基于区块链的细粒度云数据安全存储与删除方案

周由胜[①②] 陈律君*[①]

[①](重庆邮电大学计算机科学与技术学院 重庆 400065)

[②](重庆邮电大学网络空间安全与信息法学院 重庆 400065)

摘 要：在基于云计算的存储与删除服务中，由于外包数据所有权和管理分离，现有的逻辑删除机制使云上的数据很容易暴露给未经授权的用户，甚至云服务器可能未遵循用户要求删除相应数据。为此，该文提出一种细粒度的安全云端数据存储与删除方案。基于椭圆曲线构造了基于密文策略的属性基加密以实现外包数据细粒度访问控制，应用区块链实现可公开验证的安全数据删除。该文方案具有责任可追踪性以及两方删除与可验证性等特性。理论分析与实验结果表明该文方案具有较好的安全性和较高的性能，能够满足云数据共享与安全删除的需求。

关键词：云存储；安全删除；属性加密；区块链；公开验证

中图分类号：TN918 **文献标识码**：A **文章编号**：1009-5896(2021)07-1856-08

DOI: 10.11999/JEIT200399

Secure Storage and Deletion Based on Blockchain for Cloud Data with Fine-grained Access Control

ZHOU Yousheng[①②] CHEN Lüjun[①]

[①](*College of Computer Science and Technology*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

[②](*School of Cyber Security and Information Law*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

Abstract: In the storage and deletion service provided by cloud computing, due to the separation of outsourced data ownership and management, the cloud server may not follow the user's request to delete the corresponding data, and the outsourced data can be easily exposed to unauthorized users due to the widely-adopted logical deletion. Therefore, an efficient and secure cloud data storage and deletion scheme is proposed. Firstly, an attribute-based encryption based on ciphertext policy is constructed based on elliptic curves to achieve fine-grained access control. Secondly, publicly verifiable data deletion is realized by using blockchain. The proposed scheme has the characteristics of responsibility traceability, two-party deletion and verifiability. Theoretical analysis and experimental results show that the presented scheme has more desirable security and performance, and can meet the needs of cloud data sharing and secure deletion.

Key words: Cloud storage; Secure deletion; Attribute encryption; Blockchain; Public verification

1 引言

随着大量的敏感数据，如健康数据、金融数据、商业秘密、保密通信等都被外包到云端，数据安全问题引起了大量关注[1,2]。由于云服务器往往是半可信的，出于数据安全性考虑，数据所有者可能需要将某些存储在云端敏感数据删除，因而如何确保云服务运营商诚实地依照用户要求删除数据对数据拥有者而言至关重要[3–5]。除了保证云端数据的保密性和可用性外，数据拥有者如何实现安全删除其外包数据是云存储服务中需要解决的一个重要问题。

现有的数据删除方法大多基于一比特返回协议进行构造，即在假定服务器可信的情况下，数据所有者发送一个请求让云服务器从物理介质中删除数据，然后接收一个表示删除操作结果的位应答(成

收稿日期：2020-05-22；改回日期：2020-12-06；网络出版：2020-12-18

*通信作者： 陈律君 s180201047@stu.cqupt.edu.cn

基金项目：国家自然科学基金(61702067)，重庆市自然科学基金(cstc2020jcyjmsxmX0343)，重庆市留学人员回国创业创新支持计划(CX2018122)

Foundation Items: The National Natural Science Foundation of China (61702067), The Chongqing Natural Science Foundation (cstc2020jcyjmsxmX0343), The Venture & Innovation Support Program for Chongqing Overseas Returnees (CX2018122)

功/失败)[6]。如Garfinkel等人[7]提出通过删除链接到数据的系统指针的方式，但它只是删除了链接而内容仍然保留在磁盘中。Gutmann[8]提出基于随机数据覆盖存储介质的方式实现数据删除。Paul等人[9]提出了可擦除性证明(Proof of Erasability, PoE)概念，即用随机模式覆盖磁删除数据。Perito等人[10]提出安全擦除证明(PoSE-s)的方案，通过备发送一串随机模式将原始数据覆盖。通过覆盖存储介质的安全数据删除大多不支持验证而且效率较低。近年来，基于密码技术的外包数据存储与删除方案受到关注[11–17]。Perlman[12]提出了一个有保证的删除协议。张曙光等人[13]提出了一种使云服务器能够实现加密数据重复删除的方法。为了实现删除公开可验证，Yang等人[14]提出一种基于私有链的删除证据存储方案。Yu等人[15]提出利用属性基加密实现外包数据访问控制，并通过交互验证删除。Xue等人[18]提出了支持细粒度访问控制的安全删除方案，但计算代价较大，并且需要可信第三方生成重加密密钥。此外，还有部分学者考虑了基于树状存储实现安全删除[19–21]。

现有多数外包数据的安全删除方案大都假设云服务器完全可信，或依赖可信第三方协作完成安全删除，同时难以支持细粒度访问控制与删除，其安全性和效率有待于进一步提升。为此，本文提出一种基于区块链的细粒度云数据安全删除方案。首先采用基于密文策略的属性基对外包数据加密，实现数据所有者对数据的细粒度访问控制和可公开验证删除。同时，将外包的数据与属性策略相关联，通过撤销用户访问文件必不可少的属性从而确保数据删除。其次，本文提出了基于区块链的数据删除证据验证。数据所有者可以通过云服务器中已修改的密文重构默克尔哈希树(Measurement Hash Tree, MHT)，并且对比哈希链上公开的证据来验证目标数据是否已被删除。最后，本文方案基于椭圆曲线进行构造，相比传统的基于双线性映射的数据安全存储与删除方案，计算复杂度更小。在删除和验证阶段，只需要数据所有者与云服务器两方交互，不需要引入可信第三方，系统通信开销和计算开销进一步降低。

2　系统模型

2.1　安全假设

在本文中，假设云服务器为不可信实体，即云服务器可能未经数据拥有者授权删除数据或者不按照数据拥有者删除请求删除数据。假设数据拥有者为非诚实实体，即数据拥有者可能会否认自己曾经发出的数据删除请求，并诬陷云服务器未经授权删除数据，从而向云服务器索要赔偿。本方案允许被动攻击存在，即敌手可以窃听系统中的所有通信，未经授权的用户也可能相互勾结以获取明文信息。考虑到本文所提方案的应用环境，结合Ramokapane等人[22]提出的基于云的确定性删除的安全目标，将本文方案的安全目标设定为满足正确性、完整性、确定性数据删除、安全细粒度访问控制与责任追踪等要求。

2.2　系统模型

本文提出的方案包含5个实体：云服务器(Cloud Security Provider, CSP)、属性授权中心(Attribute Authorization center, AA)、数据拥有者(Data Owner, DO)、用户(Users)、区块链网络(Block Chain network, BC)。本方案的基本框架如图1所示。

(1) 云服务器(CSP)：提供存储服务、数据访问服务。此外，云服务器可根据数据拥有者请求删除数据，并将删除证据存放于区块链。

(2) 属性授权中心(AA)：为系统生成主密钥，为每个属性、合法用户生成私钥。

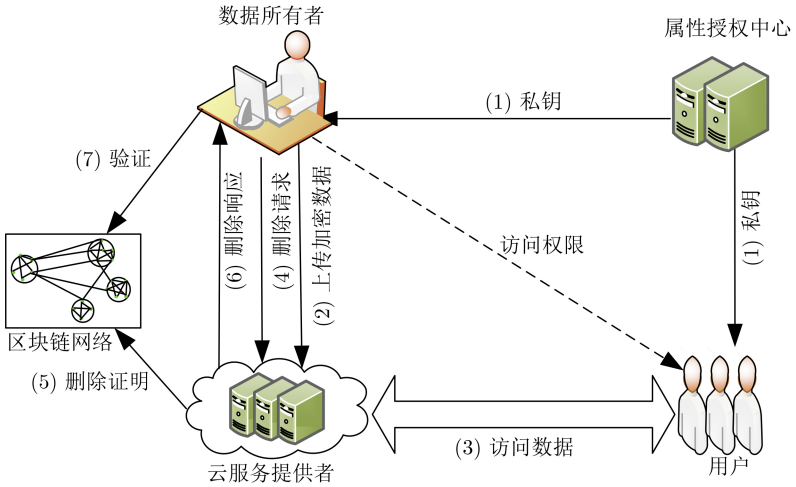


图 1 系统模型

(3) 数据拥有者(DO)：定义访问控制策略，通过定义的策略加密文件并存储到云上。此外，还可生成删除请求以及验证云服务器返回的删除结果。

(4) 用户(Users)：在云上下载并解密密文，但只有授权用户可以获取相应明文。

(5) 区块链网络(BC)：本文方案中使用联盟链以构成数据删除证据链，使用实用拜占庭算法(Practical Byzantine Fault Tolerance, PBFT)作为其共识机制。云服务器为普通节点，删除证据生成后交给所属机构的超级节点，由超级节点进行区块模拟打包，当交易记录填充完一个区块即提出出块请求。只有超级节点共同维护一份统一的包含删除证据的账本并参与共识，其他节点通过授权向超级节点申请查阅相关信息，实现删除证据公开验证服务。

2.3 算法组成

本方案包括7个算法：Setup, KeyGen, Encrypt, Decrypt, DelRequest, ReEncrypt, Verify，具体如下：

(1) Setup(1^λ)：由AA运行的系统初始化算法，将安全参数λ作为输入，输出属性公钥PK_i与系统主密钥MSK, PK_i公开而MSK由AA私密保存。

(2) KeyGen(MSK, ID, S_{ID})：由AA运行的密钥生成算法，输入主密钥MSK，用户身份标志ID以及该用户的一组属性S_{ID}，输出与用户拥有属性相关的私钥SK。

(3) Encrypt($\mathrm{PK}_i, (\boldsymbol{A}, \rho), M$)：由数据拥有者运行加密算法，该算法需要以属性公钥PK_i，访问策略$(\boldsymbol{A}, \rho)$以及明文M为输入，输出与访问策略$(\boldsymbol{A}, \rho)$相关的密文CT以及签名$\sigma_{\mathrm{SK_{D0}}}(R_j)$。这里的$R_j$为已建立的第$j$个MHT的根值，$\boldsymbol{A}$为线性秘密共享矩阵，$\rho$为一个映射，表示将矩阵$\boldsymbol{A}$的第$x$行匹配到属性$\rho(x)$。

(4) Decrypt($\mathrm{CT}, \mathrm{SK}_{\rho(x),\mathrm{ID}}$)：由用户运行的解密算法。该算法将密文CT和与该用户拥有的属性相关的私钥$\mathrm{SK}_{\rho(x),\mathrm{ID}}$作为输入，若私钥中的属性集满足密文中的访问架构，算法输出明文M，否则，算法停止。

(5) DelRequest(fname, y)：数据拥有者向云服务器申请删除数据的算法。算法将文件名fname与要更改的属性y作为算法输入，输出数据删除请求DR。

(6) ReEncrypt(CT,DR)：云服务器对密文进行重加密算法。输入密文CT与删除请求DR，输出重加密后的密文项$C'_{2,\kappa}$以及签名$\sigma_{\mathrm{SK_{CSP}}}(R'_j)$，这里的$R'_j$是更新后的MHT的根值。

(7) Verify(Resp, CT′)：数据拥有者删除验证算法。数据拥有者输入云服务器返回的删除反馈Resp，更改后的密文CT′，再结合当前哈希链的值进行验证，若验证通过，输出1，否则，输出0。

2.4 安全模型

本方案的安全模型为基于选择性访问架构安全(selective access structure security)，该模型由敌手A和挑战者C之间的游戏来定义。在游戏开始阶段，敌手A先输出一个挑战访问架构$\boldsymbol{A}^*$，接着敌手A可以发出与属性S相关的私钥查询，但这些属性不能满足访问架构$\boldsymbol{A}^*$。

模型详情如下所述：

Init： 敌手A选择一个挑战访问架构$(\boldsymbol{A}^*, \rho^*)$，此架构中属性dummy已被撤销，即用$(\boldsymbol{A}^*, \rho^*)$加密的密文已被删除。

Setup： 挑战者C执行setup算法生成系统公共参数，并为每个属性生成公私钥对。接着挑战者C将生成的公共参数发送给敌手A。

Phase 1: 敌手A向挑战者C发送与属性组S_{ID}相关的私钥查询，但所有的属性组S_{ID}不能满足$\boldsymbol{A}^*$。因为属性dummy已被撤销，则属性组S_{ID}不包括属性dummy。

Challenge: 敌手A任选两个相同长度的消息M_0, M_1发送给挑战者C，挑战者C任选$\delta \in \{0,1\}$，并基于访问架构$(\boldsymbol{A}^*, \rho^*)$加密$M_\delta$，并将加密后的密文$\mathrm{CT}^*$发送给敌手A。

Phase 2： 过程与phase 1类似，敌手A进行更多的秘钥询问。

Guess： 敌手A输出猜想，若$\delta' = \delta$，则敌手A赢得游戏。

敌手A赢得游戏的优势为$\mathrm{Adv(A)}=\left|\left[\delta=\delta'\right]-\frac{1}{2}\right|$。若对于任何拥有多项式时间的敌手A而言，若其拥有的赢得所提游戏的能力是可忽略的，则认为本文提出的方案是选择明文安全的。

3 具体方案

本节给出数据存储与安全删除方案的具体构造。为了提高算法整体性能，本文利用椭圆曲线进行构造。假设每个用户都会预先加载公共参数$\mathrm{PK}_i(1 \le i \le |U|)$以及LSSS访问矩阵。本方案的详细结构如下所示：

Setup(1^λ)：选择GF(p)为阶为P的有限域，设E是定义在GF(p)上的椭圆曲线，G为阶为r的椭圆曲线E的基。H为单向抗碰撞哈希函数。

AA任选随机数$\alpha \in Z_r$，再为$|U|$个属性选择元素$h_1, h_2, \cdots, h_{|U|} \in Z_r$，将$\mathrm{MSK} = (h_1, h_2, \cdots, h_{|U|}, \alpha)$作为系统主密钥私密保存，计算$\alpha G$, $\mathrm{PK}_i = h_i G(1 \le i \le |U|)$，并公开$\alpha G$以及属性公钥$\mathrm{PK}_i$。

KeyGen(MSK, ID, S_{ID})：

(1) AA为拥有属性组S_{ID}的用户ID生成相关属性私钥，$\mathrm{SK}_{i,\mathrm{ID}} = h_i + H(\mathrm{ID})\alpha, i \in S_{\mathrm{ID}}$。

(2) 每个DO都有ECDSA密钥对($\mathrm{PK}_{\mathrm{DO}}$,$\mathrm{SK}_{\mathrm{DO}}$)，CSP有ECDSA密钥对($\mathrm{PK}_{\mathrm{CSP}}$,$\mathrm{SK}_{\mathrm{CSP}}$)。两者密钥对生成方法类似，任选$\zeta \in Z_r^*$，计算$V = \zeta G$，则$\zeta$作为私钥私密保存，$V$作为公钥公开。

Encrypt($\mathrm{PK}_i, (\boldsymbol{A}, \rho), M$)：DO将文件加密上传至CSP, CSP再存储相应数据，详细步骤如下：

(1) DO任选唯一文件名fname $\in Z_r$，密钥$s \in Z_r$，计算$C = M + sG$，其中M是由明文m编码后椭圆曲线上的一个点。

(2) DO任选$v_2, v_3, \cdots, v_n \in Z_r$作为向量$\boldsymbol{v} = (s, v_2, v_3, \cdots, v_n)^{\mathrm{T}}$的参数，计算$\lambda_x = \boldsymbol{A}_x \cdot \boldsymbol{v} (x = 1, 2, \cdots, l)$，任选$u_2, u_3, \cdots, u_n \in Z_r$作为向量$\boldsymbol{u} = (0, u_2, u_3, \cdots, u_n)^{\mathrm{T}}$的参数，计算$w_x = \boldsymbol{A}_x \cdot \boldsymbol{u} (x = 1, 2, \cdots, l)$，最后输出密文$\mathrm{CT} = ((\boldsymbol{A}, \rho), C, \{C_{1,x} = \lambda_x G + w_x \mathrm{PK}_{\rho(x)}, C_{2,x} = w_x G\}_{1 \le x \le l})$。

(3) DO将密文项$C_{2,x}$散列运算后$H(C_{2,x})(1 \le x \le l)$作为叶子节点生成MHT，并得到该树根的值$R_j$。DO再用自己的私钥签名$R_j$得到$\sigma_{\mathrm{SK}_{\mathrm{DO}}}(R_j)$，接着DO将消息{ID,CT,$\sigma_{\mathrm{SK}_{\mathrm{DO}}}(R_j)$, fname, ind,$\Omega_{\mathrm{ind}}$}发送给CSP，其中ind为属性dummy在MHT叶子节点的索引，Ω_{ind}为属性dummy的辅助认证信息。

(4) CSP接收到DO发来的消息后，根据公钥$\mathrm{PK}_{\mathrm{DO}}$验证签名$\sigma_{\mathrm{SK}_{\mathrm{DO}}}(R_j)$，再用ind, Ω_{ind}还原MHT得到根值$\tilde{R}_j$，接着验证R_j是否等于$\tilde{R}_j$，若相等，则将{fname, ind, Ω_{ind},CT}存在云服务器上，否则，向DO返回0代表审核失败，服务中止。密文存入云服务器后，DO删除本地文件，只存入该文件，文件名fname，从而减少本地存储负担。

Decrypt($\mathrm{CT}, \mathrm{SK}_{\rho(x),\mathrm{ID}}$)：当用户需要相应的文件，可在云上下载文件的密文，若用户私钥的属性组满足密文中的访问架构，则可获得明文。用户从云服务器获得密文CT后，对于每个属性x，计算$C_{2,x}\mathrm{SK}_{\rho(x),\mathrm{ID}} = C_{2,x}(h_{\rho(x)} + H(\mathrm{ID})\alpha)$　，　$L = C_{1,x} - C_{2,x}\mathrm{SK}_{\rho(x),\mathrm{ID}}$。由LSSS特性可知，必有$c_x \in Z_r$使等式$\sum_{x \in S_{\mathrm{ID}}} c_x \cdot \boldsymbol{A}_x = (1, 0, \cdots, 0)$成立，再计算$\sum_{x \in S_{\mathrm{ID}}} c_x L$，计算结果为$sG$，由此我们可根据等式$C - \sum_{x \in S_{\mathrm{ID}}} c_x L$恢复出明文$M$。

DelRequest(fname, y)：若DO想要删除自己存储在云服务器上名为fname的文件，DO会向CSP发送数据删除请求。具体操作如下：

(1) DO任选随机数$a \in Z_r$计算$\theta = aG$。

(2) DO将删除标签$\mathrm{Tag}_{\mathrm{del}}$、预删除文件名fname、代表属性dummy的$y$, θ和当前时间戳T，以及对以上消息的签名$\mathrm{Sig}_{\mathrm{Del}} = \sigma_{\mathrm{SK}_{\mathrm{DO}}}(h(\mathrm{Tag}_{\mathrm{del}}||\mathrm{fname}||y||\theta||T))$}作为数据删除请求$\mathrm{DR} = \{\mathrm{Tag}_{\mathrm{del}}||\mathrm{fname}||y||\theta||T||\mathrm{Sig}_{\mathrm{Del}}\}$发送给CSP。

ReEncrypt(CT, DR)：接收到DR后，CSP通过对密文重加密，取消原有的访问权限。CSP具体操作步骤如下所示：

(1) 首先验证时间戳T是否在正常范围，再用$\mathrm{PK}_{\mathrm{DO}}$验证DR中签名是否成立，若不成立则向DO返回错误信息，并终止操作。若成立，则继续第(2)步操作。

(2) 任选$b \in Z_r$，计算$\eta = bG$, $\gamma = b\theta$。

(3) 利用索引ind提取相应密文组件$C_{2,x}$中属性dummy的密文$C_{2,\kappa}$，计算$C'_{2,\kappa} = C_{2,\kappa} + \gamma$。再根据更新后的$C'_{2,\kappa}$以及属性dummy的辅助认证信息$\Omega_{\mathrm{ind}}$生成新的MHT，并得到新根值$R'_j$。接着CSP用自己的私钥对$R'_j$签名得$\sigma_{\mathrm{SK}_{\mathrm{CSP}}}(R'_j)$，之后公布签名$\sigma_{\mathrm{SK}_{\mathrm{CSP}}}(R'_j)$及新的MHT，接着生成证据$\mathrm{proof}_j = (\mathrm{Sig}_{\mathrm{Del}}, \sigma_{\mathrm{SK}_{\mathrm{CSP}}}(R'_j), t_j)$，并将验证信息$\mathrm{Resp} = \{\mathrm{fname}, \mathrm{ind}, \Omega_{\mathrm{ind}}, \eta, R'_j, C_{2,\kappa}, \mathrm{proof}_j\}$发送给DO以供其验证删除操作的真实性。

接下来CSP将生成删除证据，并将其交付给所属机构的某个超级节点，超级节点将证据添加到证据链中。证据链的数据块包含两个部分，分别是区块头和区块体。区块头主要包含前一个区块哈希值h_{n-1}，MHT的根值R_n，时间戳t_n和版本信息Version。区块体包含了CSP产生的证据信息。证据链的具体结构如图2所示。

超级节点计算proof_j的哈希值$H(\mathrm{proof}_j)$作为叶子节点生成证据MHT，该根值为R_n。计算待处理数据块Block_n的哈希值$h_n = H(h_{n-1}||t_n||R_n)$，其中$t_n$表示当前区块产生的时间戳，$h_{n-1}$表示哈希链中前一个哈希值，并用MHT结构将这些$\mathrm{proof}_j$归纳到该处理块中，生成根节点$R_n$。该区块通过PBFT共识成功后，将被链接到证据链。

Verify(Resp, CT′)：DO通过CSP反馈的删除信息验证CSP是否按照其要求完成删除。具体步骤如下所示：

(1) 首先验证Resp中的签名是否成立，若不成立则向CSP返回错误信息并中止接下来的操作，若成立，计算$\gamma' = a\eta$。

(2) 在CSP检索已经被修改后的密文项$C'_{2,\kappa}$，用接收的$C_{2,\kappa}$计算$C^{\wedge}_{2,\kappa} = C_{2,\kappa} + \gamma'$。

(3) 检验$C^{\wedge}_{2,\kappa}$是否与云中现存的$C'_{2,\kappa}$相等，若相等，再用$C^{\wedge}_{2,\kappa}$以及Ω_{ind}算出新的MHT并得到根值

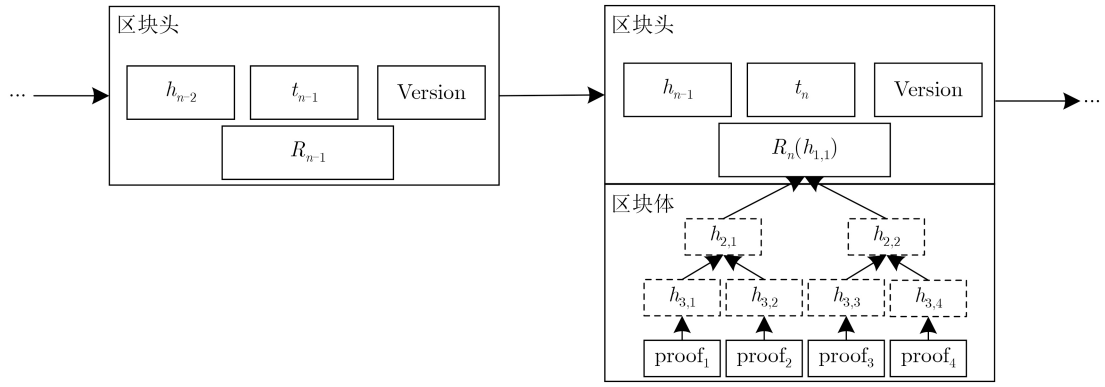


图 2 证据链结构

$\hat{R}_j$并验证是否与公布的R'_j相等。若仍然相等，则DO相信CSP已经按照自己的删除请求对相应的数据进行了删除操作。

若DO发现已要求CSP删除的数据被泄露，则可要求进行责任追踪。DO通过该数据的删除证据proof_j及证据MHT上与其相关的辅助节点计算根值$\hat{R}_u$, DO向CSP请求证据链数据，利用数据删除证据链再验证$h_u \stackrel{?}{=} H(h_{u-1}||t_u||\hat{R}_u), (u \in [n,m], m \le m)$是否成立，其中$h_m$为区块链上最新公布的哈希值。若等式成立，则说明CSP已承诺删除该数据，存在泄露行为。否则，DO未向CSP提交过该数据的删除请求，CSP无需承担数据泄露责任。

4 方案分析

4.1 安全性分析

基于选择性访问架构安全模型来证明本文所提方案的安全性。

定理 若存在PPT(Probabilistic Polynomial Time)敌手A拥有不可忽略的优势$\varepsilon > 0$攻破本文方案的选择明文安全性，则一定存在多项式时间算法B可以$\frac{\varepsilon}{2}$的优势解决ECDDH难题。

证明 挑战者$\mathcal{C}$任选阶为r的循环群P, G为循环群P的生成元，$aG, bG, Z \in P$。挑战者$\mathcal{C}$将元组$(G, aG, bG, Z$给B。接下来的过程B代替C作为挑战者。

Init: 敌手A选择一个挑战访问架构$(\boldsymbol{A}^*, \rho)$，矩阵$\boldsymbol{A}^*$的大小为$l^* \times n^*$。由访问架构加密的明文最后会被删除。

Setup: B为每个属性$1 \le i \le |U|$选择一个值$h_i^* \in Z_r$，再为每个属性计算对应的公钥$\text{PK}_i^* = h_i^* aG$。挑战者再任选$\alpha^* \in Z_r$。由于h_i^*是随机选择的，所以公共参数也是随机的。

Phase 1: 敌手A向B发送多个身份信息以及相对应的属性组$(\text{ID}^*, S_{\text{ID}}^*)$以询问私钥，B收到询问后执行相应的KeyGen算法并对相应的询问做出回答。但这些询问的S_{ID}^*不能满足$\boldsymbol{A}^*$。由于已经删除数据，因此属性dummy不包括在属性S_{ID}^*的范围内，且属性dummy相对应的密文被更改，这使得与属性dummy相关的密钥是无效的。因此，敌手对相应的密钥一无所知。

B计算$h_i^* + \alpha^*$并将其作为询问结果发送给敌手A。

Challenge: 敌手A选择两个同等长度的消息$M_0, M_1 \in P$，并将其发送给B。B根据以下步骤生成挑战密文。

B随机选$\delta \in \{0,1\}$，并任选$s^* \in Z_r$并计算$C^* = M_\delta + s^* G$。再任选参数$v_2^*, v_3^*, \cdots, v_{n^*}^* \in Z_r$，计算$\boldsymbol{v} = (s^*, v_2^*, v_3^*, \cdots, v_{n^*}^*)^{\text{T}}$，对于所有$x = 1, 2, \cdots, l^*$，计算$\lambda_x^* = \boldsymbol{A}_x^* \cdot \boldsymbol{v}$。任选参数$u_2^*, u_3^*, \cdots, u_{n^*}^* \in Z_r$，计算$\boldsymbol{u} = (0, u_2^*, u_3^*, \cdots, u_{n^*}^*)^{\text{T}}$，并对于所有$x = 1, 2, \cdots, l^*$，计算$w_x^* = \boldsymbol{A}_x^* \cdot \boldsymbol{u}$。最后B生成挑战密文$C_{1,x}^* = \lambda_x^* G + w_x^* h_{\rho(x)}^* bG$, $C_{2,x}^* = w_x^* Z$，并向敌手发送挑战密文$\text{CT}^* = ((\boldsymbol{A}^*, \rho^*), C^*, C_{1,x}^*, C_{2,x}^*)$。

Phase 2: 与phase 1类似，敌手A在不违反规则的情况下发出更多的私钥询问。

Guess: 敌手A发送一个δ的猜测δ'给B。

若$\delta' = \delta$，则B输出$\theta' = 0$代表$Z = abG$，即敌手拿到了一个有效的ECDDH元组。

若$\delta' \ne \delta$，则B输出$\theta' = 1$代表$Z = R$，即敌手拿到了一个随机元组。

显然，B生成的系统公共参数和私钥的构造方案与所提方案是相同的。

根据这个游戏规则，当$\theta = 1$时，敌手得不到任何关于δ的信息，$\Pr[\delta' \ne \delta | \theta = 1] = \Pr[\delta' = \delta | \theta = 1] = 1/2 + \varepsilon$。

由于当$\delta' \ne \delta$时，B输出$\theta' = 1$，有$\Pr[\text{B}(G, aG, bG, Z = R) = 0] - \Pr[\theta' = \theta | \theta = 1] = 1/2$。

当$\theta = 0$时，敌手A可以得到M_δ的有效密文。根据此假设，敌手的优势为ε，因此有$\Pr[\delta' \ne \delta | \theta = 0] = 1/2 + \varepsilon$。

由于当$\delta' = \delta$时，B输出$\theta' = 0$，有$\Pr[\text{B}(G, aG, bG, Z = abG) = 0] = \Pr[\theta' = \theta | \theta = 0] = 1/2 + \varepsilon$。

根据属性基加密的选择性访问架构安全模型，挑战者在此游戏的整体优势为$(\Pr[\delta'=\delta]-1/2)$，在此游戏中$\Pr[\delta'=\delta]=1/2\,[(G,aG,bG,Z=abG)=0]+1\ /\ 2[(G,aG,bG,Z=R)=0]=1/2\Pr[\theta'=\theta|\theta=0]+1/2\Pr[\theta'=\theta|\theta=1]=1/2(1/2+\varepsilon)+1/2\cdot 1/2=1/2+\varepsilon/2$。

故挑战者在此游戏的整体优势为$\Pr[\delta'=\delta]-1/2|=|1/2+\varepsilon/2-1/2|=\varepsilon/2$。　　证毕

此外，本文方案在安全特性方面与现有同类方案相比具有一定优势，具体结果如表1所示。Yang等人[14]方案与Hao等人[23]方案不采用属性加密方式，缺少细粒度访问特性，且存入云端的密文数据只能自己访问，不能分享数据，且后者方案无法提供隐私保护。Xue等人[18]方案、Yu等人[15]方案与本文方案都使用属性加密，均可提供细粒度访问控制，但是前两个方案不能进行公开验证及责任追踪，且基于双线性构造方案开销较大，综上所述，本方案在性能评估中有较好的优势。

4.2 性能分析

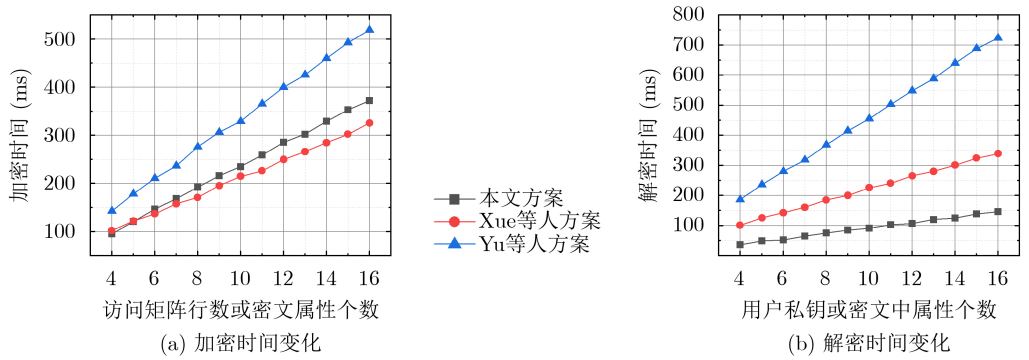
本节就时间复杂度将本文方案与现有同类方案进行分析对比，由于Setup, KeyGen等阶段由具有充足计算资源的属性中心(AA)执行的离线一次性操作，对系统运行性能影响较小，所以本节主要考虑执行较为频繁的加密、解密、删除、验证等4个阶段的计算开销，具体结果如表2所示。表2中的加密对应前文算法Encrypt，解密对应算法Decrypt，删除对应算法DelRequest和ReEncrypt，验证对应算法Verify，其中$T_{\text{p_mul}}$, $T_{\text{p\_add}}$, T_{bp}, T_{exp}, T_{mul}, T_{sig}, T_{ver}, T_{ε}, T_D, T_h分别表示单次椭圆曲线倍点运算，椭圆曲线点加运算，双线性映射运算，幂运算，乘法运算，ECDSA签名运算及ECDSA验签运算，AES加密运算及解密运算，哈希运算等运算时间。为便于描述，l代表共享生成矩阵$\boldsymbol{A}$的行数，$|\gamma|$为密文中属性个数，M_a表示满足访问策略的最少属性个数，m为当前区块链节点个数。由于Xue等人[18]方案、Yu等人[15]方案与本文方案均基于属性加密，故本节仿真实验只针对后3个方案比较，对比结果如图3(a)，图3(b)，图4(a)与图4(b)所示，分别表示加密阶段、解密阶段、删除阶段与验证阶段。本实验在Intel(R) Core(TM) i7-6700 CPU@ 3.40 GHz平台上使用JPBC库实现。椭圆曲线加密的密钥大小为160 bit, AES加密密钥大小为128 bit，安全参数设置为80，图3(a)为设置不同的访问矩阵行数或密文属性个数的加密时间对比，图3(b)为设置不同的用户私钥或密文中属性个数的解密时间对比，图4(a)为设置不同的访问矩阵行数或密文中属性个数的删除时间对比，图4(b)为设置不同用户私钥或密文中属性个数的验证时间对比。图中时间为重复50次得到的平均值。本文方案和Yu等人[15]方案是基于CP-ABE的，而Xue等人[18]方案是基于KP-ABE的。因此在加解密阶段和删除验证阶段的时间消耗受访问矩阵行数、私钥中属性个数以及密文中属性个数等不同类型参数影响，本文实验中将这些参数值都设置4～16的相同值。

表 1　安全特性对比

方案	Yang等人[14]方案	Hao等人[23]方案	Xue等人[18]方案	Yu等人[15]方案	本文方案
可公开验证	是	是	否	否	是
云数据共享	否	否	是	是	是
细粒度访问	否	否	是	是	是
责任可追踪	是	是	否	否	是
隐私保护	是	否	是	是	是

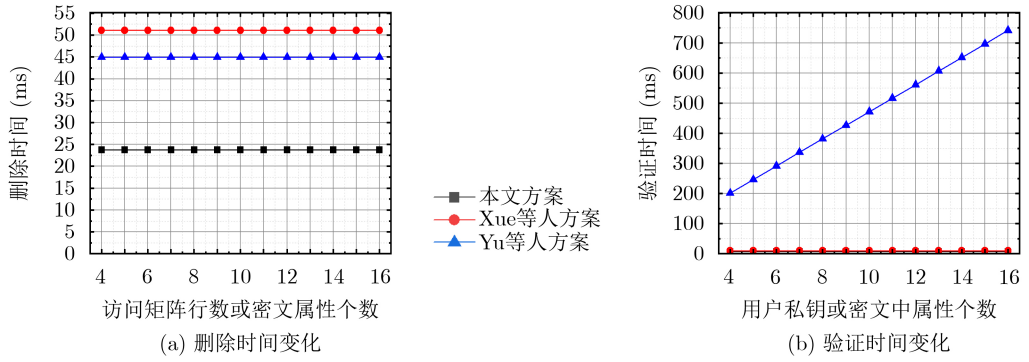
表 2　时间复杂度对比

方案	Yang等人[14]方案	Hao等人[23]方案	Xue等人[18]方案	Yu等人[15]方案	本文方案
加密	$T_{\varepsilon}+2T_h$	$T_{\text{p\_mul}}+2T_{\varepsilon}+4T_h$	$(2+2\|\gamma\|)T_{\text{exp}}+T_{\text{mul}}$ $+T_{\text{sig}}+2\|\gamma\|T_h$	$T_{\text{bp}}+3lT_{\text{exp}}+(l+1)T_{\text{mul}}$	$(5l+1)T_{\text{p\_mul}}+(l+1)T_{\text{p\_add}}$ $+T_{\text{sig}}+(2l-1)T_h$
解密	$T_{\text{sig}}+T_{\text{ver}}+T_D+3T_h$	$T_{\varepsilon}+T_D+3T_h$	$(T_{\text{bp}}+T_{\text{mul}})\|\gamma\|$ $+T_{\text{bp}}+2T_{\text{mul}}$	$(2T_{\text{bp}}+1T_{\text{exp}}+1T_{\text{mul}})M_a$ $+2T_{\text{mul}}$	$(2T_{\text{p\_mul}}+2T_{\text{p\_add}})M_a$ $+T_{\text{p\_add}}$
删除	$T_{\text{sig}}+T_{\text{ver}}$	T_{sig}	$2T_{\text{bp}}+T_{\text{exp}}+T_{\text{mul}}+T_{\text{sig}}$ $+T_{\text{ver}}+(\log_2\|\gamma\|+1)T_h$	$4T_{\text{exp}}+T_{\text{sig}}+2T_{\text{ver}}+T_h$	$3T_{\text{p\_mul}}+T_{\text{p\_add}}+2T_{\text{sig}}$ $+T_{\text{ver}}+(\log_2 l+3)T_h$
验证	$\left(\log_2 m+\frac{m+1}{2}+1\right)T_h$	T_{ver}	$T_{\text{exp}}+(\log_2\|\gamma\|+1)T_h$	$(2T_{\text{bp}}+1T_{\text{exp}}+1T_{\text{mul}})M_a$ $+1T_{\text{exp}}+2T_{\text{mul}}+T_{\text{ver}}+T_h$	$T_{\text{p\_mul}}+T_{\text{p\_add}}+T_{\text{ver}}$



(a) 加密时间变化 (b) 解密时间变化

图 3 不同访问矩阵行数或密文中属性个数下的加解密时间变化



(a) 删除时间变化 (b) 验证时间变化

图 4 不同访问矩阵行数或密文中属性个数下的删除与验证时间变化

5 结束语

为了解决云存储环境中数据可信删除问题，本文提出一种基于区块链的云数据安全存储与删除方案。本文方案不仅比同类方案更为轻量化，还实现了存储数据的细粒度访问控制，同时基于区块链的删除证据管理方式使其具有公开可验证特性。最后，对本文方案进行了安全性分析和实验分析，结果表明所提方案能更好地满足云数据安全共享与删除需求。

参 考 文 献

[1] LI Yannan, YU Yong, MIN Geyong, *et al*. Fuzzy identity-based data integrity auditing for reliable cloud storage systems[J]. *IEEE Transactions on Dependable and Secure Computing*, 2017, 16(1): 72–83. doi: 10.1109/TDSC.2017.2662216.

[2] 张玉磊, 刘祥震, 郎晓丽, 等. 云存储环境下多服务器的密钥聚合可搜索加密方案[J]. 电子与信息学报, 2019, 41(3): 674–679. doi: 10.11999/JEIT180418.

ZHANG Yulei, LIU Xiangzhen, LANG Xiaoli, *et al*. Multi-server Key Aggregation Searchable Encryption Scheme in Cloud Environment[J]. *Journal of Electronics & Information Technology*, 2019, 41(3): 674–679. doi: 10.11999/JEIT180418.

[3] ZHANG Zhiwei, TAN Shichong, WANG Jianfeng, *et al*. An associated deletion scheme for multi-copy in cloud storage[C]. International Conference on Algorithms and Architectures for Parallel Processing. Guangzhou, China, 2018: 511-526. doi: 10.1007/978-3-030-05063-4_38.

[4] HUANG Hui, CHEN Xiaofeng, WU Qianhong, *et al*. Bitcoin-based fair payments for outsourcing computations of fog devices[J]. *Future Generation Computer Systems*, 2018, 78: 850–858. doi: 10.1016/j.future.2016.12.016.

[5] 赵志远, 朱智强, 王建华, 等. 云存储环境下无密钥托管可撤销属性基加密方案研究[J]. 电子与信息学报, 2018, 40(1): 1–10. doi: 10.11999/JEIT170317.

ZHAO Zhiyuan, ZHU Zhiqiang, WANG Jianhua, *et al*. Revocable Attribute-based Encryption with Escrow-free in Cloud Storage[J]. *Journal of Electronics & Information Technology*, 2018, 40(1): 1–10. doi: 10.11999/JEIT170317.

[6] 张曙光, 咸鹤群, 王利明, 等. 云加密数据安全重复删除方法[J]. 软件学报, 2019, 30(12): 3815–3828. doi: 10.13328/j.cnki.jos.005610.

ZHANG Shuguang, XIAN Hequn, WANG Liming, *et al*. Secure Cloud Encrypted Data Deduplication Method[J]. *Journal of Software*, 2019, 30(12): 3815–3828. doi: 10.13328/j.cnki.jos.005610.

[7] GARFINKEL S L and SHELAT A. Remembrance of data passed: A study of disk sanitization practices[J]. *IEEE Security & Privacy*, 2003, 1(1): 17–27. doi: 10.1109/MSECP.2003.1176992.

[8] GUTMANN P. Secure deletion of data from magnetic and

solid-state memory[C]. The Sixth USENIX Security Symposium, San Jose, USA. 1996, 14: 77-89.

[9] PAUL M and SAXENA A. Proof of erasability for ensuring comprehensive data deletion in cloud computing[C]. International Conference on Network Security and Applications. Berlin, Germany, 2010: 340-348. doi: 10.1007/978-3-642-14478-3_35.

[10] PERITO D and TSUDIK G. Secure code update for embedded devices via proofs of secure erasure[C]. European Symposium on Research in Computer Security. Berlin, Germany, 2010: 643-662. doi: 10.1007/978-3-642-15497-3_39.

[11] HU Pengfei, NING Huansheng, QIU Tie, *et al*. Fog computing based face identification and resolution scheme in internet of things[J]. *IEEE Transactions on Industrial Informatics*, 2016, 13(4): 1910–1920. doi: 10.1109/TII.2016.2607178.

[12] PERLMAN R. File system design with assured delete[C]. The 3rd IEEE International, Security in Storage Workshop. San Francisco, USA, 2005. doi: 10.1109/SISW.2005.5.

[13] 张曙光, 咸鹤群, 王利明, 等. 无可信第三方的加密重复数据安全删除方法[J]. 密码学报, 2018, 5(3): 286-296.
ZHANG Shuguang, XIAN Hequn, WANG Liming, *et al*. Security deduplication method of encrypted data without ant additional server[J]. Journal of Cryptologic Research, 2018, 5(3): 286-296.

[14] YANG Changsong, CHEN Xiaofeng, and XIANG Yang. Blockchain-based publicly verifiable data deletion scheme for cloud storage[J]. *Journal of Network and Computer Applications*, 2018, 103: 185–193. doi: 10.1016/j.jnca.2017.11.011.

[15] YU Yong, XUE Liang, LI Yannan, *et al*. Assured data deletion with fine-grained access control for fog-based industrial applications[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(10): 4538–4547. doi: 10.1109/TII.2018.2841047.

[16] MO Zhen, XIAO Qingjun, ZHOU Yian, *et al*. On deletion of outsourced data in cloud computing[C]. 2014 IEEE 7th International Conference on Cloud Computing. Anchorage, USA, 2014: 344-351. doi: 10.1109/CLOUD.2014.54.

[17] REARDON J, RITZDORF H, BASIN D, *et al*. Secure data deletion from persistent media[C]. The 2013 ACM SIGSAC conference on Computer & communications security. Berlin, Germany, 2013: 271-284. doi: 10.1007/978-3-319-28778-2_9.

[18] XUE Liang, YU Yong, LI Yannan, *et al*. Efficient attribute-based encryption with attribute revocation for assured data deletion[J]. *Information Sciences*, 2019, 479: 640–650. doi: 10.1016/j.ins.2018.02.015.

[19] BEIMEL A. Secure Schemes for Secret Sharing and Key Distribution[M]. Haifa, International Journal of Pure & Applied Mathematics, 1996.

[20] SAHAI A and WATERS B. Fuzzy identity-based encryption[C]. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Germany, 2005: 457-473. doi: 10.1007/11426639_27.

[21] 刘忆宁, 周元健, 蓝如师, 等. 基于区块链的云数据删除验证协议[J]. 计算机研究与发展, 2018, 55(10): 2199–2207. doi: 10.7544/issn1000-1239.2018.2018.20180436.
LIU Yining, ZHOU Yuanjian, LAN Rushi, *et al*. Blockchain-Based Verification Scheme for Deletion Operation in Cloud[J]. *Journal of Computer Research and Development*, 2018, 55(10): 2199–2207. doi: 10.7544/issn1000-1239.2018.2018.20180436.

[22] RAMOKAPANE K M, RASHID A, and SUCH J M. Assured deletion in the cloud: requirements, challenges and future directions[C]. The 2016 ACM on Cloud Computing Security Workshop. New York, USA, 2016: 97-108. doi: 10.1145/2996429.2996434.

[23] HAO Feng, CLARKE D, and ZORZO A F. Deleting secret data with public verifiability[J]. *IEEE Transactions on Dependable and Secure Computing*, 2016, 13(6): 617–629. doi: 10.1109/TDSC.2015.2423684.

周由胜：男，1979年生，博士，副教授，研究方向为密码学与信息安全.

陈律君：女，1995年生，硕士生，研究方向为密码学与信息安全.

责任编辑：马秀强