基于云边协同的无证书多用户多关键字密文检索方案

杨小东[1]，田甜[1]，王嘉琪[1]，李梅娟[1]，王彩芬[2]
（1. 西北师范大学计算机科学与工程学院，甘肃 兰州 730070；
2. 深圳技术大学大数据与互联网学院，广东 深圳 518118）

摘 要：针对工业物联网环境中密文数据检索面临的单用户单关键字搜索、计算开销过大、安全等级较低等问题，提出了一种基于云边协同的无证书多用户多关键字密文检索方案。所提方案通过设定用户访问权限表并执行一次加密算法，实现了支持用户访问权限更新的多用户搜索。利用线性扫描方法进行关键字密文与陷门的匹配计算，并引入云边协同的计算模式提高计算效率，实现了关键字索引不完全包含检索关键字情况下的多关键字密文检索。基于无证书加密体制解决了密钥托管与证书管理问题，并使用数字签名技术确保了关键字密文的可认证性。安全分析结果表明，所提方案在随机预言模型下能抵抗内部关键字猜测攻击。仿真实验结果表明，与同类方案相比较，所提方案具有较高的计算效率。

关键词：密文数据检索；云边协同；多关键字；多用户；工业物联网
中图分类号：TP309.7
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022104

Certificateless ciphertext retrieval scheme with multi-user and multi-keyword based on cloud-edge collaboration

YANG Xiaodong[1], TIAN Tian[1], WANG Jiaqi[1], LI Meijuan[1], WANG Caifen[2]
1. College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
2. College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

Abstract: To solve the problems of single-user search, single-keyword search, high computational cost and low-security level of ciphertext data retrieval in the industrial Internet of things environment, a certificateless ciphertext retrieval scheme with multi-user and multi-keyword based on cloud-edge collaboration was proposed. A user access permissions table was set and once encryption algorithm was executed to realize multi-user search and update users' access permissions. A cloud-side collaborative computing model was introduced to match keyword ciphertext and keyword trapdoor efficiently by a linear scanning method when the keyword ciphertext does not entirely contain the search keyword. Key escrow and certificate management problems were solved by utilizing certificateless encryption system. Keyword ciphertext authentication was ensured by using digital signature technology. The security analysis results show that the proposed scheme can resist internal keyword guessing attacks under the random oracle model. Simulation results show that the proposed scheme has higher computational efficiency compared with similar schemes.

Keywords: ciphertext data retrieval, cloud-edge collaboration, multi-keyword, multi-user, industrial Internet of things

收稿日期：2021−12−15；**修回日期：**2022−03−11
基金项目：国家自然科学基金资助项目（No.61662069, No.61562077, No.61702552）；广东省教育厅基金资助项目（No.2020KTSCX126）；中国博士后科学基金资助项目（No.2017M610817）；甘肃省科技计划基金资助项目（No.20CX9ZA076）
Foundation Items: The National Natural Science Foundation of China (No.61662069, No.61562077, No.61702552), Educational Commission of Guangdong Province (No.2020KTSCX126), Project of China Postdoctoral Science Foundation (No.2017M610817), Gansu Science and Technology Planning Project (No.20CX9ZA076)

0　引言

随着智能无线传感器设备在工业制造环境中的广泛使用，工业物联网（IIoT, industrial Internet of things）在工业制造领域发挥着越来越重要的作用。工业物联网是一种将全球工业体系智能互联的网络，它利用数以亿计的感知器、采集器与控制器等 IIoT 设备采集数据，并通过智能分析处理 IIoT 数据实现生产制造全过程的智能跟踪、控制与预测[1]。为降低本地的存储负担与计算开销，智能设备采集的 IIoT 数据通常会被数据拥有者上传至云存储服务器。数据用户使用关键字检索技术，通过轻量级设备从云端检索并获取目标数据。然而，IIoT 数据在公共通信网络传输时，容易遭受篡改、窃取、假冒等诸多安全攻击，这可能给工业制造企业带来巨大的经济损失。数据拥有者可以通过加密 IIoT 数据，再将密文数据上传至云存储服务器解决隐私泄露、数据篡改等安全性问题，但密文形式的数据会使数据的检索操作变得困难，数据共享的灵活性显著降低。

近年来，随着 5G 技术的发展 IIoT 数据呈指数式增长。巨大的通信负担与计算压力使基于云计算技术的集中式数据处理与存储模式逐渐不能完全满足数据传输的实时性、隐私性等要求，拥有更充足的计算处理能力的云边协同计算技术更适用于当今的工业物联网环境。云边协同技术将部分计算任务迁移到边缘节点进行，可以有效减轻云存储服务器的计算压力，并快速响应数据用户需求。然而，基于云边协同计算技术的 IIoT 数据共享场景中依然存在隐私泄露、数据篡改等安全问题。如何实现云边协同计算模式下安全高效的 IIoT 数据共享、提高 IIoT 数据共享的灵活性，已经成为近年来相关学者的研究热点。

Song 等[2]提出的密文检索技术能够同时实现数据加密与密文数据检索，已被广泛应用于数据安全共享领域[3]。Boneh 等[4]利用双线性配对构造出第一个公钥密文检索方案。为解决公钥密文检索方案存在的证书管理与密钥托管问题，学者们提出了基于无证书的密文检索方案[5-8]。但目前大部分无证书密文检索方案仅支持单用户单关键字搜索，不适用于多数据采集端多数据接收端的 IIoT 场景。Golle 等[9]提出了支持多关键字搜索的密文检索方案，文献[10-12]提出了支持多用户搜索的密文检索方案，但这些方案均不能同时支持多用户多关键字搜索。Ma 等[13]提出了基于无证书的多用户多关键字密文检索方案，但该方案安全性较低，恶意的内部攻击者可能利用搜索关键字空间较小的特点对方案进行内部关键字猜测攻击（IKGA, internal keyword guessing attack）。密文检索方案的安全性是相关学者研究的另一焦点问题[14-15]。Chenam 等[16]于 2022 年提出了能够抵抗 IKGA 的多用户多关键字密文检索方案，但方案检索精度不高，在关键字索引不完全包含用户检索的关键字时无法返回搜索结果。

部分学者在云计算环境下实现了 IIoT 数据安全共享[17-18]，但基于云计算的集中式数据处理模式已经不适用于数据呈指数式增长的新型工业物联网环境。黄海平等[19]利用多个服务器完成密文检索任务，但该方案不能抵抗 IKGA。张强等[20]提出了基于边缘计算的密文检索方案，但该方案中数据加密密钥需要在安全信道中传输。在上述研究的基础上，本文针对新型 IIoT 环境提出安全性较高和检索效率较好的密文数据检索方案。本文主要贡献如下。

1) 支持多用户多关键字搜索。数据拥有者在执行数据加密算法时生成多个合法用户的身份信息密文，指定多个合法用户进行搜索。基于文件阈值表与用户访问权限表，多个边缘服务器同时进行关键字匹配计算，实现多个关键字索引未精确包含多个搜索关键字情况下的正确检索。

2) 提高了密文检索效率。与同类多用户方案相比较，本文方案在数据加密阶段计算开销较小且不随数据用户数量的增加呈线性增长。在云边协同的计算模式下，多个边缘服务器同时进行关键字匹配计算，提高了关键字匹配效率。

3) 实现了较高的安全性。本文方案基于数字签名的不可伪造性，利用数据拥有者和数据用户的私钥分别对关键字索引与搜索陷门签名，抵抗了 IKGA。基于决策线性 Diffie-Hellman 问题（DLDHP, decision linear Diffie-Hellman problem）假设，在随机预言模型下被证明能够抵抗两类攻击者的选择关键字攻击。

1　预备知识

1.1　双线性映射

设 G_1 和 G_2 是 2 个阶为大素数 p 的乘法循环群。如果一个可计算的映射 $e: G_1 \times G_1 \to G_2$ 满足以下性质，则称 e 是一个双线性映射[21]。

1) 双线性：对于任意的 $a,b\in Z_p^*$ 和 $g_1,g_2\in G_1$，等式 $e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$ 成立。

2) 非退化性：存在 $g_1,g_2\in G_1$，使 $e(g_1,g_2)\neq 1$。

3) 可计算性：对于任意的 $g_1,g_2\in G_1$，都存在一个有效算法可以计算出 $e(g_1,g_2)$。

1.2 困难问题假设

DLDHP。G_1 是由 g 生成的阶为 p 的循环群，给定 g_1、g_2、g_3、g_1^a、g_2^b 和 g_3^c，其中 $a,b,c\in Z_p^*$ 未知，DLDHP 就是判断 c 是否等于 $a+b \bmod p$。

DLDHP 假设。定义任何一个概率多项式时间算法 A 成功求解 G_1 上的 DLDHP 的概率优势为 $\mathrm{Adv}_{\mathrm{DLDHP}}(A)$。若 $\mathrm{Adv}_{\mathrm{DLDHP}}(A)$是可忽略的，则称 G_1 上的 DLDHP 假设成立。

1.3 安全目标

一般在讨论无证书公钥密码方案的安全性时，考虑两类敌手 A_1 和 A_2。A_1 无法访问系统主密钥但能进行公钥替换攻击，A_2 不能对数据用户进行公钥替换攻击但可以访问系统主密钥。针对这两类敌手，本文提出的无证书密文数据检索方案旨在达到如下 3 个安全目标。

1) 共享文件数据的机密性

即使敌手 A_1 和 A_2 截获了部分密文文件，也无法解密密文文件或猜测到密文文件对应哪些检索关键字，只有拥有文件访问权限的 IIoT 用户可以访问共享文件数据。

2) 关键字密文与搜索陷门的可认证性

在敌手 A_1 和 A_2 的攻击下，方案需能保证关键字索引密文与搜索陷门的可认证性，恶意的内部攻击者不能对方案进行内部关键字猜测攻击。

3) 选择关键字攻击下的不可区分性

在敌手 A_1 和 A_2 的选择关键字攻击下，方案需能保证关键字索引密文的不可区分性。

2 系统模型及数据结构

2.1 系统模型

本文研究 IIoT 的多用户数据共享场景，方案实体包括云服务器、边缘服务器、IIoT 中的数据拥有者和数据用户，以及密钥生成中心（KGC, key generation center)。系统模型如图 1 所示。

1) 云服务器。云服务器存储 IIoT 中的数据拥有者 D_{oj} 上传的用户访问权限表与文件访问权限表。当数据用户 D_{ui} 想要检索 D_{oj} 共享的包含某些关键字的密文文件 f_μ 时，云服务器负责判断 D_{ui} 是否为合法用户。在关键字索引不完全包含 D_{ui} 检索的关键字时，云服务器选择文件匹配值最高的密文文件，向存储此密文文件的边缘服务器发送文件传送指令。

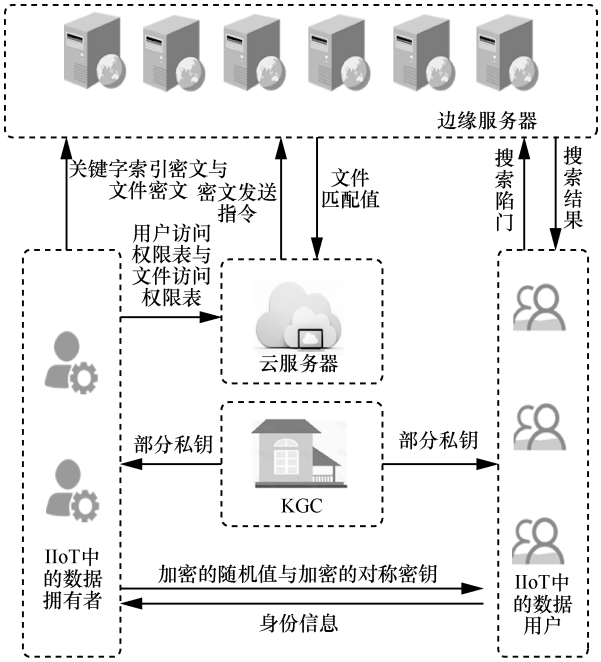


图 1 系统模型

2) 边缘服务器。边缘服务器 $S_1,S_2,\cdots,S_Z$ 存储 IIoT 中数据拥有者 D_{oj} 上传的关键字索引密文与部分文件密文。收到数据用户 D_{ui} 的搜索陷门后，$S_1,S_2,\cdots,S_Z$ 执行陷门匹配计算，生成文件匹配值并将匹配值发送给云服务器。当收到来自云服务器的密文发送指令，$S_1,S_2,\cdots,S_Z$ 发送相应的密文给 D_{ui}。

3) IIoT 中的数据拥有者。数据拥有者 D_{oj} 负责加密文件数据、生成关键字索引密文，以及将加密的随机值和对称密钥发送给合法的数据用户 D_{ui}。D_{oj} 可通过生成并更新用户访问权限表和文件访问权限表更新用户对文件数据的访问权限。

4) IIoT 中的数据用户。数据用户 D_{ui} 负责生成关键字搜索陷门并上传陷门至边缘服务器。收到密文文件 f_μ 后，D_{ui} 用私钥 $\mathrm{SK}_{D_{\mathrm{ui}}}$ 计算得到文档解密密钥 K 并解密 f_μ。

5) 密钥生成中心。密钥生成中心生成 IIoT 中数据拥有者 D_{oj} 和数据用户 D_{ui} 的部分私钥。

2.2　数据结构

本文方案使用的数据结构如下。

1) 用户访问权限表。用户访问权限表由数据拥有者 D_{oj} 上传至云服务器，数据拥有者 D_{oj} 能够对用户访问权限表执行写操作与读操作，云服务器只能对该表执行读操作。用户访问权限表中，B_i 表示合法用户 D_{ui} 的身份信息密文，D_{oj} 可通过增加或删除 B_i 对应的访问属性 $a_1,a_2,\cdots,a_x$ 更新用户 B_i 的访问属性。在用户访问权限表中增加或删除 B_i 可以直接增加或删除合法用户。

2) 文件访问权限表。文件访问权限表由数据拥有者 D_{oj} 对每个密文文件 f_μ 设定阈值 β_u 与访问属性并将该表上传至云服务器。D_{oj} 能够对文件访问权限表执行写操作与读操作，云服务器只能对文件访问权限表执行读操作。当数据用户的属性集合与 f_μ 的属性集合的交集中的元素个数 ε 大于 f_μ 的设定阈值 β_{u} 时，数据用户有权访问 f_μ。文件访问权限表如表 1 所示。

表 1　　**文件访问权限表**

文件名	阈值	属性
f_1	β_1	$a_1,a_2,\cdots,a_x$
⋮	⋮	⋮
f_n	β_n	$a_3,a_6,a_{13},\cdots$

3　方案描述

3.1　系统初始化算法

给定系统参数 1^k，KGC 选择 2 个阶为素数 q 的循环群 G_1、G_2 和双线性映射 $e:G_1\times G_1\to G_2$，其中 G_1 的生成元为 g。KGC 随机选择 $S\in Z_q^*$ 作为系统主密钥，计算系统公钥 $P=g^S$ 并选取哈希函数 $H_1:\{0,1\}^*\to G_1$，$H_2:\{0,1\}^*\to G_1$，$H_3:\{0,1\}^*\times G_1\to Z_q^*$，$H_4:\{0,1\}^*\times G_1\times G_1\times G_1\to Z_q^*$ 和 $H_0:G_2\to Z_q^*$。KGC 输出系统参数 $\text{prms}=(e,g,q,P,G_1,G_2,H_0,H_1,H_2,H_3,H_4)$。

3.2　部分密钥生成算法

KGC 输入系统参数 prms 和系统主密钥 S，执行如下步骤生成数据用户和数据拥有者的部分密钥。

1) 输入 m 个数据用户 D_{ui} 的身份 $\text{ID}_i\in\{0,1\}^*$，其中 $1\leqslant i\leqslant m$。KGC 选择 m 个随机数 $x_1,x_2,x_3,\cdots,x_m\in Z_q^*$，计算 $R_{\text{ID}_i}=g^{x_i}$，$\alpha_i=H_3(\text{ID}_i,R_{\text{ID}_i})$ 和 $D_{\text{ID}_i}=x_i+S\alpha_i(\text{mod}\ q)$，并将部分密钥 $(R_{\text{ID}_i},D_{\text{ID}_i})$ 发送给 D_{ui}。

2) 输入 n 个数据拥有者 D_{oj} 的身份 $\text{ID}_j\in\{0,1\}^*$，其中 $1\leqslant j\leqslant n$。KGC 选择 n 个随机数 $y_1,y_2,y_3,\cdots,y_n\in Z_q^*$，计算 $R_{\text{ID}_j}=g^{y_j}$，$\alpha_j=H_3(\text{ID}_j,R_{\text{ID}_j})$，$\xi=g^{S\alpha_j}$ 和 $D_{\text{ID}_j}=y_j+S\alpha_j(\text{mod}\ q)$，并将部分密钥 $(R_{\text{ID}_j},D_{\text{ID}_j})$ 及 ξ 发送给 D_{oj}。

3.3　私钥生成算法

数据拥有者 D_{oj} 随机选择 $r_j\in Z_q^*$ 作为秘密值，设定私钥 $\text{SK}_{D_{\text{oj}}}=(r_j,D_{\text{ID}_j})$。数据用户 D_{ui} 随机选择 $r_i\in Z_q^*$ 作为秘密值，设定私钥 $\text{SK}_{D_{\text{ui}}}=(r_i,D_{\text{ID}_i})$。

3.4　公钥生成算法

数据拥有者 D_{oj} 计算 $X_{D_{\text{oj}}}=g^{r_j}$，设定公钥 $\text{PK}_{D_{\text{oj}}}=(X_{D_{\text{oj}}},R_{\text{ID}_j})$。数据用户 D_{ui} 计算 $X_{D_{\text{ui}}}=g^{r_i}$，设定公钥 $\text{PK}_{D_{\text{ui}}}=(X_{D_{\text{ui}}},R_{\text{ID}_i})$。

3.5　数据加密算法

输入系统参数 prms，数据拥有者 D_{oj} 的私钥 $\text{SK}_{D_{\text{oj}}}$ 和公钥 $\text{PK}_{D_{\text{oj}}}$，数据用户 D_{ui} 的公钥 $\text{PK}_{D_{\text{ui}}}$，待加密文档 D 以及文档关键字集 $W=\{w_1,w_2,w_3,\cdots,w_l\}$，$D_{\text{oj}}$ 执行如下步骤生成文档密文和文档关键字密文。

1) D_{oj} 随机选择 $t\in Z_q^*$ 和 $t'\in Z_q^*$，计算 $A=(X_{D_{\text{oj}}}R_{\text{ID}_j}\xi)^t$ 和用户身份信息密文 $B_i=(X_{D_{\text{ui}}}^{v_i}R_{\text{ID}_i}p^{\alpha_i})^{t'}$，其中 $v_i=H_4(\text{ID}_i,P,X_{D_{\text{ui}}},R_{\text{ID}_i})$。然后，$D_{\text{oj}}$ 对每个关键字 $w_k(1\leqslant k\leqslant l)$ 计算 $e_{I_k}=H_1(w_k)$、$f_{I_k}=H_2(w_k)$ 和 $C_{I_k}=e_{I_k}^{(r_j+D_{\text{ID}_j})t}f_{I_k}^{t'}$。最后，$D_{\text{oj}}$ 输出关键字集 $W=\{w_1,w_2,\cdots,w_l\}$ 对应的关键字密文集 $C=C_1,C_2,\cdots,C_l$。

2) D_{oj} 选择对称加密密钥 $K\in Z_q^*$，计算 $C'=\text{En}_K(D)$ 并发送给边缘计算节点 $S_1,S_2,\cdots,S_Z$ 等分后的密文文档 $C'_{\frac{1}{z}}$ 和 $C'_{\frac{1}{z}}$ 对应的关键字密文 $C_\alpha,C_\beta,\cdots,C_\gamma$。每个边缘节点收到的密文数据为 $C^*=\left(A,B_1,B_2,\cdots,B_n,C_\alpha,C_\beta,\cdots,C_\gamma,C'_{\frac{1}{z}}\right)$。密文文档集 $C'_{\frac{1}{z}}$ 中包含多个密文文档，不同的密文文档中可能

包含相同的关键字密文。

3.6 增加用户算法

数据拥有者 D_{oj} 收到合法新用户 $D_{u_{\mathrm{new}}}$ 的访问请求后计算 $B_{\mathrm{new}}=X_{D_{u_{\mathrm{new}}}}^{v_{\mathrm{new}}}R_{\mathrm{ID}_{\mathrm{new}}}p^{\alpha_{\mathrm{new}}}$，并将 B_{new} 添加到用户访问权限表中。然后，D_{oj} 根据新用户 $D_{u_{\mathrm{new}}}$ 的访问权限生成 $D_{u_{\mathrm{new}}}$ 的访问属性 $u_{\mathrm{new}}=\{a_1,a_2,\cdots,a_x\}$。最后 D_{oj} 将更新后的用户访问权限表发送给云服务器。

3.7 陷门生成算法

数据用户 D_{ui} 随机选择 $r\in Z_q^*$，计算 $T_a=g^r$，$T_b=e_{I_k}{}^r$ 和 $T_c=f_{I_k}^{\frac{r}{v_ir_i}+D_{\mathrm{ID}_i}}$，其中 $e_{I_k}=H_1(w_k)$，$f_{I_k}=H_2(w_k)$。令关键字 w_k 的陷门 $T_i=(T_a,T_b,T_c)$。D_{ui} 分别计算搜索关键字 $W=\{w_1,\cdots,w_k,\cdots,w_{l'}\}$ 的搜索陷门 $T_1,\cdots,T_i,\cdots,T_{l'}$，并将 $T_1,\cdots,T_i,\cdots,T_{l'}$ 发送给各边缘计算节点 $S_1,S_2,\cdots,S_Z$。

3.8 匹配测试算法

云服务器和边缘服务器执行如下步骤匹配关键字索引密文与搜索陷门。

1) 云服务器首先对提出检索请求的用户 D_{ui} 进行合法性验证，检验用户访问权限表中是否存在 D_{ui} 的身份信息密文 B_i。若 B_i 存在则身份验证通过，继续执行下述匹配测试步骤；否则云服务器返回“⊥”，拒绝 D_{ui} 的搜索请求。

2) 边缘节点 $S_1,S_2,\cdots,S_Z$ 收到 D_{ui} 发送的搜索陷门 $T_1,\cdots,T_i,\cdots,T_l$ 后，对每个搜索陷门 T_i 进行匹配计算。

①每个边缘节点 S_r 判断等式 $e(T_a,C_{I_k})=e(A,T_b)\cdot e(B_i,T_c)$ 是否成立，等式每成立一次，则包含相应关键字的密文文件 f_μ 的匹配值 U 增加 1。

② 每个边缘节点 S_r 选择密文文件集 $C'_{\frac{1}{z}}$ 中包含数据用户检索的关键字数目最多的文件 f_μ（即文件 f_μ 的匹配值 U 最大），向云服务器上传 f_μ 的身份信息以及 f_μ 的匹配值 U。

3) 云服务器收到边缘服务器 $S_1,S_2,\cdots,S_Z$ 上传的文件身份信息与匹配值信息后，执行如下步骤找到与搜索陷门 $T_1,\cdots,T_i,\cdots,T_l$ 匹配度最高的密文文件。

① 计算 f_μ 的属性集与 D_{ui} 的属性集的交集中的元素个数，记为 ε。若 ε 大于 f_μ 的文件阈值 β_μ，将文件 f_μ 的匹配值 U 加入返回列表；否则云服务器令边缘节点 S_r 继续验证文件匹配值 U 次高的文件，直至有来自 S_r 的文件匹配值 U 加入返回列表。若无任何匹配文件，云服务器返回“⊥”，表示 D_{ui} 无法访问边缘节点 S_r 处的密文文件集 $C'_{\frac{1}{z}}$。

② 在 $S_1,S_2,\cdots,S_Z$ 返回的文件匹配值列表中，云服务器选择返回最大值 $U_{\max}$ 的边缘节点 S_r，令 S_r 发送 $U_{\max}$ 对应的密文文件 f_μ 给 D_{ui}。当 $U_{\max}$ 对应多个边缘节点时，云服务器令多个边缘节点返回相应的密文文件。

3.9 解密算法

数据拥有者 D_{oj} 和数据用户 D_{ui} 执行如下步骤解密文档密文。

1) D_{oj} 收到 D_{ui} 发送的身份信息 ID_i 后，根据用户访问权限表判断 D_{ui} 是否为合法用户。若 D_{ui} 合法，D_{oj} 随机选择 $b\in Z_q^*$，计算 $b'=\mathrm{Enc}_{\mathrm{PK}_{D_{\mathrm{ui}}}}(b)$ 和 $K'=KH_0(e(P,g)^b)$，并将 (b',K') 发送给 D_{ui}。

2) D_{ui} 收到 (b',K') 后用私钥 $\mathrm{SK}_{D_{\mathrm{ui}}}$ 解密 b' 得到 b，计算对称密钥 $K=\dfrac{K'}{H_0(e(P,g)^b)}$ 和文档明文 $M=\mathrm{Dec}_K(f_\mu)$。

4 方案分析

4.1 正确性分析

边缘服务器 $S_1,S_2,\cdots,S_Z$ 通过验证等式 $e(T_a,C_{I_k})=e(A,T_b)\,e(B_i,T_c)$ 是否成立，判断关键字密文与搜索陷门是否匹配。当 $S_1,S_2,\cdots,S_Z$ 拥有的关键字密文与数据用户发送的搜索陷门包含某相同关键字 w_k 时，$S_1,S_2,\cdots,S_Z$ 可对验证等式左右两边进行如下计算。

$$
\begin{aligned}
&e(T_a,C_{I_k})=e\left(g^r,e_{I_k}^{(r_j+D_{\mathrm{ID}_j})t}f_{I_k}^{t'}\right)=\\
&e\left(g^r,e_{I_k}^{(r_j+x_j+s\alpha_j)t}\right)e(g^r,f_{I_k}{}^{t'})=\\
&e(g,e_{I_k})^{r(r_j+x_j+s\alpha_j)t}e(g,f_{I_k})^{rt'}\\
&e(A,T_b)e(B_i,T_c)=\\
&e\left[(X_{D_{\mathrm{oj}}}R_{\mathrm{ID}_j}\xi)^t,e_{I_k}^r\right]\cdot\\
&e\left[(X_{D_{\mathrm{ui}}}^{v_i}R_{\mathrm{ID}_i}p^{\alpha_i})^{t'},f_{I_k}^{\frac{r}{v_ir_i}+D_{\mathrm{ID}_i}}\right]=\\
&e\left[g^{(r_j+x_j+s\alpha_j)t},e_{I_k}^r\right]e\left[g^{t'(v_ir_i+D_{\mathrm{ID}_i})},f_{I_k}^{\frac{r}{v_ir_i}+D_{\mathrm{ID}_i}}\right]=\\
&e(g,e_{I_k})^{r(r_j+x_j+s\alpha_j)t}e(g,f_{I_k})^{rt'}
\end{aligned}
\tag{1}
$$

(2)

由式(1)和式(2)可知，本文方案的验证等式 $e(T_a,C_{I_k})=e(A,T_b)\,e(B_i,T_c)$ 正确，满足正确性。

4.2　安全性分析

根据 1.3 节定义的安全目标，本节分析文件数据的机密性以及关键字密文与搜索陷门的可认证性。

1) 文件数据的机密性分析

文件数据的机密性由对称加密算法的安全性保证。在本文方案中，数据拥有者 D_{oj} 计算 $K'=K\cdot H_0\left(e(P,bg)\right)$ 并使用对称密钥 K 对明文文档 D 进行加密。数据用户 D_{ui} 需经过解密计算才能求得对称密钥 K。同时由于文件密文被分开存储在不同的边缘节点，每个边缘节点只拥有部分文件密文，边缘节点不容易猜测出全部的密文数据与关键字密文的对应关系。

2) 关键字密文与搜索陷门的可认证性分析

数据拥有者 D_{oj} 在生成关键字密文 $C_{I_k}=e_{I_k}^{(r_j+D_{\text{ID}_j})t}f_{I_k}^{t'}$ 的过程中使用了私钥 $\text{SK}_{D_{\text{oj}}}=(r_j+D_{\text{ID}_j})$。数据用户 D_{ui} 在生成陷门 $T_c=f_{I_k}^{\frac{r}{v_i r_i}+D_{\text{ID}_i}}$ 的过程中使用了私钥 $\text{SK}_{D_{\text{ui}}}=(r_i,D_{\text{ID}_i})$。云服务器与边缘服务器 $S_1,S_2,\cdots,S_Z$ 在没有 D_{oj} 和 D_{ui} 私钥的前提下，不能通过生成或篡改关键字密文与搜索陷门对方案进行内部关键字猜测攻击。关键字密文与搜索陷门可认证性的实质是 D_{oj} 与 D_{ui} 分别对关键字密文和搜索陷门进行了签名。基于数字签名的不可伪造性，攻击者无法对方案进行 IKGA，具体证明过程可参考文献[22]。

4.3　安全性证明

本节证明本文方案在随机预言模型下能够抵抗攻击者 A_1 和 A_2 的选择关键字攻击。

定理 1　基于 DLDHP 假设，在随机预言模型下本文方案满足选择关键字攻击下关键字密文的不可区分性。

定理 1 可通过引理 1 和引理 2 证明。

引理 1　在随机预言模型中，若敌手 A_1 能够以不可忽略的概率优势 ε 在多项式时间内攻破方案，则挑战者 C 能以不可忽略的优势 $\varepsilon'\geqslant\dfrac{\varepsilon}{4nq_t}$ 构造多项式时间算法解决 DLDHP。其中，q_t 表示陷门询问的最大执行次数，n 表示数据用户的数量。

证明　挑战者 C 通过与敌手 A_1 进行如下的交互游戏解决 DLDHP。

给定输入元组 $F=(g_1,g_2,g_3,Z,v_1,v_2,v_3)$，其中 $v_1=g_1{}^a$，$v_2=g_2{}^b$。令 $p=w_2=g_2{}^b$，$g=g_1$，$g_2=g_1^{\alpha}=g^{\alpha}$。C 执行系统初始化算法生成 $\text{prms}=(e,g,G_1,G_2,p,q,H_1,H_2,H_3,H_4)$，随机选择 $n\in Z_q^*$ 作为系统主密钥，并将 prms 发送给 A_1。

询问阶段 A_1 可向 C 进行下述的一系列询问。C 分别维护列表 H_1^{list}、H_2^{list}、H_3^{list}、H_4^{list}、L_1^{list}、L_2^{list} 和 L_3^{list}，用于回复 A_1 的各类询问，各列表初始为空。

1) H_1 询问。当 C 收到 A_1 对关键字 w_i 的 H_1 询问时，若元组 (w_i,c_i,e_i,m_i) 已经存在于 H_1^{list} 中，C 输出 e_i 作为 H_1 询问的回复。否则 C 选择 $c_i\in\{0,1\}$，当 $c_i=0$ 时，C 随机选择 $m_i\in Z_q^*$，计算 $e_i=g_1^{m_i}$；当 $c_i=1$ 时，C 随机选择 $m_i\in Z_q^*$，计算 $e_i=g_3^{m_i}$。最后 C 输出 e_i 作为回复，并将元组 (w_i,c_i,e_i,m_i) 存入 H_1^{list}。

2) H_2 询问。当 C 收到 A_1 对关键字 w_i 的 H_2 询问时，若元组 (w_i,c_i,f_i,n_i) 已经存在于 H_2^{list} 中，C 输出 f_i 作为 H_2 询问的回复。否则 C 选择 $c_i\in\{0,1\}$，当 $c_i=0$ 时，C 随机选择 $n_i\in Z_q^*$，计算 $f_i=g_2^{n_i}$；当 $c_i=1$ 时，计算 $f_i=g_3^{n_i}$，其中 $n_i=\dfrac{(\pi_j+D_{\text{ID}_j})m_i}{n}$。最后 C 输出 f_i 作为 H_2 询问的回复，并将元组 (w_i,c_i,f_i,n_i) 存入 H_2^{list}。

3) H_3 询问。C 收到 A_1 关于用户身份 ID_i 的 H_3 询问后，若元组 $(\text{ID}_i,R_{\text{ID}_i},\alpha_i)$ 已经存在于 H_3^{list} 中，C 直接输出 α_i；否则 C 随机选择 $\alpha_i\in Z_q^*$，输出 α_i 作为 H_3 询问的回复，并将元组 $(\text{ID}_i,R_{\text{ID}_i},\alpha_i)$ 存入 H_3^{list}。

4) H_4 询问。当 C 收到 A_1 关于用户身份 ID_i 的 H_4 询问时，若元组 $(\text{ID}_i,P,X_{D_{\text{ui}}},R_{\text{ID}_i},v_i)$ 存在于 H_4^{list} 中，C 输出 v_i；否则 C 随机选择 $v_i\in Z_q^*$，输出 v_i 作为 H_4 询问的回复，并将元组 $(\text{ID}_i,P,X_{D_{\text{ui}}},R_{\text{ID}_i},v_i)$ 存入 H_4^{list} 中。

5) 部分私钥询问。当 C 收到 A_1 对身份为 ID_i 的用户的部分私钥询问时，若元组 $(\text{ID}_i,x_i,R_{\text{ID}_i},D_{\text{ID}_i})$ 存在于 L_1^{list} 中，C 将元组 $(R_{\text{ID}_i},D_{\text{ID}_i})$ 返回给 A_1；否则 C 随机选择 $\alpha_i\in Z_q^*$，$x_i\in Z_q^*$，$D_{\text{ID}_i}\in Z_q^*$ 并计算 $R_{\text{ID}_i}=g_2^{D_{\text{ID}_i}}p^{-\alpha_i}$，将元组 $(\text{ID}_i,R_{\text{ID}_i},\alpha_i)$ 存入 H_3^{list}，将元组 $(\text{ID}_i,x_i,R_{\text{ID}_i},D_{\text{ID}_i})$ 存入 L_1^{list}，输出 $(R_{\text{ID}_i},D_{\text{ID}_i})$ 作为部分私钥询问的回复。

6) 公钥提取询问。当 C 收到 A_1 对身份为 ID_i 的

用户的公钥询问后，若元组 $(\mathrm{ID}_i, x_i, R_{\mathrm{ID}_i}, D_{\mathrm{ID}_i})$ 存在于 L_1^{list} 中，C 随机选择 $\pi_i \in Z_q^*$，计算 $X_{D_{\mathrm{ui}}} = g_2^{\pi_i}$，将元组 $(\mathrm{ID}_i, X_{D_{\mathrm{ui}}}, R_{\mathrm{ID}_i})$ 存入 L_2^{list} 中并输出 $(X_{D_{\mathrm{ui}}}, R_{\mathrm{ID}_i})$ 作为公钥提取询问的回复；否则 C 先执行 ID_i 的部分私钥询问生成元组 $(\mathrm{ID}_i, R_{\mathrm{ID}_i}, \alpha_i)$。然后 C 随机选择 $\pi_i \in Z_q^*$，计算 $X_{D_{\mathrm{ui}}} = g_2^{\pi_i}$，将元组 $(\mathrm{ID}_i, X_{D_{\mathrm{ui}}}, R_{\mathrm{ID}_i})$ 存入 L_2^{list} 中并输出 $(X_{D_{\mathrm{ui}}}, R_{\mathrm{ID}_i})$ 作为公钥提取询问的回复。

7) 公钥替换询问。当 C 收到 A_1 对身份为 ID_i 的用户的公钥替换询问时，C 用元组 $(\mathrm{ID}_i', X_{D_{\mathrm{ui}}}', R_{\mathrm{ID}_i}')$ 替换元组 $(\mathrm{ID}_i, X_{D_{\mathrm{ui}}}, R_{\mathrm{ID}_i})$。

8) 秘密值询问。当 C 收到 A_1 对身份为 ID_i 的用户的秘密值询问时，若元组 (ID_i, π_i) 已经存在于 L_3^{list}，C 返回 π_i；否则 C 随机选择 $\pi_i \in Z_q^*$，将元组 (ID_i, π_i) 存入 L_3^{list} 并输出 π_i 作为秘密值询问的回复。

9) 陷门询问。收到 A_1 关于关键字 w_i 的陷门询问后，C 按如下步骤回复 A_1。

① C 执行上述询问分别得到元组 (w_i, c_i, e_i, m_i)、(w_i, c_i, f_i, n_i)、$(\mathrm{ID}_i, R_{\mathrm{ID}_i}, \alpha_i)$、$(\mathrm{ID}_i, X_{D_{\mathrm{ui}}}, R_{\mathrm{ID}_i})$、$(\mathrm{ID}_i, P, X_{D_{\mathrm{ui}}}, R_{\mathrm{ID}_i}, V_i)$ 和 (ID_i, π_i)。

② C 选择随机数 $r \in Z_q^*$，计算 $T_a = g_1^r$，$T_b = g_1^{rD_{\mathrm{ID}_i}}$ 和 $T_c = g_1^{\frac{rn_i}{v_i\pi_i} + D_{\mathrm{ID}_i}}$，输出 $T_i = (T_a, T_b, T_c)$ 作为陷门询问的回复。

挑战阶段 A_1 选择 $W^* = (w_{0,1}, w_{0,2}, \cdots, w_{0,n})$ 作为挑战关键字，C 随机选择 $R = (w_{1,1}, w_{1,2}, \cdots, w_{1,n})$，令 $W_0 = W^*$，$W_1 = R$，且 A_1 不能询问 W_0 和 W_1 的陷门。然后 C 随机选择 $b \in \{0,1\}$，对所有关键字 $W_{b,i}$ 进行 H_1 询问与 H_2 询问，得到元组 $(w_{b,i}, c_{b,i}, e_{b,i}, m_{b,i})$ 和元组 $(w_{b,i}, c_{b,i}, f_{b,i}, n_{b,i})$。若所有 $c_{b,i} \neq 1$，C 终止游戏；否则 C 计算 $A = (X_{D_{\mathrm{oj}}} R_{\mathrm{ID}_j} \xi)^a$，$B_i = v_2^{(v_i\pi_i + D_{\mathrm{ID}_i})n} = g_2^{(v_i\pi_i + D_{\mathrm{ID}_i})bn} = (X_{D_{\mathrm{ui}}}^{v_i} R_{\mathrm{ID}_i} P^{\alpha_i})$。当 $c_{b,i} = 0$ 时，C 计算挑战密文 $c_{b,i} = v_1^{m_{b,i}(\pi_j + D_{\mathrm{ID}_j})} v_2^{n_{b,i}n}$；当 $c_{b,i} = 1$ 时，C 计算挑战密文 $c_{b,i} = v_3^{m_{b,i}(\pi_j + D_{\mathrm{ID}_j})}$。最后 C 发送 W_0、W_1 以及挑战密文 $(A, B_1, B_2, \cdots B_n, C_{b,1}, C_{b,2}, \cdots C_{b,l})$ 给 A_1。

猜测阶段 A_1 输出猜测值 $b' \in \{0,1\}$，若 $b \neq b'$，令 $v_3 = Z$，C 终止游戏；否则令 $v_3 = g_3^{a+b}$，C 可进行如下计算验证挑战密文的有效性。

$$A^* = X_{D_{\mathrm{oj}}} R_{\mathrm{ID}_j} \xi \tag{3}$$

$$B_i^* = v_2^{(v_i\pi_i + D_{\mathrm{ID}_i})n} = g_2^{(v_i\pi_i + D_{\mathrm{ID}_i})bn} = (X_{D_{\mathrm{ui}}}^{v_i} R_{\mathrm{ID}_i} P^{\alpha_i})^{bn} \tag{4}$$

若 $c_{b,i} = 0$，计算

$$c_{b,i} = v_1^{m_{b,i}(\pi_j + D_{\mathrm{ID}_j})} v_2^{n_{b,i}n} = g_1^{am_{b,i}(\pi_j + D_{\mathrm{ID}_j})} g_2^{bn_{b,i}n} = e_{b,i}^{a(\pi_j + D_{\mathrm{ID}_j})} f_{b,i}^{bn} \tag{5}$$

若 $c_{b,i} = 1$，计算

$$c_{b,i} = v_3^{m_{b,i}(\pi_j + D_{\mathrm{ID}_j})} = g_3^{(a+b)m_{b,i}(\pi_j + D_{\mathrm{ID}_j})} = g_3^{am_{b,i}(\pi_j + D_{\mathrm{ID}_j})} \left(g_3^{\frac{m_{b,i}(\pi_j + D_{\mathrm{ID}_j})}{n}} \right)^{bn} = e_{b,i}^{a(\pi_j + D_{\mathrm{ID}_j})} f_{b,i}^{bn} \tag{6}$$

下面，计算挑战者 C 成功解决 DLDHP 的概率优势 ε'。令事件 E_1 表示陷门询问过程中 C 没有终止，事件 E_2 表示挑战阶段中 C 没有终止，事件 E_3 表示 C 没有对 W_0 和 W_1 进行陷门询问。事件 E_1、E_2 和 E_3 相继发生。

若 A_1 能以 ε 的概率攻破本文方案，则 $\left|\Pr[b' = b] - \frac{1}{2}\right| \geqslant \varepsilon$。由 $\Pr[E_1] = \left(1 - \frac{1}{nq_t}\right)^{nq_t} \geqslant \frac{1}{4}$ 和 $\Pr[E_2 \mid E_1] = 1 - \left(1 - \frac{1}{nq_t}\right)^n \geqslant \frac{1}{nq_t}$ 可知，$\Pr[E_1 \wedge E_2] = \Pr[E_1]\Pr[E_2 \mid E_1] \geqslant \frac{1}{4nq_t}$。由 $\Pr[b' = b] = \Pr[\neg E_3] \cdot \Pr[b' = b \mid \neg E_3] + \Pr[b' = b \mid E_3]\Pr[E_3] \leqslant \Pr[\neg E_3] + \Pr[b' = b \mid E_3]\Pr[E_3] = \frac{1}{2} + \frac{1}{2}\Pr[\neg E_3]$ 可知，$\Pr[b' = b] \geqslant \Pr[b' = b \mid E_3]\Pr[E_3] = \frac{1}{2}\left(1 - \Pr[\neg E_3]\right)$，因此 $\frac{1}{2}\Pr[\neg E_3] \geqslant \left|\Pr[b' = b] - \frac{1}{2}\right| \geqslant \varepsilon$，即 $\Pr[\neg E_3] \geqslant 2\varepsilon$。挑战者 C 成功解决 DLDHP 的概率优势 $\varepsilon' \geqslant \frac{1}{2}\Pr[\neg E_3]\ \Pr[E_1 \wedge E_2] = \frac{\varepsilon}{4nq_t}$。

在上述游戏中，C 以不可忽略的概率 ε' 解决了 DLDHP，这与 DLDHP 的公认难解性矛盾。因此 A_1 攻破本文方案的概率 ε 是可忽略的值，在面对 A_1 时方案满足选择关键字攻击下关键字密文的不可区分性。证毕。

引理 2 在随机预言模型中，若敌手 A_2 能够以不可忽略的概率优势 ε 在多项式时间内攻破本文方案，则挑战者 C 可以构造算法在多项式时间内以不可忽

略的概率优势 $\varepsilon' \geqslant \frac{\varepsilon}{4nq_t}$ 解决 DLDHP。其中，q_t 表示陷门询问的最大执行次数，n 表示数据用户的数量。

证明　挑战者 C 可通过与敌手 A_2 的交互游戏解决 DLDHP。

给定元组 $F=(g_1,g_2,g_3,Z,v_1,v_2,v_3)$，其中 $v_1=g_1{}^a$，$v_2=g_2{}^b=g_2{}^s$。令 $p=v_2=g_2{}^S$，$g=g_1$，$g_2=g_1^{\alpha}=g^{\alpha}$。C 执行初始化算法生成 $\text{prms}=(e,g,G_1,G_2,p,q,H_1,H_2,H_3,H_4)$，并将秘密值 S 和 prms 发送给攻击者 A_2。此外，C 随机选择 $n\in Z_q^*$ 并秘密保存 n。

询问阶段 A_2 可对 C 进行下述一系列询问。H_1、H_2、H_3 和 H_4 询问与引理 1 相同。

1) 部分私钥询问。当 C 收到 A_2 对身份为 ID_i 的数据用户的部分私钥询问时，若元组 $(\text{ID}_i,x_i,R_{\text{ID}_i},D_{\text{ID}_i})$ 存在于 L_1^{list} 中，C 返回元组 $(R_{\text{ID}_i},D_{\text{ID}_i})$ 给 A_2；否则 C 随机选择 $x_i\in Z_q^*$ 和 $\alpha_i\in Z_q^*$，计算 $R_{\text{ID}_i}=g_2{}^{x_i}=g_1^{\alpha x_i}$，$\alpha_i=h_0(\text{ID}_i,R_{\text{ID}_i})$ 和 $D_{\text{ID}_i}=\alpha\ (x_i+\alpha_i s)$。然后 C 将元组 $(\text{ID}_i,R_{\text{ID}_i},\alpha_i)$ 和元组 $(\text{ID}_i,x_i,R_{\text{ID}_i},D_{\text{ID}_i})$ 存入 H_3^{list} 与 L_1^{list} 中，返回元组 $(R_{\text{ID}_i},D_{\text{ID}_i})$ 作为部分私钥询问的回复。

2) 公钥提取询问。当 C 收到 A_2 对身份为 ID_i 的数据用户的公钥询问时，若元组 $(\text{ID}_i,X_{D_{\text{ui}}},R_{\text{ID}_i})$ 存在于 L_2^{list} 中，C 输出元组 $(X_{D_{\text{ui}}},R_{\text{ID}_i})$ 作为公钥提取询问的回复；否则 C 随机选取 $\pi_i\in Z_q^*$ 和 $s_i\in Z_q^*$，计算 $X_{D_{\text{ui}}}=g_2{}^{\pi_i}$ 和 $R_{\text{ID}_i}=g_2{}^{s_i}$，其中 $g_2=g_1{}^{\alpha}$。最后 C 将元组 $(\text{ID}_i,X_{D_{\text{ui}}},R_{\text{ID}_i})$ 添加到 L_2^{list} 中，并输出元组 $(X_{D_{\text{ui}}},R_{\text{ID}_i})$ 作为公钥提取询问的回复。

3) 陷门询问。收到 A_2 对关键字 w_i 的陷门询问后，C 按如下步骤回复。

① C 从各列表中分别获取元组 (w_i,c_i,e_i,m_i)、(w_i,c_i,f_i,n_i)、$(\text{ID}_i,P,X_{D_{\text{ui}}},R_{\text{ID}_i},v_i)$ 以及 $(\text{ID}_i,x_i,R_{\text{ID}_i},D_{\text{ID}_i})$。

② C 随机选择 $r\in Z_q^*$，计算 $T_a=g_1{}^r$，$T_b=g_1{}^{rD_{\text{ID}_i}}$ 和 $T_c=g_1^{\frac{rn_i}{v_i\pi_i+x_i+\alpha_i s}}$，输出 $T_i=(T_a,T_b,T_c)$ 作为陷门询问的回复。

挑战阶段敌手 A_2 选择挑战关键字 $W^*=(w_{0,1},w_{0,2},\cdots,w_{0,n})$。C 选择 $R=(w_{1,1},w_{1,2},\cdots,w_{1,n})$，并令 $W_0=W^*$，$W_1=R$，A_2 不能询问 W_0 和 W_1 的陷门。然后 C 随机选择 $b\in\{0,1\}$，对所有关键字 $W_{b,i}$ 进行 H_1 询问与 H_2 询问，得到元组 $(w_{b,i},c_{b,i},e_{b,i},m_{b,i})$ 和 $(w_{b,i},c_{b,i},f_{b,i},n_{b,i})$。若所有的 $c_{b,i}\neq 1$，C 终止模拟；否则 C 计算 $A=(X_{D_{\text{oj}}}R_{\text{ID}_j}\xi)^a$ 和 $B_i=v_2^{(v_i\pi_i+x_i+s\alpha_i)n}=g_2^{(v_i\pi_i+x_i+s\alpha_i)bn}=(X_{D_{\text{ui}}}^{v_i}R_{\text{ID}_i}P^{\alpha_i})^{bn}$。当 $c_{b,i}=0$ 时，$c_{b,i}=v_1^{m_{b,i}(\pi_j+D_{\text{ID}_j})}v_2^{n_{b,i}n}$；当 $c_{b,i}=1$ 时，$c_{b,i}=v_3^{m_{b,i}(\pi_j+D_{\text{ID}_j})}$。C 发送挑战密文 $(A,B_1,B_2,\cdots,B_n,C_{b,1},C_{b,2},\cdots,C_{b,l})$、$W_0$ 和 W_1 给 A_2。

猜测阶段 A_2 输出猜测值 $b'\in\{0,1\}$，若 $b=b'$，令 $v_3=g_3^{a+b}$，C 可进行如下计算验证挑战密文的有效性；否则令 $v_3=Z$，C 终止游戏。

$$A^*=X_{D_{\text{oj}}}R_{\text{ID}_j}\xi \tag{7}$$

$$B_i^*=v_2^{(v_i\pi_i+x_i+s\alpha_i)n}=g_2^{(v_i\pi_i+x_i+s\alpha_i)bn}=(X_{D_{\text{ui}}}^{v_i}R_{\text{ID}_i}P^{\alpha_i})^{bn} \tag{8}$$

若 $c_{b,i}=0$，计算

$$c_{b,i}=v_1^{m_{b,i}(\pi_j+D_{\text{ID}_j})}v_2^{n_{b,i}n}=g_1^{am_{b,i}(\pi_j+D_{\text{ID}_j})}g_2^{bn_{b,i}n}=e_{b,i}^{a(\pi_j+D_{\text{ID}_j})}f_{b,i}^{bn} \tag{9}$$

若 $c_{b,i}=1$，计算

$$c_{b,i}=v_3^{m_{b,i}(\pi_j+D_{\text{ID}_j})}=g_3^{(a+b)m_{b,i}(\pi_j+D_{\text{ID}_j})}=g_3^{am_{b,i}(\pi_j+D_{\text{ID}_j})}\left[g_3^{\frac{m_{b,i}(\pi_j+D_{\text{ID}_j})}{n}}\right]^{bn}=e_{b,i}^{a(\pi_j+D_{\text{ID}_j})}f_{b,i}^{bn} \tag{10}$$

挑战者 C 在上述游戏中成功解决 DLDHP 的概率优势为 $\varepsilon'\geqslant\frac{1}{2}\Pr[\neg E_3]\Pr[E_1\wedge E_2]=\frac{\varepsilon}{4nq_t}$（相关计算过程与引理 1 类似），这与 DLDHP 的公认难解性矛盾。因此 A_2 攻破本文方案的概率 ε 是可忽略的值，在面对 A_2 时方案满足选择关键字攻击下关键字密文的不可区分性。证毕。

5　性能分析

5.1　功能性分析

本文方案与近些年的密文检索方案的功能性对比如表 2 所示。由表 2 可知，文献[11]方案不支持多用户多关键字搜索。文献[19]方案不具有无证书加密体制的优点。文献[13,19]方案不能抵抗 IKGA。文献[11,16]方案安全性较好，但不支持云边协同计算，在关键字索引不完全包含检索关键字的非精确匹配场景下无法高效返回正确的搜索结果。

本文方案支持多用户多关键字搜索，能够抵抗 IKGA 且引入云边协同计算技术提高了检索效率，具有功能性丰富的优点。

表 2 功能性对比

方案	抵抗 IKGA	多用户多关键字	云边协同	无证书
文献[11]方案	√	×	×	√
文献[13]方案	×	√	×	√
文献[16]方案	√	√	×	√
文献[19]方案	×	√	√	×
本文方案	√	√	√	√

5.2 计算开销分析

本节首先从理论分析角度将本文方案与同类方案在密文生成、陷门生成以及匹配测试阶段的计算开销进行对比，结果如表 3 所示。其中，E 表示一次指数运算，P 表示一次双线性对操作，h 表示一次哈希到点运算，n 表示数据用户的数量，l 表示关键字索引的数量，l' 表示数据用户检索的关键字数量。

由表 3 可知，本文方案在密文生成阶段与陷门生成阶段仅需进行指数运算和哈希到点运算，与其他同类方案相比计算开销较低。在匹配测试阶段，每进行一次关键字匹配，文献[14]方案比本文方案少计算一个双线性对运算，而文献[17]方案只需进行指数运算。本文方案在匹配测试阶段的计算开销高于文献[14]方案和文献[17]方案，但本文方案能够在关键字索引不完全包括检索关键字的非精确匹配情况下返回正确搜索结果。此外，本文方案支持多个边缘服务器同时进行关键字匹配，随着文件数量的增加，本文方案在匹配测试阶段的计算开销优势逐渐明显。

在装有 Intel Core i7-6500 2.60GH 处理器和 8 GB 内存的 Windows 系统上使用 PBC-0.47-VC 软件包进行仿真实验，并将本文方案与同类密文检索方案在密文生成、陷门生成以及测试阶段的计算开销进行对比。

图 2 展示了密文生成时间随数据用户数量的变化，为便于比较关键字索引数量，设 l =100。由图 2 可知，本文方案的密文生成时间明显少于对比方案，且密文生成时间不随数据用户的增加呈线性增长。在本文方案中，每增加一个搜索用户只需计算该用户身份密文信息 B_i 且 B_i 的计算开销较小，本文方案在密文生成阶段的计算开销优势会随用户数量的增加越来越明显。

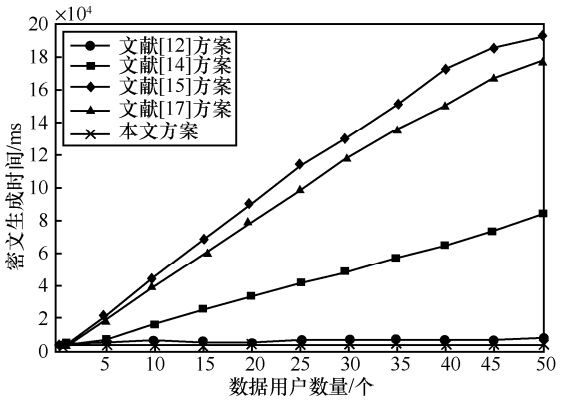


图 2 密文生成时间随数据用户数量的变化

图 3 展示了陷门生成时间随搜索关键字数量的变化。本文方案与对比方案的陷门生成时间均随用户搜索关键字数量 l' 的增加而增加，但由图 3 可知，本文方案与其他方案相比，陷门生成时间较少。

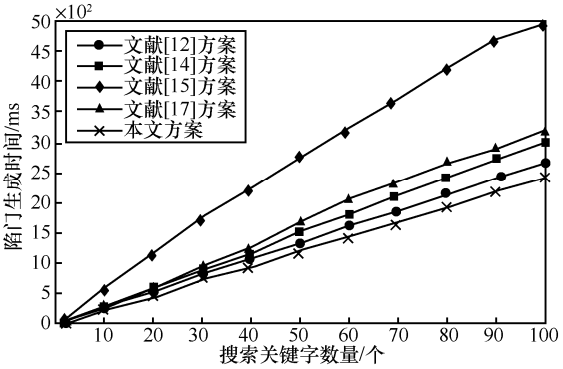


图 3 陷门生成时间随搜索关键字数量的变化

表 3 计算开销对比

方案	密文生成	陷门生成	匹配测试
文献[12]方案	$(2l+3n)E+(2l+n)P+2lh$	$3l'E+l'h$	$3l'P+l'h$
文献[14]方案	$(nl+2n)E+nlh$	$l'E+l'P+l'h$	$2l'P$
文献[15]方案	$7nlE+nlh$	$7l'E+l'h+l'P$	$3l'E+2l'P$
文献[17]方案	$3nlE+nlh+nlP$	$l'E+l'P+l'h$	$l'E$
本文方案	$(3n+2l+1)E+2lh$	$3E+2l'h$	$3l'P$

图 4(a)和图 4(b)分别比较了边缘服务器个数为 10 和 100 时，文件数量的增加对本文方案与同类方案的匹配测试时间的影响。为便于观察，假定每个文件包含 100 个关键字。随着文件数量的增加，本文方案的计算开销优势逐渐增加，这是因为本文方案支持边缘服务器同时进行陷门匹配计算，数据量越大本文方案在匹配测试阶段的计算开销优势越明显。边缘服务器数量越多，本文方案的计算效率越高，因此本文方案适用于文件数目和边缘节点众多的 IIoT 数据共享环境。

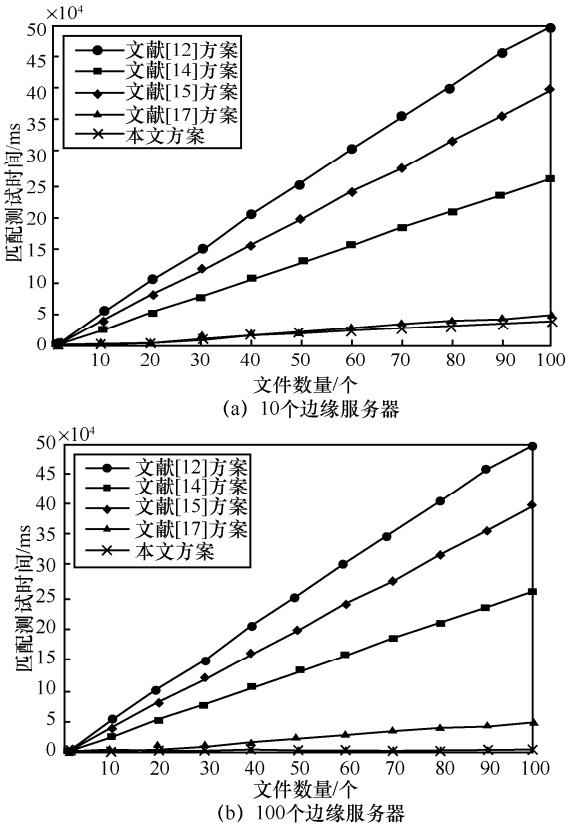


图 4　匹配测试时间随文件数量的变化

6　结束语

面向工业物联网环境，本文基于云边协同计算的思想提出了一个高效的无证书多用户多关键字密文数据检索方案。该方案实现了关键字密文认证且支持更新用户的文件访问权限，并能够在关键字索引不完全包含检索的多个关键字的情况下实现多用户密文数据检索。经过理论分析与 PBC 密码库仿真分析，本文方案计算效率较高。在随机预言模型下，本文方案能够抵抗 IKGA。下一步的工作任务是在标准模型下，设计工业物联网环境中安全高效的密文数据检索方案。

参考文献：

[1] WU H, TIAN H, NIE G F, et al. Wireless powered mobile edge computing for industrial Internet of things systems[J]. IEEE Access, 2020, 8: 101539-101549.

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

[3] LIU Z L, WENG J, LI J, et al. Cloud-based electronic health record system supporting fuzzy keyword search[J]. Soft Computing, 2016, 20(8): 3243-3255.

[4] BONEH D, DI C G, OSTROVSKY R, et al. Public key encryption with keyword search[C]//2004 International Conference on the Theory and Applications of o Cryptographic Techniques. Berlin: Springer, 2004: 506-522.

[5] PENG Y G, CUI J T, PENG C G, et al. Certificateless public key encryption with keyword search[J]. China Communications, 2014, 11(11): 100-113.

[6] ZHENG Q J, LI X X, AZGIN A. CLKS: certificateless keyword search on encrypted data[M]. Cham: Springer International Publishing, 2015.

[7] MA M M, HE D B, KHAN M K, et al. Certificateless searchable public key encryption scheme for mobile healthcare system[J]. Computers & Electrical Engineering, 2018, 65: 413-424.

[8] WU T Y, MENG C, CHEN C M, et al. On the security of a certificateless public key encryption with keyword search[C]//Advances in Intelligent Information Hiding and Multimedia Signal Processing. Berlin: Springer, 2017: 191-197.

[9] GOLLE P, STADDON J, WATERS B. Secure conjunctive keyword search over encrypted data[C]//Applied Cryptography and Network Security. Berlin: Springer, 2004: 31-45.

[10] CURTMOLA R, GARAY J, KAMARA S, et al. Searchable symmetric encryption: improved definitions and efficient constructions[J]. Journal of Computer Security, 2011, 19(5): 895-934.

[11] 张玉磊, 文龙, 王浩浩, 等. 多用户环境下无证书认证可搜索加密方案[J]. 电子与信息学报, 2020, 42(5): 1094-1101.
ZHANG Y L, WEN L, WANG H H, et al. Certificateless authentication searchable encryption scheme for multi-user[J]. Journal of Electronics & Information Technology, 2020, 42(5): 1094-1101.

[12] SUN L X, XU C X, LI C, et al. Server-aided searchable encryption in multi-user setting[J]. Computer Communications, 2020, 164: 25-30.

[13] MA M M, FAN S Q, FENG D G. Multi-user certificateless public key encryption with conjunctive keyword search for cloud-based telemedicine[J]. Journal of Information Security and Applications, 2020, 55: 102652.

[14] PAN X Y, LI F G. Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability[J]. Journal of Systems Architecture, 2021, 115: 102075.

[15] WU L B, ZHANG Y B, MA M M, et al. Certificateless searchable public key authenticated encryption with designated tester for cloud-assisted medical Internet of Things[J]. Annals of Telecommunications, 2019, 74(7/8): 423-434.

[16] CHENAM V B, ALI S T. A designated cloud server-based multi-user certificateless public key authenticated encryption with conjunctive

keyword search against IKGA[J]. Computer Standards & Interfaces, 2022, 81: 103603.

[17] PAKNIAT N, SHIRALY D, ESLAMI Z. Certificateless authenticated encryption with keyword search: enhanced security model and a concrete construction for industrial IoT[J]. Journal of Information Security and Applications, 2020, 53: 102525.

[18] MA M M, HE D B, KUMAR N, et al. Certificateless searchable public key encryption scheme for industrial Internet of things[J]. IEEE Transactions on Industrial Informatics, 2018, 14(2): 759-767.

[19] 黄海平, 杜建澎, 戴华, 等. 一种基于云存储的多服务器多关键字可搜索加密方案[J]. 电子与信息学报, 2017, 39(2): 389-396.
HUANG H P, DU J P, DAI H, et al. Multi-sever multi-keyword searchable encryption scheme based on cloud storage[J]. Journal of Electronics & Information Technology, 2017, 39(2): 389-396.

[20] 张强, 王国军, 张少波. 基于多边缘服务器的个性化搜索隐私保护方法[J]. 通信学报, 2019, 40(2): 40-50.
ZHANG Q, WANG G J, ZHANG S B. Method of privacy protection based on multiple edge servers in personalized search[J]. Journal on Communications, 2019, 40(2): 40-50.

[21] SHAO J, CAO Z F, WANG L C, et al. Proxy re-signature schemes without random oracles[R]. IACR Cryptology EPrint Archive, 2007.

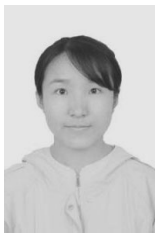
[22] HUANG Q, LI H B. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks[J]. Information Sciences, 2017, 403/404: 1-14.

[作者简介]

杨小东（1981− ），男，甘肃甘谷人，博士，西北师范大学教授，主要研究方向为信息安全及云计算安全。

田甜（1998− ），女，甘肃兰州人，西北师范大学硕士生，主要研究方向为信息安全及密码学。

王嘉琪（1997− ），女，甘肃兰州人，西北师范大学硕士生，主要研究方向为密码学及信息安全。



李梅娟（1997− ），女，甘肃临洮人，西北师范大学硕士生，主要研究方向为密码学及信息安全。

王彩芬（1963− ），女，河北安国人，博士，深圳技术大学教授，主要研究方向为密码学及信息安全。