

云环境下基于属性策略隐藏的可搜索加密方案

周艺华, 扈新宇, 李美奇, 杨宇光

(1. 北京工业大学信息学部 北京 100124; 2. 可信计算北京市重点实验室 北京 100124)

摘 要: 基于属性的可搜索加密技术可以实现对数据的细粒度访问控制, 但现有的可搜索加密方案, 关键字的搜索、访问控制、文件加密基本上是分别执行的, 导致攻击者可能跳过访问策略直接进行关键字索引匹配或文件解密; 其次, 现有方案中数据所有者需将加密文件的密钥以安全通道传给用户, 增加了数据拥有者的开销; 此外, 大多基于树型的访问控制策略是公开的, 容易造成隐私泄露。因此, 基于线性秘密分享(LSSS, linear secret sharing schemes)访问结构, 提出了一种云环境下基于属性策略隐藏的可搜索加密方案。通过将策略秘密值嵌入关键字加密与文件存储加密, 实现访问控制、关键字搜索与文件加密的有机结合; 通过聚合密钥技术实现用户无须与数据所有者交互, 即可对文件进行解密的功能, 减轻了密钥管理的负担, 存储空间提高约30%。实验结果及安全性分析表明, 所提方案具有数据存储的安全性、访问策略的隐私性、陷门的不可连接性等功能, 具有较高的密文检索效率, 与已有主流方案相比, 检索效率提高至20%以上。

关键词: 可搜索加密; 属性基加密; 策略隐藏; 聚合密钥

中图分类号: TP309

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2022019

Searchable encryption scheme based on attribute policy hiding in a cloud environment

ZHOU Yihua, HU Xinyu, LI Meiqi, YANG Yuguang

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
2. Beijing Key Laboratory of Trusted Computing, Beijing 100124, China

Abstract: Attribute-based searchable encryption technology can achieve fine-grained access control of data, but the existing searchable encryption scheme, keyword search, access control and file encryption are basically performed separately, causing the attacker to directly skip the access policy for keyword index matching and file decryption. Besides, the data owners in the existing schemes need to pass the key of the encrypted file to the user in a secure channel, which increases the cost of the data owner. Furthermore, most tree-based access control policies are open

收稿日期: 2021-11-05; 修回日期: 2022-03-02

通信作者: 扈新宇, hxy@emails.bjut.edu.cn

基金项目: 国家自然科学基金(62071015)

Foundation Item: The National Natural Science Foundation of China(62071015)

引用格式: 周艺华, 扈新宇, 李美奇, 等. 云环境下基于属性策略隐藏的可搜索加密方案[J]. 网络与信息安全学报, 2022, 8(2): 112-121.

Citation Format: ZHOU Y H, HU X Y, LI M Q, et al. Searchable encryption scheme based on attribute policy hiding in a cloud environment[J]. Chinese Journal of Network and Information Security, 2022, 8(2): 112-121.

and easy to cause privacy leakage. Therefore, based on the LSSS (linear secret sharing schemes) access architecture, the searchable encryption scheme based on attribute policy hiding in a cloud environment was proposed. Through the embedding of policy secret values into keyword encryption and file storage encryption, the combination of access control, keyword search and file encryption were realized. The aggregate key technology enables users to decrypt files without interacting with the data owner, reducing the burden of key management and increasing storage space by approximately 30%. The experimental results and security analysis show that the proposed scheme guarantees the security of stored data, privacy of access strategy and non-connectivity of trap gate. Compared with the existing mainstream scheme, the retrieval efficiency of the proposed scheme has improved to more than 20%.

Keywords: searchable encryption, attribute-based encryption, policy hiding, aggregation key

0 引言

随着云计算的快速普及,许多组织将其数据和服务外包给云,但带来了严重的隐私泄露问题。因此,很多用户将敏感数据外包给云服务之前对其进行加密。然而,加密用户的数据妨碍了云应用程序的搜索功能。为了解决此类问题,Song等^[1]首次提出了可搜索加密,它允许以加密形式外包数据,同时仍然能够进行搜索。这些方案通常包括以下步骤:数据所有者(DO, data owner)生成其加密文档和可搜索加密索引,将加密文档和加密索引外包给云。当搜索关键字时,用户(DU, data user)生成一个陷门提交到云服务器上,云服务器可以搜索加密索引并返回相关文档。

目前,大多数方案以模糊关键词^[2-4]和排序搜索关键字^[5-6]为核心,Liu等^[2]采用向量内积方式实现关键字搜索,文件解密密钥需要用户间进行密钥协商;Ahmed等^[5]采用可搜索对称加密方式,根据文档与查询的相关性对文档进行排序,任何用户通过上传搜索陷门,都可以获取所需的文档。但近年来,受基于属性的加密(ABE, attribute-based encryption)概念的启发,将访问控制与可搜索加密相结合,加密索引可以通过关键字进行搜索,搜索权限由ABE控制,用户实现了更细粒度的关键字搜索。文献[7-10]提出基于属性的可搜索加密方案,由云端直接完成访问策略的验证,若成功,则只将密文返回给用户端解密,密文的解密与策略不相关,导致攻击者可能跳过访问策略直接进行索引匹配和文件解密。

Niu等^[11]和Miao^[12]、Wu等^[13]分别将访问策略转化为访问树和“AND”门,与其方案相比,

本文方案采取线性秘密共享矩阵的访问结构,能够支持任意表达的访问策略,并进行了策略隐藏。Yan等^[14]和Waters^[15]的方案采用访问策略对每个文件进行加密,并未通过关键字搜索,虽将访问策略转化为线性秘密共享矩阵,然后进行文件的加密,但并未通过对关键字搜索,文件获取效率不高。

针对现有问题,本文主要贡献如下。

1) 利用基于属性的加密和可搜索加密技术,采取线性秘密共享矩阵构造访问策略,将访问控制与关键字搜索、文件加密相结合,保证访问策略与关键字都正确才匹配成功,进行文件存储地址的解密。同时,对属性值进行隐藏,访问策略中任何有价值的属性信息都不会透露给未经授权的接收者。

2) 不同的文件采用不同的密钥加密,用户接收一个聚合密钥,就能解密出所需的文件。

3) 陷门不可连接性。保证即使多次查询的属性或关键字相同,攻击者也无法区分。

在Linux Ubuntu64位操作系统下利用双线性对包,并用Python语言进行编程。实验结果表明,本文方案的效率高于文献[8]方案和文献[13]方案。

1 预备知识

1.1 双线性映射

令 G 和 G_T 是两个阶为素数 p 的乘法循环群,存在双线性映射 $e: G \times G \rightarrow G_T$ 满足以下3个性质^[16]。

1) 双线性。对于任意 $x, y \in G$, 存在 $a, b \in \mathbb{Z}_p$, 使等式 $e(x^a, y^b) = e(x, y)^{ab}$ 成立。

2) 非退化性。存在 $x, y \in G$, 满足 $e(x, y) \neq 1$ 。

3) 可计算性。对于任意 $x, y \in G$ ，存在有效的算法计算 $e(x, y)$ 。

1.2 线性秘密共享矩阵

令 U 是一个属性集，则

1) 选择秘密值 $s \in Z_p$ ；

2) 共享向量中每个元素对应 U 中的一个属性，共享向量中的所有元素都在域 Z_p 中；

3) 对于 U 中的每个访问结构，存在一个 $l \times n$ 的矩阵 M ，称为共享生成矩阵。

在矩阵 $M_{l \times n}$ 中，有一个映射函数 ρ ，它将第 i 行映射到 U 中的属性 Attr ，即 $\rho(i) \rightarrow \text{Attr}$ ， $\text{Attr} \in U$ ， $i = \{1, 2, \dots, l\}$ 。再生成秘密值 $s \in Z_p$ ，考虑一个列向量 $v = (s, r_2, \dots, r_n)^T$ ， $r_2, \dots, r_n \in Z_p$ 。 $\lambda = M_{l \times n} v$ 是 l 个秘密因子的向量，每个秘密因子 λ_i 对应属性 $\rho(i)$ 。通常称 $(M_{l \times n}, \rho)$ 为访问策略。

线性重构特性：假设方案的访问结构是线性秘密共享矩阵^[17]。设 S 是一个被授权的属性集，且 $I = \{i : \rho(i) \in S\}$ 。存在一组常数 $\{\omega_i \in Z_p\}_{i \in I}$ ，满足 $\sum_{i \in I} \omega_i M_i = (1, 0, \dots, 0)$ ，使如果 $\{\lambda_i\}$ 是访问结构的有效秘密因子，则可以用来计算秘密值 s ，即 $\sum_{i \in I} \omega_i \lambda_i = s$ 。

1.3 基于 LSSS 的隐藏结构

在隐藏结构中，与矩阵 M 第 i 行相关联的属性 $\{t_i : v_i\}$ 由属性类别 t_i 和属性值 v_i 组成，定义一个映射函数 $\pi : \pi(i) \rightarrow t_i$ ，将每一行 i 映射到相应的类别 t_i ，将 $\{M_{l \times n}, \pi\}$ 以明文形式作为访问结构，对属性值 v_i 加密。云服务器根据用户的属性类别匹配到相应的矩阵行数，判别是否符合访问策略，从而无法获知用户的具体属性值，保证属性值的隐藏。

1.4 困难假设

CDH (computational Diffie-Hellman) 假设^[18]。根据系统安全参数选择一个阶为素数 p 的乘法循环群 G ， g 是 G 的一个生成元，随机选择 $a, b \in Z_p$ ，如果不存在一种算法能够在多项式时间内以不可忽略的概率区分 (g^a, g^b) 和 g^{ab} ，则 CDH 假设成立。

DBDH (decisional bilinear Diffie-Hellman assumption) 假设^[19]。 g 是群 G 的一个生成元，

给定两个元组 $(g, g^a, g^b, g^c, e(g, g)^{abc})$ 和 $(g, g^a, g^b, g^c, e(g, g)^z)$ ，随机选择 $a, b, c, z \in Z_p$ ，如果不存在一种算法能够在多项式时间内以不可忽略的概率区分 $e(g, g)^{abc}$ 和 $e(g, g)^z$ ，则 DBDH 假设成立。

DDH (decisional Diffie-Hellman) 假设。 g 是群 G 的生成元，给定两个元组 (g^a, g^b, g^z) 和 (g^a, g^b, g^{ab}) ，随机选择 $a, b, z \in Z_p$ ，如果不存在概率多项式时间的攻击者以不可忽略的优势区分 g^{ab} 和 g^z ，则 DDH 假设成立。

1.5 符号描述

本节介绍本文中的一些符号和变量，符号描述如表 1 所示。

表 1 符号描述
Table 1 symbol description

符号	说明
γ	系统公共参数
Z_p	一个有限域，其中 p 是一个大的素整数
G, G_T	素数阶 p 的两个循环群
PP	系统公共参数
MSK	系统主密钥
H_1, H_2	哈希函数
k	用户密钥，解密存储地址密文
k_1	用户密钥，加密关键字
D_{x_1}, D_{x_2}	用户属性密钥
T	用户生成关键字查询密文
F_i	文件明文
D_i	文件密文
$Y_{0,i}, Y_{1,i}$	文件密钥，用户解密文件密文
C_2	文件存储地址密文
$C_{4,i}, C'_{4,i}$	数据拥有者属性密钥
C_w	数据拥有者生成关键字索引密文
Z	部分解密密钥，解密存储地址 Add 和文件 F

2 本文方案

2.1 系统模型

本系统主要包括 4 类实体，分别是数据拥有者、用户、云服务提供商 (CSP, cloud service provider)、属性授权中心 (AA, attribute authorization)。系统模型及其流程如图 1 所示。

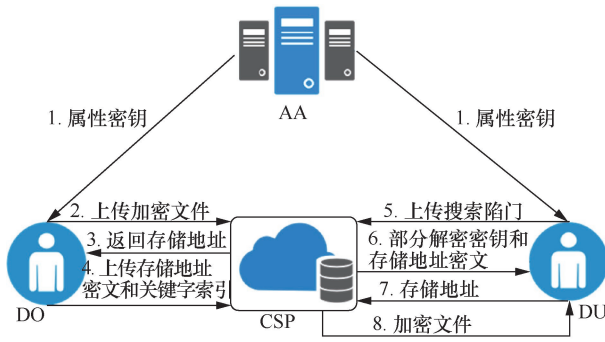


图1 系统模型及其流程
Figure 1 System model and its process

1) 属性授权中心。属性授权中心是一个完全可信的实体，负责生成系统的公钥、主密钥和公共参数。此外，负责为用户的属性分发属性密钥。

2) 数据拥有者。数据拥有者对共享文件进行加密，将密文上传到云服务器，获取文件存储地址。然后，用访问策略对关键字加密，生成关键字索引，与加密后的文件存储地址一并上传到云服务器。

3) 用户。每一用户都拥有与其属性集相关的私钥。同时，用户在 AA 的授权下，生成关键字搜索陷门，由云服务器进行搜索匹配。当符合访问策略、关键字匹配时，接收到返回的部分解密密钥和存储地址密文，解密得到文件存储地址，从而进一步访问云端。云服务器返回加密文件，用户对其进行解密，获取所需文件。

4) 云服务提供商。云服务提供商负责存储加密文件，并将所有匹配的文件密文发送给用户。同时负责关键字的匹配工作，检查用户属性值与访问策略是否匹配，关键字是否匹配，最后将存储地址密文与部分解密密钥返回给用户。

2.2 算法形式化定义

本文提出的方案包括以下算法。

1) 系统建立。Setup(1^γ) $\rightarrow$ (PP, MSK): 该算法由属性授权中心执行，输入安全参数 γ ，生成系统公共参数 PP 和主密钥 MSK。

2) 密钥生成。Keygen(PP, MSK, ϕ) $\rightarrow$ (SK): 该算法由属性授权中心执行，输入公共参数 PP、主密钥 MSK 和用户属性集 ϕ ，为用户生成私钥 SK。

3) 搜索陷门。Trapdoor(PP, SK, q) $\rightarrow T_w$: 该

算法由用户执行，输入公共参数 PP、用户密钥 SK、关键字 q ，输出陷门 T_w 。

4) 加密。Encrypt(PP, A , KW, F) $\rightarrow$ (CT, I_w , C_2): 该算法由数据拥有者执行，输入公共参数 PP、访问结构 A 、关键字集 KW、数据文件 F ，生成文件密文 CT、关键字索引 I_w 和存储地址密文 C_2 存储于云服务器。

5) 搜索。Search(PP, I_w , T_w) $\rightarrow$ (Z , $C_2/\perp$): 该算法由云服务器执行，输入公共参数 PP、关键字索引 I_w 和搜索陷门 T_w ，若用户满足访问策略并关键字匹配成功，则返回部分解密密钥 Z 和存储地址密文 C_2 ；否则，输出 $\perp$ 。

6) 解密。Decrypt(PP, SK, Z , C_2 , CT) $\rightarrow$ (Add, F): 该算法由用户执行，输入公共参数 PP、用户的私钥 SK、部分解密密钥 Z 、存储地址密文 C_2 ，解出文件存储地址 Add，然后解密文件密文 CT，输出共享数据文件 F 。

2.3 安全模型

在本文方案中，假定授权中心是完全可信的，且在为数据用户生成密钥前验证数据用户的身份，而云服务器是“诚实但好奇”的实体，即它会诚实地为用户提供服务，但也会利用其拥有的信息从密文及陷门中探寻任何私有信息。

本文方案主要考虑关键字和陷门不可区分。通过攻击者 A 和挑战者 B 之间游戏来定义，保证语义安全和密文的不可区分。

游戏 1 关键字不可区分性。

1) Setup 阶段。挑战者 B 通过调用 Setup 算法获取公共参数 PP。A 自适应地向 B 发出不同的询问。

2) 询问阶段。A 向 B 请求对应属性集的私钥，B 运行 Keygen 算法得到相应的私钥 SK 并返回给 A。

3) 挑战。执行过一定数量的上述询问后，A 向 B 发送两个大小相同的关键字 kw_0 和 kw_1 ，B 随机选择 $b \in \{0, 1\}$ ，利用 kw_b 生成加密关键字 I_{w_b} 且返回 I_{w_b} 给 A。

A 能够向 B 对任意属性集 ϕ 进行密钥询问，任意关键字 $kw(kw \neq kw_0^*, kw \neq kw_1^*)$ 进行密文询问。

4) 猜测阶段。A 输出 $b' \in \{0, 1\}$ ，当且仅当

$b' = b$ 时 A 赢得游戏 1。

本文定义在自适应选择关键字攻击下 A 成功区分密文的优势为 $\text{ADV}_A^B(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right|$ 。

游戏 2 陷门不可区分性。

1) Setup 阶段。挑战者 B 通过调用 Setup 算法获取公共参数 PP。A 自适应地向 B 发出不同的询问。

2) 密钥询问阶段。A 向 B 发出注册请求, B 运行 Keygen 算法得到相应的私钥 SK 并返回给 A。陷门询问阶段。A 对关键字 q 做出询问, B 运行 Trapdoor 算法得到搜索令牌 T_w 并返回 T_w 给 A。

3) 挑战。执行过一定数量的上述询问后, A 向 B 发送两个大小相同的关键字 q_0 和 q_1 , B 随机选择 $b \in \{0, 1\}$, 利用 q_b 生成加密关键字 T_{q_b} 且返回 T_{q_b} 给 A。A 能够向 B 对任意关键字 $q (q \neq q_0, q \neq q_1)$ 进行陷门询问。

4) 猜测阶段。A 输出 $b' \in \{0, 1\}$, 当且仅当 $b' = b$ 时 A 赢得游戏 2。A 成功赢得这个游戏的优势为 $\text{ADV}_A^B(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right|$ 。

2.4 算法设计

步骤 1 属性授权中心分发属性密钥。

1) 系统初始化

输入安全参数 γ , 给定双线性映射 $e: G \times G \rightarrow G_T$, g 为 G 的生成元, G 和 G_T 是素数阶为 p 的两个乘法循环群。此外, 定义两个哈希函数 $H_1: \{0, 1\}^* \rightarrow G, H_2: \{0, 1\}^* \rightarrow G$ 。随机选择 $\mu, \beta, a \in Z_p$, 生成系统公共参数 PP 和主密钥 MSK, PP 由 AA 公开, MSK 由 AA 秘密保存。PP = $\{e(g, g)^\mu, g^\beta, g^a, H_1, H_2\}$, MSK = $\{\mu, \beta, a\}$ 。

2) 用户密钥生成

① 用户注册时, 属性授权中心为用户选择一个随机值 $r \in Z_p$, 生成用户密钥 $k = g^\mu g^{ar}$, $k_1 = g^{\frac{ar}{\beta}}$ 。同时, 用户将其属性集 φ 以映射 $\{\text{Cate}_x: \text{Value}_x\}_{x=1}^{|\varphi|}$ 形式提交给 AA, AA 根据属性集 φ , 对每个属性选取随机值 $t \in Z_p$, 计算用户属性密钥 $D_{1,x} = g^{at}$, $D_{2,x} = g^r H_1 \cdot (\{\text{Cate}_x: \text{Value}_x\})^{-t}$ 。

② 输出 DU 密钥, 通过安全信道传送给用户。

$$\text{SK} = \begin{cases} D_{1,x} = g^{at}, D_{2,x} = \\ g^r H_1(\{\text{Cate}_x: \text{Value}_x\})^{-t}, x \in (1, 2, \dots, |\varphi|) \\ k = g^\mu g^{ar}, k_1 = g^{\frac{ar}{\beta}} \end{cases} \quad (1)$$

步骤 2 数据拥有者上传加密文件。

文件加密: DO 对每一个文件 $F_i \in F$, 文件所属 id 为 I_{id} , 为其随机选择 $\tau_i \in Z_p$, 计算加密密钥 $k_{si} = e(g, g)^{\tau_i}$, 生成密文 $D_i = \text{Enc}(F_i, k_{si})$ 。计算 $Y_{0,i} = g^{a\tau_i}$, 用于用户解密密文密钥。

DO 发送文件密文 CT = $\{D_i, Y_{0,i}\}_{i \in |F|}$ 到 CSP,

CSP 返回存储地址 Add 给 DO。

步骤 3 数据拥有者上传存储地址密文和关键字索引。

关键字加密: DO 将访问策略转换为 LSSS 访问矩阵 $M_{l \times n}$, 与 M_i 相关联的属性 $\{t_i: v_i\}_i^l$ 由属性类别 t_i 和属性值 v_i 组成, M_i 是 M 的第 i 行。映射函数 $\pi: \pi(i) \rightarrow t_i$ 将每一行 i 映射到相应的类别 t_i 。

因此, 将 $\{M_{l \times n}, \pi\}$ 以明文形式作为访问结构, v_i 隐藏在密文中, 保证属性值的隐藏。

1) 随机选择 $s \in Z_p$ 和一个向量 $\mathbf{v} = (s, r_2, \dots, r_n)^T \in Z_p$ 。

2) 计算 $\lambda = M_{l \times n} \mathbf{v}$ 。

3) 将文件存储地址 Add 加密, $C_2 = \text{Add} \cdot e(g, g)^{\mu s}, C_3 = g^s$;

对文件所属 id 进行加密, 目的是云服务器为匹配文件生成聚合密钥, $Y_{1,i} = g^{-\tau_i I_{id}}$;

对每个属性 Att 加密, $C_{4,i} = g^{a\lambda_i}, C'_{4,i} = H_1(\{\pi(i): v_i\})^{\lambda_i}, \forall i \in \text{Att}$;

用访问策略的秘密值 s 对关键字 kw 加密, $C_w = e(g^\beta, H_2(\text{kw})^s), C'_w = g^{\beta s}$;

关键字索引 $I_w = \{I_{id}, C_3, C_w, C'_w, (C_{4,i}, C'_{4,i})_{\forall i \in \text{Att}}, M_{l \times n}, \pi\}$;

DO 对关键字和存储地址加密, 将 $\{I_w, C_2\}$ 上传到 CSP。

步骤 4 用户上传搜索陷门。

陷门生成: 设关键字集为 $\{q_1, q_2, q_3, \dots, q_n\} \in Q$,

用户选取随机值 $r_a \in Z_p$ ，保证陷门的不可连接。

计算 $D'_{1,x} = D_{1,x}^{r_a} = g^{\text{atr}_a}$, $D'_{2,x} = D_{2,x}^{r_a} = g^{r_a} H_1(\{\text{Cate}_x : \text{Value}_x\})^{-r_a}$ ，对关键字 q 加密 $T = k_1^{r_a} g^{\beta^{-1}}$
 $H_2(q) = g^{\frac{\text{arr}_a}{\beta}} H_2(q)$ 。所以，陷门为

$$T_w = \begin{cases} D'_{1,x} = g^{\text{atr}_a}, D'_{2,x} = \\ g^{r_a} H_1(\{\text{Cate}_x : \text{Value}_x\})^{-r_a}, x \in (1, 2, \dots, |\phi|) \\ T = g^{\frac{\text{arr}_a}{\beta}} H_2(q) \end{cases} \quad (2)$$

并将其存储于云服务器。

步骤5 云服务器搜索

首先根据用户提交的搜索陷门，并根据用户属性密钥检测是否符合访问策略，进一步验证关键字是否匹配。

1) 根据用户上传的属性类别 Cate_x 和数据拥有者上传的映射函数 $\pi: \pi(i) \rightarrow t_i$ ，找到第 x 个属性在矩阵 $\mathbf{M}_{l \times n}$ 中对应的第 i 行。计算 F_t ，当符合访问策略时，存在一组常数 $\{\omega_i \in Z_p\}_{i \in |\phi|}$ ，使 $\sum_{i=1}^{|\phi|} \lambda_i \omega_i = s$ 。

$$F_t = \prod_{i \in |\phi|} e(C_{4,i}, D'_{2,x}) e(C'_{4,i}, D'_{1,x})^{\omega_i} = \prod_{i \in |\phi|} e(g^{a\lambda_i}, g^{r_a} H_1(\{\text{Cate}_x : \text{Value}_x\})^{-r_a}) \cdot e(H_1(\{\pi(i): v_i\})^{\lambda_i}, g^{\text{atr}_a})^{\omega_i} = \prod_{i \in |\phi|} e(g, g)^{r_a a \lambda_i \omega_i} = e(g, g)^{r_a s} \quad (3)$$

其中， $\sum_{i=1}^{|\phi|} \lambda_i \omega_i = s$ 。

2) 计算

$$H = \frac{e(C_w, T)}{F_t} = \frac{e(g^{\beta s}, g^{\frac{\text{arr}_a}{\beta}} H_2(q))}{e(g, g)^{r_a s}} = \frac{e(g^{\beta s}, g^{\frac{\text{arr}_a}{\beta}}) e(g^{\beta s}, H_2(q))}{e(g, g)^{r_a s}} = e(g^{\beta s}, H_2(q)) \quad (4)$$

验证 H 是否等于 $C_w = e(g^{\beta}, H_2(kw)^s)$ ，若等式成立，表明搜索成功；若等式不成立，则表明搜索失败。用户的属性集不满足访问策略或者搜索时关键字不同，都会导致算法终止，返回 $\perp$ 。

此外，为符合条件的文件 CT，生成聚合密钥

$$k_{\text{agg}} = (g^a g^{\prod_{j \in [CT]} I_{idj}})^{-1} = g^{\frac{1}{a - \prod_{j \in [CT]} I_{idj}}}, \text{ 并 计 算 } C_{\text{pub},i} = Y_{1,i}^{\prod_{j \in [CT]} I_{idj}} = g^{-\tau_i \prod_{j \in [CT]} I_{idj}}, \text{ 用来解密文件密文。}$$

部分解密密钥

$$Z = \begin{cases} F_t = e(g, g)^{r_a s}, C_3 = g^s \\ k_{\text{agg}} = g^{\frac{1}{a - \prod_{i \in [CT]} I_{idj}}}, C_{\text{pub},i} = g^{-\tau_i \prod_{j \in [CT]} I_{idj}} \end{cases} \quad (5)$$

将 $\{Z, C_2\}$ 返回给用户，用来解密存储地址

Add 和文件 F 。

步骤6 用户解密

1) 解密存储地址

用户使用私钥 k 、 r_a 、 F_t 对存储地址密文 C_2 进行解密，保证只有用户自己才能计算，且需要用到访问结构中的秘密值 s ，防止攻击者跳过访问策略。用户计算

$$\text{Add}' = \frac{C_2(F_t)^{1/r_a}}{e(C_3, k)} = \frac{\text{Add} \cdot e(g, g)^{s s} (e(g, g)^{r_a s})^{1/r_a}}{e(g^s, g^{\mu} g^{ar})} = \text{Add} \quad (6)$$

2) 解密文件密文

当用户解出存储地址 Add 后，用户访问云服务器，云服务器返回存储地址对应的密文 $\text{CT} = \{D_i, Y_{0,i}\}_{i \in [CT]}$ ，用户计算

$$e(Y_{0,i} C_{\text{pub},i}, k_{\text{agg}}) = e(g^{a\tau_i} g^{-\tau_i \prod_{j \in [CT]} I_{idj}}, g^{\frac{1}{a - \prod_{j \in [CT]} I_{idj}}}) = e(g, g)^{\tau_i} = k_{si} \quad (7)$$

解密 $F_i = \text{Dec}(D_i, k_{si})$ ，得到相应文件 F_i 。云服务器只需返回一个聚合密钥 k_{agg} 给用户，无须为每个文件都返回一个解密密钥，用户就可以解密出所有符合搜索条件的文件。

3 安全性分析

3.1 关键字安全

定义1 如果 DBDH 假设成立，所提方案在关键字攻击下和选择明文攻击博弈中是安全的。没有多项式对手可以用挑战结构 $(\mathbf{M}^*, \rho^*)$ 来破坏所提方案。

通过一个挑战者 B 和一个攻击者 A 模拟游戏来证明安全性。

1) 初始化: 挑战者 B 随机选择 $\theta \in \{0,1\}, \theta=1$ 时, $ZZ = e(g, g)^c$; $\theta=0$ 时, $ZZ = e(g, g)^\mu$, $\mu, c \in Z_p$, 生成系统公共参数和主密钥, 将 $PP = \{ZZ, g^\beta, g^a\}$ 发送给攻击者。

2) 阶段: A 根据自己的属性集 S 适应地向 B 进行询问用户私钥, 要求 S 不满足 (M^*, ρ^*) , B 首先检查属性集 S , 对每个属性值计算 SK, 将 SK 发送给 A。

3) 挑战: A 提交两个长度相等的关键字 kw_0 和 kw_1 。B 随机选择 $c \in \{0,1\}$, 随机选择 $s \in Z_p$ 和向量 $v = (s, r_2, \dots, r_n)^T \in Z_p$ 。计算 $\lambda_i = M_i v$, 若 $\theta=1$, $C_2^* = \text{Add} \cdot e(g, g)^{cs}$; 若 $\theta=0$, $C_2^* = \text{Add} \cdot e(g, g)^{\mu s}$ 。 $C_3^* = g^s, C_{4,i}^* = g^{\lambda_i}, C_{4,i}^* = H_1(\{\pi(i): v_i\})^{\lambda_i}, \forall i \in \text{Att}$, $C_w^* = e(g^\beta, H_2(kw)^s), C_w^* = g^{\beta s}$ 。

关键字索引 $I_w^* = \{C_3^*, C_w^*, C_w^*, (C_{4,i}^*, C_{4,i}^*)_{\forall i \in \text{Att}}\}$, B 生成关键字挑战密文, 密文为 $\{I_w^*, C_2^*\}$, 将密文返回给 A。

4) 猜测。因为 A 询问到的属性集均不满足访问策略, 故 A 必须恢复关键字信息来判定是 kw_0 还是 kw_1 。A 输出其猜测 c' 。若 $c' = c$, A 输出 $c' = c$ 的概率为 $\Pr[c' = c | \theta = 1] = 1/2$ 。

因此攻击者打破 DBDH 问题的优势是可忽略的。

$$\text{Adv}_A(\varepsilon) =$$

$$\left| \frac{1}{2} \Pr[c' = c | \theta = 0] + \frac{1}{2} \Pr[c' = c | \theta = 1] - \frac{1}{2} \right| = \frac{1}{2} \left(\frac{1}{2} + \varepsilon \right) + \frac{1}{4} - \frac{1}{2} = \frac{1}{2} \varepsilon$$

3.2 陷门安全

定义 2 如果 DDH 假设成立, 所提方案在选择明文攻击博弈中是安全的。

通过一个挑战者 B 和一个攻击者 A 模拟游戏来证明安全性。

1) 初始化: 挑战者 B 随机选择 $\theta \in \{0,1\}, \theta=1$ 时, $ZZ = g^c$; $\theta=0$ 时, $ZZ = g^a$, $a, c \in Z_p$ 。生成系统公共参数和主密钥, 将 $PP = \{e(g, g)^\mu, g^\beta, ZZ\}$ 发送给攻击者。

2) 密钥询问阶段。B 首先检查属性集 S , 对每个属性值计算 SK, 将 SK 发送给 A。若 $\theta=0$, $SK = \{D_{1,x} = g^a, D_{2,x} = g^r(H_1(\{\text{Cate}_x : \text{Value}_x\}))^{-r_a}, x \in (1, 2, \dots, |S|), k = g^\mu g^{ar}, k_1 = g^{\frac{ar}{\beta}}\}$, 陷门询问阶段。A 生成搜索陷门

$$T_w =$$

$$\begin{cases} D'_{1,x} = g^{ar_a} \\ D'_{2,x} = g^{r_a} H_1(\{\text{Cate}_x : \text{Value}_x\})^{-r_a}, x \in (1, 2, \dots, |\varphi|) \\ T = g^{\frac{ar_a}{\beta}} H_2(q) \end{cases}$$

3) 挑战: A 提交两个长度相等的关键字 q_0 和 q_1 。B 随机选择 $c \in \{0,1\}$, B 计算陷门 T_w , 并返回给 A。

4) 猜测。A 输出其猜测 c' 。若 $c' = c$, A 输出 $c' = c$ 的概率为 $\Pr[c' = c | \theta = 0] = 1/2 + \varepsilon$ 。若 $c' \neq c$, A 输出 $c' = c$ 的概率为 $\Pr[c' = c | \theta = 1] = 1/2$ 。

因此, 攻击者打破 DDH 问题的优势是可忽略的。

$$\text{Adv}_A(\varepsilon) =$$

$$\left| \frac{1}{2} \Pr[c' = c | \theta = 0] + \frac{1}{2} \Pr[c' = c | \theta = 1] - \frac{1}{2} \right| = \frac{1}{2} \left(\frac{1}{2} + \varepsilon \right) + \frac{1}{4} - \frac{1}{2} = \frac{1}{2} \varepsilon$$

3.3 密文存储安全性

1) DO 对文件加密, 将密文 $CT = \{D_i, Y_{0,i}\}_{i \in [F]}$ 上传到 CSP, CSP 返回文件对应的存储地址 Add, 但 CSP 无法知晓每个加密文件对应的 I_{id} , 无法解密密文。

2) DO 对返回的存储地址加密, 并将其与对应的关键字索引和文件 I_{id} 一并上传到 CSP, 当攻击者或者云服务器由于好奇心学习符合匹配条件的密文时, 由于存储地址加密, CSP 拥有 I_{id} 也无法匹配到对应的加密文件, CSP 无法直接返回密文。因此, 用户端需要解密存储地址密文, 并通过 CSP 返回的部分解密密钥, 计算加密文件密钥, 才能获取对应的文件, 保证密文存储的安全性。

3.4 策略隐私

在本文方案中, 将 DO 的属性值进行隐藏, 只公开所对应的属性类别以及对应的矩阵行号, 以防止属性信息泄露。属性采用哈希函数加密, 攻击者无法恢复有价值的属性信息。

定理 1 如果 CDH 假设成立, 本文方案在离线字典攻击下是真正的匿名性。

分析: 攻击者已知 g^s, g^a , 但攻击者无法计算 g^{as} , 因此攻击者不能通过 $e(g^{as}, g^{ra}) = \prod_{i \in [q]} e(C_{4,i}, D'_{2,x}) e(C'_{4,i}, D'_{1,x})^{\alpha_i}$ 发动离线字典攻击。

4 性能分析和实验仿真

4.1 功能特性比较

将本文方案与文献[8,11-13]方案进行对比, 如表 2 所示。文献[11-13]都是基于关键字的可搜索加密, 但不支持任何表达式的访问策略。与文献[8]相比, 本文方案对不同文件分配不同加密密钥, 采用聚合密钥方式对密文解密。

表 2 功能对比
Table 2 Functional Comparison

方案	访问策略	可搜索加密	策略隐藏	文件解密
文献[11]	“AND” 门	√	×	对称密钥
文献[12]	访问树	√	×	对称密钥
文献[13]	“AND” 门	√	√	对称密钥
方案[8]	LSSS	√	√	对称密钥
本文方案	LSSS	√	√	聚合密钥

注: “×” 表示不具有特定功能或未使用某种技术; “√” 表示满足。

4.2 理论分析

将本文方案与现有方案在系统初始化、密钥生成、索引生成、陷门生成 4 个方面进行计算量对比, 结果如表 3 所示。 $|G|, |G_T|, |Z_P|$ 分别表示 G, G_T, Z_P 元素的长度, 分别用 $|S|, |X|, |N|$ 表示用户的属性集、数据拥有者的属性集和系统总属性集。 L 表示陷门中搜索关键字的数量, W 表示关键字密文中关键字的数量。由表 3 可知, 本文方案与文献[12]方案计算开销代价基本相同, 优于文献[11]方案和文献[8]方案; 但在系统初始化方面, 计算量略高于文献[13]方案。

表 3 计算量对比
Table 3 Comparison of the calculated quantities

方案	Setup	KeyGen	Encrypt	Trapdoor
文献[11]方案	$5 G + G_T + 5 Z_P $	$(2 S + 2) G $	$(2 X + 2W + 1) G $	$(2 S + 3 + L) G $
文献[12]方案	$3 G + G_T + 3 Z_P $	$(2 S + 3) G $	$2 G_T + (2 X + 2W + 1) G $	$(L + 2 S) G $
文献[13]方案	$4 G + 2 Z_P $	$(S + 5) G $	$ G_T + (X + W + 4) G $	$(S) G + (L + 1) G_T $
文献[8]方案	$(4 + N) G + (1 + N) Z_P $	$(2 S + 3) G $	$ G_T + (3 X + W + 3) G $	$(S + 3 + 2L) G $
本文方案	$4 G + G_T + 3 Z_P $	$(2 S + 2) G $	$ G_T + (2 X + 2W + 1) G $	$(L + 2 S) G $

4.3 实验分析

本文实验是基于 Linux 系统, 在 8 GB 的内存, 4 核处理器下运行。采用 Python 语言, 利用 pbc 等密码学工具库进行编程, 实验使用的双线性对包是基于 512 bit 椭圆曲线群, G 和 G_T 中元素的大小是 128 byte。实验对比了 3 个方案的两个主要算法 (密钥生成、关键字加密) 的计算开销, 结果如图 2 所示。此外, 固定属性个数为 20 个, 通过改变关键字个数, 对比索引生成阶段和搜索阶段的计算开销, 结果如图 3 所示。本文方案效率高于文献[13]方案和文献[8]方案。在文件加密方面, 与不采用聚合密钥方

式加解密文件相比, 本文方案效率较高, 结果如图 4 所示。

5 结束语

本文提出了云环境下基于属性策略隐藏的可搜索加密方案。本文方案采用搜索、访问控制与文件加密三者相结合的思想, 利用 LSSS 技术实现访问控制, 并将策略进行隐藏, 保护了用户和数据拥有者的隐私。云服务器将聚合密钥发送给用户, 数据拥有者无须与用户交互, 用户端减轻了密钥管理的负担, 同时, 给出了安全性证明和性能分析。实验结果表明, 本文方案有效地提高了密文检索效率。

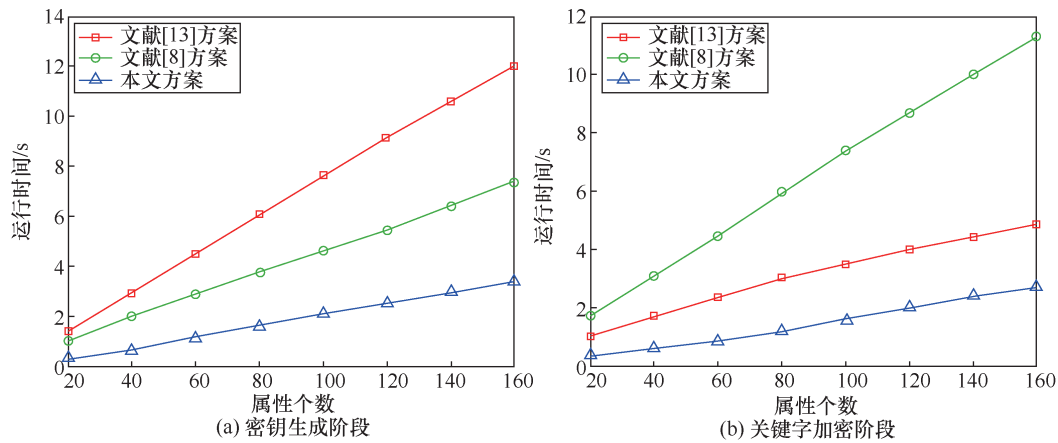


图2 密钥生成阶段和关键字加密阶段计算开销

Figure 2 The key generation phase and the keyword encryption phase calculation overhead

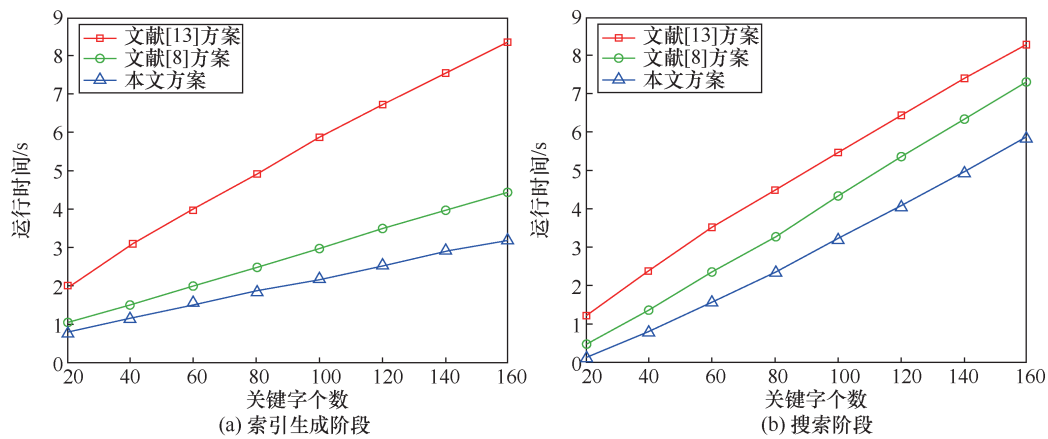


图3 索引生成阶段和搜索阶段计算开销

Figure 3 Index generation phase and search phase calculation overhead

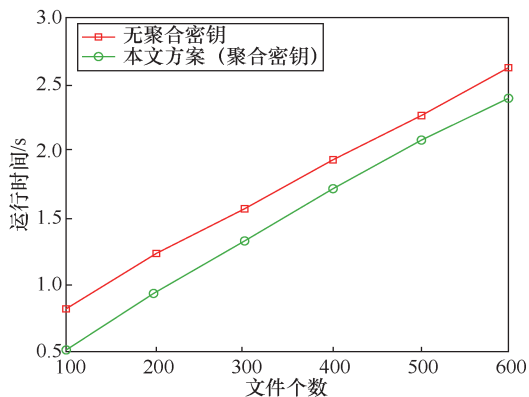


图4 无聚合密钥与本文方案(聚合密钥)计算开销

Figure 4 No aggregate key and the present scheme (aggregate key) calculation overhead

参考文献:

- [1] SONG D, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]//Proceeding 2000 IEEE Symposium on Security and Privacy. 2000: 44-55.
- [2] LIU Z Y, XU C G, YAO Z G. Forward secure searchable encryption with conjunctive-keyword supporting multi-user[C]//3rd International Conference on Security and Privacy in New Computing Environments. 2021: 423-440.
- [3] GE X R, YU J, HU C, et al. Enabling efficient verifiable fuzzy keyword search over encrypted data in cloud computing[J]. IEEE Access, 2018(6): 45725-45739.
- [4] LIN W Y, CUI H L, LI B C, et al. Privacy-preserving similarity search with efficient updates in distributed key-value stores[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(5): 1072-1084.
- [5] AHMED MA, RAMACHANDRAM S, KHAN KUR. Conjunctive keyword forward secure ranked dynamic searchable encryption over outsourced encrypted data[C]//8th International Symposium on Security in Computing and Communications (SSCC 2020). 2020: 197-212.
- [6] ZHANG Y, LI Y, WANG Y F. Efficient searchable symmetric encryption supporting dynamic multikeyword ranked search[J]. Security and Communication Networks, 2020.
- [7] 高诗尧, 陈燕俐, 许玉岚. 云环境下基于属性的多关键字可搜索加密方案[J]. 计算机科学, 2021. 9(6): 1-15.

- GAO S Y, CHEN Y L, XU Y L. Property-based multi-keyword searchable encryption schemes in a cloud environment[J]. Computer Science, 2021, 9(6): 1-15.
- [8] LIU X Y, LU T T, HE X M, et al. Verifiable attribute-based keyword search over encrypted cloud data supporting data deduplication[J]. IEEE Access, 2020(8): 52062-52074.
- [9] 张玉磊, 刘文静, 刘祥震, 等. 基于授权的多服务器可搜索密文策略属性基加密方案[J]. 电子与信息学报, 2019, 41(8): 1808-1814.
- ZHANG Y L, LIU W J, LIU X Z, et al. Authorization-based multi-server can search for a cipheric policy attribute-based encryption scheme[J]. Journal of Electronics and Informatics, 2019, 41(8): 1808-1814.
- [10] SONG X Y, ZHOU Z, DUAN W J, et al. Attribute-based searchable encryption scheme with fuzzy keywords[J]. Computer Science, 2021: 44-61.
- [11] 牛淑芬, 谢亚亚, 杨平平, 等. 区块链上基于云辅助的属性基可搜索加密方案[J]. 计算机研究与发展, 2021, 58(4): 811-821.
- NIU S F, XIE Y Y, YANG P P, et al. Cloud-based attribute bases on the blockchain are searchable for encryption schemes[J]. Computer Research and Development, 2021, 58(4): 811-821.
- [12] MIAO Y B, MA J F, LIU X M, et al. Attribute-based keyword search over hierarchical data in cloud computing[J]. IEEE Transactions on Services Computing, 2020, 13(6): 985-998.
- [13] WU A, ZHENG D, ZHANG Y, et al. Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing[J]. Sensors, 2018, 18(7).
- [14] YAN X X, NI H, LIU Y, et al. Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR[J]. Computer Science and Information Systems, 2019, 13(6): 831-847.
- [15] WATERS B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization[J]. Lecture Notes in Computer Science, 2011, (6571): 53-70.
- [16] BONEH D, FRANKLIN M. Identity-based encryption from the weil pairing[J]. Lecture Notes in Computer Science, 2001, (2139): 213-229.
- [17] ZHANG W, ZHANG Z S, XIONG H, et al. PHAS-HEKR-CP-ABE: partially policy-hidden CP-ABE with highly efficient key revocation in cloud data sharing system[J]. Journal of Ambient Intelligence and Humanized Computing, 2021(4): 1-15.
- [18] MA M M, LUO M, FAN S Q, et al. An efficient pairing-free certificateless searchable public key encryption for cloud-based IIoT[J]. Wireless Communications and Mobile Computing, 2020: 1-11.
- [19] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine grained access control of encrypted data[C]. Proceedings of the 13th ACM Conference on Computer and Communication Security, 2006: 89-98.

[作者简介]



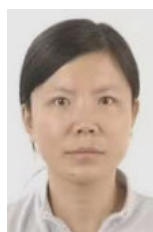
周艺华（1969—），男，山东济南人，北京工业大学副教授，主要研究方向为网络与信息安全。



扈新宇（1998—），女，河北邢台人，北京工业大学硕士生，主要研究方向为信息安全、隐私保护。



李美奇（1998—），女，山西运城人，北京工业大学硕士生，主要研究方向为信息安全、隐私保护。



杨宇光（1976—），女，北京工业大学教授、博士生导师，主要研究方向为信息安全。