计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190600079

用户身份可追踪的云共享数据完整性审计方案

张 茜 王 箭
南京航空航天大学计算机科学与技术学院 南京 211106
(zhangxi105@nuaa.edu.cn)

摘 要 云端共享数据完整性审计用于验证一个用户群组共享在云端的数据的完整性。与单用户的数据完整性审计相比,群组共享数据的完整性审计需要考虑用户撤销、身份隐私保护等问题。如果数据出现争议或其他情况,还需要对数据的来源进行追踪,目前已有的云共享数据完整性审计方案尚未能很好地处理这个问题。为了实现数据源的追踪,并保证高效的用户撤销和用户身份隐私的保护,文中提出基于群签名算法的云共享数据完整性审计方案。当需要追踪数据块签名者的身份时,群管理员可利用自己的私钥对数据块签名者的身份进行追踪,且他人无法得知该签名者的身份。该方案中的私钥更新机制能很好地支持用户撤销,极大缩减了用户撤销过程中的计算和通信开销。安全性分析和实验结果表明,该方案是安全、高效的。

关键词:云存储;共享数据;完整性审计;群签名;可追踪性
中图法分类号 TP309

Public Integrity Auditing for Shared Data in Cloud Supporting User Identity Tracking

ZHANG Xi and WANG Jian
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Abstract Public integrity auditing for shared data in the cloud is used to verify the integrity of data shared by a group of users. Compared with the integrity auditing forsingle-user data, the integrity auditing for shared data of a group needs to consider more issues, such as efficient user revocation, identity privacy protection and so on. If there is a dispute or other situation in the data, the source of the data needs to be tracked, and existing integrity auditing schemes for shared cloud data have not yet handled this problem well. In order to track the source of data and ensure efficient user revocation and the protection of user's identity privacy, an integrity auditing scheme based on group signature algorithm for shared cloud data is proposed. When it is necessary to track the identity of the signer of a data block, the group manager can track it by using his/her private key and others cannot know the identity of this signer. The private key update mechanism in this scheme can well support user revocation, and greatly reduce the computation and communication overhead during the user revocation process. Safety analysis and experimental results show that the scheme is safe and efficient.

Keywords Cloud storage, Shared data, Integrity auditing, Group signature, Traceability

1 引言

数据共享是云存储中一种广泛的应用形式,通过云端数据共享服务,用户可以将自己的数据共享在一个群组范围内,具体地说,在一个工作组内,当一个用户将自己的数据上传至云端后,群组内的其他用户都可以访问并修改这些数据,这不仅实现了信息共享,还减小了用户本地数据存储的负担。尽管云服务商承诺为用户提供一个安全可靠的环境,但由于软硬件故障或人为因素的存在,数据可能会丢失或损毁,同时,云服务商为避免因自己的声誉被损害而造成经济损失,可能会向用户隐瞒真实情况,甚至为了节省空间、获得更高效益而删除用户不常使用的数据[1]。

为验证云端数据的完整性,一些针对云端数据的完整性审计方案被提出[2-18]。在这些方案中,数据的完整性依赖于与其相关的认证器的正确性,且存在一个第三方验证者 TPA 无需下载全部数据即可检验存储在云端的数据的完整性,但是,文献[2-5]只适用于云端个人数据的完整性审计。在实际应用中,多用户间共享数据存储是一种非常重要的应用形式,与云端个人数据的完整性验证相比,云端共享数据的完整性验证会带来诸多方面的挑战,如用户的身份隐私保护、群组用户的加入与撤销等。由于共享数据块的认证器是由群组中不同的用户生成的,审计者可以根据用户处理的数据块数量或数据块被修改的频率,来推导出哪个成员更重要或哪个数据块更有价值。为保护用户的身份隐私,Wang 等首次提出了针对共享数据完整性审计的方案[6],该方案基于环签名来构造数据块认证器,验证者在验证时使用群组公钥进行验证,保护了用户的身份隐私。考虑到安全因素,当一个用户离开群组时,其应当从组内撤销,且无法再访问云端共享数据或上传

到稿日期:2019-06-17 返修日期:2019-09-24 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
通信作者:王箭(wangjian@nuaa.edu.cn)

数据至云端。为此，Wang 等[7]提出了支持用户撤销的共享数据完整性审计方案，文献[8-9]中的方案在效率和计算开销方面进行了进一步的优化，但在用户撤销阶段的计算开销和通信开销仍然与来自被撤销用户的数据块总数呈线性关系，计算开销较大，而文献[11]引入新的密钥更新机制，很好地解决了这一问题。

但是，完全的身份隐私[19-20]可能会引发新的问题：共享群组内的某个成员可以篡改数据而无需担心被发现。在实际应用中，公司雇员为了利益故意修改某些重要数据的情况屡见不鲜，即群组用户可能会出于个人利益而篡改相关数据，这可能会导致共享用户因数据不一致而产生争议，甚至会造成公司的经济损失。因此，既要实现用户的身份隐私保护，又要能够追踪恶意修改数据的成员用户身份对完善共享数据的完整性审计方法十分重要。为实现数据源的追踪以及用户身份隐私的保护，文献[12]提出了可以对数据来源进行追踪的共享数据完整性审计方案，但在该方案中，当有用户从群组撤销时，其余用户需要对所有数据块进行重签名，通信与计算开销较大，且存在替代攻击的风险。

本文提出了基于群签名的用户身份可追踪的云共享数据完整性审计方案。具体来说，本文的贡献归纳如下。

(1)基于 BBS 群签名算法[13]，本文构造了一个支持群组管理员追踪数据块签名者身份的云端共享数据完整性审计方案。群组管理员可利用自己的私钥对数据块签名者的身份进行追踪，同时保证他人无法得知用户的身份隐私。

(2)该方案提出了一个私钥更新机制，当群组中有用户撤销时，未撤销用户的部分私钥会被更新从而支持之后的存储审计任务。此外，被撤销用户无法继续访问或上传云端数据，该用户被撤销之前所生成的认证器也无需重新计算，因此用户撤销的开销被大大削减。

(3)本文给出了方案的安全性分析和实验结果分析，结果证明所提方案是安全且高效的。

本文第 2 节提出系统模型和设计目标；第 3 节介绍本方案的算法定义和相关的密码学知识；第 4 节详细描述提出的方案；第 5 节和第 6 节分别给出本方案的安全性分析和性能评估；最后总结全文。

2 系统模型和设计目标

2.1 系统模型

如图 1 所示，本方案系统模型包括 4 种实体：群组用户、群组管理员、云服务器和可信第三方审计者 TPA。

(1)群组用户。群组包含多名用户，群组成员可以加入或退出用户组，用户之间通过云存储共享数据。合法用户是诚实的，且不会向他人泄漏任何隐私信息。

(2)群组管理员。群组管理员负责群组用户的加入和撤销，被撤销的用户无法上传或访问云端数据。

(3)云服务器。云服务器为群组用户提供大量的存储空间和计算资源，并提供数据存储和数据共享服务。云服务器是半可信的，它不会恶意破坏云端数据，但会向用户隐瞒数据被破坏的事实，从而避免其名誉及经济损失。

(4)可信第三方审计者 TPA。TPA 可以代替群组用户对云端数据进行完整性审计。当 TPA 需要审计数据的完整性时，先向云服务器发送审计挑战，云服务器根据审计挑战生成相应的可证明数据完整性的证明信息，并将该信息返回给 TPA，TPA 验证该证明信息是否正确，如果正确则说明云服务器完整地存储了用户数据，否则说明数据被损坏。

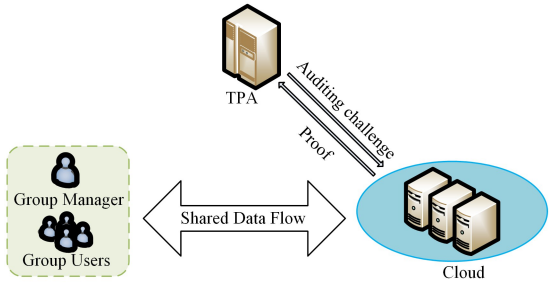

图 1 系统模型

Fig. 1 System model

2.2 设计目标

为了实现支持用户身份追踪的云端共享数据完整性审计，本方案应满足以下设计目标。

(1)正确性。如果云服务器、群组成员和 TPA 是诚实的且遵循指定步骤，那么 TPA 能够正确检测出审计数据的完整性。

(2)高效性。在审计过程中，TPA 无需下载云端的全部数据，即可对云端群组数据的完整性进行审计。

(3)审计可靠性。云服务器只有存储完整的群组数据，才能通过 TPA 的审计。

(4)身份隐私。在数据完整性审计过程中，TPA 无法获取每个数据块认证器所对应的用户身份信息。

(5)可追踪性。当数据出现争议时，群组管理员可以对群组用户的身份进行追踪。

3 算法定义与预备知识

3.1 算法定义

一个支持用户身份追踪的云端共享数据完整性审计方案包括以下 7 种算法。

(1)$KeyGen(1^k)\rightarrow(gpk,gsk,gmsk)$：密钥生成算法。由群组管理员执行，输入安全参数 k，输出群组密钥(gpk,gsk)和管理员私钥 $gmsk$。

(2)$Join(i)\rightarrow gusk[i]$：用户加入算法。由群组管理员执行，输入一个新用户 i，输出相应的用户私钥 $gusk[i]$。

(3)$AuthGen(gsk,gusk,m)\rightarrow\sigma$：认证器生成算法。由群组用户执行，输入用户私钥 $gusk[i]$、群组私钥 gsk 和数据块 m，输出认证器 σ。

(4)$ProofGen(F,\{\sigma_k\}_{k=1,\cdots,n},chal)\rightarrow P$：证明生成算法。由云服务器执行，输入共享文件 F、认证器集合$\{\sigma_k\}_{k=1,\cdots,n}$和审计质询 $chal$，输出能够证明云服务器拥有完整共享数据的审计证明 P。

(5)$ProofVerify(gpk,chal,P)\rightarrow\{True,False\}$：证明验证算法：由 TPA 执行，输入群公钥 gpk、审计质询 $chal$ 和数据持有证明 P，TPA 验证该证明有效则输出 $True$，否则输出 $False$。

(6)$Open(m_k,\sigma_k,gmsk)\rightarrow i$：身份追踪算法。由群管理员执行，输入数据块 m_k 和相应的认证器 σ_k、群管理员私钥 $gm-$

sk,输出签名者身份 i。

(7)$Rvoke(i,UR)\rightarrow gsk_{new}$:用户撤销算法。由群管理员执行,输入被撤销用户 i、当前用户撤销数 UR,输出新的群私钥 gsk_{new},并由群管理员向未撤销的群组用户发布。

3.2　预备知识

(1)双线性映射

设 G_1,G_2 为两个阶为大素数 p 的乘法循环群。定义双线性映射为 $\hat{e}:G_1\times G_1\rightarrow G_2$,且其满足以下性质。

1)可计算性:对于任意的 $P,Q\in G_1$,存在有效算法可以轻易计算 $\hat{e}(P,Q)^{ab}$;

2)双线性:对于任意的 $P,Q\in G_1$,$a,b\in Z_p^*$,满足 $\hat{e}(P^a,Q^b)=\hat{e}(P,Q)^{ab}$;

3)非退化性:存在 $P,Q\in G_1$,使 $\hat{e}(P,Q)\neq 1$。

(2)计算 Diffie-Hellman 问题

设 G_1 是阶为大素数 p 的循环乘法群,给定 $g,g^a,h\in G_1$,其中 $a\in Z_p^*$,计算 $h^a\in G_1$。

(3)离散对数问题

设 G_1 是阶为大素数 p 的循环乘法群,给定 $g,g^x\in G_1$,计算 $x\in Z_p^*$。

4　提出的方案

4.1　符号说明

设 G_1,G_2 为大素数 p 阶循环乘法群,g 为 G_1 的生成元,$\hat{e}:G_1\times G_1\rightarrow G_2$ 为双线性映射。存在 3 个安全散列函数 H_1,$H_2:\{0,1\}^*\rightarrow Z_p$,$h:\{0,1\}^*\rightarrow G_1$,$ID$ 表示群组标识,用户总数为 d。假设要上传的文件 F 被划分成 n 个数据块 $m_k\in Z_p^*$,每个块标识为 id_k,$k\in[1,n]$。

4.2　方案描述

(1)算法 $KeyGen(1^k)\rightarrow(gpk,gsk,gmsk)$

群管理员随机挑选 $h\in G_1$ 和 $\eta,\gamma,\lambda_1,\lambda_2\in Z_p^*$,令 $w=g^{\gamma}\in G_1$,取 $u,v\in G_1$,使得 $u^{\lambda_1}=v^{\lambda_2}=h$。管理员私钥为 $gmsk=(\lambda_1,\lambda_2)$,并秘密保存 γ,设置用户撤销数 $UR=0$,并随机挑选群私钥 $gsk=(\pi_{UR},\tau_{UR})$,$\pi_{UR},\tau_{UR}\in Z_p^*$,通过安全信道发送给群成员,计算公开参数 $\Omega_{UR}=g^{\pi_{UR}}$,$\mathcal{R}_{UR}=g^{\tau_{UR}}$ 和 $a=h(ID)$,群公钥 $gpk=(g,h,u,v,w,\eta,\Omega_{UR},R_{UR},a)$。

(2)算法 $Join(i)\rightarrow gusk[i]$

对于用户 i,$1\leqslant i\leqslant d$,群管理员随机挑选 $x_i\in Z_p^*$,且 $x_i+\gamma\neq 0$,令 $A_i=g_1^{1/(\gamma+x_i)}\in G_1$,则用户 i 的私钥为 $gusk[i]=(A_i,x_i)$,管理员将私钥 $gusk[i]$ 安全传输给用户 i,并把该用户加入群成员列表。

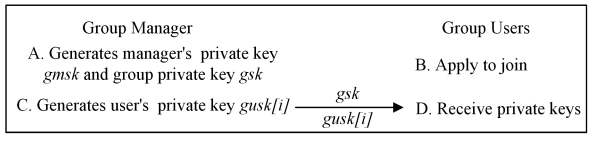

图 2　密钥生成过程

Fig. 2　Process of private key generation

(3)算法 $AuthGen(gsk,gusk,m)\rightarrow\sigma$

已知群公钥 $gpk=(g,h,u,v,w,\eta,\Omega_{UR},\mathcal{R}_{UR},a)$ 和用户 i 的私钥 $gusk[i]=(A_i,x_i)$,对于数据块 $m_k\in Z_p^*$ $(k\in[1,n])$,用户 i 计算相应的认证器 σ_k。该用户执行以下操作。

1)随机挑选 $\alpha_k,\beta_k,r_{k,\alpha_k},r_{k,\beta_k},r_{k,x_i},r_{k,\delta_{k,1}},r_{k,\delta_{k,2}}\in Z_p$。

2)计算 $T_{k,1}$,$T_{k,2}$ 和 $T_{k,3}$:

$T_{k,1}=u^{\alpha_k}$,$T_{k,2}=v^{\beta_k}$,$T_{k,3}=A_ih^{\alpha_k+\beta_k}$。

3)计算 $\delta_{k,1}=x_i\alpha_k$,$\delta_{k,2}=x_i\beta_k$。

4)计算 $R_{k,1}$,$R_{k,2}$,$R_{k,3}$,$R_{k,4}$ 和 $R_{k,5}$:

$R_{k,1}=u^{r_{k,\alpha_k}}$,$R_{k,2}=v^{r_{k,\beta_k}}$,

$R_{k,3}=e(T_{k,3},g)^{r_{k,x_i}}e(h,w)^{-r_{k,\alpha_k}-r_{k,\beta_k}}e(h,g)^{-r_{k,\delta_{k,1}}-r_{k,\delta_{k,2}}}$,

$R_{k,4}=T_{k,1}^{r_{k,x_i}}\cdot u^{-r_{k,\delta_{k,1}}}$,$R_{k,5}=T_{k,2}^{r_{k,x_i}}\cdot v^{-r_{k,\delta_{k,2}}}$。

5)计算签名 $\theta_k=a^{\pi_{UR}H_2(id_k)+m_k\pi_{UR}\tau_{UR}}$。

6)计算数据块 m_k 的挑战值 c_k:

$c_k=\eta^{m_k}\cdot H_1(id_k,\theta_k,T_{k,1},T_{k,2},T_{k,3},R_{k,1},R_{k,2},R_{k,3},R_{k,4},R_{k,5})$。

7)计算 s_{k,α_k},s_{k,β_k},s_{k,x_i},$s_{k,\delta_{k,1}}$ 和 $s_{k,\delta_{k,2}}$:

$s_{k,\alpha_k}=r_{k,\alpha_k}+c_k\alpha_k$,$s_{k,\beta_k}=r_{k,\beta_k}+c_k\beta_k$,$s_{k,x_i}=r_{k,x_i}+c_kx_i$,$s_{k,\delta_{k,1}}=r_{k,\delta_{k,1}}+c_k\delta_{k,1}$,$s_{k,\delta_{k,2}}=r_{k,\delta_{k,2}}+c_k\delta_{k,2}$。

8)计算身份标签 $B=a^{\tau_{UR}}$。

9)输出认证器:

$\sigma_k=(T_{k,1},T_{k,2},T_{k,3},c_k,R_{k,3},\theta_k,s_{k,\alpha_k},s_{k,\beta_k},s_{k,x},s_{k,\delta_{k,1}},s_{k,\delta_{k,2}},UR,B)$,上传数据块 m_k 和 σ_k 认证器到云端服务器,并删除本地数据。

云服务器通过检查 $e(\mathcal{R}_{UR},a)=e(g,B)$ 来验证身份标签 B,若身份标签验证正确则允许该用户访问,否则视该用户为非法用户并拒绝其请求。

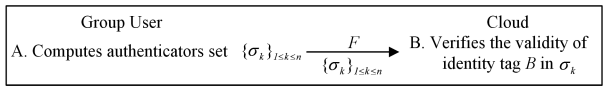

图 3　认证器生成过程

Fig. 3　Process of authenticator generation

(4)算法 $ProofGen(F,\{\sigma_k\}_{k=1,\cdots,n},chal)\rightarrow P$

为审计云端共享数据的完整性,TPA 接到用户审计请求后,执行以下步骤:

1)随机挑选 c 个元素组成集合 J,其中 $J\subset[1,n]$;

2)对于每个 $j\in J$,生成一个随机值 $y_j\in Z_p^*$;

3)发送审计挑战 $chal=\{j,y_j\}_{j\in J}$ 到云端服务器。

云服务器接到审计挑战 $chal$ 后生成数据持有证明 P:

1)计算 $\hat{c}=\prod_{j\in J}c_j^{y_j}$,$\hat{R}_3=\prod_{j\in J}R_{j,3}^{y_j}$,$\hat{\theta}=\prod_{j\in J}\theta_j^{y_j}$ 和 $\hat{m}=\sum_{j\in J}y_jm_j$;

2)发送 $P=\{\hat{c},\hat{R}_3,\hat{\theta},\hat{m},\{\sigma_j\}_{j\in J},\{id_j\}_{j\in J}\}$ 给 TPA。

(5)算法 $ProofVerify(gpk,chal,P)\rightarrow\{True,False\}$

已知审计证明 $P=\{\hat{c},\hat{R}_3,\hat{\theta},\hat{m},\{\sigma_j\}_{j\in J},\{id_j\}_{j\in J}\}$,审计挑战 $chal=\{j,y_j\}_{j\in J}$,TPA 根据 UR 值对数据块分组进行审计,执行以下步骤来验证证明的正确性。

1)计算 $\tilde{R}_{j,1}$,$\tilde{R}_{j,2}$,$\tilde{R}_{j,4}$ 和 $\tilde{R}_{j,5}$

$\tilde{R}_{j,1}=u^{s_{j,\alpha_j}}/T_{j,1}^{c_j}$,$\tilde{R}_{j,2}=v^{s_{j,\beta_j}}/T_{j,2}^{c_j}$,$\tilde{R}_{j,4}=T_{j,1}^{s_{j,x_i}}/u^{s_{j,\delta_{j,1}}}$,$\tilde{R}_{j,5}=T_{j,2}^{s_{j,x_i}}/v^{s_{j,\delta_{j,2}}}$

2)验证

$$\hat{R}_3\overset{?}{=}e\left(\prod_{j\in J}\left(T_{j,3}^{s_{j,x_i}}\cdot h^{-s_{j,\delta_{j,1}}-s_{j,\delta_{j,2}}}\cdot g^{-c_j}\right)^{y_j},g\right)\cdot e\left(\prod_{j\in J}\left(h^{-s_{j,\alpha_j}-s_{j,\beta_j}}\cdot T_{j,3}^{c_j}\right)^{y_j},w\right)\tag{1}$$

$$\hat{c} \overset{?}{=} \overset{\wedge}{\eta}^{m} \cdot \prod_{j\in J} H_1(id_j, \theta_j, T_{j,1}, T_{j,2}, T_{j,3}, \widetilde{R}_{j,1}, \widetilde{R}_{j,2}, R_{j,3}, \widetilde{R}_{j,4}, \widetilde{R}_{j,5})^{y_j} \tag{2}$$

$$e(g, \overset{\wedge}{\theta}) \overset{?}{=} e(\Omega_{UR}, a^{\sum_{j\in J} y_j H_2(id_j)} \cdot B^{\hat{m}}) \tag{3}$$

若以上等式成立，则说明数据持有证明 P 有效。

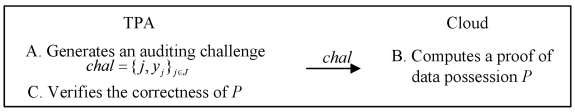

图 4 审计过程

Fig. 4 Process of auditing

(6)算法 $Open(m_k, \sigma_k, gmsk) \rightarrow i$

给出数据块 m_k 和认证器 σ_k，群组管理员利用自己的私钥 $gmsk=(\lambda_1, \lambda_2)$ 可以揭示该数据块签名者的身份：

1)验证认证器 σ_k 是 m_k 的合法签名；

2)解密用户 i 的 $A_i = T_{k,3}/(T_{k,1}^{\lambda_1} \cdot T_{k,2}^{\lambda_2})$；

3)A_i 为用户 i 私钥的一部分，群组管理员即可揭示数据块 m_k 所对应的签名者身份。

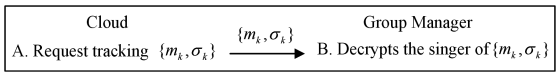

图 5 追踪过程

Fig. 5 Process of tracing

(7)算法 $Revoke(i, UR) \rightarrow gsk_{new}$

用户撤销时，管理员设置 $UR=UR+1$，更新群私钥 $gsk(\pi_{UR}, \tau_{UR})$，并将其通过安全信道发送给群成员，计算公开参数 $\Omega_{UR}=g^{\pi_{UR}}$，$\mathcal{R}_{UR}=g^{\tau_{UR}}$ 并发布，TPA 维系一个 $\{UR, \Omega_{UR}, \mathcal{R}_{UR}\}$ 记录表。

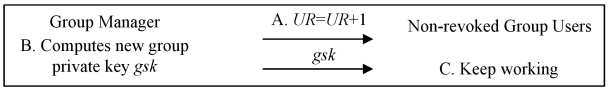

图 6 用户撤销过程

Fig. 6 Process of user revocation

5 安全分析

定理 1(正确性) 如果云服务器、群组成员和 TPA 是诚实的且遵循指定步骤，则 TPA 可以正确检测出共享数据的完整性。

证明：根据双线性映射的性质，验证等式的正确性可以通过以下步骤来证明。

式(1)的验证过程如下：

因为

$$left = \prod_{j\in J}[e(T_{j,3}, g)^{r_{j,x_i}} \cdot e(h, w)^{-r_{j,\alpha_j}-r_{j,\beta_j}} \cdot e(h, g)^{-r_{j,\delta_{j,1}}-r_{j,\delta_{j,2}}}]^{y_j}$$
$$= e[\prod_{j\in J}(T_{j,3}^{r_{j,x_i}} \cdot h^{-r_{j,\delta_{j,1}}-r_{j,\delta_{j,2}}})^{y_j}, g] \cdot e[\prod_{j\in J}(h^{-r_{j,\alpha_j}-r_{j,\beta_j}})^{y_j}, w]$$

其中

$$e[\prod_{j\in J}(T_{j,3}^{r_{j,x_i}} h^{-c_j(\delta_{j,1}+\delta_{j,2})} \cdot g^{-c_j})^{y_j}, g] \cdot e[\prod_{j\in J}(h^{-c_j(\alpha_j+\beta_j)} \cdot T_{j,3}^{c_j})^{y_j}, w]$$
$$= e[\prod_{j\in J}(T_{j,3}^{r_{j,x_i}} h^{-c_j(\delta_{j,1}+\delta_{j,2})} \cdot g^{-c_j})^{y_j}, g] \cdot e[\prod_{j\in J}(h^{-c_j(\alpha_j+\beta_j)} \cdot T_{j,3}^{c_j})^{y_j}, g^{\gamma}]$$
$$= e[\prod_{j\in J}(T_{j,3}^{c_j(x_i+\gamma)} h^{-c_j(\delta_{j,1}+\delta_{j,2})-c_j(\alpha_j+\beta_j)\cdot\gamma} \cdot g^{-c_j})^{y_j}, g]$$
$$= e[\prod_{j\in J}(T_{j,3}^{c_j(x_i+\gamma)} h^{-c_j(\alpha_j+\beta_j)\cdot(x_i+\gamma)} \cdot g^{-c_j})^{y_j}, g]$$
$$= e[\prod_{j\in J}((A_i h^{(\alpha_j+\beta_j)})^{-(x_i+\gamma)} h^{(\alpha_j+\beta_j)\cdot(x_i+\gamma)} \cdot g)^{-c_j y_j}, g]$$
$$= e[\prod_{j\in J}((g^{1/(x_i+\gamma)})^{-(x_i+\gamma)} \cdot g)^{-c_j y_j}, g] = e(1, g)$$

所以

$$right = e[\prod_{j\in J}(T_{j,3}^{r_{j,x_i}+c_j x_i} \cdot h^{-r_{j,\delta_{j,1}}-c_j\delta_{j,1}-r_{j,\delta_{j,2}}-c_j\delta_{j,2}} \cdot g^{-c_j})^{y_j}, g] \cdot e[\prod_{j\in J}(h^{-r_{j,\alpha_j}-c_j\alpha_j-r_{j,\beta_j}-c_j\beta_j} \cdot T_{j,3}^{c_j})^{y_j}, w]$$
$$= e[\prod_{j\in J}(T_{j,3}^{r_{j,x_i}} h^{-r_{j,\delta_{j,1}}-r_{j,\delta_{j,2}}})^{y_j}, g] \cdot e[\prod_{j\in J}(T_{j,3}^{c_j x_i} h^{-c_j(\delta_{j,1}+\delta_{j,2})} \cdot g^{-c_j})^{y_j}, g] \cdot e[\prod_{j\in J}(h^{-r_{j,\alpha_j}-r_{j,\beta_j}})^{y_j}, w] \cdot e[\prod_{j\in J}(h^{-c_j(\alpha_j+\beta_j)} \cdot T_{j,3}^{c_j})^{y_j}, w]$$
$$= e[\prod_{j\in J}(T_{j,3}^{r_{j,x_i}} h^{-r_{j,\delta_{j,1}}-r_{j,\delta_{j,2}}})^{y_j}, g] \cdot e[\prod_{j\in J}(h^{-r_{j,\alpha_j}-r_{j,\beta_j}})^{y_j}, w] \cdot e(1, g)$$
$$= e[\prod_{j\in J}(T_{j,3}^{r_{j,x_i}} h^{-r_{j,\delta_{j,1}}-r_{j,\delta_{j,2}}})^{y_j}, g] \cdot e[\prod_{j\in J}(h^{-r_{j,\alpha_j}-r_{j,\beta_j}})^{y_j}, w]$$
$$= left$$

式(2)的验证过程如下：

$$\hat{c} = \prod_{j\in J} c_j^{y_j}$$
$$= \prod_{j\in J}(\eta^{m_j} \cdot H_1(id_j, \theta_j, T_{j,1}, T_{j,2}, T_{j,3}, R_{j,1}, R_{j,2}, R_{j,3}, R_{j,4}, R_{j,5})^{y_j}$$
$$= \prod_{j\in J}[\eta^{m_j y_j} \cdot (H_1(id_j, \theta_j, T_{j,1}, T_{j,2}, T_{j,3}, R_{j,1}, R_{j,2}, R_{j,3}, R_{j,4}, R_{j,5})^{y_j}]$$
$$= \overset{\wedge}{\eta}^{m} \cdot \prod_{j\in J} H_1(id_j, \theta_j, T_{j,1}, T_{j,2}, T_{j,3}, u^{r_{j,\alpha_j}}, v^{r_{j,\beta_j}}, R_{j,3}, T_{j,1}^{r_{j,x_i}} \cdot u^{-r_{j,\delta_{j,1}}}, T_{j,2}^{r_{j,x_i}} \cdot v^{-r_{j,\delta_{j,2}}})^{y_j}$$
$$= \overset{\wedge}{\eta}^{m} \cdot \prod_{j\in J} H_1(id_j, \theta_j, T_{j,1}, T_{j,2}, T_{j,3}, u^{s_{j,\alpha_j}}/T_{j,1}^{c_j}, v^{s_{j,\beta_j}}/T_{j,2}^{c_j}, R_{j,3}, T_{j,1}^{s_{j,x_i}}/u^{s_{j,\delta_{j,1}}}, T_{j,2}^{s_{j,x_i}}/v^{s_{j,\delta_{j,2}}})^{y_j}$$
$$= \overset{\wedge}{\eta}^{m} \cdot \prod_{j\in J} H_1(id_j, \theta_j, T_{j,1}, T_{j,2}, T_{j,3}, \widetilde{R}_{j,1}, \widetilde{R}_{j,2}, R_{j,3}, \widetilde{R}_{j,4}, \widetilde{R}_{j,5})^{y_j}$$

式(3)的验证过程如下：

$$e(g, \overset{\wedge}{\theta}) = e(g, \prod_{j\in J}(a^{\pi_{UR}H_2(id_j)+m_j\pi_{UR}\tau_{UR}})^{y_j})$$
$$= e(g^{\pi_{UR}}, \prod_{j\in J} a^{y_j H_2(id_j)+m_j y_j \tau_{UR}})$$
$$= e(\Omega_{UR}, \prod_{j\in J} a^{y_j H_2(id_j)} \cdot \prod_{j\in J} B^{m_j y_j})$$
$$= e(\Omega_{UR}, a^{\sum_j y_j H_2(id_j)} \cdot B^{\sum_j m_j y_j})$$
$$= e(\Omega_{UR}, a^{\sum_j y_j H_2(id_j)} \cdot B^{\hat{m}})$$

以上 3 个公式是正确的，因此 TPA 可以正确验证云端共享数据的完整性。

定理 2(可检测性) 本方案中，如果云服务器存储的文件有 n 个数据块，其中有 m 个数据块被恶意删除或修改，被挑战的数据块数量为 c，则数据损毁的检测率至少为 $1-[(n-m)/n]^c$。

证明：在审计过程中，被挑战的数据块中只要有一个受损，TPA 便能够检测出来。假设被挑战的数据块中有 X 个受损数据块，P_X 表示损毁数据的检测概率，则：

$$P_X = P\{X \geqslant 1\} = 1 - P\{X=0\}$$
$$= 1 - \frac{n-m}{n} \cdot \frac{n-1-m}{n-1} \cdot \cdots \cdot \frac{n-c+1-m}{n-c+1}$$

即 $P_X \geqslant 1-[(n-m)/n]^c$。也就是说,如果 1 000 000 个共享数据块中有 1%的数据块被损毁,那么验证者只需随机挑选 300 或 460 个数据块,检测率则至少可以达到 95%或 99%。

定理 3(审计可靠性)　本方案中,云服务器只有完整地存储群组用户的共享数据才可以通过 TPA 的审计。

证明:采用文献[2]知识证明的方法构造一个知识提取器。如果云服务器没有存储完整的数据但可以通过 TPA 验证,则意味着我们可以通过知识提取器与方案之间的反复交互来提取完整的被挑战数据块。通过以下游戏进行证明[14-15]。

游戏 0:挑战者运行 *KeyGen* 算法生成系统公开参数和群组密钥,并将公开参数发送给敌手。敌手挑选一系列数据块 $m_1,\cdots,m_n$ 发送给挑战者来询问相应的认证器,挑战者计算并回复这些数据块对应的认证器。在此之后,挑战者向敌手发起挑战,敌手回复 $P=\{\hat{c},\hat{R}_3,\hat{\theta},\hat{m},\{\sigma_j\}_{j\in J},\{id_j\}_{j\in J}\}$ 作为数据持有证明,如该证明正确,则挑战者失败。

游戏 1:游戏 1 与游戏 0 类似,不同之处在于挑战者保存对敌手质询的全部应答。挑战者观察它与敌手间的挑战实例,如果发现某个证明中的聚合标签 $\hat{\theta}'$ 不等于 $\prod_{j\in J}\theta_j^{y_j}$,则挑战者失败。

分析:假设诚实的证明者提供的正确证明为 $P=\{\hat{c},\hat{R}_3,\hat{\theta},\hat{m},\{\sigma_j\}_{j\in J},\{id_j\}_{j\in J}\}$,由方案的正确性可知以下验证等式成立:

$$e(g,\hat{\theta})=e(\Omega_{UR},a^{\sum_{j\in J}y_jH_2(id_j)}\cdot B^{\hat{m}}) \tag{4}$$

假设敌手伪造的证明为 $\{\hat{c},\hat{R}_3,\hat{\theta}',\hat{m}',\{\sigma_j\}_{j\in J},\{id_j\}_{j\in J}\}$,由于伪造是成功的,则以下验证等式成立:

$$e(g,\hat{\theta}')=e(\Omega_{UR},a^{\sum_{j\in J}y_jH_2(id_j)}\cdot B^{\hat{m}'}) \tag{5}$$

其中 $\hat{m}'\neq\hat{m}$,否则 $\hat{\theta}'=\hat{\theta}$ 与假设相驳,令 $\Delta\hat{m}=\hat{m}'-\hat{m}(\Delta\hat{m}\neq 0)$。如果敌手以不可忽略的概率赢得了游戏,则意味着模拟器可以解决 CDH 难题,具体过程如下。

给定 $g,g^a,h\in G_1$,模拟器的目标是计算 h^a,随机选择两个元素 $x,y\in Z_p^*$,并设置 $B=g^xh^y$。

在密钥生成阶段,挑战者设置 $\Omega_{UR}=g^{\alpha}=g^{\pi_{UR}}$ 并将其公开,挑战者不知道 π_{UR},τ_{UR} 等私有参数,但知道系统的公开参数。

为了回复敌手的随机预言查询,模拟器为挑战中的每个 j 选择一个相应的随机值 $r_j\in Z_p^*$,并定义随机预言值:$a^{H_2(id_j)}=g^{r_j}/(g^{xm_j}\cdot h^{ym_j})$(其中 $a=h(ID)\in G_1$,设 $a=g^t,t\in Z_p^*$)。因此,可得到 $a^{H_2(id_j)}\cdot B^{m_j}=g^{r_j}/(g^{xm_j}\cdot h^{ym_j})\cdot B^{m_j}=g^{r_j}/(g^{xm_j}\cdot h^{ym_j})\cdot g^{xm_j}\cdot h^{ym_j}=g^{r_j}$,也就是模拟器可以计算出数据块 m_j 的标签 $\theta_j=a^{\pi_{UR}H_2(id_j)+m_j\pi_{UR}\tau_{UR}}=a^{(H_2(id_j)+m_j\tau_{UR})\pi_{UR}}=(a^{H_2(id_j)}\cdot B^{m_j})^{\pi_{UR}}=(g^{\pi_{UR}})^{r_j}$。

式(5)除以式(4)可以得到 $e(\hat{\theta}'/\hat{\theta},g)=e(B^{\Delta\hat{m}},\Omega_{UR})=e((g^xh^y)^{\Delta\hat{m}},\Omega_{UR})$,因此 $e(\hat{\theta}'\cdot\hat{\theta}^{-1}\Omega_{UR}^{-x\Delta\hat{m}},g)=e(h,\Omega_{UR})^{y\Delta\hat{m}}=e(h^a,g)^{y\Delta\hat{m}}$,根据该式可以得出 $h^a=(\hat{\theta}'\cdot\hat{\theta}^{-1}\Omega_{UR}^{-x\Delta\hat{m}})^{1/(y\Delta\hat{m})}$。其中,游戏失败的概率与 $y\Delta\hat{m}=0(\bmod p)$ 的概率相同,$y\Delta\hat{m}=0(\bmod p)$ 的概率为 $1/p$,这个概率是可忽略的。也就意味着,如果敌手赢得游戏 0 和游戏 1 的概率存在不可忽略的差异,构造的模拟器就可以解决 CDH 难题[14-15]。

游戏 2:游戏 2 与游戏 1 类似,不同之处在于挑战者保存并观察挑战过程中的所有挑战与应答实例。对于每个实例,如果聚合的消息 $\hat{m}$ 不等于期盼的 $\sum_{j\in J}y_jm_j$,则挑战者失败。

分析:假定来自诚实证明者的正确证明为 $P=\{\hat{c},\hat{R}_3,\hat{\theta},\hat{m},\{\sigma_j\}_{j\in J},\{id_j\}_{j\in J}\}$,由方案的正确性可知验证等式 $e(g,\hat{\theta})=e(\Omega_{UR},a^{\sum_{j\in J}y_jH_2(id_j)}\cdot B^{\hat{m}})$ 成立。假设来自敌手的应答为 $\{\hat{c},\hat{R}_3,\hat{\theta}',\hat{m}',\{\sigma_j\}_{j\in J},\{id_j\}_{j\in J}\}$,若伪造成功,则验证等式 $e(g,\hat{\theta}')=e(\Omega_{UR},a^{\sum_{j\in J}y_jH_2(id_j)}\cdot B^{\hat{m}'})$ 成立。由游戏 1 可知 $\hat{\theta}'=\hat{\theta}$,令 $\Delta\hat{m}=\hat{m}'-\hat{m}(\Delta\hat{m}\neq 0)$。如果敌手以不可忽略的概率赢得了游戏,那么构造的模拟器就可以解决 DL 难题[14-15]。

给定 $g,h\in G_1$,模拟器的目标是计算 α 的值以满足 $h=g^{\alpha}$,选择两个随机数 $x,y\in Z_p^*$,设置 $B=g^xh^y$。根据以上两个验证等式可得到:$e(\Omega_{UR},a^{\sum_{j\in J}y_jH_2(id_j)}\cdot B^{\hat{m}})=e(g,\hat{\theta})=e(g,\hat{\theta}')=e(\Omega_{UR},a^{\sum_{j\in J}y_jH_2(id_j)}\cdot B^{\hat{m}'})$,因此 $B^{\hat{m}}=B^{\hat{m}'}$,故 $1=B^{\Delta\hat{m}}=(g^xh^y)^{\Delta\hat{m}}=g^{x\Delta\hat{m}}\cdot h^{y\Delta\hat{m}}$。其中 $\Delta\hat{m}\neq 0(\bmod p)$,否则 $\hat{m}=\hat{m}'(\bmod p)$,这与假设矛盾。因此,DL 难题的解法为:

$$h=g^{\frac{-x\Delta\hat{m}}{y\Delta\hat{m}}}=g^{\frac{-x}{y}}$$

即 $\alpha=-x/y$。此时 $y=0$ 的概率是 $1/p$,这是可忽略的。因此,解决 DL 难题的概率为 $1-1/p$,这与在 G_1 中计算 DL 难题是困难的假设矛盾。

这意味着,如果敌手赢得游戏 1 和游戏 2 的概率存在不可忽略的差异,构造的模拟器就可以解决 DL 难题。因此,这些游戏之间的差异是可忽略的。

最终,我们可以构造一个知识提取器来提取所有被挑战的数据块 $m_j(j\in J,|J|=c)$。对于相同的数据块 $m_j(j\in J,|J|=c)$,选择 c 个不同的系数 $y_j(j\in J,|J|=c)$ 并执行 c 次不同的挑战,可以得到 c 个关于变量 $m_j(j\in J,|J|=c)$ 的独立线性等式。通过求解这些等式,知识提取器可计算并提取 $m_j(j\in J,|J|=c)$。这意味着,如果云服务器可以通过 TPA 的审计验证,则它必须真实地存储群组用户的完整数据。

定理 4(身份隐私)　已知数据块 m_k 及其认证器 σ_k,只有群组管理员可以追踪数据块所对应的签名者的身份,对于验证者来说,分辨出该数据块签名者的身份在计算上是不可行的。

证明:群组管理员可以通过使用自身的私钥 $gmsk=(\lambda_1,\lambda_2)$ 来恢复数据块 m_k 签名者的身份:因为 $T_{k,3}/(T_{k,1}^{\lambda_1}\cdot T_{k,2}^{\lambda_2})=A_i\cdot h^{\alpha_k+\beta_{1k}}/(u^{\alpha_k\cdot\lambda_1}\cdot v^{\beta_k\cdot\lambda_2})=A_i\cdot h^{\alpha_k+\beta_k}/h^{\alpha_k+\beta_k}=A_i$,其中 A_i 是用户 i 私钥的一部分。如果一个验证者成功选择 c_k,满足 $c_k=\alpha_k+\beta_k$,那么该验证者就能够得到用户的部分私钥 A_i,也就意味着可以通过计算 $T_{k,3}/h^{c_k}=A_i\cdot h^{\alpha_k+\beta_k}/h^{c_k}$ 来揭示出该数据块签名者的身份。但是已知参数 $u,v,h,T_{k,1}=u^{\alpha_k},T_{k,2}=v^{\beta_k},h^{c_k}\in G_1$,求解 $c_k=\alpha_k+\beta_k$ 的问题等价于求解 G_1 中的 DL 难题。因此,对于验证者来说,揭示消息块 m_k 签名者的身份在计算上是不可行的。

定理 5(抗替代攻击) 攻击者想把用户 j 对数据块 m_k' 的合法签名 θ_k' 诬陷成用户 i 对数据块 m_k' 的签名在计算上是不可行的。

证明:攻击者想把数据块 m_k' 的签名 θ_k' 诬陷给用户 i 时,需要挑选 $R_{k,1}', R_{k,2}', R_{k,4}', R_{k,5}'$ 生成 $c'_k = \eta^{m_k'} \cdot H_1(id_k', \theta_k', T_{k,1}, T_{k,2}, T_{k,3}, R_{k,1}', R_{k,2}', R_{k,3}, R_{k,4}', R_{k,5}')$,同时需要提供相应的 $s'_{k,\alpha_k}, s'_{k,\beta_k}, s'_{k,x_i}, s'_{k,\delta_{k,1}}, s'_{k,\delta_{k,2}}$ 来构造认证器 $\sigma_k' = (T_{k,1}, T_{k,2}, T_{k,3}, c_k', R_{k,3}, \theta_k', s'_{k,\alpha_k}, s'_{k,\beta_k}, s'_{k,x}, s'_{k,\delta_{k,1}}, s'_{k,\delta_{k,2}}, UR, B)$。其中,$s'_{k,\alpha_k}$ 和 s'_{k,β_k} 的值需要分别被用来求解 $R'_{k,1} = u^{s'_{k,\alpha_k}} / T_{k,1}^{c_k'}$ 和 $R'_{k,2} = u^{s'_{k,\beta_k}} / T_{k,2}^{c_k'}$,该问题等价于求解 G_1 中的 DL 难题。因此,攻击者将用户 j 对数据块 m_k' 的合法签名 θ_k' 诬陷成用户 i 的签名在计算上是不可行的。

6 性能分析

6.1 功能分析

表 1 列出了本文方案与 Panda 方案[8]、Knox 方案[12]和文献[11]中的审计方案在功能上的对比结果。

表 1 功能对比

Table1 Comparison results of function

	身份追踪	抗替代攻击	用户撤销	身份隐私
Panda 方案	×	×	√	√
Knox 方案	√	×	×	√
文献[11]	×	×	√	√
本文方案	√	√	√	√

由表 1 可以看出,本文方案可以同时支持签名者身份追踪、抗替代攻击、高效的用户撤销以及身份隐私的保护功能。在 Panda 方案和文献[11]中,由于群组中的每个用户使用相同的群组密钥生成数据块认证器,这在一定程度上保证了用户的身份隐私,但群组管理员无法对组内具体的用户身份进行追踪,也无法抵抗替代攻击。Knox 方案虽然可以支持用户身份的追踪,但认证器中数据块签名没有和与用户身份有关的挑战值 c_k 进行绑定,因此存在替代攻击的风险。另外,Knox 方案中群组有用户撤销时,未撤销用户需要对属于自己的数据块进行重签名,属于撤销用户的数据块也需要由群组管理员重新计算,因此 Knox 方案不支持高效的用户撤销。

6.2 数据分析

本文针对方案中最消耗资源的几个阶段(认证器生成阶段、证明生成阶段、证明验证阶段和用户撤销阶段)进行了计算开销和通信开销的分析,如表 2 所列。

表 2 不同阶段的计算开销和通信开销

Table 2 Computation overhead and communication overhead in different phases

阶段	计算开销	通信开销
Authenticator Generation	$n(15Exp+16Mul+9Add+3Pair+Hash)$	$14n\|p\|$
Proof Generation	$3cExp+(4c-3)Mul+(c-1)Add$	$c\cdot(\|id\|+\|p\|)$
Proof Verify	$(8c+11)Exp+(5c+6)Mul+(3c-1)Add+4Pair+cHash$	$(13c+4)\cdot\|p\|+c\cdot\|id\|$
UserRevocation	$2Exp+Add$	$2\|p\|$

Exp_{G_1} 和 Mul_{G_1} 分别表示 G_1 中的幂运算和乘法运算;Exp_{G_2} 和 Mul_{G_2} 分别表示 G_2 中的幂运算和乘法运算;$Exp_{Z_p^*}$, $Mul_{Z_p^*}$, $Add_{Z_p^*}$ 和 $Hash_{Z_p^*}$ 分别表示 Z_p^* 中的幂运算,乘法运算、加法运算和 $Hash$ 运算;$Pair$ 表示线性操作;$|p|$ 表示 G_1, G_2 和 Z_p^* 中元素的大小;$|id|$ 为每个数据块标识的大小。

(1)认证器生成阶段

1)计算开销。本阶段计算开销 $n(11Exp_{G_1}+3Exp_{G_2}+Exp_{Z_p^*}+3Mul_{G_1}+2Mul_{G_2}+11Mul_{Z_p^*}+7Add_{Z_p^*}+3Pair+Hash_{Z_p^*})$,即 $n(15Exp+16Mul+9Add+3Pair+Hash)$,其中 n 表示共享文件数据块总数。

2)通信开销。本阶段群组用户和云服务器之间的通信开销为 $14n|p|$ bit。

(2)证明生成阶段

1)计算开销:本阶段云服务器生成数据持有证明 P 的计算开销为 $cExp_{G_1}+cExp_{G_2}+cExp_{Z_p^*}+(c-1)Mul_{G_1}+(c-1)Mul_{G_2}+(2c-1)Mul_{Z_p^*}+(c-1)Add_{Z_p^*}$,可以简化成 $3cExp+(4c-3)Mul+(c-1)Add$。

2)通信开销。本阶段的通信开销主要来于 TPA 生成的挑战,审计挑战 $chal=\{j, y_j\}_{j\in J}$ 规模为 $c\cdot(|id|+|p|)$ bit。

(3)证明验证阶段

1)计算开销。接收到数据持有证明 P 后,TPA 需要验证 P 的正确性,计算开销为 $(7c+10)Exp_{G_1}+(c+1)Exp_{Z_p^*}+(3c+5)Mul_{G_1}+Mul_{G_2}+2cMul_{Z_p^*}+(3c-1)Add_{Z_p^*}+4Pair+cHash_{Z_p^*}$,化简得到 $(8c+11)Exp+(5c+6)Mul+(3c-1)Add+4Pair+cHash$。

2)通信开销。本阶段的通信开销来自云服务器生成的数据持有证明 P,数据持有证明 P 的规模为 $(13c+4)\cdot|p|+c\cdot|id|$ bit。

(4)用户撤销阶段

1)计算开销。本阶段的计算开销来自于群组新私钥 gsk_{new} 和相应公开参数的生成,因此,当群组中有用户撤销时,群组管理员的计算开销为 $2Exp_{G_1}+Add_{Z_p^*}$,即 $2Exp+Add$。

2)通信开销。在用户撤销阶段,群组管理员和一个群组用户之间的通信开销为 $2|p|$ bit。

6.3 实验结果

本文对提出的审计方案的性能进行分析。实验在 2.70 GHz 处理器、4 GB 内存的 Linux 服务器上,利用 C 语言并结合 GMP 和 PBC 函数库进行。实验设置基域大小为 512 bit,Z_p^* 中的元素大小为 160 bit,假定共享云文件的大小为 20 MB,被划分成 1000000 个数据块,群组用户数 $d=10$。

由定理 2 可知,如果有 1%的数据块被损毁,要使检测率达到 95%或 99%,审计任务中随机采样的数据块数量应为 $c=300$ 或 $c=460$。审计过程主要包括 3 个阶段:审计挑战的产生、审计证明的生成和审计证明的验证。从表 3 可以看出,当 $c=300$ 时,本方案的计算开销为 6.28 s,通讯开销为 3.45 kB;当 $c=460$ 时,本方案的计算开销为 7.353 s,通讯开销为 4.03 kB。如果用户想要获得一个更高的错误检测率,则 TPA 需要消耗更多的时间和带宽来完成审计任务。

表 3 审计性能

Table 3 Auditing performance

Number of Selected Blocks/c	Computation Cost/s	Communication Cost/kB
300	6.28	34.46
460	7.35	40.32

图7给出了在用户撤销阶段，本文方案与Panda方案、Knox方案和文献[11]中的审计方案的用户端计算开销。对于Panda方案和Knox方案来说，当群组中有一个用户撤销时，群组管理员需要为组内用户生成一个新的群组私钥，用户使用新密钥对共享数据块进行重签名操作，两个方案的计算开销较大；对于文献[11]中的方案来说，未撤销用户需要执行一个加法运算，得到新的群组私钥；而在本方案中，当群组中有用户撤销时，群管理员需要对群组密钥进行更新，但群组内每个用户无需执行任何操作，且TPA仍然能够对之前的认证器进行认证。

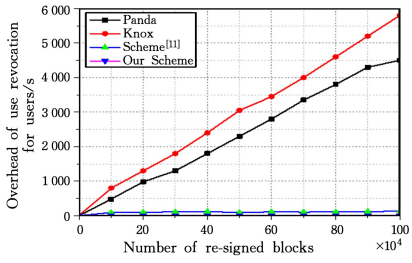

图7　群组用户端的用户撤销计算开销

Fig. 7　Computation overhead of user revocation on group user side

结束语　本文提出了一种基于群签名的用户身份可追踪的云共享数据完整性审计方案。在所提方案中，当需要对指定数据块签名者的身份进行追踪时，群管理员可以追踪签名者的身份，且他人无法得知群组用户的身份隐私。为了防止非法用户对共享数据的访问，用户在上传或访问云端数据前需要进行身份认证。此外，本方案支持高效的用户撤销，当组内有用户撤销时，云服务器或未撤销用户无需对任何数据块进行重签。安全性分析和实验结果表明，本方案是安全、高效的。本文方案根据用户撤销数 UR 值对数据块进行分组验证，这意味着审计效率随 UR 的增加而降低，当群组用户撤销过于频繁时，方案的审计效率也会随之降低，因此，如何进一步提高用户撤销时的审计效率是接下来的工作重点。

参 考 文 献

[1] REN K, WANG C, WANG Q. Security Challenges for the Public Cloud[J]. IEEE Internet Computing, 2012, 16(1): 69-73.

[2] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possessionat untrusted stores[C]// Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM, 2007: 598-609.

[3] ERWAY C C, KÜPÇÜ A, PAPAMANTHOUS C, et al. Dynamic provable data possession[J]. ACM Transactions on Information and System Security(TISSEC), 2015, 17(4): 15.

[4] ZHU Y, WANG H, HU Z, et al. Dynamic audit services for integrity verification of outsourced storages in clouds[C]// Proceedings of the 2011 ACM Symposium on Applied Computing. ACM, 2011: 1550-1557.

[5] CAO N, YU S, YANG Z, et al. LT codes-based secure and reliable cloud storage service[C]// 2012 Proceedings IEEE INFOCOM. IEEE, 2012: 693-701.

[6] WANG B, LI B, LI H. Oruta: Privacy-preserving public auditing for shared data in the cloud[J]. IEEE Transactions on Cloud Computing, 2014, 2(1): 43-56.

[7] WANG B, LI H, LI M. Privacy-preserving public auditing for shared cloud data supporting group dynamics[C]// 2013 IEEE International Conference on Communications (ICC). IEEE, 2013: 1946-1950.

[8] WANG B, LI B, LI H. Panda: Public auditing for shared data with efficient user revocation in the cloud[J]. IEEE Transactions on Services Computing, 2013, 8(1): 92-106.

[9] JIANG T, CHEN X, MA J. Public integrity auditing for shared dynamic cloud data with group user revocation[J]. IEEE Transactions on Computers, 2015, 65(8): 2363-2373.

[10] YU J, WANG H. Strong key-exposure resilient auditing for secure cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(8): 1931-1940.

[11] ZHANG Y, YU J, HAO R, et al. Enabling Efficient User Revocation in Identity-based Cloud Storage Auditing for Shared Big Data[J]. IEEE Transactions on Dependable and Secure Computing, 2018, PP(99): 1-1.

[12] WANG B, LI B, LI H. Knox: privacy-preserving auditing for shared data with large groups in the cloud[C]// International Conference on Applied Cryptography and Network Security. Springer, Berlin, Heidelberg, 2012: 507-525.

[13] BONEH D, BOYEN X, SHACHAM H. Short group signatures[C]// Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 2004: 41-55.

[14] REN K, WANG C, WANG Q. Security challenges for the public cloud[J]. IEEE Internet Computing, 2012, 16(1): 69-73.

[15] SHACHAM H, WATERS B. Compact proofs of retrievability[J]. Journal of Cryptology, 2013, 26(3): 442-483.

[16] YANG G, YU J, SHEN W, 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.

[17] SHEN W, YU J, XIA H, et al. Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium[J]. Journal of Network and Computer Applications, 2017, 82: 56-64.

[18] SOOKHAK M, YU F R, ZOMAYA A Y. Auditing big data storage in cloud computing using divide and conquer tables[J]. IEEE Transactions on Parallel and Distributed Systems, 2017, 29(5): 999-1012.

[19] ZHANG Y, YU J. ID-based Cloud Storage Integrity Detection Scheme[J]. Computer Engineering, 2018, 44(3): 8-12, 18.

[20] YU J, HAO R, ZHAO H. IRIBE: Intrusion-resilient identity-based encryption[J]. Information Sciences, 2016, 329: 90-104.

ZHANG Xi, born in 1995, postgraduate, is a member of China Computer Federation. Her main research interests include cloud computing security and applied cryptography.

WANG Jian, born in 1968, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include key management, cryptographic protocoland privacy protection.