DOI:10.16515/j.cnki.32-1722/n.2019.01.001

高效的云端群组数据完整性验证

阎 浩,柳亚男,邱 硕,张 正

(金陵科技学院网络安全学院,江苏 南京 211169)

摘 要:云存储服务为群组用户提供了一种高效共享数据的模式。为保证云端群组数据的完整性,提出一个基于无证书的云端群组共享数据完整性验证方案(CLG-PDP)。CLG-PDP方案具有两个主要特点:一是实现了对群组内任意成员生成的共享数据的完整性公开验证;二是避免了PKI中的证书管理问题和基于身份密码机制中的密钥托管问题。理论分析和仿真实验结果表明CLG-PDP方案是高效可用的。

关键词:云存储;数据共享;完整性验证;无证书密码体制;安全性;高效

中图分类号:TP309 **文献标识码**:A **文章编号**:1672-755X(2019)01-0001-05

An Efficient Public Integrity Checking Scheme for Shared Group Data on Cloud Storage

YAN Hao, LIU Ya-nan, QIU Shuo, ZHANG Zheng

(Jinling Institute of Technology, Nanjing 211169, China)

Abstract: Cloud storage service supplies with an efficient method to share data within a group. To ensure the integrity of data shared on cloud storage, the paper proposes a certificateless provable data possession scheme(CLG-PDP). CLG-PDP has two main characters: first, it realizes the public integrity checking for the group data no matter which group member is the data generator; second, it eliminates the problems of certificate management in PKI and the key escrow in identity-based cryptography. Theoretical analysis and experiment results exhibit that CLG-PDP is efficient and feasible.

Key words: cloud storage; shared data; integrity verification; certificateless cryptography; security; efficient

云存储技术的发展及应用普及,为用户提供了一种高效的数据共享模式,使群组用户协同工作变得更加容易。但在云存储中,服务提供商(Cloud Storage Provider,CSP)不是完全可信的,不能保证用户数据的安全存储[1]。例如,因软硬件故障导致云数据丢失的事件时有发生,甚至部分CSP将不常访问的云数据移至线下存储或者删除以节省存储空间。为了维护自身信誉和逃避赔偿,CSP不会主动将数据被破坏的事实告知数据拥有者。因此,为数据拥有者提供一种检查数据完整性的机制是十分重要的。

2007年,Ateniese等提出一种可证明数据持有(Provable Data Possession,PDP)模型[2]。基于概率检测的思想,PDP通过检测目标数据中随机抽取的部分数据块的正确性,以极高的概率获知目标数据整体的完整性。基于这种思路,许多学者和研究人员提出了多种安全性和性能更高的云数据完整性验证PDP

收稿日期:2019-03-02

基金项目:江苏省高等学校自然科学研究面上项目(17KJD520003);金陵科技学院高层次人才科研启动基金(jit-b-201639,jit-b-201726);网络安全专项项目(2017YFB0802800)

作者简介:阎浩(1980—),男,江苏沛县人,副教授,硕士,主要从事应用密码学、云安全技术研究。

模型[3-7]。但其中大多数模型仅仅面向个人用户数据，不能解决群组共享数据的完整性验证问题。与个人用户数据相比，群组共享数据的完整性验证存在更多挑战。群组内所有成员都可以为共享数据生成验证标签，不同成员生成的验证标签必然存在差异。特别是当数据更新的时候，重新生成新标签是必然的。当验证数据完整性时，所有相关标签都可能被抽测到，必然涉及这些标签的聚合问题，以及标签生成者的判定问题，增加了数据完整性验证的复杂度。

为解决群组共享数据的完整性验证问题，Wang 等基于群签名技术提出一个支持群组数据完整性验证的 PDP 方案[8]。方案将所有成员看成一个群，每个群成员使用群签名技术为数据生成验证标签。在验证完整性时借助群签名验证技术实现对标签的合法性验证，从而验证被检测数据的完整性。但该方案在标签生成和验证阶段需要较大的计算代价，效率较低。为提高性能，Wang 等基于代理重签名技术提出了一个支持群组数据完整性验证的 PDP 方案[9]，并解决了群组成员的撤销问题。但是该方案的安全性证明不够严谨，方案的安全性存在疑问。Yu 等也提出了一个群组数据完整性验证方案[10]，其中考虑了群组的动态性。Yuan 等基于多项式认证标签提出一个多用户场景下的数据完整性验证方案[11]。但这两个方案存在密钥托管的安全问题。前述方案都是基于传统 PKI 技术实现的，因此数字证书的管理是方案的巨大负担，使方案的应用受到较大限制。为解决以上问题，本文受文献[12]的启发提出了一个基于无证书密码机制的云端群组数据的完整性验证方案(CLG-PDP)。

1 预备知识

1.1 双线性映射

$\mathbb{G}_1$ 和 $\mathbb{G}_2$ 是两个乘法循环群，阶为大素数 q。g 是群 $\mathbb{G}_1$ 的生成元。$e:\mathbb{G}_1\times\mathbb{G}_1\rightarrow\mathbb{G}_2$ 是一个双线性映射，且满足如下性质。1)可计算性：对于 $\forall u,v\in\mathbb{G}_1$，$e(u,v)$ 的结果是可以高效计算的。2)双线性：对于 $\forall u,v\in\mathbb{G}_1$，$\forall a,b\in Z_q^*$，有 $e(u^a,v^b)=e(u,v)^{ab}$。3)非退化性：$\exists u,v\in\mathbb{G}_1$ 满足 $e(u,v)\neq 1_{\mathbb{G}_2}$。

1.2 系统模型

CLG-PDP 方案的系统模型包含 3 个实体：用户群组、云服务提供商(CSP)和验证者。

1)用户群组。其中包含多个成员用户，群组内的每个用户均享有平等的数据生成、访问、更新以及删除的权限。群组拥有一个群组管理员，负责群组的建立及群组成员管理工作。同时群组管理员扮演无证书密码体系中的密钥生成中心(KGC)角色，负责为每个成员生成部分私钥。

2)云存储提供商(CSP)。为用户提供数据存储和管理服务，具有强大的存储能力和计算能力，能满足不同类型的数据存储和管理需求。

3)验证者。负责验证存储在云端的数据的完整性。CLG-PDP 方案支持公开验证，因此验证者可以是任意云用户。

图 1 展示了 CLG-PDP 方案的系统模型。系统假设 CSP 是半可信的(Semi-trust)，即它可以忠实地执行验证协议，但是会试图隐瞒数据不完整的事实。

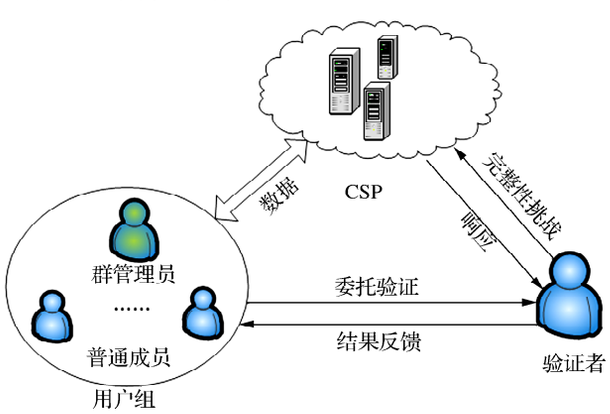

图 1 CLG-PDP 系统模型

1.3 方案的形式化定义

CLG-PDP 方案包括 8 个多项式时间算法：

1)系统建立算法：*Setup*。该算法由群组管理员执行，用于建立系统。算法输入参数是一个安全参数 k，输出系统主私钥 msk 和系统公开参数 $params$。

2)部分私钥生成算法：*PartialKeyGen*。该算法由群组管理员运行，用于为群组成员生成部分私钥。算法输入参数有主私钥 msk 和群组成员 u_i 的身份信息 ID_i，输出为成员 u_i 的部分私钥 D_i。

3)秘密值生成算法：*SecretValueGen*。群组成员运行该算法生成自己的秘密值。若成员 u_i 运行该算法，则算法输出 u_i 的秘密值，表示为 S_i。需要注意的是，任意群组成员的私钥信息均由部分私钥和秘密值

两个部分构成，如成员 u_i 的私钥为(D_i,S_i)。

4)公钥生成算法：$PublicKeyGen$。该算法由群组成员运行生成自己的公钥信息。算法的输入为群组成员的秘密值，输出为该成员的公钥信息。假设成员 u_i 运行该算法，则输入为 u_i 的秘密值 S_i，输出为 u_i 的公钥，表示为 PK_i。

5)标签生成算法：$TagGen$。该算法由群组成员运行为群组共享的数据生成验证标签。算法的输入参数为用户的部分私钥、秘密值以及目标数据，输出为目标数据的验证标签。群组内任意成员均可运行此算法。假设成员 u_i 运行该算法为数据块 m_j 生成标签，则需要输入 u_i 的部分私钥 D_i、秘密值 S_i 和目标数 m_j，输出为 m_j 的标签，表示为 T_j。

6)挑战算法：$Challenge$。由验证者运行用于发起一次数据完整性验证。其输入为要挑战的数据块数量 c，输出为挑战信息 $chal$。

7)证据生成算法：$ProofGen$。由 CSP 运行用于生成完整性证据。其输入为要挑战的数据 F，对应的验证标签集 T 和挑战信息 $chal$，输出为完整性证据 P。

8)验证算法：$Verify$。由验证者运行用于检查证据的正确性。其输入挑战信息 $chal$、完整性证据 P 以及所有群组成员的公钥集合 PK。如果证据 P 是正确的，算法输出 1，否则输出 0。

2　具体构造及安全证明

2.1　方案构造

假设一个群组中包含了 z 个成员，令 ID_i 表示群组成员 $u_i(1\leqslant i\leqslant z)$的唯一身份标识。不失一般性，将群组成员 u_1 设置为群组管理员，负责系统建立、群组管理及为其他群组成员生成部分私钥。假设待存储的数据为 F，将其分割成 n 个数据块表示为 $F=(m_1,m_2,\cdots,m_n)$，其中任意的 $m_i\in Z_q^*$。CLG-PDP 方案具体构造如下。

系统建立：$Setup$。设定安全参数 k，随机选择一个大素数 q 满足$|q|=k$。$\mathbb{G}_1$ 和$\mathbb{G}_2$ 是两个阶为 q 的乘法循环群，g 是群$\mathbb{G}_1$ 的生成元，$e:\mathbb{G}_1\times\mathbb{G}_1\rightarrow\mathbb{G}_2$ 是一个双线性映射。选择两个 Hash 函数 H_1 和 H_2，以及一个伪随机置换 π，一个伪随机函数 ϕ。定义如下：

$H_1:\{0,1\}^*\rightarrow\mathbb{G}_1^*$

$H_2:\{0,1\}^*\rightarrow\mathbb{G}_1^*$

$\pi:Z_q^*\times\{1,\cdots,n\}\rightarrow\{1,\cdots,n\}$

$\phi:Z_q^*\times Z_q^*\rightarrow Z_q^*$

随机选择一个整数 $s\in Z_q^*$ 作为主私钥，计算 $P_0=g^s$。将主私钥 s 保密，并公开系统参数 $params=(q,g,\mathbb{G}_1,\mathbb{G}_2,e,P_0,H_1,H_2,\phi,\pi)$。

部分私钥生成：$PartialKeyGen$。任意群组成员可以提交自己的身份信息到管理员 u_1 请求部分私钥，u_1 收到用户 u_i 请求后，计算其部分私钥 $D_i=H_1(ID_i)^s$ 并将 D_i 返回给用户 u_i。注意这里要求身份信息 ID_i 在群组内是唯一的。

秘密值生成：$SecretValueGen$。群组成员 u_i 随机选择一个整数 $x_i\in Z_q^*$，并设置自己的秘密值 $S_i=x_i$。

公钥生成：$PublicKeyGen$。群组成员 u_i 计算自己的公钥信息为 $PK_i=g^{S_i}=g^{x_i}$，并将其发布。

标签生成：$TagGen$。群组内的数据是共享的，每一个群组成员均可以访问数据，并且可以为数据生成验证标签。不失一般性，假设成员 $u_i(1\leqslant i\leqslant z)$为数据块 $m_j(1\leqslant j\leqslant n)$生成标签，则标签生成算法为：

$$T_j=D_i^{m_j}\cdot H_2(\omega_j)^{S_i} \tag{1}$$

其中 $\omega_i=F_{id}\parallel n\parallel j$，$F_{id}$ 表示数据的唯一标识。为记录数据块、标签和标签生成者的关系，群组管理员 u_1 维护一张公开的标签生成日志文件。当成员生成了标签后，维护日志文件，插入信息记录(ω_j,ID_i)。当成员修改数据块时，要重新为该数据块生成新的标签，此时需要更新日志记录文件中对应的元组信息。最后群组成员将最新的数据块和对应的标签上传到 CSP 中保存。CSP 可以通过公式(2)验证标签的正确性：

$$e(T_j,g)=e(H_2(\omega_2),PK_i)\cdot e(H_1(ID_i)^{m_j},P_0) \tag{2}$$

若 CSP 发现上传的数据和标签不匹配，则可以拒绝接收该数据和标签。

完整性挑战：$Challenge$。验证者随机选择两个整数(k_1,k_2)，其中 $k_1,k_2\in Z_q^*$。得到挑战信息 $chal=(c,k_1,c_2)$，并将其发送到 CSP。

证据生成：$ProofGen$。CSP 收到挑战信息 $chal=(c,k_1,k_2)$后，利用伪随机置换和伪随机函数计算得到挑战集合 $C=\{(v_i,a_i)\mid i\in[1,c]\}$，其中 $v_i=\pi(k_1,i)$，$a_i=\phi(k_2,i)$。CSP 查看公开的标签生成日志文件，根据标签生成者的身份信息，将挑战集合划分为多个子集合。不失一般性，假设集合 C 中的标签由 d 个成员生成，那么 C 将被划分为 d 个子集合，表示为 $C=\{c_1,\cdots,c_d\}(1\leqslant d\leqslant z)$，其中子集合 C_j 表示其中的数据标签均为用户 $u_{l_j}(1\leqslant j\leqslant d,1\leqslant l_j\leqslant z)$生成的。令 c_j 表示子集合 C_j 中的元组个数，则 $c=\sum_{j=1}^{d}c_j$，$C=C_1\cup C_2\cup\cdots\cup C_d$，且对于任意的 $j\neq j'$有 $C_j\cap C_{j'}=\varnothing$。CSP 计算出每一个子集合的子证据$(\overline{T}_j,\overline{F}_j)$：$\overline{T}_j=\prod_{v_i\in C_j}T_{v_i}^{a_i},\overline{F}_j=\sum_{v_i\in C_j}a_i m_{v_i}$。

CSP 得到完整的证据为：$P=(\overline{T},\overline{F})$，其中 $\overline{T}=\{\overline{T}_1,\cdots,\overline{T}_d\}$，$\overline{F}=\{\overline{F}_1,\cdots,\overline{F}_d\}$。

完整性验证：$Verify$。验证者收到证据 P 后，首先计算挑战集合 $C=\{(v_i,a_i)\mid i\in[1,c]\}$，其中 $v_i=\pi(k_1,i)$，$a_i=\phi(k_2,i)$。查看公开的标签生成日志文件，根据标签生成者将挑战集合划分为 d 个子集合。提取出日志文件中所有与挑战数据块对应的 ω_{v_i} 值。最后验证者检查公式(3)是否成立：

$$e\Big(\prod_{j=1}^{d}\overline{T}_j,g\Big)=\prod_{j=1}^{d}\Big(e\Big(\prod_{v_i\in C_j}H_2(\omega_{v_i})^{a_i},PK_{l_j}\Big)\Big)\cdot e\Big(\prod_{j=1}^{d}H_1(ID_{l_j})^{\overline{F}_j},P_0\Big)\tag{3}$$

如果成立输出 1，否则输出 0。

2.2 安全性证明

如果 CSP 和验证者诚实的执行验证协议，那么 CLG-PDP 方案的正确性可以证明如下：

$$\begin{aligned}
e\Big(\prod_{j=1}^{d}\overline{T}_j,g\Big)&=\prod_{j=1}^{d}(\overline{T}_j,g)=\prod_{j=1}^{d}e\Big(\prod_{v_i\in C_j}T_{v_i}^{a_i},g\Big)\\
&=\prod_{j=1}^{d}e\Big(\prod_{v_i\in C_j}(H_1(ID_{l_j})^{sa_i m_{v_i}}\cdot(H_2(\omega_{v_i}))^{x_{l_j}a_i}),g\Big)\\
&=\prod_{j=1}^{d}\Big(e\Big(\prod_{v_i\in C_j}H_1(ID_{l_j})^{a_i m_{v_i}},g^s\Big)\cdot e\Big(\prod_{v_i\in C_j}H_2(\omega_{v_i})^{a_i},g^{x_{l_j}}\Big)\Big)\\
&=\prod_{j=1}^{d}\Big(e(H_1(ID_{l_j})^{F_j},P_0)\cdot e\Big(\prod_{v_i\in C_j}H_2(\omega_{v_i})^{a_i},PK_{l_j}\Big)\Big)\\
&=\prod_{j=1}^{d}e\Big(\prod_{v_i\in C_j}H_2(\omega_{v_i})^{a_i},PK_{l_j}\Big)\cdot e\Big(\prod_{j=1}^{d}H_1(ID_{l_j})^{\overline{F}_j},P_0\Big)
\end{aligned}$$

3 仿真实验及性能分析

为验证 CLG-PDP 方案的效率，本文基于 PBC[13] 和 GMP[14] 实现了方案原型系统，并进行仿真实验。实验的运行环境是：VM 虚拟机系统，Linux 操作系统，4 G 内存，20 G 硬盘；主机是联想 LaptopL440，Win7 系统，CPU 是 Corei7-4712MQ @2.3 GHz，8 G 内存。实验中使用了 PBC 库中 A 类的椭圆曲线，阶为 160 bit，能够达到 RSA1024 byte 的安全水平。为了获得更准确的结果，每组实验都循环了 50 次。

在标签生成实验中，模拟了一个包含 50 名成员的群组，生成的标签个数为 100 000～1 000 000。不失一般性，实验中假设标签是由群组内成员按平均分配的方式生成的，这种假定并不影响实验结果的分析。标签生成的实验结果如图 2 所示。可以看出，标签生成的代价和数据块数量呈线性关系。生成 1 000 000 个数据标签时，仅需要 112 s，在实际的应用中是完全可接受的。

在 CLG-PDP 方案中，完整性挑战、证据生成和验证操作是一个不断循环的过程，也是方案的核心业务。方案整体性能取决于该阶段的性能。而验证算法的性能对用户影响最大，因此实验重点验证了该阶段的性能。如图 3 所示，挑战的数据块数目越大，验证时所花费的代价越大。当挑战的数据块数量一定时，验证代价随群组的递增而线性增加。对于一个拥有 100 名成员的群组，当挑战的数据块数量为 460 时，完整性验证需要的时间约为 3.18 s，当挑战的数量为 300 时，所需时间约为 2.28 s。可以看出该计算

代价在实际应用中是高效可行的。

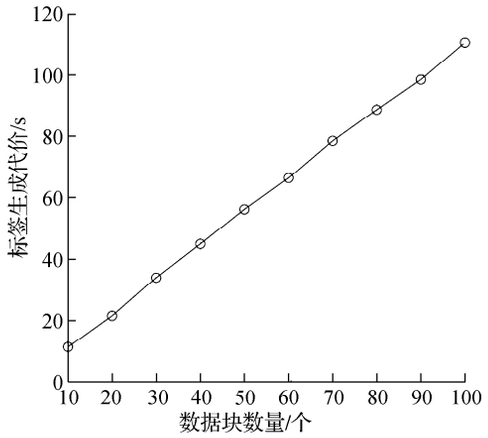

图 2　标签生成代价

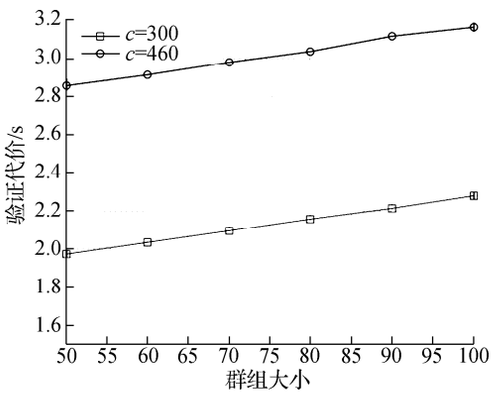

图 3　验证阶段开销

4　结　语

本文提出了一个基于无证书密码体制的、支持群组共享数据的云端数据完整性公开验证方案。本文形式化地定义了方案的系统模型，给出了方案的具体构造，并证明了方案的安全性。通过理论分析和原型系统的仿真实验证明了方案的有效性和高效性。

参考文献：

[1] Buyya R，Yeo C S，Venugopal S，et al. Cloud computing and emerging IT platforms：Vision，hype，and reality for delivering computing as the 5th utility[J]. Future Generation Computer Systems，2009，25(6)：599－616

[2] Ateniese G，Burns R，Curtmola R，et al. Provable data possession at untrusted stores[C]. New York：Proceeding of 14th ACM Conference on Computer and Communications Security，2007：598－609

[3] Erway C，Küpçü A，Papamanthou C，et al. Dynamic provable data possession[C]. New York：Proceeding of 16th ACM Conference on Computer and Communications Security，2009：213－222

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

[5] Yang K，Jia X. An efficient and secure dynamic auditing protocol for data storage in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems，2013，24(9)：1717－1726

[6] Yan H，Li J，Han J，et al. A novel efficient remote data possession checking protocol in cloud storage[J]. IEEE Transactions on Information Forensics and Security，2017，12(1)：78－88

[7] Wang H，He D，Tang S. Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud [J]. IEEE Transactions on Information Forensics and Security，2016，11(6)：1165－1176

[8] Wang B，Li B，Li H. Knox：privacy-preserving auditing for shared data with large groups in the cloud[C]. New York：Proceeding of 10th International Conference Applied Cryptography and Network Security，2012：507－525

[9] Wang B，Li B，Li H. Panda：Public auditing for shared data with efficient user revocation in the cloud[J]. IEEE Transactions on Service Computing，2015，8(1)：92－106

[10] Yu Y，Mu Y，Ni J，et al. Identity privacy-preserving public auditing with dynamic group for secure mobile cloud storage [C]. Berlin：Proceeding of 8th International Conference on Network and System Security，2014：28－40

[11] Yuan J，Yu S. Public integrity auditing for dynamic data sharing with multiuser modification[J]. IEEE Transactions on Information Forensics and Security，2015，10(8)：1717－1726

[12] GMP. The GNU multiple precision arithmetic library[EB/OL]. (2018－03－20)[2019－02－25]. http://gmplib.org/

[13] Li J，Yan H，Zhang Y. Certificate less public integrity checking of group shared data on cloud storage[J]. IEEE Transactions on Services Computing，2017，10. 1109/TSC. 2018. 2789893

[14] The pairing-based cryptography library(PBC)[EB/OL]. (2018－03－20)[2019－02－25]. https://crpto. stanford. edu/pbc/download. html

（责任编辑：湛　江）