doi:10.11959/j.issn.1000-436x.2016260

面向云端群组数据的轻量级完整性验证方案

刘云飞[1]，王勇军[1]，付绍静[1,2]
（1. 国防科技大学计算机学院，湖南 长沙 410073；2. 密码科学技术国家重点实验室，北京 100878）

摘 要：为保护群组用户存储在云中的数据安全，设计了一个可以保护群组用户隐私的数据完整性验证方案。该方案可以高效地检测存储在云中的群组共享数据，并支持数据的动态更新，利用环签名的特性来隐藏数据块所对应的签名者的身份。也就是说，第三方验证者无法在验证时窥探到用户身份等隐私信息。通过使用聚合的方法生成数据标签，减少了标签的存储开销，同时支持群组数据的动态操作，使群组中的用户可以轻松的修改云中的群组数据。

关键词：云存储；数据完整性；身份隐私；聚合标签
中图分类号：TP309 **文献标识码**：A

Lightweight integrity verification scheme for cloud based group data

LIU Yun-fei[1], WANG Yong-jun[1], FU Shao-jing[1,2]
(1. College of Computer, National University of Defense Technology, Changsha 410073, China;
2. Sate Key Laboratory of Cryptology, Beijing 100878, China)

Abstract: In order to protect the security of the data stored in the cloud by group users, a data integrity verification scheme was designed which can protect the privacy of the group users. The scheme can efficiently detect the shared data in the cloud and support the dynamic updating of the data, and use the characteristic of the ring signature to hide the identity of the signer corresponding to the data block. That is, the third-party verifier can not spy on the users identity and other private information when validating. The aggregated approach is used to generate data labels, which reduces the storage cost of labels and supports the dynamic operation of group data, so that the users in the group can easily modify the cloud group data.

Key words: cloud storage, data integrity, identity privacy, aggregated labels

1 引言

随着云存储技术的发展，用户在享受云端数据服务所提供便利的同时，云端数据的外包和共享为保障用户数据安全带来了诸多新的挑战。由于用户在本地不保留任何数据副本，而云服务器又不是完全可信的，所以存储在云端的数据完整性不能得到保证。如何高效地验证云端数据的完整性成为亟待解决的问题。

云存储中群组共享数据的情况给数据完整性的验证带来了新的挑战。与单个用户不同的是，群组用户一个显著的特点就是各用户在群组的地位和作用不同，可能是原始用户也可能是普通用户，对数据使用的情况也不尽相同，这就涉及用户身份隐私的问题。由于存储在云上的不同数据块可能会由不同的用户计算数据标签，而在完整性验证过程中又需要用到该标签对应用户的公钥，因此，对于第三方验证者而言，在使用用户公钥验证数据完整

收稿日期：2016-09-01
基金项目：国家自然科学基金资助项目（No.61572026,No.61472439,No.61379052）；密码科学技术国家重点实验室开放课题基金资助项目；湖南省创新平台开放基金资助项目（No.13K025（2014-2016））
Foundation Items: The National Natural Science Foundation of China (No.61572026, No.61472439, No.61379052), The Open Foundation of State Key Laboratory of Cryptology, The Open Fund Project of Innovation Platform for Universities in Hunan Province (No.13K025(2014-2016))

性的过程中就有可能推测出某些数据块对群组中的某位用户来说更为重要，也可能会根据用户计算数据标签的数量来推测该用户在群组中的地位。这样的隐私信息泄露无疑也是很危险的，在验证云端群组共享数据完整性的过程中尽可能保护用户的身份隐私信息成为一个重要的问题。

为验证云服务器中数据的完整性，Ateniese 等[1]于 2007 年提出了可证明数据拥有方案。该方案引入了公开的第三方验证者，可以在不下载全部数据的情况下利用用户的公钥验证用户存储在云服务器上的数据完整性，但该方案不支持数据的动态更新（用户需要更新数据时，新的数据块及其对应的标签都要重新计算）。此外，Ateniese 等[2]对所有线性的同态标签构造可证明数据拥有方案的可行性进行了证明。Shacham 等[3]利用 BLS（Boneh-Lynn-Shacham）签名方案设计了一个数据拥有性证明方案[4]，BLS 是利用双线性映射构造的数字签名方案，比一般的签名方案长度短，更能节省计算和存储的开销。Ateniese 等[5]提出了一种基于对称密钥的可证明数据拥有方案，可以在对数据完整性进行验证的同时支持数据的动态更新，但由于签名和验证使用的是对称密钥，用于验证的密钥是不能泄露的，所以该方案不支持第三方公开验证，而且验证次数会受到对称密钥数量的限制。为了解决在公开验证模型下支持数据动态更新的问题，Wang 等[6]在利用 BLS 签名[4]的同时，与 Markle 散列树相结合设计了一个数据拥有性证明的公开验证方案。Erway 等[7]的数据拥有性方案则是基于认证字典的，该方案也能够支持数据更新。为了减小公开验证方案中存储验证信息的开销，Zhu 等[8, 9]利用散列索引表和分离结构的方法提出了一种公开验证方案，其中，散列索引表起到支持数据更新的作用。Wang 等[10,11]提出了一种公开验证方案，可以有效保护数据的隐私性，即验证者在完成数据完整性验证时无法获知数据的内容。该方案为了支持对多个任务的同时验证还引入了聚合签名[12]的方法，有效地降低了验证过程中的计算开销。Zhu 等[13]则对数据完整性验证技术在混合云中的应用进行了讨论。针对多用户或群组用户共享数据的完整性验证问题，Tate 等[14]基于可信硬件提出了一个数据完整性验证方案，可以支持数据更新操作。Wang 等设计了 Oruta[15]、Knox[16]以及一种可以撤销用户权限的可证明数据拥有性证明方案[17]。Oruta 是基于环签名方案设计的，可以保护用户的隐私，存在的不足是群组用户的数量增加会导致验证信息和计算时间的增长。Knox 方案虽然克服了环签名导致的不足，但它所使用的群签名方案比较复杂，导致标签的计算和存储开销较大。

本文设计了一个保护用户身份隐私的数据完整性验证方案。在高效的验证云存储中数据完整性的同时保护用户的身份隐私安全，为了更好地适应在移动终端上使用，在构造数据标签的过程中使用聚合的方法，并改进了聚合标签的生成方式，减少了标签的存储开销，同时还支持群组数据的动态更新操作。

2　预备知识

2.1　双线性映射

假设 G_1 和 G_2 是 2 个阶为大素数 p 的循环群，g 为群 G_1 的生成元，$e:G_1\times G_1\to G_2$ 为双线性映射，e 满足以下 3 个性质。

1) 可计算性：存在一个有效的算法 $e(u,v)$，其中，$u,v\in G_1$。

2) 双线性：对任意的 $u,v\in G_1$ 和任意的 $a,b\in Z_p^*$，满足 $e(u^a,v^b)=e(u,v)^{ab}$。

3) 非退化性：$e(g,g)\neq 1$。

2.2　离散对数假设

离散对数问题（DLP，discrete logarithm problem），设 $a\in Z_p^*$，已知 $g,g^a\in G_1$，求 a。

定义离散对数假设（discrete logarithm assumption），对于一个攻击者 A，该攻击者在多项式时间内只能以可以忽略的优势求解出离散对数问题，具体表示为

$$\Pr[A(g,g^a)=(a):a\xleftarrow{R}Z_p^*]\leqslant\varepsilon$$

其中，ε 表示为一个可以忽略的优势。

简单地说，就是在离散对数假设下，计算离散对数问题是计算上不可行或是计算困难的。

3　验证模型

同 Ateniese 等提出的数据拥有性证明模型类似，该方案的验证模型包括 3 个部分，即：群组用户、云服务器和第三方验证者，如图 1 所示。群组用户包括原始用户和一些普通群组用户。原始用户负责将原始数据上传到云服务器上并与普通用户共享这些数据。原始用户和普通用户都是群组中的成员。群组中的各个成员都可以访问并使用在云服

务器上的数据，如果对某些数据块做过修改，该成员就需要重新对这些数据块进行签名并将数据块和签名一起上传到云服务器上。第三方验证者是一个专业提供云端数据完整性验证服务的第三方机构。当用户需要验证自己存储在云中的数据的完整性时，首先要向第三方验证者发送数据验证协商信息，第三方验证者收到用户的验证请求之后，就会向云服务器发送一个挑战（即验证询问）。云服务器收到挑战之后 ，根据挑战信息计算一个验证证明（即示证响应）并将它发送给第三方验证者。验证者检测验证证明的正确性，如果正确就说明存储在云端的数据是完整的，否则，就说明数据的完整性可能遭到了破坏。

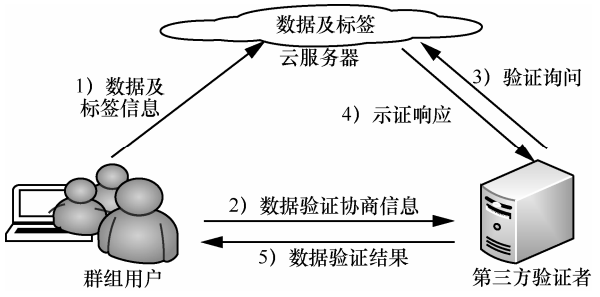

图 1 验证模型

4 保护用户隐私的轻量级云端群组数据完整性验证方案

4.1 方案构造

在验证环签名时，验证者无法区分签名者具体是环中的哪一位成员。因此利用环签名的这一特性来计算数据标签，从而达到隐藏用户身份信息的目的。方案主要由 5 个步骤组成，分别是 GenKey、TagBlock、GenChal、GenProof 和 CheckProof。内容如下。

1) GenKey（生成密钥）：用户端运行的一个用以初始化方案的多项式时间算法。它输入一个安全参数，输出一对用户的公钥和私钥。

2) TagBlock（生成标签）：用户端运行的一个多项式时间算法。用户利用自己的私钥和其他用户的公钥为数据块计算签名，生成数据块的标签。

3) GenChal（生成挑战）：验证者将抽样的数据块与对应的安全参数作为验证挑战发送给云服务器。

4) GenProof（生成证明）：收到验证者发来的挑战后，云服务器生成一个验证证明，证明它对数据的拥有性。该证明包含公钥、数据块的有序集合、验证挑战和标签的有序集合。

5) CheckProof（检验证明）：验证者对云服务器生成的验证证明进行正确性验证。输入相应的公私钥、验证挑战和验证证明，如果数据拥有证明是数据完整性的正确证明，该算法返回 success，否则，返回 failure。

具体方案如下。

设 G_1、G_2、G_T 是阶为 p 的乘法循环群。g_1、g_2 分别是群 G_1、G_2 的生成元。映射 $e:G_1\times G_2\to G_T$ 为双线性映射，$\psi:G_2\to G_1$ 是同构运算，$\psi(G_2)=G_1$。$H_1:\{0,1\}^*\to G_1$，$H_2:\{0,1\}^*\to Z_q$ 和 $h:G_1\to Z_p$ 为 3 个散列函数。群组中用户的总数为 d，共享数据 M 被分为 n 块，$M=(m_1,\cdots,m_n)$，每一个数据块 m_j 又被进一步分为 Z_p 中的 k 个元素，$m_j=(m_{j,1},\cdots,m_{j,k})$。

1) 生在密钥及数据标签

GenKey：用户 u_i 随机选择一个元素 $x_i\in Z_p$，计算 $\lambda_i=g_2^{x_i}$，用户 u_i 的公钥是 $pk_i=\lambda_i$，私钥是 $sk_i=x_i$，原始用户随机生成公共聚合密钥 $pak=(\xi_1,\cdots,\xi_k)$，其中，$\xi_i=g_1^{\alpha_i},\alpha\in Z_p$。

TagBlock：令 $(pk_1,\cdots,pk_d)=(\lambda_1,\cdots,\lambda_d)$ 为 d 个群组用户的公钥，对任一数据块 $m_j=(m_{j,1},\cdots,m_{j,k})$，用户 u_s 的私钥为 sk_s，u_s 利用如下方法计算环签名。

① 利用公共聚合密钥 pak 将数据块 m_j 聚合为 $\prod_{l=1}^{k}\xi_l^{m_{j,l}}$，并且计算

$$\beta_j=H_1(\omega_j)\prod_{l=1}^{k}\xi_l^{m_{j,l}}\in G_1,\omega_j=H_2(m_j)\|V_j$$

其中，$V_j=nj$ 是数据块 m_j 的初始虚拟索引值，$n\in N^*$ 是用户确定的系统参数，ω_j 为数据块 m_j 的标识符。

② 随机选择 $a_{j,i}\in Z_p$，对任意 $i\neq s$ 令 $\sigma_{j,i}=g_1^{a_{j,i}}$，然后计算

$$\sigma_{j,s}=\left(\frac{\beta_j}{\psi(\prod_{i\neq s}\lambda_i^{\alpha_{j,i}})}\right)^{\frac{1}{x_s}}\in G_1$$

则数据块 m_j 的标签就是 $\sigma_j=(\sigma_{j,1},\cdots,\sigma_{j,d})$。

2) 挑战应答

GenChal：每进行一次数据完整性验证，验证者都要随机地从物理索引值中选取 c 个值组成一个集合 $J=\{s_1,\cdots,s_c\}$，这里假定 $s_1\leqslant\cdots\leqslant s_c$。对于集合 $J=\{s_1,\cdots,s_c\}$ 中的每个元素 s_j，验证者随机选择一个值 $y_j\in Z_q$ 与之对应，其中，$j\in J$，随后验证

者将 $\{(j,y_j)\}_{j\in J}$ 作为验证挑战发给云服务器。

GenProof：在接到验证者发送的挑战请求 $\{(j,y_j)\}_{j\in J}$ 后，云服务器生成相应的示证响应以向验证者证明抽样数据块的完整性。

① 选择一个随机数 $\tau_l\in Z_q$，并且计算 $\delta_l=\xi_l^{\tau_l}\in G_1$，$l\in[1,k]$。

② 计算 $\mu_l=\sum_{j\in J}y_jm_{j,l}+\tau_l h(\delta_l)\in Z_P, l\in[1,k]$。

③ 聚合签名 $\Phi_i=\prod_{j\in J}\sigma_{j,i}^{y_j}$，$i\in[1,d]$。

④ 将示证响应 $\{\delta,\mu,\Phi,\{\omega_j\}_{j\in J}\}$ 返回给验证者。

其中，$\delta=(\delta_1,\cdots,\delta_k)$，$\mu=(\mu_1,\cdots,\mu_k)$，$\Phi=(\Phi_1,\cdots,\Phi_d)$。

3) 证据检验

CheckProof：根据示证响应 $\{\delta,\mu,\Phi,\{\omega_j\}_{j\in J}\}$，验证挑战 $\{(j,y_j)\}_{j\in J}$，公共聚合密钥 $pak=(\xi_1,\cdots,\xi_k)$，和所有群组用户的公钥 $(pk_1,\cdots,pk_d)=(\lambda_1,\cdots,\lambda_d)$，验证者验证下面的等式

$$e(\prod\nolimits_{j\in J}H_1(\omega_j)^{y_j}\prod\nolimits_{l=1}^{k}\xi_l^{\mu_l},g_2)\stackrel{?}{=}(\prod\nolimits_{i=1}^{d}e(\Phi_i,\lambda_i))e(\prod\nolimits_{l=1}^{k}\delta_l^{h(\delta_l)},g_2)\tag{1}$$

如果等式成立，则可以认定在云服务器上存储的数据是完整的；如果不成立，则数据可能已遭到篡改或损坏。

4) 数据更新

修改（modification）：当用户修改第 j 个数据块 m_j 之后，数据块 m_j 变为 m_j'，此时 m_j' 的虚拟索引值不变，仍然是 V_j，通过计算 $\omega_j'=H_2(m_j')\|V_j$ 得到新的 ω_j'，用户根据 ω_j' 和 m_j' 生成新的标签信息 σ_j'。用户将 $\{m_j',\omega_j',\sigma_j'\}$ 上传到云服务器。其他数据块及相应的标签信息保持不变，云端共享数据块的总数仍然为 n。

插入（insertion）：用户将新的数据块 m_j' 插入到共享数据中，假定此时 m_j' 插入到了 m_{j-1} 和 m_j 之间，用户首先需要计算新数据块 m_j' 的虚拟索引值 $V_j'=\frac{V_{j-1}+V_j}{2}$，并计算 $\omega_j'=H_2(m_j')\|V_j'$，然后用户为新插入的数据块 m_j' 计算新的标签信息 σ_j'，并将 $\{m_j',\omega_j',\sigma_j'\}$ 上传到云服务器。其他数据块及标签信息保持不变，共享数据块的总数加 1。

删除（deletion）：用户若需要删除存储在云服务器上的数据块 m_j 时，其相应的标识符 ω_j 和标签信息 σ_j 也要随之从云服务器上删除，其他共享数据块的内容和标签信息仍然保持不变，共享数据块的总数减 1。

如图 2 所示，左侧为原始共享数据块，右侧为分别进行了修改、插入、删除操作后的共享数据块。

4.2 安全性分析

1) 正确性分析

第三方验证者可以正确验证存储在云服务器中的数据完整性。

证明 由于双线性映射的特殊性质，式(1)可以按以下方式进行计算验证。

$$(\prod_{i=1}^{d}e(\Phi_i,\lambda_i))e(\prod_{l=1}^{k}\delta_l^{h(\delta_l)},g_2)$$
$$=(\prod_{i=1}^{d}e(\prod_{j\in J}\sigma_{j,i}^{y_j},\lambda_i))e(\prod_{l=1}^{k}\delta_l^{h(\delta_l)},g_2)$$

索引	数据块	虚拟索引	标识符
1	m_1	V_1	ω_1
2	m_2	V_2	ω_2
3	m_3	V_3	ω_3
4	m_4	V_4	ω_4
5	m_5	V_5	ω_5
6	m_6	V_6	ω_6
7	m_7	V_7	ω_7
8	m_8	V_8	ω_8
⋮	⋮	⋮	⋮
n	m_n	V_n	ω_n

修改m_2

插入m'_4

删除m_6

索引	数据块	虚拟索引	标识符
1	m_1	V_1	ω_1
2	m'_2	V_2	ω'_2
3	m_3	V_3	ω_3
4	m_4	$\frac{V_3+V_4}{2}$	ω_4
5	m_5	V_4	ω_5
6	m_6	V_5	ω_6
7	m_7	V_7	ω_7
8	m_8	V_8	ω_8
9	m_9	V_9	ω_9
⋮	⋮	⋮	⋮ ⋮
n	m_n	V_n	ω_n

图 2　分别对共享数据块进行修改、插入、删除操作

$$=(\prod_{j\in J}(\prod_{i=1}^{d}e(\sigma_{j,i},\lambda_i)^{y_j}))e(\prod_{l=1}^{k}\xi_l^{\tau_l h(\delta_l)},g_2)$$

$$=(\prod_{j\in J}e(\beta_j,g_2)^{y_j})e(\prod_{l=1}^{k}\xi_l^{\tau_l h(\delta_l)},g_2)$$

$$=e(\prod_{j\in J}(H_1(\omega_j)\prod_{l=1}^{k}\xi_l^{m_{j,l}})^{y_j},g_2)e(\prod_{l=1}^{k}\xi_l^{\tau_l h(\delta_l)},g_2)$$

$$=e(\prod_{j\in J}H_1(\omega_j)^{y_j}\prod_{l=1}^{k}\xi_l^{\sum_{j\in J}m_{j,l}\cdot y_j}\prod_{l=1}^{k}\xi_l^{\tau_l h(\delta_l)},g_2)$$

$$=e(\prod_{j\in J}H_1(\omega_j)^{y_j}\prod_{l=1}^{k}\xi_l^{\mu_l},g_2)$$

2) 不可伪造性分析

如果离散对数假设成立，云服务器想要利用伪造的验证证明通过验证者的验证在计算上是困难的。

证明 如果一个不可信的云服务器想要在数据不完整的情况下成功伪造一个验证证明，就必须赢得下面的安全游戏。

第三方验证者在验证数据完整性时将数据 $\{(j,y_j)\}_{j\in J}$ 作为验证挑战发送给云服务器，如果云服务器上存储的数据都是正确的完整的，那么云服务器就会通过正确的数据 M 生成一个验证证明 $\{\delta,\mu,\Phi,\{\omega_j\}_{j\in J}\}$，显然这个验证证明可以通过式(1)的验证。如果云服务器上的数据是被篡改过的或不完整的，那么云服务器只能通过不正确的数据 M' 伪造一个验证证明 $\{\delta,\mu',\Phi,\{\omega_j\}_{j\in J}\}$，其中，$\mu'=(\mu_1',\cdots,\mu_k')$，$\mu_l'=\sum_{j\in J}y_j m_{j,l}'+\tau_l h(\delta_l)\in Z_P$。设 $\Delta\mu_l=\mu_l'-\mu_l$，$1\leqslant l\leqslant k$，因为 $M\neq M'$，所以 $\{\Delta\mu_l\}_{1\leqslant l\leqslant k}$ 中的元素至少有一个不为零。如果这个伪造的验证证明能够通过式(1)的验证，就判断云服务器赢得了这个安全游戏，整个数据完整性验证过程是无效的，无法判定云服务器上数据的完整性。否则，就认为云服务器输掉了游戏，验证者能够正确验证数据的完整性而不会被云服务器所欺骗。

首先假设云服务器赢得了该安全游戏，那么根据式(1)，有

$$e(\prod_{j\in J}H_1(\omega_j)^{y_j}\prod_{l=1}^{k}\xi_l^{\mu_l'},g_2)$$

$$=(\prod_{i=1}^{d}e(\Phi_i,\lambda_i))e(\prod_{l=1}^{k}\delta_l^{h(\delta_l)},g_2)$$

由于 $\{\delta,\mu,\Phi,\{\omega_j\}_{j\in J}\}$ 是正确的验证证明，可以得知

$$e(\prod_{j\in J}H_1(\omega_j)^{y_j}\prod_{l=1}^{k}\xi_l^{\mu_l},g_2)$$

$$=(\prod_{i=1}^{d}e(\Phi_i,\lambda_i))e(\prod_{l=1}^{k}\delta_l^{h(\delta_l)},g_2)$$

还可以进一步得知

$$\prod_{l=1}^{k}\xi_l^{\mu_l'}=\prod_{l=1}^{k}\xi_l^{\mu_l},\prod_{l=1}^{k}\xi_l^{\Delta\mu_l}=1$$

$g,h\in G_1$ 是 2 个随机的元素，根据循环群的性质可知存在一个 $x\in Z_p$ 使 $h=g^x$。一般地，已知 $g,h\in G_1$，每一个 ξ_l 可以随机的按照 $\xi_l=g^{\eta_l}\cdot h^{\gamma_l}\in G_1$ 计算生成，η_l 和 γ_l 是 Z_p 中的随机元素。然后，有

$$\prod_{l=1}^{k}\xi_l^{\Delta\mu_l}=\prod_{l=1}^{k}(g^{\eta_l}h^{\gamma_l})^{\Delta\mu_l}=g^{\sum_{l=1}^{k}\eta_l\Delta\mu_l}h^{\sum_{l=1}^{k}\gamma_l\Delta\mu_l}=1$$

对于已知的 $g,h^x\in G_1$，可以进一步得出

$$h=g^{\frac{\sum_{l=1}^{k}\eta_l\Delta\mu_l}{\sum_{l=1}^{k}\gamma_l\Delta\mu_l}}=g^x,x=\frac{\sum_{l=1}^{k}\eta_l\Delta\mu_l}{\sum_{l=1}^{k}\gamma_l\Delta\mu_l}$$

如果 x 无解，除非分母 $\sum_{l=1}^{k}\gamma_l\Delta\mu$ 为零。但是根据上述安全游戏中的定义，$\{\Delta\mu_l\}_{1\leqslant l\leqslant k}$ 中的元素不全为零，又因为 γ_l 是从 Z_p 中随机选取的一个元素，可知 γ_l 为零的概率为 $\frac{1}{p}$，所以 $\frac{\sum_{l=1}^{k}\eta_l\Delta\mu_l}{\sum_{l=1}^{k}\gamma_l\Delta\mu_l}$ 中分母为零的概率也是 $\frac{1}{p}$。因此，x 有解的概率为 $1-\frac{1}{p}$。即如果云服务器赢得了安全游戏，那么能在 G_1 中求解出 DL 问题的概率为 $1-\frac{1}{p}$，由于 p 是一个大素数，所以这个概率是不可忽略的。这与 DL 假设是相矛盾的。因此，云服务器想要利用伪造的验证证明通过验证者的验证在计算上是困难的。

3) 隐私性分析

假设群组用户中用户的数量为 d，在进行一次完整性验证时，验证者抽取的数据块的数量为 c，则验

证者最多能以$\frac{1}{d^c}$的概率区分所有用户的身份。

证明　设 A 是一个攻击者，它能推测出群组中任意用户u_s身份的概率为$\Pr[A(\sigma)=u_s]$，其中，σ是利用用户u_s $(1\leqslant s\leqslant d)$的私钥$sk_s$和其他用户公钥计算生成的环签名。对于任意$h\in G_1$，已知$\{g_1^{a_1},\cdots,g_1^{a_d}:a_i\in Z_p\}$，其中，$i\neq s$，选取$a_s$使$\{\prod_{i=1}^{d}g_1^{a_i}=h\}$的分布与$\{g_1^{a_1},\cdots,g_1^{a_d}:\prod_{i=1}^{d}g^{a_i}=h\}$的分布是一样的。因此，已知$\sigma=(\sigma_1,\cdots,\sigma_d)$，攻击者 A 能够分辨出$\sigma_s(1\leqslant s\leqslant d)$的概率最多不超过$\frac{1}{d}$。由此可知，对一个群组用户签名的数据，若存在一个攻击者 A，则 A 最多只能以不超过$\frac{1}{d}$的概率区分出该数据块的签名者在群组中的身份信息。在数据完整性验证中，通常会随机选择c个独立的数据块，其中，$c\in[1,n]$，因此，第三方验证者在验证过程中最多能够以$\frac{1}{d^c}$的概率区分所有数据块签名所对应签名者的身份信息。

5　开销分析

5.1　计算开销

云服务器收到验证者发来的验证挑战之后，会根据验证挑战计算一个验证证明$\{\delta,\mu,\Phi,\{\omega_j\}_{j\in J}\}$。计算该证明所需要的计算开销为$(dc+1)Exp_{G_1}+dcMul_{G_1}+ckMul_{Z_p}+kHash_{Z_p}$，为判断验证证明$\{\delta,\mu,\Phi,\{\omega_j\}_{j\in J}\}$是否正确，验证者通过式(1)进行验证。该过程所涉及的总共的运算开销为$(c+2)Exp_{G_1}+(c+2)Mul_{G_1}+dMul_{G_T}+cHash_{G_1}+(d+2)Pair$，相关符号描述如表 1 所示。

表 1　完整性验证过程中的计算开销所对应的操作

符号	操作
Exp_{G_1}	在G_1中计算一个幂指数运算
Mul_{G_1}或Mul_{G_T}	在G_1或G_T中计算一个乘法运算
Mul_{Z_p}	在Z_p中计算一个乘法运算
$Hash_{Z_p}$	在Z_p中计算一个散列运算
$Pair$	在$e:G_1\times G_2\to G_T$计算一个线性对运算

5.2　通信开销

进行一次数据完整性验证，云服务器和第三方验证者要进行两次通信，分别是验证者向服务器发送验证挑战和服务器向验证者发送验证证明。在这两次通信中，验证挑战$\{(j,y_j)\}_{j\in J}$产生的通信开销为$c(|q|+|n|)$ bit，验证证明$\{\delta,\mu,\Phi,\{\omega_j\}_{j\in J}\}$产生的通信开销为$(2k+d)|p|+c|q|$ bit，其中，$|p|$表示群G_1阶的大小，$|q|$表示Z_q阶的大小，$|n|$表示一个索引的大小。

6　结束语

本文针对群组用户在对云存储中的共享数据进行完整性验证时身份隐私的保护问题设计了一个数据完整性验证方案。该方案基于可信第三方进行公开验证，利用环签名的特殊属性计算同态可验证的数据标签，验证者在验证过程中可以确定数据标签的正确性，但无法揭示该标签对应的用户身份，有效地保护了用户的隐私。

参考文献：

[1] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//ACM CCS. 2007:598-610.

[2] ATENIESE G, KAMARA S, KATZ J. Proofs of storage from homomorphic identification protocols[C]//ASIACRYPT 2009: 319-333.

[3] BONEH D, LYNN B, SHACHAM H. Short signature from the weil pairing[C]//ASIACRYPT 2001. 2001: 514-532.

[4] SHACHAM H, WATERS B. Compact proofs of retrievability[C]//ASIACRYPT 2008. 2008: 90-107.

[5] ATENIESE G, PIETRO R D, MANCINI L V, et al. Scalable and efficient provable data possession[C]//4th ICST SecureComm. 2008.

[6] WANG Q, WANG C, LI J, et al. Enabling public verifiability and data dynamic for storage security in cloud computing[C]//ESORICS 2009. 2009: 355-370.

[7] ERWAY C, KUPCU A, PAPAMANTHOU C, et al. Dynamic provable data possession[C]//Proceedings of ACM CCS. 2009: 213-222.

[8] ZHU Y, WANG H, HU Z, et al. Dynamic audit services for integrity verification of outsourced storage in cloud[C]//ACM SAC. 2011: 1550-1557.

[9] ZHU Y, AHN G J, HU H, et al. Dynamic audit services for outsourced storage in clouds[C]//IEEE Transactions on Services Computing, 2013, 6(99): 1.

[10] WANG C, WANG Q, REN K, et al. Privacy-preserving public auditing for data storage security in cloud computing[C]//IEEE INFOCOM. 2010: 525-533.

[11] WANG C, CHOW S S, WANG Q, et al. Privacy-preserving public auditing for secure cloud storage[J]. IEEE Transactions on Computers, 2013, 62(2): 362-375.

[12] BONEH D, GENTRY C, LYNN B, et al. Aggregate and verifiably encrypted signatures from bilinear maps[C]//EUROCRYPT 2003. 2003: 416-432.

[13] ZHU Y, WANG H, HU Z, et al. Poster: efficient provable data possession for hybird clouds[C]//ACM CCS. 2010.

[14] TATE S R, VISHWANATHAN R, EVERHART L. Multi-user dynamic proofs of data possession using trusted hardware[C]//ACM CODASPY. 2013: 353-364.

[15] WANG B, LI B, LI H. Oruta: privacy-preserving public auditing for shared data in the cloud[C]//IEEE 5th International Conference on Cloud Computing (CLOUD'12). Piscataway, NJ: IEEE, 2012: 295-302.

[16] WANG B, LI B, LI H. Knox: privacy-preserving auditing for shared data with large groups in the cloud[C]//10th International Conference on Applied Cryptography and Network Security (ACNS'12). 2012: 507-525.

[17] WANG B, LI B, LI H. Public auditing for shared data with efficient user revocation in the cloud[C]//2013 IEEE International Conference on Computer Communication (INFOCOM'13). Los Alamitos, 2013: 2904-2912.

作者简介：

刘云飞（1985–），男，河南睢县人，国防科技大学硕士生，主要研究方向为网络安全。

王勇军（1971–），男，江西高安人，博士，国防科技大学教授，主要研究方向为网络安全、系统安全等。

付绍静(1984–)，男，江西玉山人，博士，国防科技大学副教授，主要研究方向为密码学。