多方参与高效撤销组成员的共享数据审计方案

田俊峰 井 宣*

(河北大学网络空间安全与计算机学院 保定 071000)

(河北省高可信信息系统重点实验室 保定 071000)

摘 要：针对云平台上共享数据的完整性验证问题，该文提出一种多方参与高效撤销组成员的共享数据审计方案(SDRM)。首先，通过Shamir秘密共享方法，使多个组成员共同参与撤销非法组成员，保证了组成员间的权限平等。然后，结合代数签名技术，用文件标识符标识数据拥有者的上传数据记录和普通组成员的访问记录，使数据拥有者能够高效更新其所有数据。最后对方案的正确性、安全性和有效性进行理论分析和实验验证，结果表明，该文方案的计算复杂度与被撤销组成员签名的文件块数之间相互独立，达到了高效撤销组成员的目的。并且，随数据拥有者数量增加，该方案更新数据效率较NPP明显提升。

关键词：共享数据；Shamir秘密共享；代数签名；文件标识符

中图分类号：TN919; TN918 **文献标识码**：A **文章编号**：1009-5896(2020)06-1534-08

DOI: 10.11999/JEIT190468

Shared Data Auditing Scheme for Efficient Revocation of Group Members via Multi-participation

TIAN Junfeng JING Xuan

(*School of Cyberspace Security and Computer Institute*, *Hebei University*, *Baoding* 071000, *China*)

(*Hebei Key Laboratory of High Confidence Information Systems*, *Hebei University*, *Baoding* 071000, *China*)

Abstract: In the view of the integrity verification problem of data sharing on the cloud platform, a Shared Data auditing scheme for efficient Revocation of group Members via multi-participation (SDRM) is proposed. First, through the Shamir secret sharing method, multiple group members participate in revoking the illegal group members, ensuring the equal rights between the group members. Second, this scheme combines with algebraic signature technology, the file identifier identifies the data owner's upload data record and the normal group member's access record, enabling the data owner to update efficiently all of its data. Finally, theoretical analysis and experimental verification of the correctness, safety and effectiveness of the scheme show that the scheme meets the requirement of efficient cancellation of group members, at the same time, as the number of data owners increases, the efficiency of updating data in this scheme is significantly higher than that of NPP.

Key words: Shared data; Shamir secret sharing; Algebraic signature; File identifier

1 引言

为使用户能够验证云端数据的完整性，研究者们提出了数据持有性证明(Provable Data Possession, PDP)方案。2007年，Ateniese等人[1]首次提出了可证明数据拥有模型，无需检索全部数据便能够验证云端数据的完整性。然而，该方案只支持用户对文件的静态操作，却不支持动态操作。在随后的方案中，Ateniese等人[2]利用对称密钥，实现了支持部分动态操作的PDP方案。Wang等人[3]利用梅克尔哈希树(Merkle Hash Tree, MHT)，通过叶子结点存储相应文件块的哈希值来实现数据的更新与验证，提出了基于MHT的全动态PDP方案。在一些实际场景中，云端数据不仅支持数据拥有者动态更新，而且能够被多个用户共享。

为了实现组成员间数据共享，一些研究者基于PDP模型，提出了共享数据的审计方案[4]。Wang等人[5]首先提出了一种利用环签名技术的共享数据审计方案，保护了组成员的身份隐私安全，但由于该方案计算认证标签(又称认证器)的复杂度与组的大小成线性关系，导致了认证标签的生成和数据完

收稿日期：2019-06-24；改回日期：2019-12-30；网络出版：2020-01-11

*通信作者： 井宣 abidble@gmail.com

基金项目：国家自然科学基金(61802106)

Foundation Item: The National Natural Science Foundation of China (61802106)

整性验证的效率低下。Worku等人[6]使用随机遮掩技术进行数据隐藏，通过组密钥签名来构建同态线性认证标签，降低了计算复杂度，保护了组成员的数据隐私安全。随后Shen等人[7]通过指定代理来代替组成员计算认证标签，保护组成员身份隐私和数据隐私的同时，实现了组成员的轻量级计算。最近黄龙霞等人[8]基于逻辑层次树和代理重签名，实现了有效的组密钥管理，从而保护了组成员的身份隐私安全。随后文献[9]又通过消除证书管理，实现了无证书的共享数据审计方案。上述方案虽然在组成员隐私保护层面作了相关研究，但均未考虑组成员本身存在非法访问共享数据的可能[10]。

为使方案支持撤销非法组成员，Wang等人[11]采用代理重签名技术，通过向云端发送重签名参数，将被撤销组成员签名的认证标签，转换为合法组成员签名的认证标签，使非法组成员的密钥失效，最终撤销非法组成员。但其缺点是无法抵抗云端与非法组成员的共谋攻击。Yuan等人[12]使用了基于多项式的认证标签，实现了支持撤销多用户的共享数据审计方案，但同样无法抵抗共谋攻击。最近Luo等人[13]利用Shamir秘密共享方法，将重签名参数划分为秘密份额分发给不同的代理，达到了高效撤销非法组成员的目的，同时抵抗了共谋攻击。上述方案虽然以较高的效率实现了撤销非法组成员，但其计算复杂度与被撤销组成员签名的文件块数线性正相关，当审计云端的批量数据时，合法组成员仍有很大计算负担。

Jiang等人[14]基于向量承诺，提出了一种由验证者本地撤销非法组成员的方案，该方案虽然更加高效，但却忽略了验证者的安全隐患。Zhang等人[15]的方案中，每次撤销组成员之后，只需要组管理者执行一次累加操作，进一步提升了效率，同时降低了通信开销。但该方案忽略了组管理者管理上的不可控因素(如包庇组成员的非法行为或其被恶意攻击等)。为了解决上述问题，Fu等人[16]提出了一个多管理者的审计方案NPP(New Privacy-aware Public auditing mechanism)，通过构造二叉树记录文件块使用历史，由多管理者共同合作，从而撤销非法组成员。但在该方案中，数据拥有者充当组管理者，拥有者的数量越少，管理权就会越集中，所以不能真正保证组成员的权限平等。另外，该方案采用环签名的方法计算认证标签，数据拥有者更新或撤销数据时，计算量随拥有者数量的增加而呈指数级增长。

本文在NPP的基础上进行改进，提出了一种多方参与高效撤销组成员的共享数据审计方案(Shared Data auditing scheme supports efficient Revocation of group Members via multi-participation, SDRM)。主要贡献如下：

(1) 建立了多个数据拥有者共享数据的系统模型，提出了新的支持本地撤销组成员的共享数据审计方案；

(2) 参考Shamir秘密共享方法[17]，为各个组成员分发不同的秘密份额，各个组成员使用共享数据之后，利用秘密份额来盲化文件标识符，保证了文件标识符的安全性。当组成员的数量超过Shamir门限值时，各组成员就能通过文件标识符追溯使用记录，进而共同撤销非法组成员，保证了组成员的权限平等；

(3) 结合代数签名技术[18]，用文件标识符计算共享数据的认证标签，标识了数据拥有者的上传数据记录和普通组成员的访问记录，方便了多数据拥有者高效更新云数据；

(4) 通过进行实验验证，证明本方案能够实现高效撤销组成员和数据拥有者高效更新数据，具有可行性。

2　系统模型

如图1所示的系统模型包含5类实体：组成员、数据拥有者、密钥生成器(Private Key Generator, PKG)、第三方审计者(Third Party Auditor, TPA)和云端。

(1) 组成员：一个组由多个组成员组成，任何组成员都可以访问所在组的共享数据，并且能够注册或离开组。为验证共享数据的完整性，组成员向TPA发送审计请求，TPA向组成员返回审计结果；

(2) 数据拥有者：一个组有1个或多个数据拥有者，数据拥有者拥有组成员的所有权限，能够将其数据上传到云端供其他组成员共享，并且能够更新和撤销其所有数据；

(3) PKG：通常被认为是可信任的实体，负责生成系统公共参数和组成员的密钥；

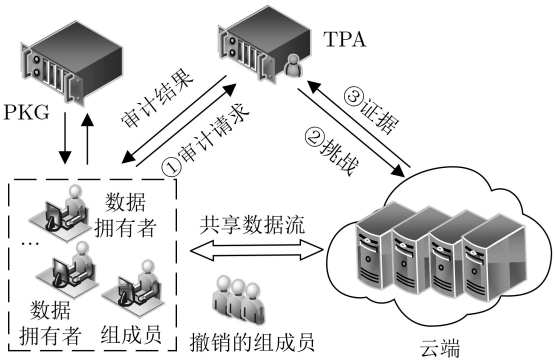

图 1 系统模型

(4) TPA：负责代表组成员验证共享数据的完整性。TPA向云端发起挑战后，云端向TPA返回拥有共享数据的证据。TPA通过判断该证据的正确性，来验证共享数据的完整性；

(5) 云端：为用户提供了共享数据的存储服务，为组成员间共享数据提供了平台。

当组成员想要验证云数据完整性时，向TPA发送审计请求，然后TPA向云端发起挑战。收到此挑战后，云端向TPA返回拥有共享数据的证据。最后TPA验证该证据的正确性，并把审计结果返回给组成员。

3 预备知识

3.1 符号说明

m：数据块集合$m=\{m_1,m_2,\cdots,m_i,\cdots,m_n\}$，数据被分为$n$个块，每个块由$m_i(1\le i\le n)$表示，$m_i\in Z_p^*$；

U：用户集合$U=\{U_1,U_2,\cdots,U_k,\cdots,U_s\}$，方案有$s$个组成员，每个组成员由$U_k(1\le k\le s)$表示；

n：数据块数量；

s：组成员数量；

κ：参与者数量；

ssk：授权私钥；

X：组私钥；

$F[k][i]$：对应U_k所上传m_i的标识符；

τ：所有组成员秘密份额的总和；

u_k：文件标识符被U_k使用后的参数；

u_k'：文件标识符被U_k盲化后的参数；

u_k^t：U_k第t次更新u_k后，所有u_k'的总和为u_k^t；

$\mathrm{SSig}_{\mathrm{key}}(\cdot)$：由key加密的数字签名；

F_t: U_k第t次更新u_k后，所有文件标识符总和；

F_{t-1}: U_k第t−1次更新u_k后，所有文件标识符总和；

F_{t-2}: U_k第t−2次更新u_k后，所有文件标识符总和。

3.2 基础知识

(1) 双线性对映射：令G_1和G_2是两个阶为大素数p的循环群，并且g_1,g_2是群G_1的生成元。双线性对映射为$e:G_1\times G_1\to G_2$，并具有以下属性：

(a) 双线性：对任意$g_1,g_2\in G_1$和$a,b\in Z_p^*$，有$e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$；

(b) 非退化性：$e(g_1,g_2)\ne 1$；

(c) 可计算：存在有效的算法来计算这个配对。

(2) CDH(Computational Diffe-Hellman)问题：对于$x,y\in Z_p^*$，给定$g,v=g^x$和$g^y\in G_1$，计算$v^y\in G_1$。当以不可忽略的概率难以计算出上述问题时，CDH假设成立。

(3) 代数签名：代数签名是一种具有代数性质的哈希函数。一个文件F的代数签名可以由Galois域(Galois Field, GF)[18]中的位符号表示。对于一个被分成$n(n\in N)$个部分的文件块F，可以计算它的代数签名：$S_g(F)=\sum_{i=1}^{n}F[i]\cdot g^{i-1}$，其中，$F[i]$表示文件的第$i(i\in N)$个文件块，$g$是阶为大素数$p$的循环群$G_1$的生成元。并且对于两个大小相同的文件$F$和$G$, F+G的代数签名为$S_g(F+G)=S_g(F)+S_g(G)$。

(4) Shamir秘密共享：Shamir提出的(Ξ,Γ)秘密共享方案是基于Lagrange插值公式构造的，假设一个组中有$\Gamma(\Gamma\in N)$个参与者，并且分发的秘密值$\zeta\in Z_p$。门限值为$\Xi(1\le\Xi\le\Gamma,\Xi\in N)$的秘密共享方式描述如下：

(a) 秘密分发者在GF中任意选取$\Xi-1$个随机元素$a_i\in Z_p(i$=1,2,$\cdots,\Xi-1)$，并构建插值多项式：$L(x)=\zeta+a_1x+\cdots+a_{\Xi-1}x^{\Xi-1}$，其中$p$是一个大素数且$p$>$\zeta$，秘密值$\zeta=L(0)=a_0$；

(b) 秘密分发者为每个参与者计算秘密值$y_i=L(x_i)\in Z_p,1\le i\le\Gamma$，并把$(x_i,y_i)$发送给参与者；

(c) 至少Ξ个参与者能够一起恢复秘密ζ：$\zeta=L(0)=\sum_{i=1}^{\Xi}y_i\prod_{f=1,f\ne i}^{\Xi}\frac{-x_f}{x_i-x_f}$。

4 SDRM方案

方案中μ,g是群G_1,G_2中的生成元，$H:Z_p{}^*\{0,1\}^*\to G_1$是随机密码散列函数。

(1) Setup: PKG为每个组成员生成授权私钥和秘密份额；

(a) PKG随机生成授权私钥ssk$\in Z_p^*$；

(b) PKG随机生成组私钥$X\in G_1$，计算各个组成员的秘密份额(x_k,y_k)如下：

(i) PKG随机选择$a_i\in Z_p(i\in[1,\kappa-1])$，并构建插值多项式$L(x)=X+a_1x+\cdots+a_{\kappa-1}x^{\kappa-1}$，其中$L(0)$=$X$；

(ii) PKG随机生成s个随机值$x_k\in G_1$, $k=1,2,\cdots,s$，进而计算$y_k=L(x_k)$，并将(x_k,y_k)分发给各个组成员；

(iii) PKG计算$\tau=\sum_{k=1}^{s}x_k$，然后计算τ的数字签名$\mathrm{SSig}_X(\tau||\mathrm{SSig}_X(\tau))$作为公共参数。

(c) PKG保存X，将ssk分发给各个组成员。

(2) Data prepare：各组成员计算文件块标识符信息，并将其发送给组内其他组成员。

如图2所示，数据拥有者U_1, U_2在上传文件之前，对文件分块并按顺序标号，U_1的数据为$m_1,m_2,\cdots,m_a$，标识符为$F[1][1],F[1][2],\cdots,F[1][n]$, U_2的数据为$m_{a+1},m_{a+2},\cdots,m_b$，标识符为$F[2][1],F[2][2],\cdots,F[2][n]$, $1\le a\le n,1\le b-a\le n$，

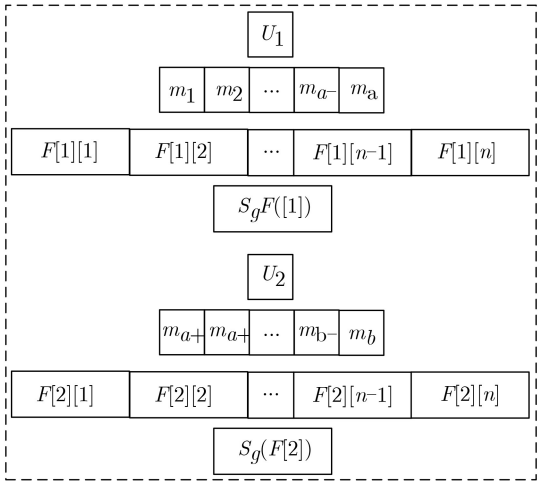

图 2 文件处理图

$n \in N$, $F[k][i] = H(m_i)$，对于文件块数少于n的部分，其标识符由0补齐、对于非数据拥有者的组成员，其初始标识符由全1表示。为了方便计算，对应$U_k(1 \le k \le s)$的文件标识符由代数签名$S_g(F[k])$表示。

$$S_g(F[k]) = \sum_{i=1}^{n} F[k][i] \cdot g^{i-1} \tag{1}$$

组成员计算出$S_g[F[k]]$后，将其发送给组内其他组成员，各组成员由Identity verify阶段的身份验证方式验证发送者的身份信息，验证通过后接收该文件标识符。

(3) Identity verify：各组成员采用身份标签[19]互相验证身份。

(a) 组成员随机选取$r_k \in Z_p^*$，计算验证值g^{r_k}；

(b) 组成员设置$\alpha_0 = U_k || g^{r_k}$，计算身份标签$\alpha$。其中$\mathrm{SSig}_{\mathrm{ssk}}(\alpha_0)$为授权私钥ssk对$\alpha_0$进行的签名

$$\alpha = \alpha_0 || \mathrm{SSig}_{\mathrm{ssk}}(\alpha_0) \tag{2}$$

(c) 组成员计算$\beta = \mu^{r_k}$，将α和β发送给其他组成员；

(d) 根据授权私钥，接收者通过验证$\mathrm{SSig}_{\mathrm{ssk}}(\alpha_0)$的正确性，来验证发送者的身份标签$\alpha$。如果验证通过，接收者解析$\alpha_0$来获取验证值$g^{r_k}$；

(e) 接收者通过$e(\mu, g^{r_k}) = e(\beta, g)$来验证发送者的身份信息，如果验证通过，接收者接收$U_k$的文件标识符。

(4) Authenticator generation：各个组成员下载文件块之后，向其他组成员发送新的文件标识符。各组成员上传文件之前，根据最新文件标识符来计算认证标签，然后将文件块和认证标签发送到云端进行存储。

为了体现文件的使用历史，(1)可以表示为：使用/访问第k个数据拥有者的第i个文件块后的文件标识符。例如，组成员U_2使用U_1的第3块数据后，首先需要更新文件标识符被U_2使用后的参数u_2

$$u_2 = \sum_{i=1}^{n} F[2][i] \cdot g^{i-1} + F[1][3] \tag{3}$$

然后，U_2利用秘密份额x_2盲化u_2

$$u_2' = x_2 + u_2 = x_2 + \left(\sum_{i=1}^{n} F[2][i] \cdot g^{i-1} + F[1][3] \right) \tag{4}$$

最后，U_2将u_2'发送给其他组成员。对于没有访问过数据的组成员，都根据各自的秘密份额x_k盲化文件标识符：$u_k' = x_k + u_k = x_k + \sum_{i=1}^{n} F[k][i] \cdot g^{i-1}$，并将$u_k'$发送给其他组成员。如果数据拥有者更新或撤销其所有数据，则按照Data prepare中的算法重新计算(1)，进而计算$u_k' = x_k + u_k = x_k + S_g[F[k]]$。

(a) 各个组成员由Identity verify验证发送者身份信息，验证通过后，由最新更新u_k的组成员计算全部的u_k'之和u_k^t。然后，该组成员计算公钥$\mathrm{pk}_1 = g^{u_k^t}$，并存储$u_k^t$

$$u_k^t = \sum_{k=1}^{s} (x_k + u_k) \tag{5}$$

(b) 组成员根据接收到的最新u_k^t计算认证标签σ_i，并将(m_i, σ_i)发送到云端

$$\sigma_i = (H(i) \cdot \mu^{m_i})^{u_k^t} \tag{6}$$

(c) 云端验证σ_i: $e(\sigma_i, g) = e(H(i) \cdot \mu^{m_i}, \mathrm{pk}_1)$，若等式成立，则接收并存储$(m_i, \sigma_i)$，否则拒绝接收。

(5) Proof generation：TPA向云端发起挑战，云端生成证据以证明其存储着共享数据。

(a) TPA生成审计挑战：

(i) 随机选择包含c个元素的集合I，其中$I \subseteq [1, n]$；

(ii) 对应于$i \in I$，生成随机值$v_i \in Z_p^*$；

(iii) 向云端发送审计挑战$\mathrm{chal} = \{i, v_i\}_{i \in I}$。

(b) 在收到来自TPA的审计挑战后，云端生成存储共享数据的完整性证据：

(i) 计算$\sigma = \prod_{i \in I} \sigma_i^{v_i}$, $\rho = \sum_{i \in I} v_i m_i$；

(ii) 向TPA发送审计证据$\mathrm{prf} = \{\rho, \sigma\}$。

(6) Verification：TPA接收来自云端的证据prf，并验证证据的正确性。如果式(7)正确，则验证通过。

$$e(\sigma, g) = e\left(\prod_{i \in I} H(i)^{v_i} \cdot \mu^{\rho}, \mathrm{pk}_1 \right) \tag{7}$$

(7) User revocation：通过使用记录表追溯非法组成员，由多个组成员共同撤销非法组成员。

如图3所示，每个组成员均存有一个使用记录表，记录其使用文件块后的文件标识符u_k^t。例如，U_1第t次更新u_k后，记录的文件块标识符为u_1^t。如果此时组成员U_1对m_{c-1}的使用过程产生争议，由标识符$F[3][c–1]$, U_1得到m_{c-1}为最初由U_3上传的第c–1个文件块。

各组成员根据秘密份额$\{(x_k,y_k)\}_{1\le k\le \kappa}$，共同计算关于秘密份额$x$的多项式(8)，然后计算多项式$L(x)=\sum_{k=1}^{\kappa} y_k\cdot\lambda_k(x)$，进而得到组私钥$X=L(0)=\sum_{k=1}^{\kappa} y_k\lambda_k(0)$。组成员根据$X$就能解析出$\mathrm{SSig}_X(\tau\,||\mathrm{SSig}_X(\tau))$中的$\tau$。

$$\lambda_k(x)=\prod_{f=1,f\ne k}^{\kappa}\left(\frac{-x_f}{x-x_f}\right) \tag{8}$$

根据u_1^t,g和τ，可以计算$F_t=u_1^t-\tau=\sum_{k=1}^{s}\sum_{i=1}^{n}F[k][i]\cdot g^{i-1}$。然后计算$F_{t-1}=u_3^{t-1}-\tau=\sum_{k=1}^{s}\sum_{i=1}^{n}F[k][i]\cdot g^{i-1}$。最后比较$F[3][c-1]$与$F_t-F_{t-1}$的值是否相同，若相同，则$U_3$为使用文件块$m_{c-1}$的组成员；若不相同，根据使用记录表向前迭代计算$F_{t-2}$，比较$F[3][c-1]$与$F_{t-1}-F_{t-2}$的值是否相同，依次类推，直到找出所有使用该数据的组成员，撤销非法组成员。数据拥有者U_3计算并存储新的u_3^t, PKG重发密钥，组成员更新身份验证信息，被撤销的组成员将无法通过其他组成员的验证。

本文采用区块链共识算法，将各个组成员的使用记录表作为交易记录，来解决使用记录表可信性低的问题。第5.3节对共识算法在本文的应用标准作了简要描述，详细应用过程参考文献[20]。

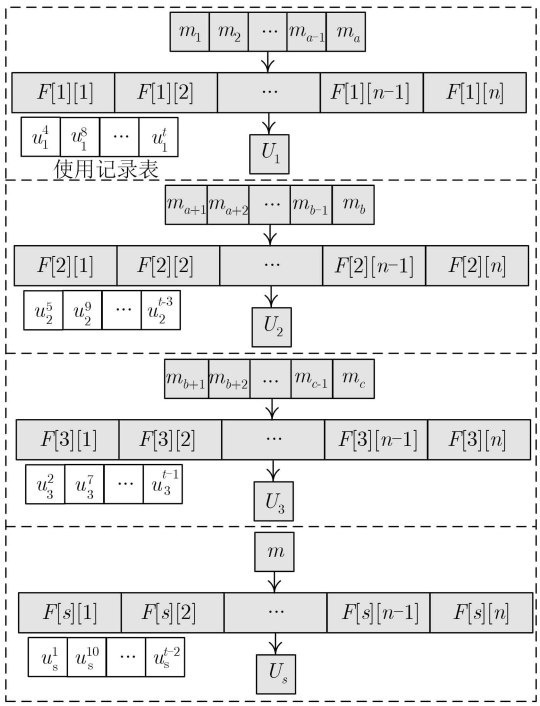

图 3 文件块使用记录

5 安全性分析

本节从审计正确性和审计健壮性方面分别给出了安全性分析。然后，证明了使用记录表的可信性。最后，说明了本文能够保证组成员的权限平等。

5.1 共享数据完整性审计的正确性

云端返回证据prf后，TPA验证等式(7)，从而完成共享数据的完整性验证。过程为

$$\begin{aligned}e(\sigma,g)&=e\left(\prod_{i\in I}\sigma_i^{v_i},g\right)\\&=e\left(\prod_{i\in I}\left((H(i)\cdot\mu^{m_i})^{u_k^t}\right)^{v_i},g\right)\\&=e\left(\prod_{i\in I}H(i)^{v_i}\cdot\mu^{\sum_{i\in I}m_iv_i},g^{u_k^t}\right)\\&=e\left(\prod_{i\in I}H(i)^{v_i}\cdot\mu^{\rho},g^{u_k^t}\right)\\&=e\left(\prod_{i\in I}H(i)^{v_i}\cdot\mu^{\rho},pk_1\right)\end{aligned} \tag{9}$$

5.2 共享数据完整性审计的健壮性

命题1 若以不可忽略的概率难以解决CDH问题，那么在本文所提出的审计方案中，云端只有真正存储组成员的数据时才能通过TPA的验证。

证明 挑战者C存储一个列表来记录所有被查询的认证标签，C观察攻击者A的每次挑战和响应的过程。如果A能够提供证明prf通过C的验证，并且C发现σ与$\prod_{i\in I}\sigma_i^{v_i}$不相等，则$C$失败。

分析 假设诚实的证明者生成了正确的证明$\mathrm{prf}=\{\rho,\sigma\}$。从方案的正确性考虑，验证方程式(10)成立

$$e(\sigma,g)=e\left(\prod_{i\in I}H(i)^{v_i}\cdot\mu^{\rho},pk_1\right) \tag{10}$$

假设A伪造了证明$\mathrm{prf}=\{\rho',\sigma'\}$，因为$A$通过了$C$的验证，所以验证方程式(11)成立

$$e(\sigma',g)=e\left(\prod_{i\in I}H(i)^{v_i}\cdot\mu^{\rho'},\mathrm{pk}_1\right) \tag{11}$$

通过分析本方案的审计过程可知，如果$\rho=\rho'$，那么$\sigma=\sigma'$，而这与上述假设相矛盾。由此定义$\Delta\rho=\rho'-\rho\ne 0$，以下通过构建模拟器来说明，如果$A$通过了审计过程，模拟器将以不可忽略的概率解决CDH问题。给定$g,g^{u_k^t},h\in G_1$，模拟器按如下方式模拟输出$h^{u_k^t}$。

模拟器选择两个随机值$a,\varepsilon\in Z_p^*$，设定$\mu=g^ah^{\varepsilon}$。

每当挑战第i个文件块，模拟器选择一个随机值$r_i \in Z_p^*$，并在i处执行随机预言模型$H(i)=g^{r_i}/(g^{am_i}\cdot h^{\varepsilon m_i})$，因此，模拟器能够计算$\sigma_i'=(g^{r_i}/(g^{am_i}\cdot h^{\varepsilon m_i})\cdot\mu^{m_i})^{u_k^t}=(g^{r_i})^{u_k^t}=(g^{u_k^t})^{r_i}$。

通过等式(11)除以等式(10)，得$e(\sigma'/\sigma,g)=e(\mu^{\Delta\rho},pk_1)=e((g^a h^{\varepsilon})^{\Delta\rho},pk_1)$，进而得到等式(12)

$$e(\sigma'/\sigma\cdot pk_1^{-a\Delta\rho},g)=e(h,pk_1)^{\varepsilon\Delta\rho} \tag{12}$$

通过等式(12)可得$h^{u_k^t}=(\sigma'/\sigma\cdot pk_1^{-a\Delta\rho})^{1/\varepsilon\Delta\rho}$。所以，当且仅当$1/\varepsilon\Delta\rho=0 \bmod p$，模拟器能够计算出$h^{u_k^t}$。又因为$1/\varepsilon\Delta\rho=0 \bmod p$的可能性为$1/p$，所以计算出$h^{u_k^t}$的概率可以忽略不计。也就是说，如果$A$通过了审计过程，那么构建的模拟器可以解决CDH问题。而这与若以不可忽略的概率难以解决CDH问题的假设相矛盾。因此，按照设计方式执行，云端只有在真正存储组成员的数据时才能通过TPA的验证。　　证毕

5.3 使用记录表的可信性

命题2　要保证使用记录表的可信性，需要满足共识算法的容错节点数量为f=(d–1)/(2+P)。其中d为组成员个数(节点总数)，P为节点既故障又恶意的概率。

证明　由于Raft共识算法只考虑故障节点，而PBFT共识算法既考虑故障节点又考虑恶意节点，故以Raft和PBFT为基准进行证明。Raft算法对由d个组成员组成的组提供f=(d–1)/2的容错能力，而PBFT算法对由d个组成员组成的组提供f=(d–1)/3的容错能力。因此，设定既故障又恶意节点的概率为P，可以得出既故障又恶意的节点数量为f=(d–1)/(2+P)。所以，当P=0时，满足Raft的容错能力。此时至少需要3个共识节点才能够保证区块链的可信性。也就是说，3名组成员就能够保证使用记录表的可信性。当P=1时，满足PBFT的容错能力，此时组成员个数至少为4。(其中“使用记录表的可信性”中“可信”指的是，通过使用记录表记录的内容是正确的，各个组成员记录的过程是可追溯的。“区块链的可信性”中“可信”指的是，区块链公开公认，具有不可抵赖和防篡改等特性)。　　证毕

5.4 组成员的权限平等

本文基于安全的Shamir秘密共享[17]进行说明。由于秘密值X通过插值多项式被分为了多个秘密份额，并分发送给了组内的各个成员。因此至少κ个组成员妥协于攻击者，攻击者才能够获取足够的秘密份额来生成X。假设共有100个组成员，门限值κ设为60，此时攻击者至少需要获取60个秘密份额来恢复出X，在计算上几乎不可行。因此本方案能够保证组成员的权限平等。

6 实验评估

实验在Ubuntu 16.04操作系统中实现，处理器为Intel Core i7 3.4 GHz，内存为8 GB。程序由PBC(Pairing-Based Cryptography) library中的库函数模拟密码学运算，Z_p^*中元素大小为$|p|$=160 bit。共享文件的大小为20 MB，分为100000块。采用以下符号来表示方案中的具体开销：Mul_{G_1},Mul_{Z_p}分别代表G_1,Z_p^*中的乘法运算时间，Exp_{G_1},Exp_{Z_p}分别代表G_1,Z_p^*中的指数运算时间，Add_{Z_p}代表Z_p^*中的加法运算时间，Hash_{G_1},Hash_{Z_p}分别代表G_1,Z_p^*中的哈希运算时间，Pair代表$e:G_1\times G_1\to G_2$的运算时间。

由于BLS(Boneh-Lynn-Shacham)签名技术具有计算轻便的特性，目前较为广泛的应用于轻量级PDP方案中，来降低认证标签生成开销和审计开销[14,15]。本文采用BLS签名技术计算文件块的认证标签，并与同样基于BLS签名的审计方案[14,15]进行对比，对比结果如表1所示。(分析文献[14,15]方案可知，被挑战数据块的数量c=460时，足以验证数据的完整性。因此，考虑到审计阶段的计算效率，测试审计过程时，选取的挑战文件块数为460个。)

文献[14,15]方案的主要创新点是高效地实现了本地撤销非法组成员。与之对比，本方案在没有增加额外审计开销的同时，在撤销组成员的方式上更加安全。下面通过(1)来说明SDRM能够高效地实现本地撤销非法组成员，通过(2)来说明SDRM支持数据拥有者高效更新数据。

(1) 撤销组成员阶段开销：在支持撤销组成员的共享数据审计方案[12–16]中，文献[12–15]更加关注

表 1　计算开销对比

方案	标签生成	审计阶段
文献[14]	$n(2\mathrm{Exp}_{G_1}+\mathrm{Mul}_{G_1}+\mathrm{Hash}_{G_1})$	$7\mathrm{Pair}+\mathrm{Mul}_{Z_p}+9\mathrm{Exp}_{G_1}+5\mathrm{Hash}_{Z_p}+3c(\mathrm{Mul}_{G_1}+\mathrm{Exp}_{G_1}+\mathrm{Mul}_{Z_p})=3.362$ s
文献[15]	$n(2\mathrm{Exp}_{G_1}+\mathrm{Mul}_{G_1}+\mathrm{Hash}_{G_1})$	$c\mathrm{Hash}_{G_1}+2\mathrm{Hash}_{Z_p}+(2c+2)\mathrm{Mul}_{G_1}+(2c+3)\mathrm{Exp}_{Z_p}+2\mathrm{Pair}+(c-1)\mathrm{Add}_{Z_p}+c\mathrm{Mul}_{Z_p}=3.213$ s
本文SDRM	$n(2\mathrm{Exp}_{G_1}+\mathrm{Mul}_{G_1}+\mathrm{Hash}_{G_1})$	$c(2\mathrm{Exp}_{G_1}+\mathrm{Hash}_{Z_p}+2\mathrm{Mul}_{G_1}+\mathrm{Mul}_{Z_p}+\mathrm{Add}_{Z_p})+2\mathrm{Pair}+\mathrm{Exp}_{Z_p}=3.202$ s

提高撤销组成员的效率问题。所以通过对比目前针对性更强的文献[12–15]方案，展现SDRM的高效性。文献[12,13]都采用重签名的方式，来达到撤销非法组成员的目的。相较于文献[12]，文献[13]采用外包算法提高了效率。其计算开销分别为λ(Pair+Exp_{G_1})和$\lambda\mathrm{Exp}_{G_1}$，随$\lambda$增加而线性增长($\lambda$是被撤销组成员签名的文件块数)。实验在批量文件块数量下进行，并选取横坐标为被撤销组成员签名的文件块数，从$\lambda=1$递增到$\lambda=5\times10^4$。如图4所示，$\lambda=1\times10^4$时，开销分别为29.5 s, 14.7 s，当λ增加到$\lambda=5\times10^4$时，开销分别为147.5 s, 73.5 s。

与文献[12,13]方案不同，文献[14,15]方案与本方案的撤销效率均与被撤销组成员签名的文件块数无关，更加稳定和高效。文献[14]采用验证者本地撤销组成员，每撤销一名非法组成员后，更新各合法组成员公钥A，再由TPA验证$e(A_iv^{\alpha}/A,u)=e(u^{\alpha},v)$，撤销开销主要来自TPA，约为10.2 ms,。文献[15]采用组管理者本地撤销组成员，每撤销一名组成员后，由组管理者更新撤销组成员的累加器RN=RN+1，开销约为0.007 ms。SDRM同样采用本地撤销组成员。每撤销一名组成员后，由数据拥有者U_k计算其第t次更新u_k后，所有u'_k的总和u_k^t，开销约为7.2 ms。

(2) 数据拥有者更新数据开销：目前较为新颖的共享数据审计方案中，NPP[16]支持多数据拥有者更新数据。NPP采用环签名的方法计算认证标签：(a) 组成员U_k随机选取$x_k\in Z_p^*$，计算$\omega_k=g^{x_k}\in G_1$。将U_k的私钥设为x_k，公钥设为ω_k。(b) 给定所有l个组成员的公钥$(\omega_1,\omega_2,\cdots,\omega_l)$、组成员$U_s$的私钥$x_s$和第$i$个文件块$m_i\in Z_p^*$，组成员$U_s$随机选取$\alpha_k\in Z_p^*$ $(1\le k\le l,k\ne s)$，然后计算$\beta=(H(i)\cdot g^{m_i})\in G_1$，最后计算$\sigma_s=\left(\dfrac{\beta}{\psi(\prod_{1\le k\le l,k\ne s}\omega_k^{\alpha_k})}\right)^{1/x_s}\in G_1$($\psi$为

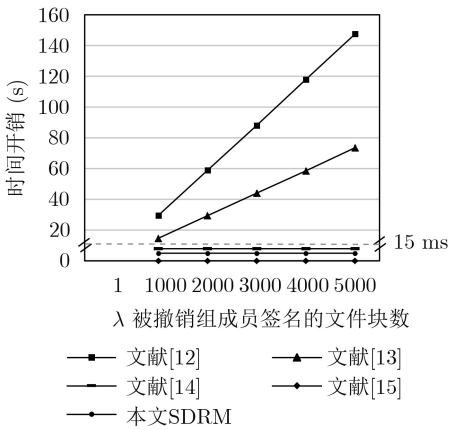

图 4 撤销组成员阶段开销

中间参数)。文件块m_i的环签名$\sigma=\{\sigma_1,\sigma_2,\cdots,\sigma_l\}$即为共享数据的认证标签。所以在NPP中，认证标签的计算开销与组成员数量l呈线性关系。当组内有d个数据拥有者时，开销就与dl正相关。因此随数据拥有者数量d增加，数据拥有者的更新数据开销呈指数级增长。与之相比，SDRM方案中，数据拥有者统一更新数据时，首先计算$S_g[F[k]]$，然后计算$u'_k=x_k+u_k=x_k+S_g[F[k]]$，开销为$n(\mathrm{Hash}_{G_1}+\mathrm{Mul}_{Z_p}+\mathrm{Add}_{Z_p})$。当组内$d$个数据拥有者时，需要计算$u_k^t=\sum_{k=1}^{d}(x_k+u_k)=\sum_{k=1}^{d}(x_k+S_g[F[k]])=\sum_{k=1}^{d}(x_k)+S_g\left(\sum_{k=1}^{d}F[k]\right)$，开销约为6 min。如图5所示实验结果显示，与现有方法NPP相比，SDRM明显缩短了更新数据时间。

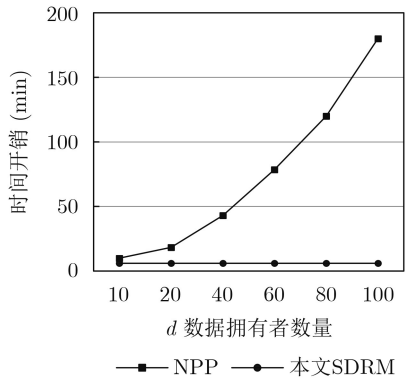

图 5 数据拥有者更新数据开销

7 结束语

本文提出了一种多方参与高效撤销组成员的共享数据公共审计方案SDRM。基于Shamir秘密共享和代数签名，SDRM可以相对公平的达到高效撤销组成员的目的，并且可以使多数据拥有者高效更新数据。此外，SDRM可以在保证安全性的前提下，验证共享数据的完整性。然而，本文所提到的区块链共识算法只有在满足容错能力下，才能保证使用记录表的可信性。如何进一步降低节点的出错率，是未来需要完成的工作。

参 考 文 献

[1] ATENIESE G, BURNS R, CURTMOLA R, *et al*. Provable data possession at untrusted stores[C]. The 14th ACM Conference on Computer and Communications Security, Alexandria, USA, 2007: 598–609. doi: 10.1145/1315245.1315318.

[2] ATENIESE G, DI PIETRO R, MANCINI L V, *et al*. Scalable and efficient provable data possession[C]. The 4th International Conference on Security and Privacy in Communication Networks, Istanbul, Turkey, 2008. doi:

10.1145/1460877.1460889.

[3] WANG Qian, WANG Cong, REN Kui, *et al*. Enabling public auditability and data dynamics for storage security in cloud computing[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2011, 22(5): 847–859. doi: 10.1109/TPDS.2010.183.

[4] BONEH D and SHACHAM H. Group signatures with verifier-local revocation[C]. The 11th ACM Conference on Computer and Communications Security, Washington, USA, 2004: 168–177. doi: 10.1145/1030083.1030106.

[5] WANG Boyang, LI Baochun, and LI Hui. Oruta: Privacy-preserving public auditing for shared data in the cloud[C]. The 5th IEEE International Conference on Cloud Computing, Honolulu, USA, 2012: 295–302. doi: 10.1109/CLOUD.2012.46.

[6] WORKU S G, XU Chunxiang, ZHAO Jining, *et al*. Secure and efficient privacy-preserving public auditing scheme for cloud storage[J]. *Computers & Electrical Engineering*, 2014, 40(5): 1703–1713. doi: 10.1016/j.compeleceng.2013.10.004.

[7] SHEN Wenting, YU Jia, XIA Hui, *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. doi: 10.1016/j.jnca.2017.01.015.

[8] 黄龙霞, 张功萱, 付安民. 基于层次树的动态群组隐私保护公开审计方案[J]. 计算机研究与发展, 2016, 53(10): 2334–2342. doi: 10.7544/issn1000-1239.2016.20160429.
HUANG Longxia, ZHANG Gongxuan, and FU Anmin. Privacy-preserving public auditing for dynamic group based on hierarchical tree[J]. *Journal of Computer Research and Development*, 2016, 53(10): 2334–2342. doi: 10.7544/issn1000-1239.2016.20160429.

[9] HUANG Longxia, ZHANG Gongxuan, and FU Anmin. Certificateless public verification scheme with privacy-preserving and message recovery for dynamic group[C]. Australasian Computer Science Week Multiconference, Geelong, Australia, 2017: 761–766. doi: 10.1145/3014812.3014890.

[10] PLANTARD T, SUSILO W, and ZHANG Zhenfei. Fully homomorphic encryption using hidden ideal lattice[J]. *IEEE Transactions on Information Forensics and Security*, 2013, 8(12): 2127–2137. doi: 10.1109/TIFS.2013.2287732.

[11] WANG Boyang, LI Baochun, and LI Hui. Panda: Public auditing for shared data with efficient user revocation in the cloud[J]. *IEEE Transactions on Services Computing*, 2015, 8(1): 92–106. doi: 10.1109/TSC.2013.2295611.

[12] YUAN Jiawei and YU Shucheng. Efficient public integrity checking for cloud data sharing with multi-user modification[C]. 2014 IEEE Conference on Computer Communications, Toronto, Canada, 2014: 2121–2129. doi: 10.1109/infocom.2014.6848154.

[13] LUO Yuchuan, XU Ming, HUANG Kai, *et al*. Efficient auditing for shared data in the cloud with secure user revocation and computations outsourcing[J]. *Computers & Security*, 2018, 73: 492–506. doi: 10.1016/j.cose.2017.12.004.

[14] JIANG Tao, CHEN Xiaofeng, and MA Jianfeng. Public integrity auditing for shared dynamic cloud data with group user revocation[J]. *IEEE Transactions on Computers*, 2016, 65(8): 2363–2373. doi: 10.1109/TC.2015.2389955.

[15] ZHAGN Yue, YU Jia, HAO Rong, *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, 17(3): 608–619. doi: 10.1109/TDSC.2018.2829880.

[16] FU Anmin, YU Shui, ZHANG Yuqing, *et al*. NPP: A new privacy-aware public auditing scheme for cloud data sharing with group users[J]. *IEEE Transactions on Big Data*, To be published. doi: 10.1109/TBDATA.2017.2701347.

[17] SHAMIR A. How to share a secret[J]. *Communications of the ACM*, 1979, 22(11): 612–613. doi: 10.1145/359168.359176.

[18] SCHWARZ T S J S and MILLER E L. Store, forget, and check: Using algebraic signatures to check remotely administered storage[C]. The 26th IEEE International Conference on Distributed Computing Systems, Lisboa, Portugal, 2006: 12. doi: 10.1109/ICDCS.2006.80.

[19] 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*, 2019, 16(1): 72–83. doi: 10.1109/TDSC.2017.2662216.

[20] 田俊峰, 李天乐. 基于TPA云联盟的数据完整性验证模型[J]. 通信学报, 2018, 39(8): 113–124. doi: 10.11959/j.issn.1000-436x.2018144.
TIAN Junfeng and LI Tianle. Data integrity verification based on model cloud federation of TPA[J]. *Journal on Communications*, 2018, 39(8): 113–124. doi: 10.11959/j.issn.1000-436x.2018144.

田俊峰：男，1965年生，博士，教授，研究方向为信息安全、分布式计算和网络技术.

井　宣：男，1994年生，硕士生，研究方向为信息安全、分布式计算和网络技术.