

基于属性加密的块级云数据去重方案

葛文婷, 李卫海, 俞能海

(中国科学技术大学网络空间安全学院, 安徽 合肥 230026)

摘要: 针对已有的云数据去重方案主要集中在文件级去重。提出了一种基于属性加密的支持数据块级去重的方案, 对文件级和数据块级做双粒度去重, 并由属性加密实现数据共享。在混合云架构上设计算法, 私有云根据文件标签和数据块标签进行重复性检测和一致性检测, 并由块级标签建立默克尔树, 支持对用户进行所有权证明。用户上传密文, 私有云应用线性秘密共享技术, 向密文添加访问结构和辅助信息, 并为新的拥有权限的用户更新整体的密文信息。由私有云做代理重加密和代理解密, 在无法获得明文的情况下承担大部分计算, 减轻用户的计算时间开销。处理好的密文和标签存入公有云中, 由私有云进行存取。安全性分析表明, 所提方案在私有云可达到 PRV-CDA (privacy chosen-distribution attacks) 安全。分别对固定分块大小改变属性个数和固定属性个数改变分块大小两种情况进行仿真实验, 应用 4 种椭圆曲线加密测试密钥生成、加密和解密计算时间, 结果符合线性秘密共享的特性。仿真实验和开销分析表明所提方案可提升去重效率, 并降低计算时间开销。

关键词: 数据去重; 云存储; 属性加密; 所有权证明; 线性访问结构

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2023066

Block level cloud data deduplication scheme based on attribute encryption

GE Wenting, LI Weihai, YU Nenghai

School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230026, China

Abstract: Due to the existing cloud data deduplication schemes mainly focus on file-level deduplication. A scheme was proposed, based on attribute encryption, to support data block-level weight removal. Double granularity weight removal was performed for both file-level and data block-level, and data sharing was achieved through attribute encryption. The algorithm was designed on the hybrid cloud architecture. Repeatability detection and consistency detection were conducted by the private cloud based on file labels and data block labels. A Merkle tree was established based on block-level labels to support user ownership proof. When a user uploaded the cipher text, the private cloud utilized linear secret sharing technology to add access structures and auxiliary information to the cipher text. It also updated the overall cipher text information for new users with permissions. The private cloud

收稿日期: 2023-02-17; 修回日期: 2023-05-28

通信作者: 李卫海, whli@ustc.edu.cn

基金项目: 国家重点研发计划 (2018YFB0804101)

Foundation Item: The National Key R&D Program of China (2018YFB0804101)

引用格式: 葛文婷, 李卫海, 俞能海. 基于属性加密的块级云数据去重方案[J]. 网络与信息安全学报, 2023, 9(5): 106-115.

Citation Format: GE W T, LI W H, YU N H. Block level cloud data deduplication scheme based on attribute encryption[J]. Chinese Journal of Network and Information Security, 2023, 9(5): 106-115.

served as a proxy for re-encryption and proxy decryption, undertaking most of the calculation when the plaintext cannot be obtained, thereby reducing the computing overhead for users. The processed cipher text and labels were stored in the public cloud and accessed by the private cloud. Security analysis shows that the proposed scheme can achieve PRV-CDA (Privacy Choose-distribution attacks) security in the private cloud. In the simulation experiment, four types of elliptic curve encryption were used to test the calculation time for key generation, encryption, and decryption respectively, for different attribute numbers with a fixed block size, and different block sizes with a fixed attribute number. The results align with the characteristics of linear secret sharing. Simulation experiments and cost analysis demonstrate that the proposed scheme can enhance the efficiency of weight removal and save time costs.

Keywords: deduplication, cloud storage, attribute-based-encryption, proof of ownership, linear secret sharing scheme

0 引言

随着云计算技术的发展,使用云存储服务的用户日益增多,而大量重复数据的存储增加了网络传输带宽与存储的开销,降低了存储利用率。因此,大规模云存储系统往往需要去除重复的冗余信息,即去重技术^[1],以节省存储开销。在去重技术中,通用方式是为文件计算特定的标签并存储于服务器,对待上传文件进行重复性检测,进而判断用户是否需要上传文件。对于重复性文件,服务器无须再重复存储,从而达到节省带宽和提高存储利用率的目的。去重多见于文件级^[2]的去重,即只检测整个文件是否重复。几乎相近的文件显然会重复占用存储空间。文献[3]的BL-MLE (block-level message-locked encryption) 方案将文件分割成固定大小的数据块,实现数据块级的去重,并能检验数据块与文件的一致性。文献[4]在 BL-MLE 的基础上提出一种高效更新数据的方案,但未考虑数据共享。目前未见属性加密应用于块级去重,而属性加密可实现高效的数据共享和细粒度的访问控制。

用户为保护数据安全与隐私,往往会将数据加密后再上传云存储系统,这使得云存储系统在去重时必须考虑用户对数据的加密。属性加密(ABE, attribute-based-encryption)实现了一对多的访问控制以实现数据共享。现有的 ABE 方案有密钥策略的基于属性加密(KP-ABE, key-policy attribute-based encryption)^[5]和密文策略的基于属性加密(CP-ABE, ciphertext-policy attribute-based encryption)^[6]两大分支,都可用于基于属性的细粒度访问控制。文献[7]将属性加密应用到文件级的密文数据去重,上传者无须提前获知待共享数

据用户的身份即可通过属性值来达到秘密共享数据的目的。文献[8]表明,线性访问结构中的密文和加密时间随访问结构的增大呈线性增长,而在树形访问结构中呈指数型增长。线性访问结构与树形访问结构可相互转化。在数据去重中,访问结构大小与用户量相关,因此本文采用线性秘密共享方案(LSSS, linear secret sharing scheme)^[9]。

云服务器通过标签进行重复性检测,但若攻击者截获标签,则可冒充文件拥有者。因此在去重操作前,需对用户进行所有权证明(PoW, proof of ownership)^[10],在文件上传时建立默克尔树(MT, Merkle tree)存储于云服务器,仅需与客户交互少量信息即可证明用户是文件拥有者。服务器中的数据是动态变化的,文献[11]实现对动态数据更便捷的所有权证明。

由于当访问结构发生变化时,加密后的密文会发生变化,使用代理重加密^[12]可以在原密文上生成新的密文,不需要解密再加密,而引入代理加密和代理解密,在云服务器上进行大部分复杂计算,可以减小用户客户端的计算时间开销^[13]。

尽管已有支持去重的 ABE 方案,但数据块级的基于 ABE 的数据去重较为少见。本文提出了一种基于线性秘密共享的块级云数据去重方案。该方案将线性秘密共享嵌入块级去重算法,支持文件级和数据块级的双粒度去重,并实现数据块级的数据共享;支持对用户进行所有权证明,引入代理重加密实现更高效的访问结构更新,引入代理解密减轻客户端的计算负担;在进行所有权证明时,用私有云本身存储的块级标签建立默克尔树对用户发起所有权证明,可节省存储开销。经仿真测试,该方案可提升去重效率,并节省计算时间开销。

1 预备知识

本节介绍几项重要的基础知识：双线性映射是标签一致性检测的重要方式，也应用于密文设计；LSSS 访问结构的定义和性质；建立默克尔树是本文所有权证明的方式。决策 ($q-1$) 假设是安全性证明时用到的重要假设。

1.1 双线性映射

设 G 、 G_T 均是阶为素数 p 的乘法群，当映射 $e: G \times G \rightarrow G_T$ 同时满足以下 3 个性质时，称 $e: G \times G \rightarrow G_T$ 为双线性映射^[14]。

1) 双线性：对于 $\forall u \in G, v \in G, \forall a, b \in \mathbb{Z}_p$ ，满足 $e(u^a, v^b) = e(u, v)^{ab}$ 。

2) 非退化性：存在 $u \in G, v \in G$ ，满足 $e(u, v) \neq 1_{G_T}$ 。

3) 计算性： $\forall u \in G, v \in G$ ，存在有效的多项式时间算法，可计算 $e(u, v)$ 的值。

1.2 LSSS 访问结构

设 $\{P_1, P_2, \dots, P_n\}$ 是由 n 个参与者组成的实体集合，集合 $A \subseteq 2^{\{P_1, P_2, \dots, P_n\}}$ ，如果对于 $\forall B, C, B \in A, B \subseteq C$ ，有 $C \in A$ ，那么 A 就是单调的。如果集合 A 是 $\{P_1, P_2, \dots, P_n\}$ 的非空子集，即 $A \subseteq 2^{\{P_1, P_2, \dots, P_n\}} \setminus \{\emptyset\}$ ，那么 A 是一个访问结构， A 中包含的集合被称为授权集，不包含在 A 中的集合被称为非授权集。

一个定义在实体集 P 上的 LSSS^[9] 满足以下两点：① 所有实体的共享组成 Z_p 上的一个向量；② 存在一个 $l \times n$ 的共享矩阵 M 和从 $\{1, 2, \dots, l\}$ 到 P 的映射，随机选取 $v = (s, r_2, \dots, r_n)^T \in Z_p$ ， $s \in Z_p$ 是需要共享的秘密， $r_2, \dots, r_n \in Z_p$ 是随机选取的值，那么 M_v 是根据访问结构得到的关于 s 的 l 个共享组成的向量，其中 $(M_v)_i$ 属于实体 $\rho(i)$ ，记作 λ_i 。

由以上定义得到每一个线性秘密共享方案都具有线性重构的性质。假设一个对应访问结构 A 的线性秘密共享方案，对于任何授权集合 $S \in A$ ，定义 $I = \{i: \rho(i) \in S\} \subset \{1, 2, \dots, l\}$ ，如果 $\{\lambda_i\}$ 是秘密 s ，根据共享矩阵的有效分享，则存在一个常数集 $\{\omega_i \in Z_p\}_{i \in I}$ ，使得 $\sum_{i \in I} \omega_i \lambda_i = s$ ；对于任何非授权集，存在向量 $w \in Z_p$ ，使得 $w \cdot (1, 0, \dots, 0)^T = -1$ ，

$$w \cdot M_i = 0, \quad i \in I。$$

1.3 默克尔树

默克尔树^[15]是一棵存储 hash 值的树。将文件划分为多个数据块，每个叶子节点存储每个数据块的 hash 值，非叶子节点存储其子节点串联字符串再进行 hash 运算的值。默克尔树可用于数据的对比和验证，实现可证明拥有。

可证明拥有由以下步骤组成：初始化、生成挑战信息、生成响应信息和验证响应信息。具体的实现过程如下。

1) 初始化：验证者生成验证数据。

验证者计算文件 f 的默克尔树—— $MT_{H,b}(f)$ ，并将 $MT_{H,b}(f)$ 的根节点值作为验证信息。其中 H 代表计算默克尔树的节点值时用到的哈希函数， b 表示默克尔树的叶子节点的数量。

2) 生成挑战信息：当用户声称其拥有文件 f 时，验证者生成挑战信息——随机生成 u 个文件 f 的叶子节点，编号为 $l_1, l_2, \dots, l_u$ ，发送给用户。

3) 生成响应信息：用户收到验证者发送的挑战信息之后，生成响应信息。

用户计算文件 f 的默克尔树—— $MT_{H,b}(f)$ ，并将叶子节点 $l_1, l_2, \dots, l_u$ 的值以及其旁路径上的节点值，一起发送给验证者。

4) 验证响应信息：验证者收到用户生成的响应信息后，利用子节点 $l_1, l_2, \dots, l_u$ 的值以及其旁路径上的节点值，计算默克尔树的根节点值。若这样计算得到的根节点值等于服务器自身保存的根节点值，则验证者认为用户确实拥有文件 f ，即用户通过了可证明拥有。

1.4 决策 ($q-1$) 假设

对于任何基于概率的多项式时间的算法，给定

$$\begin{aligned} \vec{y} &= g, g^s \\ g^{a^i}, g^{b_j}, g^{sb_j}, g^{a^i b_j}, g^{a^i / b_j^2} & \quad \forall (i, j) \in [q, q] \\ g^{a^i / b_j} & \quad \forall (i, j) \in [2q, q] \quad i \neq q+1 \\ g^{a^i b_j / b_j^2} & \quad \forall (i, j, j') \in [2q, q, q] \quad j \neq j' \\ g^{sa^i b_j / b_j}, g^{sa^i b_j / b_j^2} & \quad \forall (i, j, j') \in [q, q, q] \quad j \neq j' \end{aligned}$$

其中， $a, \mu, b_1, \dots, b_q$ 均匀且独立地从 Z_p 中选取，且有 $g \in G, Z \in G_T$ ，那么区分 $(\vec{y}, e(g, g)^{a^{q+1}\mu})$ 与 $(\vec{y}, Z)$ 是困难的^[16]。

2 属性加密块级去重方案

本节提出一种基于线性秘密共享的数据块级去重方案。

2.1 系统模型

基于 LSSS 的块级去重系统模型如图 1 所示，主要由以下三方构成。

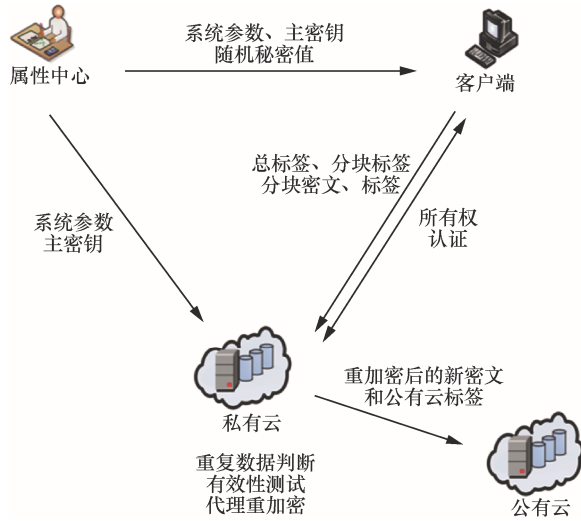


图 1 基于 LSSS 的块级去重系统模型
Figure 1 Block-level deduplication system model based on LSSS

1) 属性中心 (AA, attribute authority)

第三方可信机构，为系统生成系统参数、主密钥和公钥。为拥有访问权限的用户生成属性私钥，用户可与云服务器计算得到其有访问权限的数据的明文。

2) 客户端：包括第一位数据上传者和其他用户。

客户端借助云服务器实现数据备份和存储，并将复杂的计算外包给云服务器。属性中心为用户分配与其访问权限相关的属性，使拥有权限的用户可通过分配到的属性访问其有权访问的数据。

3) 云存储服务器：包含公有云和私有云。私有云验证用户对文件是否拥有所有权，负责解密和重加密计算，拒绝客户重复上传服务器中已存储的数据，并将密文等数据上传至公有云。公有云负责存储大量密文数据。在同一服务器范围内，数据具有唯一性。

在用户上传新文件时，先在用户端通过比较文件级标签，检验文件是否重复。若文件重复，

则在通过所有权证明后进行文件级的去重，该步骤直接在客户端进行，无须上传服务器；若文件不重复，则将文件划分成固定大小的数据块，对分块后的数据进行块级去重。此时，属性中心生成属性集、主密钥、公共参数和私钥发送给用户，用户加密不重复的数据块并上传到私有云，私有云对该加密数据进行有效性测试和一致性测试，并将通过测试的块密文存储至公有云。对于公有云上的数据，另外建立一个标签表来查找数据。对于重复的数据块，用户无须再次上传，系统通过属性加密分配给该用户权限，属性中心生成代理密钥发送给私有云，使私有云代理重加密解密出一个中间值，再由私有云发送给用户，由该拥有权限的用户解密出明文。

2.2 方案详细设计

基于线性秘密共享的块级云数据密文去重方案由以下多项式时间算法组成。

2.2.1 初始化阶段

系统建立 $(I^1) \rightarrow (PP, MK)$

该算法由属性中心运行，生成系统公共参数和主密钥，为系统初始化设置。

属性中心设定安全参数 λ ， G ， G_T 均是阶为素数 p 的群，生成元为 g ，双线性映射 $e: G \times G \rightarrow G_T$ 。选择抗碰撞哈希函数 $H_0: G_T \rightarrow Z_p$ ， $H_1: M \rightarrow Z_p$ ， $H_2: Z_p \rightarrow G$ ， $H_3: G_T \rightarrow K$ ，选择随机元素 $u, h, w, v \in G$ 和 $\alpha \in Z_p$ ，从 G_1 中随机选取 k 个元素 $u_1, u_2, \dots, u_k$ ，系统公共参数为 $PP = (H_0, H_1, H_2, g, u, u_1, \dots, u_k, h, w, v, e(g, g)^\alpha)$ ，主密钥设置为 $MK = \alpha$ 。

2.2.2 文件上传阶段

(1) 加密 $(PP, m[i], (M, \rho)) \rightarrow (T, ct[i])$

该算法由客户端运行，先生成文件级标签、数据块标签，发送给私有云，由私有云检索标签列表并进行一致性检验，判断该用户是否为第一位数据上传者。若该用户是数据上传者，则继续计算数据块密文发送给私有云。

上传者输入公共参数 PP ，LSSS 访问结构 (M, ρ) ，消息 $M = m[1] \| m[2] \| \dots \| m[n]$ ， M 是一个 $l \times n$ 阶矩阵， $\rho(i)$ 表示矩阵第 i 行对应的属性，即 $M \in Z_p^{l \times n}$ ， $\rho: [l] \rightarrow Z_p$ 。随机选取秘密共享值

μ , 随机选取 $y_2, y_3, \dots, y_n \in Z_p$, 作为向量 $\mathbf{y} = (\mu, y_2, y_3, \dots, y_n)^T \in Z_p^n$, 得到共享向量 $\mathbf{v} = (v_1, v_2, \dots, v_l)^T = \mathbf{M} \cdot \mathbf{y}$ 。随机选取 $\beta \in G_T, z_1, z_2, \dots, z_l \in Z_p$ 。计算标签值 $k_{\text{mas}} = H_1(M), k_i = H_2(m[i])$ 。生成文件标签 $T_0 = g^{k_{\text{mas}} \mu}$, 将数据块密文划分为 q 份 $\{c[i][j]\}_{1 \leq j \leq q}$, 辅助信息 $\text{aux}_i = e(k_i, T_0)$, 生成数据块标签 $T_i = (k_i \sum_{j=1}^q u_j^{c[i][j]})^{k_{\text{mas}}}$ 。标签 $L = g^{k_i} h^{H_0(\beta)}$ 。

$c[i] = \text{Enc}(H_3(\beta), m[i]), B = g^\mu, E = \beta \cdot e(g, g)^{\alpha \mu}, C_i = w^{v_i} v^{z_i}, D_i = g^{z_i}, E_i = (u^{\rho(i)} h)^{-z_i}, F_i = g^{v_i}$, Enc 和 Dec 是一个安全对称加密方案下的加密和解密算法。最终得到密文为

$$ct[i] = ((\mathbf{M}, \rho), c[i], B, E, \{C_i, D_i, E_i, F_i\}_{i \in [1, l]}) \quad (1)$$

(2) 数据块密文相等性检验

该算法由私有云运行。若私有云发现相同标签, 则用两数据块对应的文件级标签再次验证两数据块密文是否相等。若相等, 则可拒绝上传重复数据。

给定分别属于文件标签为 T_0 和 T'_0 的数据块密文标签 T_i 和 T'_i , 若 $e(T_i, T'_0) = e(T'_i, T_0)$ 成立, 可说明两个数据块密文是相等的。

(3) 数据块标签与密文的一致性检验

该算法由私有云运行, 验证数据块标签与密文是否一致。

由双线性映射的性质, 对于单个密文块 $c[i]$, 计算并验证

$$e(T_i, g^\mu) = \text{aux}_i \cdot e(\prod_{j=1}^q u_j^{c[i][j]}, T_0) \quad (2)$$

若成立, 则可通过一致性验证。

(4) 所有权证明

私有云进行标签比较后, 若判定文件 f 重复, 则对该用户发起所有权证明。若该用户通过了所有权证明, 则判定其为拥有访问权限的客户, 需更新属性列表。

私有云中存储着 $\text{MT}_{H_2, b}(f)$, 即私有云本身拥有全部 $k_i = H_2(m[i])$ 的值并由此生成默克尔树。随机生成 u 个叶子节点 $l_1, l_2, \dots, l_u$, 发送给数据上传者, 数据上传者计算 $\text{MT}_{H_2, b}(f)$, 并将叶子节点 $l_1, l_2, \dots, l_u$ 的值以及其旁路径上的节点值, 一起发

送给私有云, 由私有云计算判断该数据上传者所有权。

(5) 代理密钥 $(\text{PP}, \text{MK}, x_{\text{PC}}, x_U, A) \rightarrow P_x K_U$

当数据不重复上传至服务器时, 属性中心生成代理密钥, 并将代理密钥发送给私有云。

输入公共参数 PP, 主密钥 MK, 私有云选取 $x_{\text{PC}} \in Z_p$, 用户选取 $x_U \in Z_p$ 和属性集 $A = \{A_1, A_2, \dots, A_l\}$ 。算法随机选择 $r, r', r_1, \dots, r_l \in Z_p, \forall A_i \in A$, 计算代理密钥。

$$P_x K_U = (K_0 = g^{x_{\text{PC}} + \alpha x_U} w^{r'}, K_1 = g^r, K_2 = g^{r'}, \forall A_i \in A: \{K_{i,1} = g^{r_i}, K_{i,2} = (u^{A_i} h)^{r_i} v^{-r'}\}) \quad (3)$$

(6) 重加密 $(\text{PP}, ct[i], (\mathbf{M}', \rho')) \rightarrow (ct'[i])$

对于新加入的数据拥有者, 更新属性列表, 则密文需要改变。对私有云上的数据进行代理重加密, 使数据不需要解密再加密, 实现更为方便的数据共享。

私有云输入公共参数 PP, LSSS 结构 $(\mathbf{M}', \rho')$, 密文 $ct[i]$, $\mathbf{M}'$ 是一个 $l_1 \times n_1$ 矩阵。对于 $\mathbf{M}'$ 的每行 $\mathbf{M}'_i = (m'_{i,1}, \dots, m'_{i,n_1}), i \in [1, l_1]$, 随机选取 $\mathbf{y}' = (\mu', y'_2, y'_3, \dots, y'_n)^T \in Z_p^{n_1}, z'_i \in Z_p, \mu' = \mu + \bar{\mu}$, 对于 $i \in [1, l_1]$, $B' = g^{\mu'}, E' = \beta \cdot e(g, g)^{\alpha \mu'}, C'_i = w^{m'_{i,1} \cdot y'_2} v^{z'_i}, D'_i = g^{z'_i}, E'_i = (u^{\rho'(i)} h)^{-z'_i}, F'_i = g^{m'_{i,1} \cdot y'_n}$, 输出新密文为

$$ct'[i] = ((\mathbf{M}', \rho'), c[i], B', E', \{C'_i, D'_i, E'_i, F'_i\}) \quad (4)$$

标签 $L' = L$ 。私有云将重加密后的密文 $ct'[i]$ 和标签 L' 发送至公有云。公有云删除原来的密文 $ct[i]$, 存储新密文 $ct'[i]$ 。

2.2.3 解密阶段

块级去重解密阶段如图 2 所示, 拥有访问权限的用户向 AA 提出下载请求, AA 计算代理密钥并发送给私有云, 由私有云解密出中间值再由用户解密出明文。

(1) 代理解密 $(x_{\text{PC}}, P_x K_U, ct'[i]) \rightarrow (\text{ME})$

对于有数据访问权限的用户, 从服务器上下载数据时, 需要由私有云进行代理解密。

私有云输入私钥 x_{PC} 与属性集 A 相关的代理密钥 $P_x K_U = (K_0, K_1, K_2, \forall A_i \in A: \{K_{i,1}, K_{i,2}\})$, 元组 $(B', E', C'_i, D'_i, E'_i, F'_i)$ 。计算属性集 A 中共享矩阵

M' 的行集合, $I = \{i: \rho'(i) \in A\}$ 。如果集合 A 是被策略授权的集合, 则存在使 $\sum_{i \in I} w_i M'_i = (1, 0, \dots, 0)$ 成立的常数集合 $\{w_i \in Z_p\}_{i \in I}$, 其中, M'_i 是矩阵 M' 的第 i 行。私有云执行代理解密如下。

$$\frac{e(B', K_0)}{\prod_{i \in I} (e(C'_i, K_2) e(E'_i, K_{i,1}) e(D'_i, K_{i,2}) e(F'_i, K_1)^{x_{PC}})^{w_i}} = \frac{e(g, g)^{\alpha u' x_U} e(g, g)^{r \mu' x_{PC}} e(g, w)^{\mu' r'}}{\prod_{i \in I} e(g, g)^{\gamma_j' w_i x_{PC}} e(g, w)^{r' y_i' w_i}} = e(g, g)^{\alpha u' x_U} = \text{ME} \quad (5)$$

私有云解出一个中间值 ME, 没有用户的私钥不能继续解密, 保护了数据的安全性。

(2) 用户解密 ($\text{SK}_U, E', \text{ME}, ct'$) $\rightarrow (m[i])$

用户最终的解密部分。用户输入私钥 $\text{SK}_U = x_U$, 密文 E' , 中间值 ME, 解密如下。

$$\frac{E'}{(e(g, g)^{\alpha u' x_U})^{x_U^{-1}}} = \frac{\beta \cdot e(g, g)^{\alpha u'}}{e(g, g)^{\alpha u'}} = \beta \quad (6)$$

计算 $m[i] = \text{Dec}(H_3(\beta), ct')$, 即可解密出明文。

如图 2 所示, 解密阶段私有云代理解密, 承担了大部分计算量, 且仅有用户可得到明文。

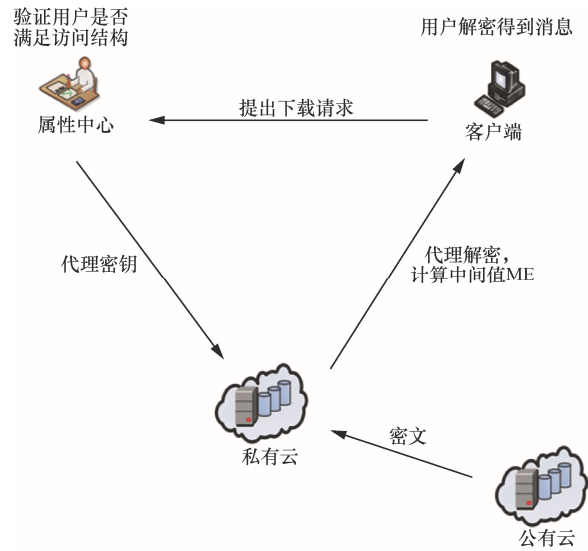


图2 块级去重解密阶段
Figure 2 Block-level deduplication decryption phase

3 安全性分析

定理 1 假设群 G 满足 $(q-1)$ 假设, SE 是一个

安全对称加密方案, L 是一个安全忠诚方案, 那么本文方案在考虑公有云的视角下是选择性不可区分的。

证明 由 LSSS 进行属性加密的安全性证明文献中, Rouselakis-Waters 方案^[17]在 $(q-1)$ 假设下是选择性不可区分的。本文方案的证明与其相似, 仅需要考虑与该方案不同的部分。注意, 本文方案的假设保证了添加的部分 $c[i] = \text{Enc}(H_3(\beta), m), L = g^{H_1(M)} h^{H_0(\beta)}$ 不会揭露任何信息, 其他同 Rouselakis-Waters 方案的证明, 可以得知该方案在公有云的视角下是选择性不可区分的。

定理 2 假设群 G_1, G_2 满足决策 $(q-1)$ 假设和决策 BDH (bilinear Diffie-Hellman) 假设, SE 是一个安全对称加密方案, 那么该方案是 PRV-CDA 安全的。

证明

PRV-CDA 安全性由加密安全性和重加密安全性构成。加密安全性同定理 1, 只需考虑添加的部分不会暴露原来关于 M 的信息, 其他的证明可以参照 Rouselakis-Waters 方案的证明。

在考虑到重加密安全性时, 需要回顾本文重加密方案的过程。假设有某个攻击者算法破坏了 PRV-CDA 安全性, 本文将构造出一个挑战者算法解决 BDH 问题, 以此来导出矛盾。

最初, 挑战者调用组生成算法, 输入安全参数, 选择一个随机的组元素 $g \xleftarrow{\$} G$ 和 $q+2$ 个随机指数 $a, s, b_1, b_2, \dots, b_q \xleftarrow{\$} Z_p$, 然后向攻击者发送 (p, G, G_T, e) 和以下所有元素。

$$\begin{aligned} &g, g^s \\ &g^{a^i}, g^{b_j}, g^{sb_j}, g^{a^i b_j}, g^{a^i / b_j^2} \quad \forall (i, j) \in [q, q] \\ &g^{a^i / b_j} \quad \forall (i, j) \in [2q, q] \quad i \neq q+1 \\ &g^{a^i b_j / b_j^2} \quad \forall (i, j, j') \in [2q, q, q] \quad j \neq j' \\ &g^{sa^i b_j / b_j}, g^{sa^i b_j / b_j^2} \quad \forall (i, j, j') \in [q, q, q] \quad j \neq j' \end{aligned}$$

在挑战阶段, 挑战者随机选取 $b \xleftarrow{\$} \{0, 1\}$, 如果 $b=0$, 挑战者发送给攻击者 $e(g, g)^{sa^{q+1}}$, 否则发送一个随机项 $R \xleftarrow{\$} G_T$ 。最终攻击者输出一个猜测 $b' \in \{0, 1\}$ 。

考虑到一个挑战者算法需要满足：在给出 (g, g^a, g^b, g^c, Z) 时，当 $Z = e(g, g)^{abc}$ 时，输出 1，当 Z 在 G_T 上均匀分布时，输出 0。挑战者在 Z_p 中随机选择一个 x ，在 G 中随机选择 $\tilde{u}, \tilde{h}, \tilde{v}$ 并且计算 $w = g^x$ 。此时的公共参数是 $(H_0, H_1, H_2, g, u, h, w, v, e(g^a, g^b))$ 。其中 H_0, H_1, H_2 为上面所定义的形式。这意味着，其中的密钥 $\alpha = ab$ 对挑战者来说是未知的。挑战者将公共参数发送给攻击者，其中具体参数的计算公式如下。

$$g = g \quad (7)$$

$$w = g^a \quad (8)$$

$$v = g^{\tilde{v}} \cdot \prod_{(j,k) \in [\ell, n]} (g^{a^k/b_j})^{M_{j,k}^*} \quad (9)$$

$$u = g^{\tilde{u}} \cdot \prod_{(j,k) \in [\ell, n]} (g^{a^k/b_j^2})^{M_{j,k}^*} \quad (10)$$

$$h = g^{\tilde{h}} \cdot \prod_{(j,k) \in [\ell, n]} (g^{a^k/b_j^2})^{-\rho^*(j)M_{j,k}^*} \quad (11)$$

$$e(g, g)^\alpha = e(g^a, g^{a^q}) \cdot e(g, g)^{\tilde{\alpha}} \quad (12)$$

当攻击者输出一个结构 (M^*, ρ^*) 时，挑战者首先从 M_0^* 和 M_1^* 中选择一个 $M_b^*, b \in \{0, 1\}$ ；之后从 Z_p 中随机选取 $\tilde{c}, y_2, \dots, y_n$ 并且让 $v = (c, y_2, \dots, y_n), \tilde{v} = (\tilde{c}, y_2, \dots, y_n)$ ；最后，挑战者随机选取 $\beta \in G_T, z_1, \dots, z_l \in Z_p$ 。

注意 $Z = e(g, g)^{abc} = e(g^a, g^b)^c$ ，这意味着其输出的分布和在攻击者视角下是一样的。

挑战 攻击者会输出一对等长的信息 (m_0, m_1) 。相当于随机选取 $b \leftarrow \{0, 1\}$ ，然后计算 $C = m_b \cdot T \cdot e(g, g^s)^{\tilde{\alpha}}, C_0 = g^s$ ，其中 T 为挑战项， g^s 是该假设对应的项。

设置 $\tilde{y} = (s, sa + \tilde{y}_2, sa^2 + \tilde{y}_3, \dots, sa^{n-1} + \tilde{y}_n)^\perp$ ， $\tilde{y}_2, \tilde{y}_3, \dots, \tilde{y}_n \leftarrow \mathbb{Z}_p$ 。由于秘密 s 对 $\mathcal{A}$ 是隐藏的，并且 $\tilde{y}_i$ 是随机选取的，理论上秘密 s 和向量 $\mathbf{y}$ 是合理的。由于 $\lambda = M^* \mathbf{y}$ ，可以得到

$$\lambda_\tau = \sum_{i \in [n]} M_{\tau,i}^* sa^{i-1} + \sum_{i=2}^n M_{\tau,i}^* \tilde{y}_i = \sum_{i \in [n]} M_{\tau,i}^* sa^{i-1} + \tilde{\lambda}_\tau \quad (13)$$

对每个 $\tau \in [\ell]$ ，有 $\tau \in [\ell], \tilde{\lambda}_\tau = \sum_{i=2}^n M_{\tau,i}^* \tilde{y}_i$ 。对每个 $\mathcal{B}$ ， $t_\tau = -sb_\tau$ 。 b_i 是理论上对攻击者隐藏的

信息。由以上的信息，挑战者可以计算得到密文信息，并发送给攻击者。

如果攻击者输出猜测 $b' = b$ ，挑战者会输出 1，而这意味着 $Z = e(g, g)^{abc}$ 。当 $Z = e(g, g)^{abc}$ 是由假设中的安全 SE 方案给出时，攻击者的视角和上述模拟中是一样的。而当 Z 是均匀分布时， b 的信息对攻击者来说，是完全未知的。在这个意义下，如果攻击者破坏了 PRV-CDA 安全性，它要么解决了 BDH 问题，要么破坏了 SE 方案的安全性或者破坏了安全忠诚方案的安全性，而这都是不可能的。

证毕。

4 仿真实验与性能评估

本节给出对本文方案的运行性能评估，选取 SS512、MNT159、MNT201、MNT224 这 4 种椭圆曲线，对本文提出的基于属性加密的块级去重方案，分别测试固定分块的算法运行时间，以及各方案对于不同大小的分块的运行时间。

4.1 实验环境

1) 硬件环境

CPU: AMD Ryzen 5 4600H, Radeon Graphics

3.00 GHz

内存: 16 GB RAM SAMSUNG

硬盘: 512 GB SSD SAMSUNG2

2) 开发环境

开发操作系统: Ubuntu 16.04 (64 bit)

开发语言及工具: Python 3.5、GMP-6.2.0 库、PBC-0.5.14 库、Charm-0.50 库、Pycharm、VMware。

4.2 仿真结果

4.2.1 固定分块大小对不同属性个数的计算时间

选取 SS512、MNT159、MNT201、MNT224 这 4 种椭圆曲线对固定分块大小为 1 kB 的基于属性加密的块级去重方案测试运行时间，如图 3 所示。这 4 种曲线分别对应 112 bit、80 bit、80 bit 和 100 bit 的安全性。

实验结果显示，当属性个数增加时，密钥生成、加密、重加密和解密的计算时间线性增长。这与 LSSS 的特性相符合。在加解密和重加密的计算中，SS512 拥有最小的计算时间开销，而 MNT224 的计算时间开销最大。在密钥生成算法中，SS512 的计算时间开销最大，但对于 100 个属

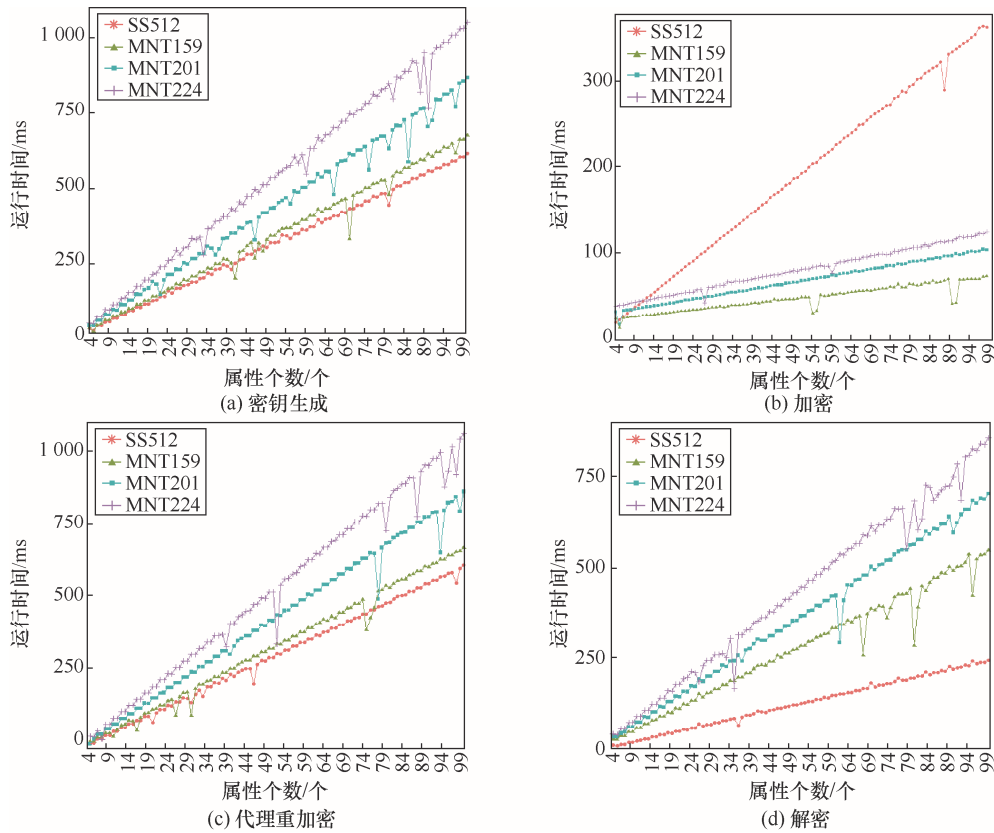


图3 固定分块大小对不同属性个数的运行时间
Figure 3 Fixed block size running time for different number of attributes

性个数也仅占用 360 ms, 相比加密解密和重加密有较大的时间消耗, SS512 依然是最优的方法。由实验结果可知, 对于固定 1 kB 的分块, 在属性个数为 100 的情况下, 对于最耗时的加密和重加密过程, 可将计算时间控制在 1.05 s 内, 对比文献[7]的文件级去重的实验结果, 块级的去重方案可节省对于每个分块的计算时间开销。

4.2.2 相同属性个数对不同分块大小的运行时间

对于固定的 40 个属性, 选取 SS512、MNT159、MNT201 和 MNT224 这 4 种椭圆曲线对分块大小为 1 MB 到 30 MB 的数据分块, 运行本文算法并测试运行时间, 实验结果如图 4 所示。

当属性个数固定、分块大小增加时, 密钥生成算法不受分块大小的影响, 加密和解密算法的运行时间随分块大小的增加呈线性增长, 这同样与 LSSS 的特性相符合。在密钥生成算法中, SS512 的运行时间最大, 对于固定的 40 个属性, 运行时间为 148 ms, 是可以接受的。在加密和解密算法中, SS512 的运行时间最小,

MNT224 的运行时间最大。

4 种椭圆曲线的运行时间随着分块大小的增加, 增长速率几乎是一致的。对于 40 个属性, 在加密和解密中, MNT224 的运行时间比 SS512 的运行时间约长 200 ms。因此 SS512 是可以选取的最优方案。

值得注意的是, 以上实验曲线中都存在违反整体趋势的“锯齿”。在密钥生成以及加解密时, 出于安全性考虑都随机选取了不同的随机数, 尤其选择属性参数时引入了随机量, 因此总体线性但有随机误差, 产生“锯齿”。若实验增大属性间隔可以减轻此类“锯齿”的情况。

4.3 计算时间开销分析

对本文方案的各项算法分析计算时间开销, 用 l 表示访问结构的属性数量, 用 k 表示与私钥关联的属性集的大小, 用 y 表示私有云现存的标签数量。用 Exp 表示一个指数运算, 用 Pair 表示一个双线性配对运算。由前文已知, 在计算标签和进行一致性验证时, 将数据块文件

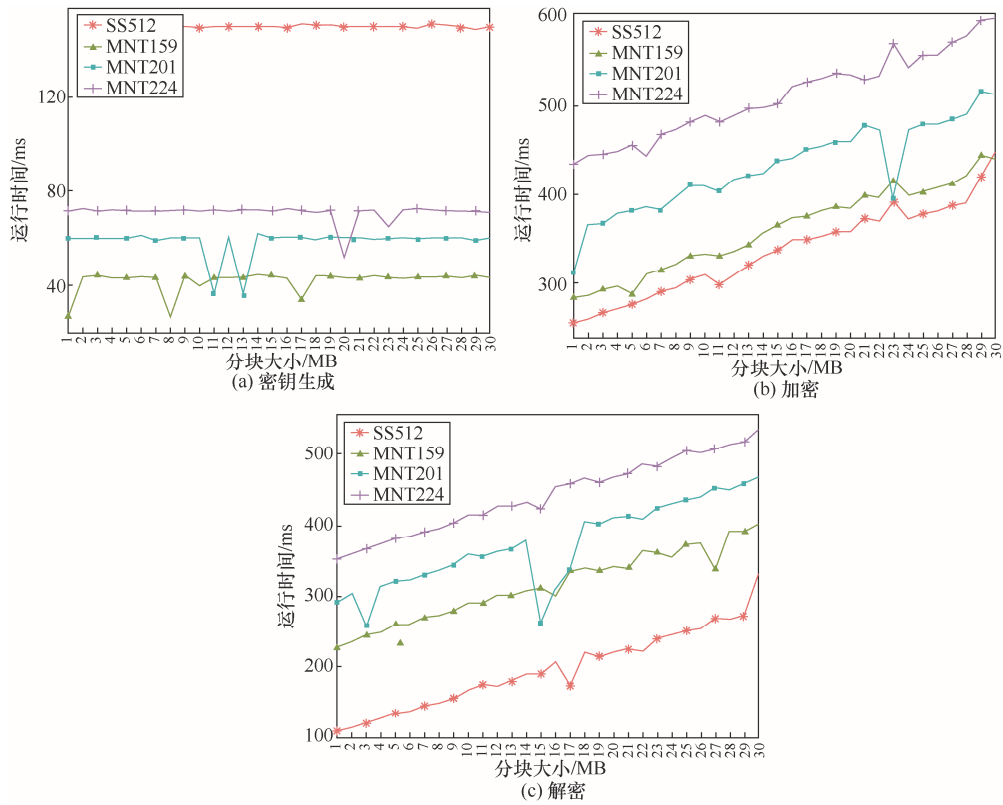


图 4 相同属性个数对不同分块大小的运行时间
Figure 4 Fixed number of attributes running time for different block sizes

划分 q 块。本文方案的各多项式算法的计算时间开销如表 1 所示。

表 1 本文方案的各多项式算法的计算时间开销
Table 1 The calculation time costs of each algorithm in proposed scheme

类型	公有云 标签	私有云 标签	加密	重加密	一致性 检验	相等性 检验	解密
Exp	2	$q+3$	$6l+1$	$7l+2$	$q+1$	0	$\leq k+2$
Pair	0	0	0	0	3	$2y$	$\leq 5k+1$

指数运算主要存在于生成标签和加解密运算，双线性配对主要存在于一致性检验、相等性检验和解密。

相对文件级去重，块级去重不可避免地增加了对数据块分块时额外产生的指数运算，以及在加解密中添加元组时产生的开销。文献[7]实现了基于属性的文件级云数据去重，各方案的计算时间开销对比如表 2 所示。

对比可知，本文方案的块级去重的额外计算时间开销均匀分配给数据上传者、私有云和需要共享数据的用户。本文方案设计的数据块标签计

算方式增加了 q 次指数运算，同时影响到一致性检验，本文方案增加了 3 次双线性配对运算。由于在密文中增加新的元组，加解密的开销分别增加了 l 次指数运算和 $2k$ 次双线性配对运算。由理论分析和前文的仿真实验可知，这样的开销增加是可以接受的。

表 2 各方案的计算时间开销对比
Table 2 The calculation time costs of each scheme

文献	数据上传者	私有云	用户
文献[7]	$(5l+6)\text{Exp}$	$(6l+7)\text{Exp}$	$\leq (k+2)\text{Exp}$
		$2y\text{Pair}$	$\leq (3k+1)\text{Pair}$
本文方案	$(5l+6)\text{Exp}$	$(7l+q+3)\text{Exp}$	$\leq (k+2)\text{Exp}$
		$(2y+3)\text{Pair}$	$\leq (5k+1)\text{Pair}$

5 结束语

随着互联网时代的快速发展，云计算凭借其高效和经济的特性正在快速增长。云环境下的存储效率和计算时间开销以及安全性，需要兼顾和平衡。

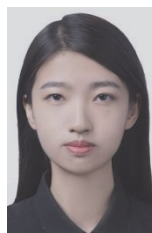
云环境下基于属性的块级云数据去重方案，是在文件级的基于属性的去重基础上，引入了块

级去重, 实现更细粒度的去重, 提高去重效率。而在属性加密基础上的块级去重, 提高了数据共享的能力。本文设计了一种用双线性配对验证完成块级去重情况下的标签与密文一致性检验的方案, 该方案由多个多项式时间算法组成。块标签可用于建立默克尔树, 对用户进行所有权证明。针对块级云数据去重的整个流程中的密钥生成、加密、代理重加密、解密算法, 在选取不同椭圆曲线的情况下, 分别对固定分块大小的不同数量的属性集, 以及固定属性集数量文件的不同分块大小, 测试了计算时间开销。实验结果验证了在LSSS属性加密下, 加解密的计算时间开销随属性个数或分块大小的增加而线性增长, 而生成密钥的计算时间开销只与属性个数有关, 随属性个数的增加线性增长。另外, 本文还对该方案进行了安全性分析, 该方案是PRV-CDA安全的。下一步工作将重点考虑允许用户自定义密钥情况下的去重, 以及海量用户块级去重后的数据所属权管理等。

参考文献:

- [1] BELLARE M, KEELVEEDH S. Interactive message-locked encryption and secure deduplication[C]//Public-Key cryptography - PKC 2015: 18th ICAR International Conference on Practice and Theory in Public Key Cryptography. 2015: 516-538.
- [2] AHAMED N, DURAIPANDIAN N. Secured data storage using deduplication in cloud computing based on elliptic curve cryptography[J]. Tech Science Press, 2022, 41: 83-94.
- [3] CHEN R, MU Y, YANG G, et al. BL-MLE: block-level message-locked encryption for secure large file deduplication[J]. IEEE Transactions on Information Forensics & Security, 2015, 10(12): 2643-2652.
- [4] ZHAO Y, CHOW S. Updatable block-level message-locked encryption[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(4): 1620-1631.
- [5] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proceedings of the 13th ACM Conference on Computer and Communications Security. 2006: 89-98.
- [6] EMURA K, MIYAJI A, OMOTE K, et al. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length[J]. International Journal of Applied Cryptography, 2009, 2(1): 46-59.
- [7] CUI H, DENG R, LI Y, et al. Attribute-based storage supporting secure deduplication of encrypted data in cloud[J]. IEEE Transactions on Big Data, 2019, 5(3): 330-342.
- [8] GOYAL V, JAIN A, PANDEY O, et al. Bounded ciphertext policy attribute based encryption[C]//35th International Colloquium on Automata, Languages and Programming (ICALP 2008). 2008: 579-591.
- [9] WATERS B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization[C]//Public Key Cryptography-PKC 2011. 2011: 53-70.
- [10] GONZALEZ-MANZANO L, ORFILA A. An efficient confidentiality-preserving proof of ownership for deduplication[J]. Journal of Network & Computer Applications, 2015, 50(4): 49-59.
- [11] HUANG K, ZHANG X S, MU Y, et al. Bidirectional and malleable proof-of-ownership for large file in cloud storage[J]. IEEE Transactions on Cloud Computing, 2021, (99): 1.
- [12] CHEN Y, HU Y, ZHU M, et al. Attribute-based keyword search with proxy re-encryption in the cloud[J]. IEICE Transactions on Communications, 2018, (8): 1798-1808.
- [13] 刘尚, 郭银章. 云计算多授权中心 CP-ABE 代理重加密方案[J]. 网络与信息安全学报, 2022, 8(3): 176-188.
- LIU S, GUO Y Z. Multi-authority based CP-ABE proxy re-encryption scheme for cloud computing[J]. Chinese Journal of Network and Information Security, 2022, 8(3): 176-188.
- [14] BONEH D, FRANKLIN M. Identity-based encryption from the Weil pairing[J]. SIAM Journal on Computing, 2003, 32(3): 586-615.
- [15] SZYDLO M. LECTURE NOTES IN COMPUTER SCIENCE 3027. Merkle Tree Traversal in Log Space and Time[C]//2004 International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2004). 2004: 541-554.
- [16] ROUSELAKIS Y, WATERS B. New constructions and proof methods for large universe attribute-based encryption[C]//ACM SigSAC Conference on Computer & Communications Security. 2013: 463-474.
- [17] ROUSELAKIS Y, WATERS B. New constructions and proof methods for large universe attribute-based encryption[C]//SigSAC Conference on Computer & Communications Security. 2013.

[作者简介]



葛文婷(1998-), 女, 江苏泰州人, 中国科学技术大学硕士生, 主要研究方向为数据安全。



李卫海(1975-), 男, 辽宁大连人, 中国科学技术大学副教授, 主要研究方向为多媒体内容安全、数据安全。



俞能海(1964-), 男, 安徽无为, 中国科学技术大学教授、博士生导师, 主要研究方向为视频处理与多媒体通信、信息检索、媒体内容安全、数据安全。