基于属性加密的高效密文去重和审计方案

马 华[①] 党乾龙*[①] 王剑锋[②] 刘振华[①]

[①](西安电子科技大学数学与统计学院 西安 710071)

[②](西安电子科技大学网络与信息安全学院 西安 710071)

摘 要：针对当前支持去重的属性加密方案既不支持云存储数据审计，又不支持过期用户撤销，且去重搜索和用户解密效率较低的问题，该文提出一种支持高效去重和审计的属性加密方案。该方案引入了第3方审计者对云存储数据的完整性进行检验，利用代理辅助用户撤销机制对过期用户进行撤销，又提出高效去重搜索树技术来提高去重搜索效率，并通过代理解密机制辅助用户解密。安全性分析表明该方案通过采用混合云架构，在公有云达到IND-CPA安全性，在私有云达到PRV-CDA安全性。性能分析表明该方案的去重搜索效率更高，用户的解密计算量较小。

关键词：属性加密；数据去重；审计；用户撤销；代理解密

中图分类号：TP309 **文献标识码**：A **文章编号**：1009-5896(2019)02-0355-07

DOI: 10.11999/JEIT170935

Efficient Ciphertext Deduplication and Auditing Scheme with Attribute-based Encryption

MA Hua[①] DANG Qianlong[①] WANG Jianfeng[②] LIU Zhenhua[①]

[①](*School of Mathematics and Statistics*, *Xidian University*, *Xi'an* 710071, *China*)

[②](*School of Network and Information Security*, *Xidian University*, *Xi'an* 710071, *China*)

Abstract: Existing attribute-based deduplication schemes can support neither auditing of cloud storage data nor revocation of expired users. On the other hand, they are less efficient for deduplication search and users decryption. In order to solve these problems, this paper proposes an efficient deduplication and auditing Attribute-Based Encryption (ABE) scheme. A third-party auditor is introduced to verify the integrity of cloud storage data. Through an agent auxiliary user revocation mechanism, the proposed scheme supports the revocation of expired users. Effective deduplication search tree is put forward to improve the search efficiency, and the proxy decryption mechanism is used to assist users to decrypt. Finally, the security analysis shows that the proposed scheme can achieve IND-CPA security in the public cloud and PRV-CDA security in the private cloud by resorting to the hybrid cloud architecture. The performance analysis shows that the deduplication search is more efficient and the computation cost of user encryption is smaller.

Key words: Attribute-Based Encryption (ABE); Data deduplication; Auditing; Users revocation; Proxy decryption

1 引言

随着大数据和云计算技术的快速发展，越来越多的企业和个人将自己的敏感数据加密后上传到云服务器。这样不仅可以降低本地的存储开销，还可以享受高质量的云存储服务[1]。在云存储中，属性加密(Attribute-Based Encryption, ABE)方案实现了数据“一对多”的细粒度访问控制。由于早期的ABE方案采用门限操作，策略表达比较单一。有学者提出了基于密文策略[2](Ciphertext-Policy, CP)和密钥策略[3](Key-Policy, KP)的ABE方案，支持灵活的访问控制策略。然而，对于这些ABE方案，当系统设置的属性值用完后，必须重建系统。若在构建系统之初就把属性值设置的很大，会非常浪费资源。文献[4]首次提出了支持大属性空间的ABE方案，该方案中的属性空间不再是多项式有界的，并且在建立系统的时候不需要设置属性个

收稿日期：2017-10-10；改回日期：2018-11-14；网络出版：2018-11-19

*通信作者： 党乾龙 xidianqldang@163.com

基金项目：国家自然科学基金(61702401, 61472470)，中国博士后科学基金(2017M613083)

Foundation Items: The National Natural Science Foundation of China (61702401, 61472470), The China Postdoctoral Science Foundation (2017M613083)

数的上限。

在云存储中，随着存储在云服务器中的数据呈爆炸式增长，存储数据的计算开销也会大大增加。因此，如何管理日益增长的海量数据成为了一个重要的问题。数据去重技术[5,6]就是一种优化云存储空间和压缩数据的方法。然而，在ABE方案中，相同的明文在不同的访问策略下会被加密成不同的密文。因此，ABE方案进行去重是非常困难的。文献[7]通过引入代理重加密技术，首次提出了支持去重的ABE方案。该方案一方面可以实现语义安全性，另一方面可以通过特定的访问策略而不是分发密钥来分享数据。用户在去重的过程中，需要云服务器搜索去重数据。文献[7]在去重搜索的过程中，双线性对的计算次数与云存储数据量呈线性关系。文献[8]通过随机化标签进行高效的云存储数据去重，该方案引入了决策树技术，在标签进行等式测试算法的过程中，双线性对计算次数降为对数级。文献[9]提出了高效和隐私保护的跨域大数据去重方案，该方案利用去重决策树将去重搜索的双线性对计算量降为常数级。但该方案是3层跨域去重架构，它的去重决策树不适用于常见的2层跨用户去重架构。

当用户将数据存储到云服务器上时，用户失去了对数据的管理权。因此，用户能够及时了解存储数据的完整性是非常重要的，云存储审计[10−12]就是检查存储在云服务器上数据的完整性。文献[10,11]是支持云存储数据公共审计的方案，然而，这两个方案都不是ABE方案，也不支持过期用户撤销。文献[13]提出了支持用户撤销和数据完整性验证的ABE方案,该方案是基于大属性空间构建的支持用户撤销和数据完整性验证的方案。此外，在ABE方案中，用户解密密文需要计算大量的指数和双线性对运算。文献[13,14]运用云服务器代理解密技术，减少了用户解密密文的计算量。

尽管以往的学者分别提出了支持去重的ABE方案[7]和支持审计与撤销的ABE方案[13]，但目前还没有一个同时支持去重、审计和用户撤销的ABE方案。此外，支持去重的ABE方案的去重搜索和用户解密效率都较低，这将影响云存储技术在实际中的应用。针对以上问题，本文提出了一种支持高效去重和审计的ABE方案。本文方案引入了第3方审计者对云存储数据进行审计，利用代理辅助用户撤销机制对过期用户进行撤销，又提出了高效去重搜索树(Efficient Deduplication Search Tree, EDST)技术将去重搜索过程中双线性对的计算量降为常数级，并通过代理解密机制辅助用户解密，用户只需1个指数运算就可以解密出明文。

2 预备知识

2.1 双线性映射

设G和G_T是两个p阶乘法循环群，p为素数。g为群G的随机生成元，若映射$e: G\times G\to G_T$能满足以下3个性质，则称$e: G\times G\to G_T$为双线性映射：

(1) 双线性：对于$\forall u, v\in G$和$a, b\in Z_P^*$，这里有$e(u^a, v^b)=e(u, v)^{ab}$。

(2) 非退化性：$e(g, g)\neq 1$。

(3) 可计算性：对于$\forall u, v\in G$，存在一个算法在有效时间内计算出$e(u, v)$。

2.2 高效去重搜索树(EDST)

高效去重搜索树由节点和枝干组成，叶子节点存储密文元组的指针，枝干和非叶子节点判断密文元组指针的移动路径。高效去重搜索树与决策树、去重决策树不同，高效去重搜索树中，只有叶子节点存储密文元组的指针，决策树和去重决策树所有节点中都存储着密文元组。因此，高效去重决策树的构造更加简单高效。

高效去重搜索树由标签T_2来生成，标签T_2是二进制字符串，下面举例说明高效去重搜索树的生成过程。假设标签T_2的长度是5 bit，标签$T_{2,i}$表示第i个数据的第2个标签。如图1(a)，第1个数据插入，标签是$T_{2,1}=00110$。从根节点开始，标签对应字符串第1位是0，则密文元组的指针移动到根节点的左子节点；同理，标签对应字符串第2位是0，则密文元组的指针移动到当前节点的左子节点；依次类推，根据标签$T_{2,1}$生成图1(a)的高效去重搜索树，该树的叶子节点存储密文元组的指针。第2个数据插入，标签为$T_{2,2}=00010$，按照上述方法在图1(a)的高效去重搜索树基础上生成图1(b)的高效去重搜索树，第2个密文元组的指针存储在第2个叶子节点中。第3个数据插入，标签为$T_{2,3}=11011$，依照上述方法在图1(b)的高效去重搜索树基础上生成图1(c)的高效去重搜索树，第3个密文元组的指针存储在第3个叶子节点中。

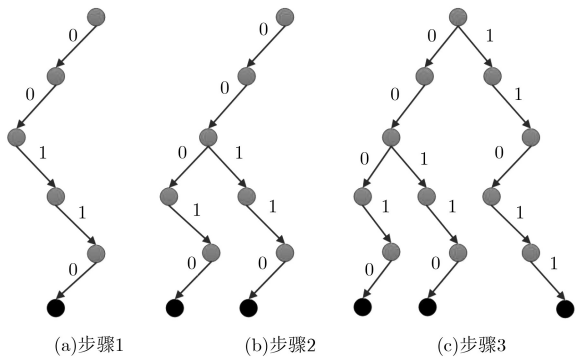

图 1 高效去重搜索树生成过程举例

3　支持高效去重和审计的属性加密方案

本节主要描述一种属性加密方案的具体结构，该方案支持高效去重和审计，并支持过期用户撤销和轻量级用户解密。

3.1　系统建立和密钥生成阶段

系统建立$(1^\lambda)\to(\mathrm{PP},\mathrm{MK})$：属性中心输入安全参数$\lambda$。设$p$为大素数，随机选择阶数为$p$，生成元为$g$的群$G$，双线性映射$e: G\times G\to G_T$。选择抗碰撞哈希函数：$H_0: G_T\to Z_p$，$H_1: M\to Z_p$，$H_2: G_T\to K$，$H_3: M'\to Z_p$，$H_4: G\to Z_p$，$H_5: Z_p^*\to G$。选择随机元素$u,h,w,v\in G$和$\alpha\in Z_p$，主密钥设置为$\mathrm{MK}=\alpha$，系统公共参数为$\mathrm{PP}=(H_0,H_1,H_2,H_3,H_4,H_5,g,u,h,w,v,e(g,g)^\alpha)$。

密钥对生成$(x_i)\to(\mathrm{PK}_i,\mathrm{SK}_i)$：用户随机选择$x_U\in Z_p$，生成自己的密钥对$(\mathrm{PK}_U=g^{x_U},\mathrm{SK}_U=x_U)$。用同样的方法生成私有云和第1个上传者的密钥对$(\mathrm{PK}_{\mathrm{PC}}=g^{x_{\mathrm{PC}}},\mathrm{SK}_{\mathrm{PC}}=x_{\mathrm{PC}})$，$(\mathrm{PK}_{\mathrm{FU}}=g^{x_{\mathrm{FU}}},\mathrm{SK}_{\mathrm{FU}}=x_{\mathrm{FU}})$。

代理密钥生成$(\mathrm{PP},\mathrm{MK},\mathrm{PK}_{\mathrm{PC}},\mathrm{PK}_U,A)\to(\mathrm{P}x\mathrm{K}_U)$：属性中心输入公共参数PP，主密钥$\mathrm{MK}=\alpha$，私有云公钥$\mathrm{PK}_{\mathrm{PC}}=g^{x_{\mathrm{PC}}}$，用户公钥$\mathrm{PK}_U=g^{x_U}$和属性集$A=\{A_1,A_2,\cdots,A_t\}$。算法随机选择$r,r',r_1,\cdots,r_t\in Z_p,\forall A_i\in A$，计算代理密钥：$\mathrm{P}x\mathrm{K}_U=\left(K_0=g^{rx_{\mathrm{PC}}+\alpha x_U}w^{r'},\ K_1=g^r,\ K_2=g^{r'},\forall A_i\in A:\{K_{i,1}=g^{r_i},K_{i,2}=(u^{A_i}h)^{r_i}v^{-r'}\}\right)$，并将代理密钥发送给私有云。

3.2　用户加密数据阶段

加密$(\mathrm{PP},m,(\boldsymbol{M},\rho))\to(\mathrm{sk}_T,\mathrm{CT},\mathrm{ZKP})$：上传者输入公共参数PP，消息$m\in M$和LSSS访问结构$(\boldsymbol{M},\rho)$，$\boldsymbol{M}$是一个$l\times n$矩阵，函数$\rho$是把矩阵$\boldsymbol{M}$的第$i$行映射到属性$\rho(i)$，即$\boldsymbol{M}\in Z_p^{l\times n}$，$\rho:[l]\to Z_p$。选一个向量$\boldsymbol{y}=(\mu,y_2,y_3,\cdots,y_n)^{\mathrm{T}}\in Z_p^n$，$\mu$是随机选取的秘密共享值，共享向量$\boldsymbol{v}=(v_1,v_2,\cdots,v_l)^{\mathrm{T}}=\boldsymbol{M}\cdot\boldsymbol{y}$，随机选取$\beta\in G_T$，$z_1,z_2,\cdots,z_l\in Z_p$，属性中心随机选择秘密值$s$发送给上传者，这里$s\in Z_p$。计算标签和密文如式(1)和式(2)。

$$\begin{aligned}\mathrm{Tag}=\Big(&T_1=(R,B)=\left(g^{H_1(m)\mu},g^{\mu}\right),\\&T_2=H_3(m\parallel e(g,g)^{\alpha s}),\\&L=g^{H_1(m)}h^{H_0(\beta)}\Big)\end{aligned}\tag{1}$$

$$\begin{aligned}\mathrm{ct}=\Big(&(\boldsymbol{M},\rho),C=\mathrm{Enc}(H_2(\beta),m),B=g^{\mu},\\&E=\beta\cdot e(g,g)^{\alpha\mu},\{C_i=w^{v_i}v^{z_i},D_i=g^{z_i},\\&E_i=(u^{\rho(i)}h)^{-z_i},F_i=g^{v_i}\}_{i\in[1,l]}\Big)\end{aligned}\tag{2}$$

这里，Enc表示对称加密算法，$H_2(\beta)$是对称加密算法的密钥。此外，上传者需要向私有云证明标签和数据的一致性，需要运用一个零知识证明。零知识证明(Zero-Knowledge Proof, ZKP)包括$(R,B,T_2,L,\delta_1,\delta_2)$，上传者随机选择$f_1,f_2\in Z_p^*$，计算零知识证明如式(3)。

$$\begin{aligned}\mathrm{ZKP}=\Big(&P_1=B^{f_1}g^{T_2},P_2=g^{f_1T_2}h^{f_2T_2},\\&c=H_4(R,B,T_2,L,P_1,P_2),\\&\delta_1=f_1T_2-c\cdot H_1(m),\\&\delta_2=f_2T_2-c\cdot H_0(\beta)\Big)\end{aligned}\tag{3}$$

在计算c时，上传者将R,B,T_2,L,P_1,P_2并在一起，即$R\parallel B\parallel T_2\parallel L\parallel P_1\parallel P_2$，再利用安全的哈希函数$H_4$计算哈希值$C=H_4(R,B,T_2,L,P_1,P_2)=H_4(R\parallel B\parallel T_2\parallel L\parallel P_1\parallel P_2)$。上传者输出陷门密钥$\mathrm{sk}_T=w^u$，密文元组$\mathrm{CT}=(\mathrm{Tag},\mathrm{ct})$和零知识证明ZKP。

3.3　私有云去重阶段

3.3.1　私有云检测数据的有效性

有效性测试$(\mathrm{PP},\mathrm{CT})\to 1/0$：私有云输入公共参数PP，密文元组CT和零知识证明ZKP。私有云利用上传者提供的零知识证明(ZKP)来验证密文的有效性。先计算$P_1=R^cB^{\delta_1}$，$P_2=L^cg^{\delta_1}h^{\delta_2}$，根据上传者提供的$(R,B,T_2,L)$和私有云计算的$(P_1,P_2)$，私有云计算$H_4(R,B,T_2,L,P_1,P_2)$并与上传者提供的$c$对比。如果$c=H_4(R,B,T_2,L,P_1,P_2)$，输出1，接受CT，存储$(T_1,T_2,L)$到私有云，并存储$(L,((\boldsymbol{M},\rho),C,B,E,\{(C_i,D_i,E_i,F_i)\}_{i\in[1,l]}))$到公有云；否则，输出0，拒绝CT。

上传者提供的零知识证明(ZKP)只在私有云进行有效性测试阶段使用，当通过有效性测试后，私有云将删除上传者提供的零知识证明。

3.3.2　高效去重搜索树构造

本文参考决策树技术和去重决策树技术构建了高效去重搜索树。设定标签T_2是60 bit的二进制字符串，在云服务器初始化时，假设私有云收到i个不同的密文元组$\{\mathrm{CT}_1,\mathrm{CT}_2,\cdots,\mathrm{CT}_i\}$。私有云根据表1的算法1构建高效去重搜索树。

3.3.3　高效去重搜索树搜索重复数据

假设一个用户U希望上传数据m到云服器，U计算密文元组CT发送给私有云。私有云先根据T_2的值按照表2的算法2在EDST上搜索重复数据，当找到$T_2=T_{2,i}$。用标签$T_1=(g^{H_1(m)\mu},g^{\mu})$和$T_{1,i}=(g^{H_1(m^*)\mu_i},g^{\mu_i})$判断上传数据和已有数据是否相同。如果$e(g^{H_1(m)\mu},g^{\mu_i})=e(g^{H_1(m^*)\mu_i},g^{\mu})$，私有云找到重复数据，对云存储中的密文进行重加密，存储重加密后的密文，删除原密文；如果

表 1 高效去重搜索树构造

算法 1 高效去重搜索树构造

假设私有云需要存储密文元组CT_i的指针到EDST叶子节点。(初始化时，当前节点从根节点开始。假设$|T_{2,i}|$=60 bit。)

步骤 1 若$1 \le j < 60$时，私有云判断标签$T_{2,i}$第j位。如果第j位是0，密文元组CT_i的指针移动到当前节点的左子节点；否则移动到右子节点。

步骤 2 若$j = 60$时，私有云判断标签$T_{2,i}$第60位。如果第60位是0，密文元组CT_i的指针移动到当前节点的左子节点，并在当前节点存储密文元组CT_i的指针；否则移动到右子节点，并在当前节点存储密文元组CT_i的指针。

表 2 数据在高效去重搜索树上去重搜索

算法 2 数据在高效去重搜索树上去重搜索

U发送密文元组CT_i到私有云，私有云收到CT_i后，它开始从根节点检查重复数据。(初始化时，当前节点从根节点开始。)

步骤 1 若$1 \le j < 60$时，私有云判断标签$T_{2,i}$第j位；

(1)如果第j位是0。若当前节点有左子节点，则密文元组CT_i的指针移动到当前节点的左子节点；否则，重复数据没有找到；

(2)如果第j位是1。若当前节点有右子节点，则密文元组CT_i的指针移动到当前节点的右子节点；否则，重复数据没有找到。

步骤 2 若$j = 60$时，私有云判断标签$T_{2,i}$第60位；

(1)如果第60位是0。若当前节点有左子节点，则密文元组CT_i的指针移动到当前节点的左子节点，转(3)步；否则，重复数据没有找到；

(2)如果第60位是1。若当前节点有右子节点，则密文元组CT_i的指针移动到当前节点的右子节点，转(3)步；否则，重复数据没有找到；

(3)密文元组CT_i的指针找到密文元组CT_j，私有云判断等式$e(g^{H_1(m)\mu_i}, g^{\mu_j}) = e(g^{H_1(m)\mu_j}, g^{\mu_i})$是否成立。若成立，则重复数据找到；否则，重复数据没有找到。

$e(g^{H_1(m)\mu}, g^{\mu_i}) \ne e(g^{H_1(m^*)\mu_i}, g^{\mu})$，私有云没有找到重复数据，存储$m$的密文。

3.3.4 去重数据重加密

重加密$(\mathrm{PP}, \mathrm{sk}_T, L, \mathrm{ct}, (\boldsymbol{M}', \rho')) \to (L', \mathrm{ct}')$：私有云输入公共参数PP，陷门密钥$\mathrm{sk}_T$，密文$((\boldsymbol{M}, \rho), C, B, E, \{(C_i, D_i, E_i, F_i)\})$，标签$L$，一个LSSS访问结构$(\boldsymbol{M}', \rho')$，$\boldsymbol{M}'$是一个$l_1 \times n_1$矩阵。随机选取$\bar{\boldsymbol{y}} = (\bar{\mu}, y_2', \cdots, y_{n_1}')^{\mathrm{T}} \in Z_p^{n_1}$。对于$\boldsymbol{M}'$的每行$\boldsymbol{M}_i' = (m_{i,1}', \cdots, m_{i,n_1}')$，$i \in [1, l_1]$，随机选取$z_i' \in Z_p$。$\boldsymbol{y}' = (\mu', y_2', \cdots, y_{n_1}')^{\mathrm{T}}$，$\mu' = \mu + \bar{\mu}$，对于$i \in [1, l_1]$，输出新密文：

$$\begin{aligned}\mathrm{ct}' = \Big(& (\boldsymbol{M}', \rho'), C' = \mathrm{Enc}(H_2(\beta), m), B' = g^{\mu'}, \\ & E' = \beta \cdot e(g,g)^{\alpha\mu'}, \{ C_i' = w^{\boldsymbol{M}_i'\boldsymbol{y}'} v^{z_i'}, D_i' = g^{z_i'}, \\ & E_i' = \left(u^{\rho'(i)} h\right)^{-z_i'}, F_i' = g^{\boldsymbol{M}_i'\boldsymbol{y}'} \}_{i \in [1,l_1]} \Big) \end{aligned} \tag{4}$$

此外，标签$L' = L$。私有云将重加密后新的密文ct'和标签L'发送给公有云。公有云删除原来的密文ct，存储新的密文ct'。

3.4 解密阶段

用户向云服务器提出下载云存储数据的请求，属性中心先验证用户的属性是否满足密文的访问结构。如果满足，属性中心生成该用户的代理密钥，并将代理密钥发送给私有云，私有云进行代理解密，用户再进行解密。

代理解密$(x_{\mathrm{PC}}, \mathrm{PxK}_U, \mathrm{ct}') \to (\mathrm{ME})$：私有云输入私钥$x_{\mathrm{PC}}$与属性集$A$相关的代理密钥$\mathrm{PxK}_U = (K_0, K_1, K_2, \forall A_i \in A : \{K_{i,1}, K_{i,2}\})$和元组$(E', B', C_{i'}', D_{i'}', E_{i'}', F_{i'}')$。计算属性集$A$中共享矩阵$\boldsymbol{M}'$的行集合，$I = \{i : \rho'(i) \in A\}$。如果集合$A$是被策略授权集合，则存在使$\sum_{i \in I} w_i \boldsymbol{M}_i' = (1, 0, \cdots, 0)$成立的常数集合$\{w_i \in Z_p\}_{i \in I}$，其中，$\boldsymbol{M}'_i$是矩阵$\boldsymbol{M}'$的第$i$行。私有云执行代理解密如式(5)。

$$\begin{aligned} & \frac{e(B', K_0)}{\prod_{i \in I} \left(e(C_i', K_2) e(E_i', K_{i,1}) e(D_i', K_{i,2}) e(F_i', K_1)^{x_{\mathrm{PC}}} \right)^{w_i}} \\ & = \frac{e(g,g)^{\alpha\mu' x_u} e(g,g)^{r\mu' x_{\mathrm{PC}}} e(g,w)^{\mu' r'}}{\prod_{i \in I} e(g,g)^{r y_i' w_i x_{\mathrm{PC}}} e(g,w)^{r' y_i' w_i}} \\ & = e(g,g)^{\alpha\mu' x_U} = \mathrm{ME} \end{aligned} \tag{5}$$

用户解密$(\mathrm{SK}_U, E', \mathrm{ME}, C') \to (m)$：用户输入私钥$\mathrm{SK}_U = x_U$，密文$E'$和中间值ME，解密如式(6)。

$$\frac{E'}{\left(e(g,g)^{\alpha\mu' x_U}\right)^{x_U^{-1}}} = \frac{\beta \cdot e(g,g)^{\alpha\mu'}}{e(g,g)^{\alpha\mu'}} = \beta \tag{6}$$

然后，用户计算$m = \mathrm{Dec}(H_2(\beta), C')$。

3.5 用户身份撤销阶段

用户身份撤销$(\Gamma, U) \to (\Gamma')$：输入用户身份$U$和代理密钥列表$\Gamma$，用户撤销算法从代理密钥列表中删除身份$U$及其对应的代理密钥，例如：$\Gamma' = \Gamma \backslash \{U, \mathrm{PxK}_U\}$。

用户的属性通过验证后，发送自己的身份到私有云，私有云存储该用户的身份。当用户成为过期用户的时候，私有云除去列表中身份和代理密钥，用户将不能访问数据。

3.6 数据拥有者进行审计阶段

该方案中，第1个数据上传者在上传密文时，将密文C分为n块即$C = C[1] \parallel C[2] \parallel \cdots \parallel C[n]$。设第$j$块密文是$C[j]$，对应的身份标识符$\mathrm{ID}_j$,其中$j = 1, 2, \cdots, n$，计算验证符$\Phi_1, \Phi_2, \cdots, \Phi_n$。第1个上传者把密文块和验证符一同发送给私有云，后续上传者找到去重文件，成为去重文件的所有者，第3方审计者在对应数据的身份列表中增加该数据拥有

者身份信息。然后，该数据拥有者就可以委托第3方审计者对云存储数据进行审计。

验证符生成$(\mathrm{SK_{FU}}, C[j], \mathrm{ID}_j) \to (\Phi_j)$：第1个上传者输入自己的私钥$\mathrm{SK_{FU}} = x_{\mathrm{FU}}$和密文块以及密文块的身份标识$(C[j], \mathrm{ID}_j)$，其中$j = 1, 2, \cdots, n$。计算验证符$\Phi_j = (H_5(\mathrm{ID}_j) \cdot u^{C[j]})^{x_{\mathrm{FU}}}$,数据拥有者将数据块和验证符一同发给私有云。

审计$(\mathrm{PK_{FU}}, \mathrm{Cs}, \Phi, \phi) \to 1/0$：第3方审计者输入第1个上传者公钥$\mathrm{PK_{FU}} = g^{x_{\mathrm{FU}}}$，随机选择挑战序列$\mathrm{Cs} = \{(C[j], b_j)\}_{j \in D}$，其中$b_j \in Z_p^*$, $S = \{s_1, s_2, \cdots, s_c\}$是集合$[1, n]$中的子集，将选择好的序列发送给私有云。私有云计算一个集合验证符$\Phi = \prod_{j \in S} \Phi_j^{b_j}$，计算密文块的线性组合$\phi = \sum_{j \in S} b_j C[j]$，将$(\Phi, \phi)$发送给第3方审计者作为存储数据完整性验证的证据。当第3方审计者收到(Φ, ϕ)，第3方审计者将会验证式(7)是否成立。

$$e\left(\prod_{j \in S} H(\mathrm{ID}_j)^{b_j} \cdot u^{\phi}, \mathrm{PK_{FU}}\right) = e(g, \Phi) \tag{7}$$

如果等式成立，输出1；否则，输出0。

4　安全性分析

4.1　数据去重阶段

文献[7]利用ZKP证明了标签(T_1, L)和密文C的一致性。本文方案在文献[7]基础上利用零知识证明方案证明了标签(T_1, T_2, L)和密文C的一致性，保证了密文和标签是一致的。

引理 1　对于零知识证明的证据(M, β),ZKP是一个安全的零知识证明系统。

证明　由于ZKP的完整性是显然的，这里关注其合理性和零知识。

合理性：假设有两个相同的元组(R, T_2, L)的副本，但是有不同的挑战c'和c与不同的响应(δ_1', δ_2')和(δ_1, δ_2)。然后(μ, M)可以从式(8)中提取：

$$\left.\begin{aligned} R &= B^{H_1(m)} = B^{\frac{\delta_1' - \delta_1}{c - c'}} \\ T^2 &= \frac{c'(\delta_2 - \delta_1) + c(\delta_1' - \delta_2')}{(f_1 - f_2)(c - c')} \\ L &= g^{H_1(m)} h^{H_0(\beta)} = g^{\frac{\delta_1' - \delta_1}{c - c'}} h^{\frac{\delta_2' - \delta_2}{c - c'}} \end{aligned}\right\} \tag{8}$$

其中由$(f_1 - f_2)$无法推测出f_1和f_2。

零知识：模拟器随机选取$\delta_1, \delta_2 \in Z_p^*$, $c \in Z_p^*$，然后私有云计算：$P_1 = R^c B^{\delta_1}$, $P_2 = L^c g^{\delta_1} h^{\delta_2}$。其中令$c = H_4(R, B, T_2, L, P_1, P_2)$。　　证毕

加密数据在私有云和公有云中的隐私性：在公有云中达到IND-CPA安全性，在私有云中达到PRV-CDA安全性。

定理 1　假设$(q-1)$假设在G中成立，Enc是安全的对称加密方案，L由安全承诺方案生成，则所提出的方案就公有云而言达到IND-CPA安全性。

证明　假设$(q-1)$假设在G中成立，文献[4]的方案达到IND-CPA安全性。本文方案的证明与方案[4]相似，除了在挑战阶段加入密文C^*和标签L^*。由于对称加密的安全性，C^*不会泄露明文数据的任何信息。因为零知识证明的特性，L^*也不会泄露明文数据的任何信息。因此，本文方案就公有云而言达到IND-CPA安全性。　　证毕

定理 2　假设$(q-1)$假设在G中成立，判定BDH假设在G中成立，Enc是安全对称加密方案，PoK是安全的零知识证明，则所提的方案就私有云而言达到PRV-CDA安全性。

证明　PRV-CDA安全性由加密算法(敌手$\mathcal{A}_1$)和重加密算法(敌手$\mathcal{A}_2$)的安全性构成。安全抵抗敌手$\mathcal{A}_1$的算法是包括两部分：密文和证明。就密文而言，除了在挑战阶段，E和L将被添加到挑战密文中，其它的证明与定理1相似。关于证明，由于零知识证明的特性，不会泄露关于m_b的任何信息。因此，这里主要研究重加密算法的安全性，将重加密算法的安全性规约到判定BDH假设。

下面描述在判定BDH假设下的安全性证明。假设存在敌手$\mathcal{A}_2$可以攻破本文系统的PRV-CDA安全性，我们可以构建一个挑战算法$\mathcal{B}$解决判定BDH问题。算法$\mathcal{B}$给出(g, g^a, g^b, g^c, Z)，如果$Z = e(g, g)^{abc}$，输出0；如果Z在G_T中是均匀的，输出1。

算法$\mathcal{B}$随机选择$x \in Z_p^*$, $u, h, v \in G$，计算$w = g^x$。将公共参数设置为$\mathrm{PP} = (H_1, H_4, g, u, h, w, v, e(g^a, g^b))$，这里$H_1$, H_4是抗碰撞哈希函数。这意味着主密钥$\alpha = ab$对于算法$\mathcal{B}$来说是未知的。

当算法$\mathcal{A}_2$输出一个访问策略$(\boldsymbol{M}^*, \rho^*)$，算法$\mathcal{B}$首先从消息空间中选择一个密文$m_b \in \{m_0, m_1\}$，其中$b \in \{0, 1\}$。然后随机选择$\tilde{c}, y_2, \cdots, y_n \in Z_p$，设$\boldsymbol{y} = (\mu, y_2, y_3, \cdots, y_n)^{\mathrm{T}}$，$\bar{\boldsymbol{y}} = (\bar{\mu}, y_2', \cdots, y_{n_1}')^{\mathrm{T}}$。算法$\mathcal{B}$随机随机选择$\beta \in G_1$, $z_1, z_2, \cdots, z_l \in Z_p$。输出陷门密钥、标签和密文元组：

$$\left.\begin{aligned} &\mathrm{sk}_T = w^c = (g^c)^x, L = g^{H_1(m_b)} h^{H_0(\beta)} \\ &C = \mathrm{Enc}(H_2(\beta), m_b), B = g^c, E = \beta \cdot Z \\ &\tilde{B} = g^{\tilde{c}}, \tilde{E} = \beta \cdot Z, C_i = w^{\boldsymbol{M}_i^* \boldsymbol{v}} v^{z_i}, D_i = g^{z_i} \\ &E_i = (u^{\rho^*(i)} h)^{-z_i}, F_i = g^{v_i} \end{aligned}\right\} \tag{9}$$

这里对于$i \in [1, l]$, C_i可以在不知道值c的情况下被计算出来。

$$C_i = w^{\boldsymbol{M}_i^* \boldsymbol{v}} v^{z_i} = w^{(cm_{i1}^* + \ldots + y_n m_{in}^*)} v^{z_i} = (w^c)^{m_{i1}^*} w^{(y_2 m_{i2}^* + \cdots + y_n m_{in}^*)} v^{z_i} = (g^c)^{x m_{i1}^*} w^{(y_2 m_{i2}^* + \cdots + y_n m_{in}^*)} v^{z_i} \tag{10}$$

因为$Z = e(g,g)^{abc} = e(g^a, g^b)^c$，则元组$\Big(L, \big((\boldsymbol{M}^*, \rho^*), C, B, E, \{(C_i, D_i, E_i, F_i)\}\big)\Big)$的分布是清楚的，对于敌手$\mathcal{A}_2$算法来说，陷门密钥$\mathrm{sk}_{\mathcal{T}}$是和重加密算法相同的。最终，算法$\mathcal{A}_2$输出一个猜测$b'$。如果$b' = b$，算法$\mathcal{B}$输出0，即$Z = e(g,g)^{abc}$；否则，输出1。

当$Z = e(g,g)^{abc}$，C^*是由安全的对称加密方案生成，L是由安全的承诺方案生成。对于算法$\mathcal{A}_2$来说这和实际游戏是相同的。当Z在$G_{\mathcal{T}}$中均匀分布，C和L是均匀生成时，对于算法$\mathcal{A}_2$来说，b的值是保密的。因此，算法$\mathcal{A}_2$可以攻破本文方案的PRV-CDA安全性，算法$\mathcal{B}$可以解决判定BDH问题。最终，证明所提方案就私有云而言达到PRV-CDA安全性。 证毕

4.2 数据审计阶段

审计方案的安全性分析与方案[4]相似，云服务器只有拥有正确的数据块和认证符才能通过第3方审计者的验证过程。假设云服务器中的数据C被损坏了，云服务器中存储损坏数据C'。云服务器计算$\Phi = \prod_{j \in S} \Phi_j^{b_j}$和$\phi' = \sum_{j \in S} b_j C'[j]$（$C'[j]$是损坏的密文块），然后发送有效证明$(\Phi, \phi)$给第3方审计者，第3方审计者根据有效证明$(\Phi, \phi)$计算等式是否成立：$e\left(\prod_{j \in S} H(\mathrm{ID}_j)^{b_j} \cdot u^{\phi'}, \mathrm{PK}_{\mathrm{FU}}\right) = e(g, \Phi)$，由于$\phi' \neq \phi$，等式不成立，云服务器不能通过第3方审计者的验证。

5 性能分析

本文方案与文献[7,13]方案进行功能和效率的比较，这里主要考虑了大属性空间、数据审计、用户撤销、数据去重、去重搜索阶段的计算量、用户解密阶段的计算量和用户私钥的长度。3个方案的比较结果如表3所示。表中使用符号如下：$\mathrm{Exp}(G)$表示G上的指数运算；$\mathrm{Exp}(G_{\mathcal{T}})$表示$G_{\mathcal{T}}$上的指数运算；Pair表示双线性对运算；$n$表示云存储中数据的数量；$k$表示用户属性集的空间；$|S|$表示一个私钥中属性集的大小。

表 3 本文方案与不同方案之间的比较

方案	大属性空间	数据审计	用户撤销	数据去重	去重搜索阶段的计算量	用户解密阶段的计算量	用户私钥的长度
文献[7]方案	√	×	×	√	$2\mathrm{Exp}(G) + 2n\mathrm{Pair}$	$\leq (k+2)\mathrm{Exp}(G) + (3k+1)\mathrm{Pair}$	$2\|S\| + 2$
文献[13]方案	√	√	√	×	/	$\mathrm{Exp}(G)$	$2\|S\| + 3$
本文方案	√	√	√	√	$2\mathrm{Exp}(G) + \mathrm{Exp}(G_{\mathcal{T}}) + 2\mathrm{Pair}$	$\mathrm{Exp}(G)$	$2\|S\| + 3$

如表3所示，本文方案同时支持大属性空间、数据审计、用户撤销和数据去重，其它方案只能实现其中的部分功能。在去重搜索阶段，文献[7]方案的计算量与云存储数据量呈线性增加的关系；文献[13]方案由于不支持去重，这里不考虑该方案的计算量；本文方案的计算量是常数级，不随着云存储数据量的增加而增加。在解密阶段，文献[7]方案的计算量随着用户属性集空间的增加而线性增加；文献[13]方案和本文方案的计算量是常数级，只需1个指数运算就可以解密出明文。在用户私钥存储阶段，本文方案与文献[13]方案相同，比文献[7]方案的用户私钥长度长1个单位。

6 结束语

随着大数据和云计算技术的飞速发展，云存储业务的重要性越来越突出。为了提高云存储服务的质量和用户解密密文的效率，本文提出了支持高效去重和审计的属性加密方案，并给出了方案的安全性分析和性能分析。在云存储服务中，本文方案不仅支持高效去重搜索，还支持云存储数据审计和过期用户撤销。在用户解密密文的过程中，利用代理解密技术，提高了解密效率。

参 考 文 献

[1] CHOO K K R, HERMAN M, IORGA M, *et al*. Cloud forensics: State-of-the-art and future directions[J]. *Digital Investigation*, 2016, 18: 77–78. doi: 10.1016/j.diin.2016.08.003.

[2] LAI Junzuo, DENG R H, and LI Yingjun. Fully secure cipertext-policy hiding CP-ABE[C]. International Conference on Information Security Practice and Experience, Guangzhou, China, 2011: 24–39. doi: https://doi.org/10.1007/978-3-642-21031-0_3.

[3] YU Shucheng, WANG Cong, REN Kui, *et al*. Achieving secure, scalable, and fine-grained data access control in cloud computing[C]. Proceedings of the 29th Conference on Information Communications, San Diego, USA, 2010: 1–9. doi: 10.1109/INFCOM.2010.5462174.

[4] ROUSELAKIS Y and WATERS B. Practical constructions and new proof methods for large universe attribute-based encryption[C]. ACM Sigsac Conference on Computer &

Communications Security, Berlin, Germary, 2013: 463–474. doi: 10.1145/2508859.2516672.

[5] BELLARE M, KEELVEEDHI S, and RISTENPAR T. Message-locked encryption and secure deduplication[C]. Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, 2013: 296–312. doi: https://doi.org/10.1007/978-3-642-38348-9_18.

[6] BELLARE M and KEELVEEDHI S. Interactive messagee-locked encryption and secure deduplication[C]. IACR International Workshop on Public Key Cryptography, Gaithersburg, USA, 2015: 516–538. doi: https://doi.org/10.1007/978-3-662-46447-2_23.

[7] CUI Hui, DENG R H, LI Yingjiu, *et al*. Attribute-based storage supporting secure deduplication of encrypted data in cloud[J]. *IEEE Transactions on Big Data*, 2017(99): 1–13. doi: 10.1109/TBdata.2017.2656120.

[8] JIANG Tao, CHEN Xiaofeng, WU Qianhong, *et al*. Secure and efficient cloud data deduplication with randomized tag[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 12(3): 532–543. doi: 10.1109/TIFS.2016.2622013.

[9] YANG Xue, LU Rongxing, CHOO K K R, *et al*. Achieving efficient and privacy-preserving cross-domain big data deduplication in cloud[J]. *IEEE Transactions on Big Data*, 2017(99): 1–12. doi: 10.1109/TBDATA.2017.2721444.

[10] YU Yong, LI Yannan, NI Jianbing, *et al*. Comments on “public integrity auditing for dynamic data sharing with multiuser modification”[J]. *IEEE Transactions on Information Forensics and Security*, 2016, 11(3): 658–659. doi: 10.1109/TIFS.2015.2501728.

[11] YANG Guangyuan, YU Jia, SHEN Wenting, *et al*. Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability[J]. *Journal of Systems and Software*, 2016, 113: 130–139. doi: 10.1016/j.jss.2015.11.044.

[12] SHEN Jian, SHEN Jun, CHEN Xiaofeng, *et al*. An efficient public auditing protocol with novel dynamic structure for cloud data[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 12(10): 2402–2415. doi: 10.1109/TIFS.2017.2705620.

[13] YAN Xuewei, MA Hua, LIU Zhenhua, *et al*. Large universe revocable fine-grained encryption with public auditing [C]. International Conference on Broadband and Wireless Computing, Communication and Applications, Asan, Korea, 2016: 823–830. doi: https://doi.org/10.1007/978-3-319-49106-6_84.

[14] 王光波, 王建华. 基于属性加密的云存储方案研究[J]. 电子与信息学报, 2016, 38(11): 2931–2939. doi: 10.11999/JEIT160064.

WANG Guangbo and WANG Jianhua. Research on cloud storage scheme with attribute-based encryption[J]. *Journal of Electronics & Information Technology*, 2016, 38(11): 2931–2939. doi: 10.11999/JEIT160064.

马　华：女，1963年生，教授，研究方向为网络与信息安全.

党乾龙：男，1993年生，硕士生，研究方向为网络与信息安全.

王剑锋：男，1985年生，讲师，研究方向为应用密码学、云安全和数据外包.

刘振华：男，1978年生，教授，研究方向为云计算中的密码理论与安全协议、密文数据的再处理.