryptNode(PK,CT,SK_u,S)\\&=e(g,g)^{bz_uq_i(0)}\\&=e(g,g)^{bz_us_i'}\end{aligned} \tag{22}$$

通过式(23)计算 E_i：

$$E_i=\frac{A_i}{e(g,C_i)}=\frac{e(g,g)^{bz_us_i'}}{e(g,g^{\frac{s_i'}{s_i}})}=e(g,g)^{bz_us_i} \tag{23}$$

之后 EN 将结果 E_i 传输给数据使用者 DU。

2)数据使用者 DU 侧的解密

在得到 E_i 后，首先计算 D_{u1}，D_{u2} 后，再计算 E_i'。

$$D_{u1}=k^a,D_{u2}=k^{z_u} \tag{24}$$

$$e(E_i',g)=\frac{E_i}{e(g,g)^b}=\frac{e(g,g)^{bz_us_i}}{e(g,g)^b}=e(g,g)^{z_us_i}$$

$$E_i'=g^{z_us_i} \tag{25}$$

通过式(26)和式(27)计算得出密钥：

$$F_i=\frac{e(E_i',D_{u1})}{e(g,D_{u2})}=\frac{e(g^{z_us_i},g^{\beta a})}{e(g,g^{\beta z_u})}=e(g,g)^{as_i} \tag{26}$$

$$\frac{\widetilde{C}_i}{F_i}=\frac{ck_iT^{s_i}}{e(g,g)^{as_i}}=ck_i,i\in[1,k] \tag{27}$$

数据使用者 DU 使用密钥 ck_i 就可以解密得到对应的文件 m_i。

3)解密低层次文件

如果 DU 想查看其属性集合 S 的低层次的授权节点的文件，则可以利用传输节点的相关密文子项，通过下式获得其有关的低层次的授权节点的值。

(1)首先边缘节点计算 $\widetilde{C}_{x,j}$，之后边缘节点将 $\widetilde{C}_{x,j}$ 传输给 DU。

$$e(\widetilde{C}_{x,j},g)=e(\widehat{C_{x,j}}',\frac{1}{C_i'})=e(g^{\frac{s_i'}{cd}},g^{\frac{s_i}{s_i'}})=e(g^{\frac{s_i}{cd}},g) \tag{28}$$

(2)DU 通过式(29)计算得出低层次的秘密值。

$$\begin{aligned}F_{i+1,j}&=\frac{F_i}{e(\widehat{C_{x,j}},\widetilde{C}_{x,j})}\\&=\frac{e(g,g)^{as_i}}{e(g^{\frac{cd}{q_{dh_j}(0)}},g^{\frac{s_i}{cd}})}\\&=e(g,g)^{aq_{ckj}(0)}\end{aligned} \tag{29}$$

依此类推，可以计算出 $e(g,g)^{as_i}$，…，$e(g,g)^{as_k}$，之后使用式(27)得出对应文件密钥 ck，解密出对应文件。

5.7 属性更新 $Revoke(pp,SK,\chi)\rightarrow(PK^*,SK^*)$

当某些属性 χ 被撤销时，系统通过更新公钥、属性密钥来更新密文，来确保拥有该属性的用户无法继续解密相关密文，具体步骤如下。

1)创世节点更新 PK、计算 TK：假设被撤销的属性集是 χ，当 DU_j 的某个属性 $x_1\in\chi$ 被撤销时，创世节点将属性值 h_{x_1} 更新为新版本 $h_{x_1'}$。同时，为仍具有属性 x_1' 的 DU 计算属性密钥，如式(30)所示：

$$PP^*=\{\cdots,h_{x_1'},\forall x_1'\in\chi\} \tag{30}$$

BCS 计算转换密钥 TK_i：

$$TK_i=\left\{\frac{g^{H_1(h_1')}}{g^{H_1(h_1)}}\right\}=g^{H_1(h_1')-H_1(h_1)},\forall h_1'\in\chi \tag{31}$$

2)CSP 更新密文：在接收到变换密钥 TK_i 后，CSP 按如下方式更新密文：

$$\begin{aligned}C_y''&=C_y'\cdot TK_i\\&=g^{dH_1(h_i)}\cdot g^{H_1(h_1')-H_1(h_i)}\\&=g^{dH_1(h_i')}\end{aligned} \tag{32}$$

输出的更新的最终密文如下：

$$CT'=\left\{\begin{matrix}CT_{\text{record}},CT_{\text{index}},CT_{\text{key}},C_j'=g^{\frac{s_j'}{s_j}}\\ \{\widehat{C_{x,j}}=g^{\frac{cd}{q_{ch_j}(0)}},\widehat{C_{x,j}}'=g^{\frac{s_i'}{cd}}\}_{\forall x\in X}\\ \{C_y=g^{dq_y(0)},C_y'=g^{dH_1(h_i')}\}_{\forall y\in Y}\end{matrix}\right\} \tag{33}$$

6 安全性分析

6.1 安全性证明

定理 1 在 DBDH 问题是困难的这一假设下，任何概率

多项式时间敌手能够成功实施选择性攻击、攻破本文方案的概率是可忽略的。

在本文方案的选择性安全游戏中，假设存在敌手$\mathcal{A}$有不可忽视的优势ϵ攻破本文方案的构造。构造一个模拟器$\mathcal{B}$，它能够以$\epsilon/2$的优势解决DBDH困难问题。$\mathbb{G}_0$的一个生成元为g，双线性映射为$e:\mathbb{G}_0\times\mathbb{G}_0\to\mathbb{G}_T$。

证明 挑战者C随机选择阶为p的群$\mathbb{G}_0$和$\mathbb{G}_T$，双线性映射：$e:\mathbb{G}_0\times\mathbb{G}_0\to\mathbb{G}_T$。随机选择$\alpha,\beta,c,d\in\mathbb{Z}_p$，$d\in\{0,1\}$，$\langle A,B,C\rangle=\langle g^a,g^b,g^c\rangle$，生成元$g\in\mathbb{G}_0$以及一个随机参数$R$，$R'\in\mathbb{G}_T$。若$b=0$，$C$设$K=e(g,g)^{abc}$，$K'=e(g,g)^{abc'}$，否则$K=R$，$K'=R'$。为更清晰地描述安全性证明，以下假设只加密两个层次文件。

准备 敌手$\mathcal{A}$向模拟器$\mathcal{B}$提交一个挑战访问结构$\mathbb{T}^*$，并将$\mathbb{T}^*$发送给挑战者C。

初始化 模拟器$\mathcal{B}$选择随机数$\alpha,\beta,a,b,c,d\in\mathbb{Z}_p$。模拟器$\mathcal{G}$计算$g^a,g^\beta,g^a,g^b,g^c,g^d,e(g,g)$。同时，设置$H_0,H_1$。模拟器$\mathcal{B}$将$PK$和用户属性节点私钥$PSK$发送给敌手$\mathcal{A}$。

查询阶段1 敌手$\mathcal{A}$向模拟器$\mathcal{B}$发送请求，$\mathcal{B}$响应$\mathcal{A}$的查询。

密钥生成阶段：$\mathcal{B}$执行密钥生成算法生成用户私钥。设损坏的节点为$\sigma=\{P_1,P_2,\cdots,P_{t-1}\}$。用户将$z_u$发给属性节点$P_i\in P_i$，$P_i$生成部分用户私钥$D_{u,i1},D_{u,i2},D_j^u,D_j^{u\prime},T$。然后，用户生成最终的私钥：

$$\begin{Bmatrix}\{D_{u1},D_{u2}\},D_w,a,b,\upsilon,\\ D_{j,h_{x_j}}=g^{b\upsilon z_u r_j H_1(h_j)}\\ D'_{j,h_{x_j}}=g^{wr_j}\end{Bmatrix}\tag{34}$$

最后，挑战者C将用户私钥发给攻击者$\mathcal{A}$，并将加密的文件和关键词一起发给$\mathcal{A}$，之后$\mathcal{A}$执行陷门生成算法创建搜索令牌。

陷门生成阶段：$\mathcal{A}$选取任意一个关键词ω'，挑战者C执行算法生成陷门，并发给攻击者$\mathcal{A}$，其陷门为$T_1=D_w$，$T_2=g^{bH_0(w')}(\omega')$，$T_3=g^{ab}$。

挑战：敌手$\mathcal{A}$向模拟器$\mathcal{B}$提交两个等长的消息$\mathcal{M}_0$和$\mathcal{M}_1$。模拟器$\mathcal{B}$将$\mathcal{M}_0,\mathcal{M}_1$提交给挑战者$C$。挑战者$C$选择一个随机值$b\in\{0,1\}$，使用挑战的访问结构加密$\mathcal{M}_b$，产生的密文子项为：$C_i'=g^{s_i}=g^c=C$，$\widetilde{C}_i=\mathcal{M}_b e(g,g)^{as_i}=\mathcal{M}_b e(g,g)^{ac}=\mathcal{M}_b\cdot Ke(g,g)^{a'c}$，$\forall y\in Leaf_{\mathbb{T}^*}:C_y,C_y'$。对于$\forall ch_j\in TN-CT(x)$，模拟器$\mathcal{B}$随机选择$\phi_j\in\mathbb{Z}_p$，计算$\bar{C}=\frac{\widetilde{C}_i}{\mathcal{M}_0}$，$\widehat{C_{x,j}}=e(g,g)^{ac'}\cdot e(g,g)^{aH_2(\bar{C})}\cdot e(g,g)^{\phi_j H_2(\bar{C})}=e(g,g)^{a'c'}\cdot K'\cdot e(g,g)^{a'+ab+\phi_j H_2(\bar{C})}$，$\widetilde{C_{x,j}}=g^{a'+ab+\phi_j}$。模拟器$\mathcal{B}$将$CT^*=\{\mathbb{T}^*,\widetilde{C}_i,C_i',C_y,C_y',\widehat{C_{x,j}},\widetilde{C_{x,j}}\}$发送给敌手$\mathcal{A}$。

查询阶段2 敌手$\mathcal{A}$重复阶段1的查询，但是敌手$\mathcal{A}$获取的私钥SK不满足挑战访问结构$\mathbb{T}^*$。

猜测 敌手$\mathcal{A}$输出b的猜想为b'，那么模拟器$\mathcal{B}$输出的猜测也是b'。当$b=0$时，模拟器$\mathcal{B}$给出了一个准确的模拟，即$K=e(g,g)^{abc}$，$K'=e(g,g)^{abc'}$，则可计算$\bar{C}=\frac{\widetilde{C}_i}{\mathcal{M}_0}$，$\widehat{C_{x,j}}=e(g,g)^{ac'}\cdot e(g,g)^{aH_2(\bar{C})}\cdot e(g,g)^{\phi_j H_2(\bar{C})}=e(g,g)^{a'c'}\cdot K'\cdot e(g,g)^{a'+ab+\phi_j H_2(\bar{C})}$，$\widetilde{C_{x,j}}=g^{a'+ab+\phi_j H_2(\bar{C})}$。此时，输出即为本文方案的完整密文。当$b=1$时，表示$T,T'$为$\mathbb{G}_T$群中的随机元素，则消息$\mathcal{M}_b$对敌手$\mathcal{A}$完全隐藏，因此无法计算出$\bar{C}$和$\widehat{C_{x,j}}$。敌手$\mathcal{A}$在此安全游戏中的优势如下：

$$\frac{1}{2}Pr[b'=b|b=0]+\frac{1}{2}Pr[b'=b|b=1]-\frac{1}{2}=$$

$$\frac{1}{2}\left(\frac{1}{2}+\epsilon\right)+\frac{1}{2}\cdot\frac{1}{2}-\frac{1}{2}=\frac{1}{2}\tag{35}$$

通过以上描述，不存在概率多项式时间敌手$\mathcal{A}$以不可忽略的优势攻破本文方案的构造，因此本文方案在DBDH假设下能达到CPA安全。

6.2 安全性分析

本文旨在解决FH-CP-ABE方案中的安全漏洞——其密文子项$\widehat{C_{(x,y),j}}=e(g,g)^{\alpha(q_{(x,y)}(0)+q_{child_j}(0)}$的构造导致层次节点的秘密值$q_{child_j}(0)$可被计算得到，进而危及整个系统。同时，也解决了传统加密搜索中存在的访问控制、效率和关键词隐私泄露问题。

本文方案引入了边缘节点(EN)的动态加密与分层解密机制。在加密阶段，核心改进是边缘节点通过引入秘密多项式，将密文子项重构为$\widehat{C_{x,j}}=g^{\frac{cd}{q_{ch_j}(0)}}$，$\widehat{C'_{x,j}}=g^{ds_i'}$，从根本上阻止了攻击者对层次秘密的直接恢复。解密时，边缘节点仅允许合法用户通过匹配属性和陷门(如$T_1=D_w$，$T_2=g^{bH_0(w')}(\omega')$，$T_3=g^{ab}$)执行递归算法，逐层恢复对称密钥$e(g,g)^{\alpha s_i}$，而非匹配路径则被阻断。这种设计结合访问树的门限约束与陷门的随机化哈希绑定，确保了方案能有效抵御合谋攻击与关键词猜测攻击。

7 隐私性分析

7.1 索引加密机制

为保护索引隐私，本文方案采用非对称加密。每个关键词索引的密文包含一个双线性对运算结果$e(g,g)^{bz_u s_i}$，其解密依赖用户私钥中的z_u。基于双线性映射的不可逆性，攻击者即使截获索引密文，也无法反推关键词与文件的映射关系，从而抵御关键词频率分析和数据集轮廓推断攻击。

7.2 陷门隐私性质

陷门生成过程通过密码学原语确保查询关键词的隐私性。

哈希函数遮蔽：关键词ω首先通过单向哈希函数$H_0(\omega)$进行转换，使其无法从陷门中被直接逆向推导。

群指数混淆：哈希值被用作群元素的指数(如$g^{bH_0(\omega)}$)，其安全性基于计算困难的离散对数难题。

随机性隔离：每次陷门生成都会引入一个独立的随机数t，确保即使对同一关键词进行多次查询，也能生成不同的陷门，从而有效防止重放攻击与统计分析。

7.3 陷门不可连接性

本文方案通过动态随机化和密码学混淆，确保攻击者无法关联不同的陷门。每次生成陷门时，都会引入独立的随机数(如t和υ)，使得陷门组件T_1和T_2即使对于同一关键词也各不相同。此外，用户的属性集在陷门中也经过了动态随机

掩码处理。这种多重随机化机制，结合关键词本身的加密隐藏，使得从陷门的内容、结构或时序上关联用户查询行为变得计算上不可行。

8 性能分析

本章从理论方面将本文方案同其他文献中的方案[14-16]进行对比，主要从以下方面展开：功能特性对比、索引存储、计算开销对比、是否实现弱中心化功能。

表 2 列出了本文方案与相关工作的功能特性对比。可以看出，多数方案侧重于特定功能的优化：文献[31,33-34]实现了高效的密文检索与云边协同，但缺乏属性撤销机制；文献[13]支持属性撤销，但无法进行密文检索；文献[32]虽功能较为全面，同时具备撤销与检索功能，但仍依赖于中心化架构，存在单点风险。相较之下，本文方案是唯一一个将属性撤销、密文检索、云边协同与弱中心化 4 个关键特性融为一体的方案，在功能完备性上具有明显优势。

表 2 功能特性对比

Table 2 Comparison of functional characteristics

方案	属性撤销	密文检索	云边端协同计算	弱中心化
文献[31]	×	√	×	×
文献[32]	√	√	√	×
文献[33]	×	√	√	×
文献[13]	√	×	√	×
文献[34]	×	√	√	×
文献[14]	×	×	√	×

为精确评估并对比各方案的理论计算开销，本节预先定义了一系列关键符号，如表 3 所列。这些符号主要分为两类：一是决定问题规模的核心系统参数，如属性与关键词的数量、访问策略的复杂度等；二是构成所有复杂算法的底层密码学“原子操作”的耗时，如一次双线性配对或指数运算所需的时间。通过将这些基础符号组合成数学表达式，能够为不同方案的加密、解密及检索等关键阶段构建出精确的计算复杂度公式，从而在统一的框架下进行公平且量化的性能对比。

表 3 符号及对应含义

Table 3 Symbols and their meanings

符号	含义
$\|S\|$	用户属性集中的属性数量
$\|T\|$	传输节点数量
E_1	在群$\mathbb{G}$进行一次指数操作所需要的时间
E_{G_T}	在群$\mathbb{G}_T$进行一次指数操作所需要的时间
E_T	一次双线性配对所需的时间
l	访问策略矩阵的行数
k	满足访问结构的行数
M	群内乘法
W_f	文件的关键词数量
W_q	用户查询关键词数量
Y	叶子节点的数量

从表 4 可以看出，在加密开销方面，本文方案在用户端(DO)的计算成本为固定的 $2E_1+E_T$，这是一个较低的常数级开销。相比之下，方案[32]的用户端加密开销为 $(l+3)E_1+2E_{G_T}$，方案[31]的为 $(2|Y|+1)E_1+W_fM+E_{G_T}$，分别随着系统属性总数 l 以及访问策略复杂度 $|Y|$ 的增加而增长。本方案通过将与属性数量 $|S|$ 相关的计算 $(2|S|+3)E_1$ 转移至边缘节点(EN)执行，有效降低了用户端的计算负担。

表 4 计算开销对比

Table 4 Comparison of computational overhead

方案	加密		陷门生成		搜索	解密	
	DO	EN	DU	EN	—	DU	EN
文献[31]	$(2\|Y\|+1)E_1+W_fM+E_{G_T}$	—	$(2S+1)E_1+W_qM$	—	$(2\|S\|+1)E_T+\|S\|E_{G_T}$	$3E_T+\|S\|E_{G_T}$	—
文献[32]	$(l+3)E_1+2E_{G_T}$	$(l+2)E_1$	$(2\|S\|+1)E_1$	$(\|Y\|+2)E_1+E_{G_T}$	$(\|S\|+1)E_T+E_1+E_{G_T}$	E_{G_T}	$(l+2)E_T$
文献[33]	$4E_1+E_T$	$(S+5)E_1+E_{G_T}$	$4E_1$	E_{G_T}	$(3+2k)E_T$	$3E_T$	$(2\|S\|+3)E_T+S\cdot E_{G_T G_T}$
文献[13]	$2E_1+2E_T+2E_{G_T}$	$(2\|Y\|+\|S\|)E_1$	—	—	—	$4E_T$	$(4+\|S\|)E_T+\|S\|E_{G_T}$
文献[34]	$(2l+1)E_1+E_T$	—	$(W_f+1)E_1$	—	$2W_fE_T$	E_T	$(\|S\|+2)E_T$
文献[14]	$(5l+3)E_1+E_{G_T}$	—	—	—	—	$2E_1+E_{G_T}$	$E_{G_T}+4E_T$
本文方案	$2E_1+E_T$	$(2\|S\|+3)E_1$	$4E_1$	—	$(2k+3)E_T$	$3E_T$	$2E_T$

在搜索阶段，本文方案的计算开销为 $(2k+3)E_T$，与满足访问结构的用户属性数量 k 呈线性关系。其他方案的搜索开销也与属性数量 $|S|$ 或 k 线性相关。各方案在复杂度上属于同一量级。

在解密阶段，本文方案的计算开销由用户(DU)$3E_T$ 与边缘节点(EN)$2E_T$ 组成，总计为固定的 $5E_T$。相比之下，方案[33]的总解密开销为 $(2|S|+3)E_T+S\cdot E_{G_T}$，方案[32]的为 $E_{G_T}+(l+2)E_T$，分别与用户属性数量 $|S|$ 和系统属性数量 l 呈线性增长关系。因此在属性集庞大的场景下，本文方案具有更稳定和高效的性能。

9 实验分析

本实验基于 64 位 Ubuntu 16.04 虚拟机环境开展，采用 Python 3.7 作为开发语言，并集成 Charm-0.43 及适配 Charm-0.44 的 PBC 库，曲线类型为 SS512。实验主机为笔记本电脑，搭载 12 代 Intel(R) Core(TM) i5-12500H 处理器(主频 2.50GHz)，配备 16GB 内存。

为确保实验数据的可靠性，有效降低随机误差的影响，本文方案所有实验结果均通过 100 次独立实验取均值获得。在两项仿真实验中，系统研究了加密、解密及搜索过程的时间开销与关键词数量、属性数量之间的变化关系。实验设定关键词数量 $n\in\{2,4,6,8,10\}$，用户数量 $S\in\{10,20,30,40,50\}$。

具体仿真实验结果如图 2－图 6 所示，其中绿色曲线代表 ABMKS[14,31]方案，蓝色曲线代表 LFGS[15,32]方案，紫色曲线对应 SFMS-CC[16,33]方案，HAC-SDS 为本文方案。

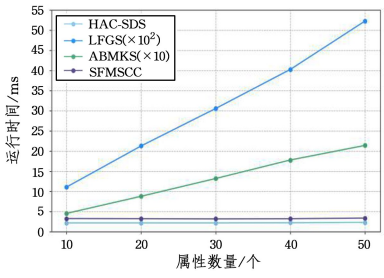

图 3 加密运行时间随属性数量变化(电子版为彩图)

Fig. 3 Change in encryption running time with the number of attributes

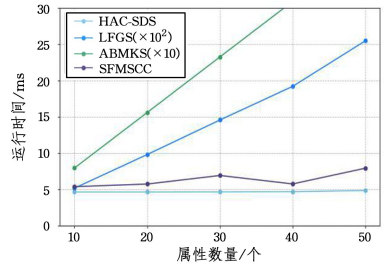

图 4 搜索运行时间随属性数量变化(电子版为彩图)

Fig. 4 Change in search running time with the number of attributes

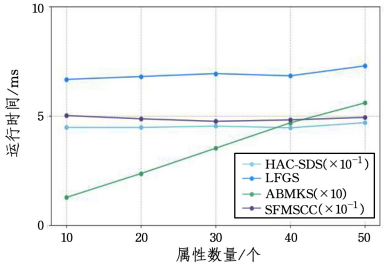

图 5 解密运行时间随属性数量变化(电子版为彩图)

Fig. 5 Change in decryption running time with the number of attributes

在图 3－图 5 中，固定关键词数量为 10，重点分析本文方案与对比方法在数据用户属性数量变化时，对加密时间的影响。对图中曲线的深入分析揭示，本文方案之所以性能最优，根本原因在于其创新的云-边-端协同计算架构。它通过计算卸载，从根本上解决了传统方案的可扩展性难题。与此形成鲜明对比的是，ABMKS 和 LFGS 等传统方案的性能趋势随着属性数量的增加而呈线性恶化，这是因为它们要求资源受限的用户终端独立完成所有与属性数量成正比的、计算密集型的密码学操作：在加密阶段，需要为每一个属性执行复杂的运算来嵌入访问策略；在搜索和解密阶段，同样需要遍历用户的属性集来生成陷门和重构密钥。而本文方案则将这些繁重的任务转移至计算能力更强的边缘节点，使得用户终端仅需执行一个固定的、计算开销极小的操作，因此其性能曲线在所有阶段都呈现为一条近乎水平的直线，实现了与系统复杂度无关的恒定时间开销，展现了在真实应用场景下无与伦比的效率和可扩展性。

图 6－图 8 揭示了本文方案在处理多关键词场景下的卓越性能，其核心优势在于采用了先进的聚合索引与聚合陷门技术。这一技术能够将任意数量的关键词高效地编码成一个单一且固定大小的索引(加密时)或陷门(搜索时)，因此其计算开销与关键词数量无关，呈现出恒定的(O(1))性能趋势，即图中的水平线。这与 LFGS 方案形成鲜明对比，后者需要为每一个关键词单独进行密码学运算，导致其计算开销随关键词数量线性增长(O(n))。而在解密阶段，所有方案均呈现恒定开销，因为解密过程与用于检索的关键词数量完全解耦；一旦文件被定位，其解密开销仅与访问策略相关，而与检索方式无关。尽管如此，HAC-SDS 凭借其在加密和搜索阶段的压倒性效率优势，证明了其在构建高效、可扩展的多关键词可搜索加密系统中的巨大价值。

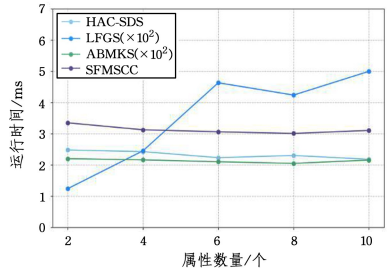

图 6 加密运行时间随关键词数量变化

Fig. 6 Change in encryption running time with the number of keywords

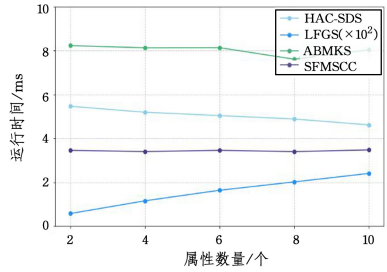

图 7 搜索运行时间随关键词数量变化

Fig. 7 Change in search running time with the number of keywords

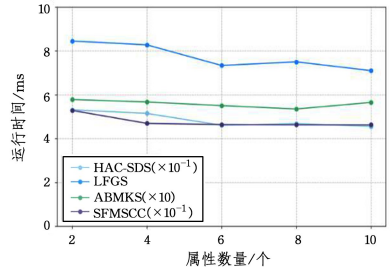

图 8 解密运行时间随关键词数量变化

Fig. 8 Change in decryption running time with the number of keywords

综合图 9－图 11 的空间开销对比分析可知，尽管 HAC-SDS 方案在私钥与密文存储上高于部分对比方案，但这并非设计的冗余，而是一个为实现架构级安全与高级功能所做出的让步。具体而言，较高的存储开销是两大核心创新的直接体现：一是为了构建基于区块链的弱中心化密钥管理体系，从而根除传统中心化方案的单点故障与密钥托管风险；二是为了实现支持精细化分层访问控制的复杂树状加密结构，提供远超扁平化策略的灵活性与管理便利性。如图 11 所示，所提方案在陷门生成开销上表现均衡，远优于 LFGS 等方案，证明

了其设计的针对性与高效性。因此，HAC-SDS 的空间开销是为其卓越的分布式安全性与高级功能性付出的必要代价，是为解决复杂应用场景核心挑战而做出的合理设计。

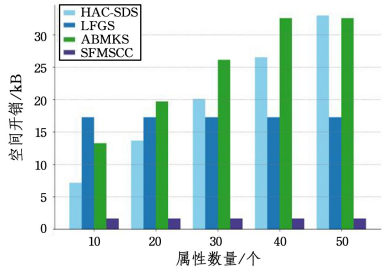

图 9 私钥生成空间开销(DU 侧)对比

Fig. 9 Comparison of private key generation space overhead (DU side)

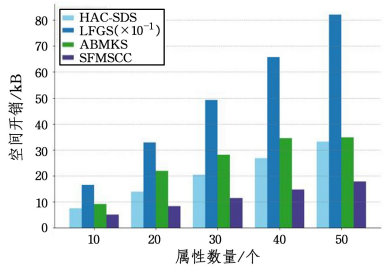

图 10 加密空间开销(DO 侧)对比

Fig. 10 Comparison of encryption space overhead (DO side)

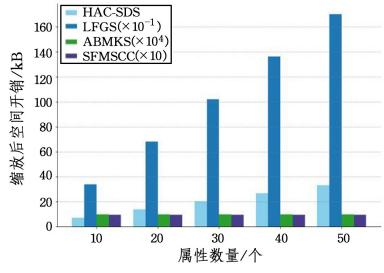

图 11 陷门生成空间开销(DU 侧)对比

Fig. 11 Comparison of trapdoor generation space overhead (DU side)

结束语 本文成功构建了一种融合云边端协同、动态加密索引与区块链技术的新型可搜索加密方案，有效解决了传统方案在终端开销、检索效率和密钥管理安全上的核心挑战。然而，该方案在通信开销与网络延迟、区块链性能瓶颈以及系统部署复杂性等方面仍存在不足。因此，未来的研究将重点聚焦于通过智能任务调度优化协同效率，利用轻量级区块链突破性能瓶颈，以及采用自动化部署降低系统复杂性，从而推动本文方案向更高效的下一代安全云服务架构演进。

参 考 文 献

[1] ZHAO Y F. Application of big data and cloud computing in electronic information systems[J]. Integrated Circuit Application, 2025, 42(1): 122-123.

[2] D'ORAZIO C J, CHOO K K R. Circumventing iOS security mechanisms for APT forensic investigations: A security taxonomy for cloud apps[J]. Future Generation Computer Systems, 2018, 79: 247-261.

[3] BROWN A J, GLISSON W B, ANDEL T R, et al. Cloud forecasting: Legal visibility issues in saturated environments[J]. Computer Law & Security Review, 2018, 34(6): 1278-1290.

[4] ZHANG X, ZHOU Y, WU D, et al. A survey on privacy-preserving caching at network edge: Classification, solutions, and challenges[J]. ACM Computing Surveys, 2025, 57(5): 1-38.

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

[6] GREEN M, HOHENBERGER S, WATERS B. Outsourcing the decryption of ABE ciphertexts[C]// 20th USENIX security symposium(USENIX Security 11). 2011.

[7] GUO L F, XING X M, GUO H. An efficient, traceable, and revocable attribute-based encryption scheme in cloud storage[J]. Journal of Cryptologic Research, 2023, 10(1): 131-145.

[8] YAN L, WANG G, YIN T, et al. Attribute-based searchable encryption: A survey[J]. Electronics, 2024, 13(9): 1621.

[9] ZHOU X B, JIANG, R. A fine-grained data encryption and sharing scheme for cloud-fog integrated environments[J]. Journal of Cryptologic Research, 2023, 10(6): 1295-1318.

[10] REN J, ZHANG L, WANG B. Decentralized multi-authority attribute-based searchable encryption scheme[J]. International Journal of Network Security, 2021, 23(2): 332-342.

[11] SAHAI A, WATERS B. Fuzzy identity-based encryption[C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2005: 457-473.

[12] WANG S, ZHOU J, LIU J K, et al. An efficient file hierarchy attribute-based encryption scheme in cloud computing[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(6): 1265-1277.

[13] ZHENG K, DING C, WANG J. A secure data-sharing scheme for privacy-preserving supporting node-edge-cloud collaborative computation[J]. Electronics, 2023, 12(12): 2737.

[14] LIU J, LI Y, SUN R, et al. EMK-ABSE: Efficient multikeyword attribute-based searchable encryption scheme through cloud-edge coordination[J]. IEEE Internet of Things Journal, 2022, 9(19): 18650-18662.

[15] XIA J, CHENG G, GU S, et al. Secure and trust-oriented edge storage for Internet of Things[J]. IEEE Internet of Things Journal, 2019, 7(5): 4049-4060.

[16] SONG D X, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]// Proceeding 2000 IEEE Symposium on Security and Privacy. IEEE, 2000: 44-55.

[17] ZHENG Q, XU S, ATENIESE G. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data[C]// IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014: 522-530.

[18] HUANG Q, HUANG P, LI H, et al. A more efficient public-key authenticated encryption scheme with keyword search[J]. Journal of Systems Architecture, 2023, 137: 102839.

[19] FAN K, CHEN Q, SU R, et al. MSIAP: A dynamic searchable encryption for privacy-protection on smart grid with cloud-edge-

end[J]. IEEE Transactions on Cloud Computing,2021,11(2):1170-1181.

[20] ZHANG W,ZHANG Z,XIONG H,et al. PHAS-HEKR-CP-ABE:partially policy-hidden CP-ABE with highly efficient key revocation incloud data sharing system[J]. Journal of Ambient Intelligence and Humanized Computing,2022,13(4):613-627.

[21] LUO F C,AL-KUWARI S,WANG H Y,et al. Revocable attribute-based encryption from standard lattices [J]. Computer Standards & Interfaces,2023,84:103698.

[22] LIU J,LI Y,SUN R,et al. EMK-ABSE:Efficient multikeyword attribute-based searchable encryption scheme through cloud-edge coordination[J]. IEEE Internet of Things Journal,2022,9(19):18650-18662.

[23] LI C,LI J,ZHANG K,et al. Verifiable cloud-based data publish-subscribe service with hidden access policy[J]. IEEE Transactions on Cloud Computing,2023,11(4):3737-3749.

[24] HUANG B H,HUANG P R,ZHAO W H,et al. Multi-keyword searchable encryption scheme upporting attribute revocation in cloud storage[J]. Computer Engineering,2021,47(11):29-36.

[25] NIU S F,SONG M,FANG L Z,et al. Cloud storage data sharing based on attribute encryption in smart healthcare[J]. Journal of Electronics & Information Technology,2022,44(1):107-117.

[26] CHASE M. Multi-authority attribute based encryption [C]// Theory of Cryptography:4th Theory of Cryptography Conference. Berlin:Springer,2007:515-534.

[27] LEWKO A,WATERS B. Decentralizing attribute-based encryption[C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer,2011:568-588.

[28] CUI J,BIAN F,ZHONG H,et al. An anonymous and outsourcing-supported multiauthority access control scheme with revocation for edge-enabled IIoT system[J]. IEEE Systems Journal,2022,16(4):6569-6580.

[29] YANG X,LI W,FAN K. A revocable attribute-based encryption EHR sharing scheme with multiple authorities in blockchain [J]. Peer-to-Peer Networking and Applications,2023,16(1):107-125.

[30] SASIKUMAR A,RAVI L,DEVARAJAN M,et al. Blockchain-assisted hierarchical attribute-based encryption scheme for secure information sharing in industrial internet of things[J]. IEEE Access,2024,12:12586-12601.

[31] CUI Y,GAO F,SHI Y,et al. An efficient attribute-based multi-keyword search scheme in encrypted keyword generation[J]. IEEE Access,2020,8:99024-99036.

[32] MIAO Y,MA J,LIU X,et al. Lightweight fine-grained search over encrypted data in fog computing[J]. IEEE Transactions on Services Computing,2018,12(5):772-785.

[33] ZHENG K,ZHOU Z,LIU J,et al. Secure Fine-Grained Multi-Keyword Ciphertext Search Supporting Cloud-Edge-End Collaboration in IoT[J]. Chinese Journal of Electronics,2025,34(1):266-281.

[34] LI Y F,ZHANG G P,LIN L B,et al. An attribute-based encryption scheme supporting accountability and verifiable outsourced decryption[J]. Journal of Guangdong University of Technology,2024,41(4):106-113.

ZHENG Kaifa, born in 1989, Ph.D, researcher. His main research interests include privacy computing, privacy protection and information security.

HE Qiang, born in 1991, Ph.D, professor, Ph.D supervisor. His main research interests include edge computing, computing power network, terminal-edge-cloud collaboration, intelligent network, artificial intelligence (model interpretability, security), machine learning algorithms(graph neural network, multi-agent), etc.

(责任编辑:何杨)