对基于身份云数据完整性验证方案的分析与改进

王少辉[1,2]，潘笑笑[1,2]，王志伟[1,2]，肖甫[1,2]，王汝传[1,2]

（1. 南京邮电大学计算机学院、软件学院、网络空间安全学院，江苏 南京 210003；
2. 江苏省无线传感网高技术研究重点实验室，江苏 南京 210003）

摘 要：个人或企业将数据外包给远程云服务器，在获得运营便利的同时也失去了对数据的本地控制权，无法直接保证数据的完整性和隐私性。分析了 Zhang 等和 Xu 等提出的基于身份云数据完整性验证方案的安全性。分析表明 Zhang 等所提方案易遭受密钥恢复攻击，云服务器利用存储的用户数据可恢复出用户的私钥，而 Xu 等所提方案不能满足健壮性的安全要求。在 Xu 等方案的基础上，提出了一个改进的云数据完整性验证方案，分析表明新方案可提供健壮性和隐私性的安全需求，并且可提供与 Xu 等所提方案相同的通信和计算开销。

关键词：云存储；基于身份密码系统；数据完整性；隐私性

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000-436x.2018229

Analysis and improvement on identity-based cloud data integrity verification scheme

WANG Shaohui[1,2], PAN Xiaoxiao[1,2], WANG Zhiwei[1,2], XIAO Fu[1,2], WANG Ruchuan[1,2]

1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2. Key Laboratory of Jiangsu High Technology Research for Wireless Sensor Network, Nanjing 210003, China

Abstract: Many individuals or businesses outsource their data to remote cloud. Cloud storage provides users the advantages of economic convenience, but data owners no longer physically control over the stored data, which introduces new security challenges, such as no security guarantees of integrity and privacy. The security of two identity-based cloud data integrity verification schemes by Zhang et al and Xu et al respectively are analysed. It shows that Zhang et al.'s scheme is subjected to secret key recovery attack for the cloud servers can recover user's private key only utilizing stored data. And Xu et al.'s scheme cannot satisfy security requirements of soundness. Based on Xu et al.'s scheme, a modified identity-based cloud data integrity verification scheme is proposed. A comprehensive analysis shows the new scheme can provide the security requirements of soundness and privacy, and has the same communication overhead and computational cost as Xu et al.'s scheme.

Key words: cloud storage, identity-based cryptosystem, data integrity, privacy

1 引言

云存储具有低成本、高扩展、海量存储、便捷存取的优点，越来越多的个人或企业选择将数据存放在云端，这样用户可以节约大量的存储、管理以及维护成本，但是随之带来的问题是用户无法对数据直接控制，从而不能确保存储在云中数据的完整性和隐私性。云环境下导致数据损坏的因素主要包括以下方面。1）云服务提供商不完全可信。出于经济原因，云服务提供商可能会删除用户很少或从

收稿日期：2017−12−22；**修回日期**：2018−10−11

基金项目：国家自然科学基金资助项目（No.61373006, No.61373139, No.61672016, No.61872192）；江苏省科技支撑计划基金资助项目（No.61003236）

Foundation Items: The National Natural Science Foundation of China (No.61373006, No.61373139, No.61672016, No.61872192), The Scientific of Technolgical Suppovt Project of Jiangsu Province (No.61003236)

未访问的数据，以便可以节省空间存储其他用户数据从而获取更多收入。2）由于云服务器的故障、管理失误或对手恶意攻击，云服务器中存储的数据可能会被破坏。但是，云服务提供商为了维护良好的声誉可能会故意隐藏数据丢失的事实。在云存储中，数据完整性和隐私性已成为用户最关心的问题，因为即使很少一部分数据的损坏也可能会造成不可估量的损失。因此，如何解决云存储中所面临的数据完整性验证问题已经成为了目前学术界和企业界研究的一个热点，具有较高的研究价值和应用前景。

Deswarte 等[1]在 2003 年提出了基于散列函数的云数据完整性验证方案，它允许客户端使用相同的元数据进行多次挑战，但是该协议必须对存储的整个文件进行取幂，计算成本过高。Oprea 等[2]在 2005 年提出了一个支持块级数据完整性验证的方案，但是该方案仍存在过高的通信成本和计算成本。Ateniese 等[3-4]在 2007 年最先对可证明数据持有审计方案(PDP，provable data possession)进行了形式化建模。该方案引入第三方审计者(TPA，third-party auditor)提供验证服务，从而能够有效减轻用户的负担。方案可以使 TPA 在不下载数据到本地的情况下验证数据的完整性，并提出分块思想降低生成验证元的计算成本。同时，Juels 等[5]最先对数据可恢复证明问题（POR, proof of retrievability）进行了形式化建模，提出了基于哨兵的 POR 验证机制，该机制不仅能够识别远程数据是否损坏而且能够对已经损坏的数据文件进行恢复。2008 年，Shacham 等[6-7]提出了两种使用纠错码技术的高效紧缩的 POR 方案。这两种 POR 方案都支持任意次验证并且支持无状态认证，即认证过程中认证者不需要保存认证状态，最后通过引用同态验证标签将证明信息有效地缩小为一个较小的值。

上述方案均在公钥基础设施 PKI（public key infrastructure）架构下设计。Shamir[8]在 1984 年创新性地提出了基于身份的密码系统，用于解决传统 PKI 密码体制带来的证书生成、验证、存储和注销等问题。在基于身份密码体制中，将标示用户身份的唯一信息（如用户的 IP 地址）作为用户的公钥，用户的私钥由可信的私钥生成中心（PKG, public key generator）利用用户的身份信息和自己的主密钥计算得到。2001 年，Boneh 等[9]利用双线性对构建了第一个安全实用的基于身份的加密方案，随后基于身份的加密方案设计引起了密码学界的广泛关注。

目前，大部分基于身份云数据完整性验证方案主要利用双线性运算这一工具，如 2016 年，Zhang 等[10]基于 Waters 签名，利用双线性运算在标准模型中提出了一个基于身份的公共验证方案。该方案具有固定的通信开销和计算成本。双线性运算比 RSA 算法中的模幂运算需要更多的计算开销。为了提高验证效率，Yu 等[11]利用 RSA 算法提出了基于身份的高效远程数据完整性验证方案。随后 Zhang 等[12]指出 Yu 等的方案仅适用于单个用户场景，而且没有考虑密钥泄露的问题。Zhang 等通过对密钥在固定时间间隔进行更新，提出了一个能够抵御密钥泄露攻击的数据完整性验证方案。最近 Xu 等[13]指出了 Yu 等[11]的方案没有考虑数据的隐私性问题，它容易受到来自 TPA 或外部攻击者的数据恢复攻击。

本文对基于身份云数据完整性验证方案进行了研究，首先对文献[12]和文献[13]所提方案的安全性进行了分析，通过分析指出文献[12]方案容易遭受私钥恢复攻击，云服务器通过分析用户存储的数据可以恢复并得到用户私钥；而文献[13]所提方案不满足健壮性要求，即服务器即使没有完整无误地保存用户数据，也可以成功伪造证据通过 TPA 的验证。进而在文献[13]方案的基础上提出了一个改进的基于身份云数据完整性验证方案，通过分析证明该方案能有效满足隐私性和健壮性要求，而在效能方面，新方案具有与文献[13]方案相同的通信、计算成本。

2　系统模型和安全需求

2.1　系统模型

如图 1 所示，基于身份云数据完整性验证模型主要由如下 4 个部分组成。

1) 用户：用户是拥有大量数据需要存储在云服务器的个人或组织，通常用户带宽、存储和计算能力有限。

2) 云服务器：拥有大量的存储空间和强大的计算能力以维护用户存储的数据，并为用户提供数据访问服务。一般假设云服务器并不完全可信，因为云服务器可能会为了自己的利益删除很少被访问的用户数据。

3) 第三方审计者(TPA)：代替用户与云服务器完成交互，能够检查存储在云服务器上用户数据的完整性。TPA 由可信部门监督和管理，提供可信、

公正公平的审计结果。但是假定 TPA 是诚实而好奇(curious but honest)的，因此必须保证 TPA 无法通过审计过程获得用户的隐私数据。

4) 私钥生成中心(PKG)：是完全可信的机构，根据用户的身份信息为其生成用户私钥。

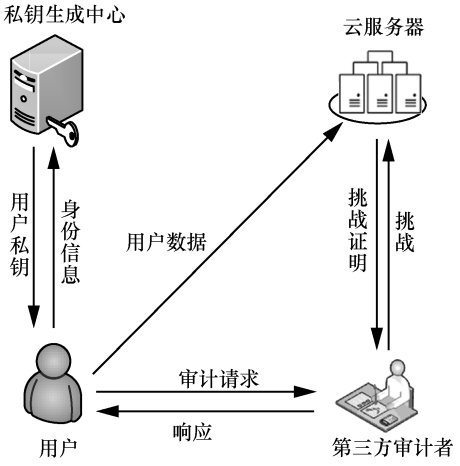

图 1 系统模型

基于身份云数据完整性验证方案通常由以下 6 个算法构成，其中算法 4)～算法 6) 构成了 Audit ($TPA \leftrightarrow$ 云服务器) 算法，Audit 由 TPA 和云服务器按照算法 4)~算法 6)的步骤交互完成数据的完整性验证服务。

1) Setup $(1^k) \to (params, mpk, msk)$：由 PKG 执行的概率算法。该算法以安全参数 k 作为输入，输出公共系统参数 *params* 以及 PKG 的主公钥、私钥对 (mpk, msk)。

2) Extract $(params, mpk, msk, ID) \to (sk_{ID})$：由 PKG 和用户交互执行的概率算法。该算法以系统参数 *params*、PKG 的密钥对 (mpk, msk) 以及用户传输的身份信息 $ID \in \{0,1\}^*$ 作为输入，输出用户私钥 sk_{ID}。

3) TagGen $(M, sk_{ID}) \to (\zeta)$：通过该概率算法，用户为云存储数据生成认证标签。通常用户会将外包数据 M 分割成小的数据块 m_i，利用用户私钥 sk_{ID}，为每个 m_i 生成一个数据认证标签 ζ_i。最后用户将外包数据 M 和认证标签 $\zeta = (\zeta_1, \zeta_2, \cdots, \zeta_n)$ 一并存储在云服务器中。

4) Challenge $(params, ID, Fname) \to (C)$：以系统公共参数 *params*、用户的身份 *ID* 和文件名 *Fname* 作为输入，TPA 执行该算法输出一次验证服务的挑战信息 C。

5) ProofGen $(params, ID, C, M, \zeta) \to (P)$：云服务器接收到来自 TPA 的挑战信息 C，利用存储的相关数据文件 M和认证标签 ζ，云服务器生成相应的挑战证明信息 P。

6) Verify $(C, P, ID, mpk, params) \to (0/1)$：TPA 对云服务器的完整性证明信息 P 进行验证，如果数据文件被完整的保存则输出审计结果为 1，否则输出 0。

2.2 安全需求

首先给出可忽略概率的概念。

定义 1(可忽略概率优势) 给定一个任意的多项式 $f(n)$，对于任何正整数 N，当 $n > N$ 时，都有 $\Pr[n] < \frac{1}{f(n)}$，则概率优势 $\Pr[n]$ 是可以忽略的。

云数据完整性验证方案需要满足隐私性和健壮性的安全需求，下面本文给出两个安全需求的形式化定义。

定义 2(隐私性) 本文通过挑战者和敌手之间的如下游戏来定义隐私性，游戏过程如下所示。

1) 初始化：挑战者运行 Setup 算法生成系统参数 *params*、主公私钥对 (mpk, msk)，然后将 *params* 和 *mpk* 发送给敌手，秘密保存主私钥 *msk*。

2) 质询：敌手自适应地向挑战者进行如下质询，包括 Extract 质询、TagGen 质询和 Audit 质询。

① Extract 质询：敌手将任意身份 *ID* 发送给挑战者，挑战者执行 Extract 算法得到并返回相应私钥 sk_{ID}。

② TagGen 质询：敌手向挑战者发送身份信息 *ID* 和数据 M^i，而挑战者执行 TagGen 算法为敌手生成认证标签 ζ^i。

③ Audit 质询：通过此质询，敌手和服务器将执行交互，执行数据的完整性验证算法。

3) 挑战者选择身份信息 *ID* 和数据文件 M_1，要求敌手没有对 *ID* 进行过 Extract 质询，也未对文件 M_1 进行过 TagGen 质询。挑战者生成文件 M_1 在身份 *ID* 下的标签 $\zeta \leftarrow \text{TagGen}(M, sk_{ID})$。

4) 敌手与挑战者进行 Audit 验证交互，此时敌手扮演 TPA 的角色，而挑战者扮演云服务器的角色，最终敌手输出数据文件 M^*。

假设数据文件 M_1 具有大的信息熵，如果通过上述游戏，敌手成功的概率为

$$\Pr[M^* \leftarrow \text{敌手}(pk) : M_1 = M^*]$$

是可以忽略的。本文称基于身份云数据完整性验证方案满足隐私性，

定义 3(健壮性)　通过挑战者和敌手之间的如下游戏来定义健壮性，其中初始化和质询操作同隐私性的定义，这里不再赘述。

1) 对于已经进行 TagGen 质询的文件 F，敌手可以通过指定数据拥有者的身份 ID 和文件名 Fn 来执行 ProofGen 算法，此时挑战者扮演 TPA 的角色，而敌手扮演云服务器的角色输出对于指定身份 ID_i 和文件名 Fn 的证据信息 P。

2) 敌手选择文件名为 Fn' 的文件和身份为 ID' 的用户，要求敌手未对身份 ID' 进行 Extract 质询并且也未对 (ID', Fn') 进行过 TagGen 质询。最终对挑战者的验证挑战 C，敌手输出完整性证据信息 P'。

如果概率为

$$\Pr[(\text{Verify}(param, C, ID', Fn', P') = 1]$$

是可以忽略的，则称基于身份云数据完整性验证方案满足健壮性。

3　对两个云数据完整性方案分析

本节对文献[12]和文献[13]中所提方案进行详尽地安全分析，分析指出文献[12]方案存在私钥恢复攻击，服务器能利用存储的数据和标签信息得到用户在 t 时刻的私钥。而文献[13]则不满足健壮性的要求，即使服务器没有存储数据，也能生成合法的证明信息通过 TPA 的验证。

3.1　对文献[12]方案的安全分析

文献[12]的方案主要针对用户私钥的泄露问题，这里只简单介绍与安全性分析相关的 Setup、Extract 和 TagGen 等算法，其他算法具体可以参考文献[12]。

1) Setup。选取安全参数 1^k，PKG 产生 2 个 k bit 安全大素数 p_0和q_0，计算 RSA 模 $N_0 = p_0 q_0$，PKG 随机选择满足 $\gcd(e, \varphi(N_0)) = 1$ 的素数 e，其中 $\varphi(N_0)$ 是欧拉函数，并计算 $d \in Z_{N_0}$，使 $ed \equiv 1 \bmod(N_0)$。同时 PKG 选择两个 Hash 函数 $H_1 : \{0,1\}^* \to \{0,1\}^{l_1}$ 和 $H_2 : \{0,1\}^* \to Z_{N_0}$，其中，$l_1 < 80$。系统公布公共参数 $Para = \{k, l, e, N_0, H_1, H_2\}$，而主密钥为 (p_0, q_0, d)。

2) Extract。当身份为 $ID \in \{0,1\}^*$ 的用户 U 在时间段 t 向 PKG 请求私钥时，PKG 利用主密钥 d 计算并安全发送用户私钥为 $S_{ID}^t = H_1(ID)^{d^{T+1-t}} \bmod \varphi(N_0)$，初始时刻 t 为 0。

3) TagGen。在 t 时间段，身份为 ID 的用户执行以下步骤将数据文件 M 外包到云端。

① 用户随机选择公私钥对为 (sk, pk) 的安全签名算法 $\sum$。将外包文件 M 分割为 $M = M_1 \| M_2 \| \ldots \| M_n$，对任意 $i = 1, 2, \cdots, n$，$M_i \leqslant 2^k$。

② 令 $\tau = FName \| n \| ID_i$，其中 $FName$ 是文件名，然后用户对 τ 生成签名，并将 $sig = \tau \| \sum.sign_{sk}(\tau)$ 作为文件的身份标识信息。

③ 对于数据文件 $FName$，用户随机选择 $r \in Z_N$，计算 $R = r^{e^{T+1-t}}$，$i = 1, 2, \cdots, n$。M_i 的认证标签计算如下 $\zeta_i = r^{H_2(FName \| R \| index_i)} \cdot (S_{ID}^t)^{M_i} \bmod N_0$，用户将 $\left\{M, t, \{\zeta\}_{i=1}^n, R, \tau, sig\right\}$ 上传到云服务器后删除本地存储。

安全分析：下面指出上述方案易遭受私钥恢复攻击，攻击方法如下所示。

由 TagGen 算法，知道文件块 M_i 的认证标签为 $\zeta_i = r^{H_2(FName \| R \| index_i)} \cdot (S_{ID}^t)^{M_i} \bmod N_0$，这里将 $H_2(FName \| R \| index_i)$ 简记为 x_i，即 $\zeta_i = r^{x_i} \cdot (s_{ID}^t)^{M_i} \bmod N_0$，而云服务器可计算 s_{ID}^t 的值。

对于不同的消息 M_1 和 M_2 及其认证标签 $\zeta_1 = r^{x_1} (s_{ID}^t)^{M_1}$，$\zeta_2 = r^{x_2} (s_{ID}^t)^{M_2}$，云服务器计算为

$$(\zeta_1)^{x_2} = r^{x_1 x_2} (s_{ID}^t)^{M_1 x_2} \bmod N_0 \tag{1}$$

$$(\zeta_2)^{x_1} = r^{x_1 x_2} (s_{ID}^t)^{M_2 x_1} \bmod N_0 \tag{2}$$

然后式(1)除以式(2)得

$$(\zeta_1)^{x_2} (\zeta_2)^{-x_1} = \left(s_{ID}^t\right)^{M_1 x_2 - M_2 x_1}$$

同理考虑消息块 M_3、M_4，可得

$$(\zeta_3)^{x_4} (\zeta_4)^{-x_3} = \left(s_{ID}^t\right)^{M_3 x_4 - M_4 x_3}$$

如果 $(M_1 x_2 - M_2 x_1, M_3 x_4 - M_4 x_3) = 1$，利用数论知识可知存在整数 (α_1, α_2)，使 $\alpha_1 (M_1 x_2 - M_2 x_1) + \alpha_2 (M_3 x_4 - M_4 x_3) = 1$，从而云服务器可以计算得到 t 时刻用户的私钥 s_{ID}^t 如下所示。

$$s_{ID}^t = [(\zeta_1)^{x_2} (\zeta_2)^{-x_1}]^{\alpha_1} [(\zeta_3)^{x_4} (\zeta_4)^{-x_3}]^{\alpha_2}$$

3.2　对文献[13]方案的安全分析

文献[13]提出的基于身份云数据完整性认证方案由以下算法构成。

1) Setup。PKG 选择两个随机素数 p'和q'，计

算 RSA 模 $N'=p'q'$，并随机选择一个素数 α，计算 $\beta\equiv\alpha^{-1}\bmod\varphi(N')$。同时定义 3 个 Hash 函数：$H_1:\{0,1\}^*\to Z_{N'},H_2:\{0,1\}^*\to Z_N^*,H_3:\{0,1\}^*\to\{0,1\}^l$。PKG 选择伪随机函数 $f_1:Z_N\times Z_N\to Z_N$ 和伪随机序列 $f_2:Z_N^*\times\{1,2,\cdots,N\}\to\{1,2,\cdots,N\}$。系统主密钥为 β，而可公开公钥为 $mpk=\left(N',\alpha,H_1,H_2,H_3,f_1,f_2\right)$。

2) Extract。给定用户身份 $ID\in\{0,1\}^*$，PKG 生成并安全发送用户私钥 $s_{ID}=H_1(ID)^{\beta}\bmod(N')$。

3) TagGen。为了将文件 F 存储在远程云服务器上，用户首先选择签名公私钥对为 $\{pk,sk\}$ 的签名算法 $SSig()$，同时选择 2 个素数 p、q 以及一个随机素数 e，计算 $N=pq$ 和 $d\equiv e^{-1}\bmod\varphi(N)$。用户随机选择一个随机数 $\gamma\in Z_N^*$，计算 $T=\gamma^{\alpha}\bmod N'$，$t=H_3(T\|e\|pk)$，$s=\gamma\cdot s_{ID}^t$，这里 (t,s) 是对 $\left(pk,e\right)$ 的基于身份签名。给定文件 F，用户将其分割成 n 个小的数据块 $F=\{m_1\|m_2\|\cdots\|m_n\}$，令 $w_0=Fn\|n\|\mu$，其中，Fn 是文件名，μ 是一个大随机值，记 $w=w_0\|SSig_{sk}(w_0)$ 为文件 F 的标识信息。用户为数据块 m_i 计算标签：$\zeta_i=(H_2(Fn\|i)\ \mu^{m_i})^d\bmod N$，记 $\zeta=\left(\zeta_i\right)_{1\leqslant i\leqslant n}$，将 $\{F,\zeta,w,t,s,pk,e\}$ 存储在云服务器上，并删除本地存储。

4) Challenge。TPA 和云服务器执行以下步骤完成挑战。TPA 首先向云服务器发送验证请求。云服务器向 TPA 返回 $\left(t,s,pk,e\right)$。TPA 计算 $T'=s^{\alpha}H_1(ID)^{-t}\bmod N'$ 并验证 $t=H_3\left(T'\|e\|pk\right)$ 是否成立，如果成立则继续，否则停止并返回 0。TPA 使用签名公钥 pk 验证 w 签名的有效性，无效则停止并返回 0。否则 TPA 随机选择 $k\in Z_N^*$ 和 $c\in[1,n]$，然后生成挑战集 $Q=\{(i_j,v_j)\}$，其中，$i_j=f_1\left(k,j\right)$，$v_j=f_2(k,j)$，$i\in[1,c]$。TPA 将挑战 $chal=(c,k)$ 发给云服务器。

5) ProofGen。云服务器接收到 TPA 的挑战 $chal=(c,k)$ 后，首先计算挑战集 $Q=\left\{\left(i_j,v_j\right)\right\}$，然后随机选择 $\eta\in Z_N^*$，并计算 $V=\eta+\sum_j v_jm_j$ 且 $\sigma=\prod_j\zeta_j^{v_j}\bmod N$。云服务器将证明信息 $P=\left(V,\sigma,\mu^{\eta}\right)$ 发送给 TPA。

6) Verify。TPA 验证下面的等式是否成立。

$$\sigma^e=(\prod_j H_2(Fn\|i_j)^{v_j})\mu^V\mu^{-\eta}\bmod N$$

如果成立则返回 1，表明验证通过。否则表明数据没有被服务器保存完整，返回 0。

安全分析：下面给出对上述方案的攻击，可以看出方案[13]不能满足健壮性的安全需求，云服务器可以在任意修改甚至删除用户数据后仍能生成合法的证明消息 P 通过 TPA 的验证，令 TPA 相信服务器正确存储了用户数据。

在 ProofGen 算法中，云服务器接受到来自 TPA 的挑战 $chal=(c,k)$ 后，首先在 Z_N^* 中选择随机数 l，并计算 $V=le$ 和 $\sigma=\mu^l$，云服务器令 $\mu^{\eta}=(\prod_j H_2(Fn\|i_j)^{v_j})$。因为云服务器只是发送 μ^{η} 的值，此时其不知道对应 η 的具体值。云服务器将伪造的挑战证明信息 $P=(V,\sigma,\mu^{\eta})$ 发送给 TPA。

在 Verify 算法中，TPA 诚实的执行验证过程，此时

$$\begin{aligned}&\prod_j H_2(Fn\|i_j)^{v_j}\mu^V\cdot\mu^{-\eta}\bmod N\\&=\mu^V\bmod N=\mu^{le}\bmod N\end{aligned}$$

而 $\sigma=\mu^l$，显然验证等式成立。因此，TPA 相信用户的数据被完好存储，而实际上云服务器在生成验证信息 P 时，并没有用到存储的用户数据。

4 新的改进方案

在本节中，基于文献[13]的方案，本文提出了一个改进的基于身份云数据完整性验证方案，并给出了改进方案的安全性分析和效能比较。

4.1 改进方案的设计

改进的基于身份云数据完整性认证方案由 6 个算法组成，其中，算法 Setup、Extract、TagGen 和 Challenge 与文献[13]的方案一致，这里不再赘述，ProofGen 和 Verify 算法改进如下所示。

1) ProofGen。云服务器接收到 TPA 的挑战信息 $chal=(c,k)$ 后，首先计算挑战集 $Q=\left\{\left(i_j,v_j\right)\right\}$。并在 Z_N^* 选择随机值 η，然后计算 $\theta=H_1(\mu^{\eta})$，$V=\eta+\theta\cdot\sum_j v_j\cdot m_j$ 且 $\sigma=\prod_j\zeta_j^{v_j}\bmod N$。云服务器将证明信息 $P=\left(V,\sigma,\mu^{\eta}\right)$ 发送给 TPA。

2) Verify。TPA 验证下面等式是否成立

$$\left(\sigma^e\right)^{\theta}=\prod_j H_2(Fn\|i_j)^{v_j\cdot\theta}\mu^V\cdot\mu^{-\eta}\bmod N$$

如果成立则返回 1，表明验证通过。否则返回 0 表明数据没有被服务器保存完整。

4.2　安全性分析

本节中，本文分别对方案的正确性、健壮性和隐私性进行说明。

正确性。由方案可知，此时

$$\begin{aligned}\left(\sigma^{e}\right)^{\theta}&=\left(\prod_{j}\zeta_{j}^{v_j}\right)^{e\cdot\theta}\bmod N\\&=\left(\prod_{j}\left[\left(H_2\left(Fn\|i_j\right)\mu^{m_j}\right)^{d}\right]^{v_j}\right)^{e\cdot\theta}\bmod N\\&=\left(\prod_{j}\left(H_2\left(Fn\|i_j\right)\right)^{v_j\cdot\theta}\prod_{j}\mu^{v_j\cdot m_j\cdot\theta}\right)^{e\cdot d}\bmod N\\&=\prod_{j}H_2(Fn\|i_j)^{v_j\cdot\theta}\mu^{\left(\eta+\theta\cdot\sum_{j=1}^{c}v_j\cdot m_j\right)}\mu^{-\eta}\bmod N\\&=\prod_{j}H_2(Fn\|i_j)^{v_j\cdot\theta}\mu^{V}\mu^{-\eta}\bmod N\end{aligned}$$

从而如果云服务器正确保存用户数据，生成的验证信息能够通过 TPA 的验证。

健壮性。新方案的健壮性基于如下的 RSA 假设。

定义 4(RSA 假设)　给定 (N,y,e)，其中，N 是两个大素数的乘积 $N=pq$，$e>1$ 且 $(e,\varphi(N))=1$，则在多项式时间内找到 $x\in Z_N$，满足 $\left(x\right)^{e}=y$ 的概率是可以忽略的。

定理 1　如果 RSA 假设成立，则改进的基于身份云数据完整性验证方案满足健壮性安全需求。

证明　文献[13]的健壮性证明分为两部分，首先证明 TagGen 算法中对 $\left(pk,e\right)$ 的基于身份签名是不可伪造的，然后证明方案的健壮性。本文在随机预言模型下证明新方案满足健壮性的安全需求，对基于身份签名方案的不可伪造性证明可参考文献[13]。

给定 RSA 假设的参数 $N=pq$，其中，p、q 是两个大素数，$(e,\varphi(N))=1$，给定 $y\in Z_N^{*}$，挑战者计算 $y^{\frac{1}{e}}$。

挑战者首先设定 $\mu=y^{2}$，按如下方式应答敌手对随机预言 $H_1(\cdot)$ 和 $H_2(\cdot)$ 的质询，并保存相应的输入输出序列。

1) 对 $H_2(Fn\|i)$，挑战者随机选择 $r_i\in Z_N^{*}$，并返回 $r_i^{e}\mu^{-m_i}\bmod N$，即消息块 m_i 对应的标签为 r_i。

2) 对敌手的 $H_1(\mu^{\eta_i})$ 质询，挑战者返回随机值 $\theta_i\in Z_N^{*}$。

从而挑战者可以与敌手进行 Audit 的交互，假设最终对挑战者的挑战信息集 $Q=\left\{\left(i_j,v_j\right)\right\}$，攻击者返回正确的挑战响应信息 $P=(V,\sigma,\mu^{\eta})$，令 $M=\sum_j v_j\cdot m_j$，本文假设敌手对不同消息 $M^{*}\neq M$ 生成的证明信息。在随机预言模型下，敌手必曾质询过 $H_1(\mu^{\eta})$ 和 $H_2(Fn\|i_j)$ 的相关值，则挑战者计算为

$$\begin{aligned}\sigma^{\theta}&=((\prod_j H_2(Fn\|i_j))^{v_j\theta}\mu^{V}\mu^{-\eta})^{d}\\&=(\prod_j(r_j^{e}\mu^{-m_j})^{v_j\cdot\theta}\mu^{\theta M^{*}})^{d}=\prod_j(r_j)^{v_j\theta}(\mu^{\theta(M^{*}-M)})^{d}\end{aligned}$$

即

$$\left(\frac{\sigma}{\prod_j(r_j)^{v_j\cdot}}\right)^{e}=\mu^{(M^{*}-M)}=y^{2(M^{*}-M)}$$

由于 e 通常选择大的随机素数，则以不可忽略的概率优势等式 $(e,2(M^{*}-M))=1$ 成立，从而存在 α,β，使得 $\alpha e+\beta 2(M^{*}-M)=1$，从而

$$y^{\frac{1}{e}}=(y^{\alpha e+\beta 2(M^{*}-M)})^{\frac{1}{e}}=y^{\alpha}\left(\frac{\sigma}{\prod_j(r_j)^{v_j}}\right)^{\beta}$$

即挑战者输出 $x=y^{\alpha}\left(\frac{\sigma}{\prod_j(r_j)^{v_j}}\right)^{\beta}$，这与 RSA 问题的困难性假设矛盾，从而新方案满足健壮性的安全需求。

隐私性。最后我们简要证明新方案满足隐私性的安全需求。

定理 2　在新方案中，给定验证证明信息 $P=\left(V,\sigma,\mu^{\eta}\right)$，TPA 可以获得用户存储消息在计算上不可行。

证明　可以看出如果服务器将 $\sum_j v_j m_j$ 直接发送给 TPA，则 TPA 可以通过求解线性方程组的方法获得消息块 m_j。为保护用户数据的隐私性，云服务器利用随机值 η 对 $\sum_j v_j m_j$ 的值进行了盲化，即 $V=\eta+\theta\sum_{j=1}^{c}v_j m_j$。此时，只有当 TPA 获得了 η 的值，才能继续采用求解方程组的方法获得消息。但是给定 μ^{η}，由离散对数的困难性假设，TPA 无法计算得到 η 的值，进而其不能通过响应验证信息来计算出数据块 m_j，因此新方案满足隐私性的安全需求。

4.3 效率分析

分别从通信成本和计算成本这两个方面将新方案与文献[13]中的方案进行了比较。假设在两种方案中，存储相同的数据文件 F，而 TPA 产生的挑战信息 *chal* 一样。从通信开销来看，新方案在各阶段和文献[13]的方案一致，故两种方案具有相同的通信开销。

计算成本。分别用 T_{add}，T_{mul}，T_{inv}，T_{exp} 表示在 Z_N 中执行一次模加、模乘、模逆和模幂运算所需的时间，而 T_{hash} 表示在 Z_{N} 或 Z_{N}^* 中执行 Hash 运算所需的时间。新方案与文献[13]方案的计算成本比较如表 1 所示，其中，TagGen 和 Challenge 算法两种方案具有相同的计算开销。

对 ProofGen 算法，文献[13]方案中云服务器需要计算 $V=\eta+\sum_{j=1}^{c}v_j\cdot m_j$，$\sigma=\prod_{j=1}^{c}\delta_j^{v_j} \bmod N$ 和 μ^{η}，共执行 c 次模加运算，$2(c-1)$ 次模乘运算和 c 次模幂运算。而新方案中服务器执行的计算有 $\theta=H_1(\mu^{\eta})$ 、 $V=\eta+\theta\cdot\sum_{j=1}^{c}v_j\cdot m_j$ 和 $\sigma=\prod_{j=1}^{c}\delta_j^{v_j} \bmod N$，其中，服务器执行 c 次加法运算、$2c-1$ 次乘法运算、1 次 Hash 运算和 c 次幂运算。

在 Verify 算法中，文献[13]方案中的 TPA 需验证等式 $\sigma^e=\prod_{j=1}^{c}H_2(Fn\|i_j)^{v_j}\mu^V\cdot\mu^{-\eta} \bmod N$，共需要执行 Z_N 中的 $c+1$ 次模乘运算，1 次模逆运算，c 次 Hash 运算和 $c+2$ 次模幂运算。在本文的方案中 TPA 计算 $\left(\sigma^e\right)^{\theta}=\prod_{j=1}^{c}H_2(Fn\|i_j)^{v_j\cdot\theta}\mu^V\mu^{-\eta} \bmod N$，需要在 Z_N 中执行 $2c+2$ 次模乘运算，1 次模逆运算，$c+1$ 次 Hash 运算和 $c+2$ 次模幂运算。

可以看出在 ProofGen 算法和 Verify 算法中，本文的新方案与文献[13]方案具有相同的计算复杂度。

4.4 实验结果

本文的方案与文献[13]方案在 TagGen 和 Challenge 阶段的计算时间相同，下面给出两种方案在 ProofGen 和 Verify 两种算法中的性能实验比较。本文使用 Java 语言对两种方案的 ProofGen 算法和 Verify 算法进行实现。模拟在 Win7 系统中实现，处理器为 Intel 3230 M，主频为 2.6 GHz，内存为 8 GB，选择的文件大小为 1 MB，分块大小为 4 KB，指定 TPA 每次分别随机选择不同数量的数据块检查外包数据的完整性。为了减少误差，每个阶段记录 100 次执行时间，最终取平均值。表 2 给出了 TPA 每次随机选择 200 个数据块检查外包数据完整性的实验结果对比情况，可以看出两种方案在 ProofGen 算法和 Verify 算法中具有几乎相同的运行时间。

表 2 TPA 选择 200 个挑战块计算时间

算法	文献[13]方案/ms	新方案/ms
ProofGen	≈592	≈594
Verify	≈561	≈569

当 TPA 选择不同的挑战块数目时，图 2 给出了在 ProofGen 算法中新方案和文献[13]方案的计算时间比较，可以看出新方案和文献[13]方案的计算成本几乎相同。

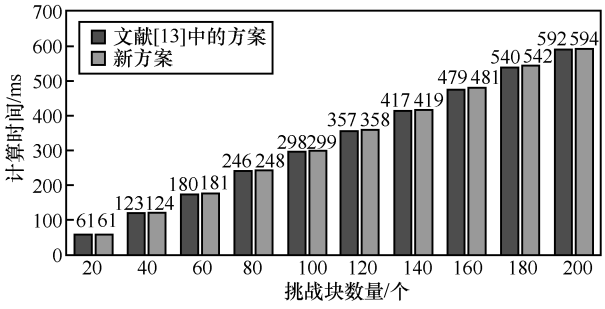

图 2 ProofGen 阶段的计算时间对比

同样地，当 TPA 选择不同的挑战块数量块时，两种方案中 Verify 算法的计算时间比对如图 3 所示。从图 3 中可以看出新方案额外计算开销与选取数据块的数量线性相关。在一次完整性验证过程中选取的挑战块数量有限，新方案在可以提供更高的安全需求的同时，具有和 Xu 等方案基本一致的计算开销。

表 1 计算成本比较

算法	文献[13]方案	新方案
TagGen	$(n+1)T_{\text{mul}}+T_{\text{inv}}+(n+1)T_{\text{hash}}+(2n+1)T_{\text{exp}}$	$(n+1)T_{\text{mul}}+T_{\text{inv}}+(n+1)T_{\text{hash}}+(2n+1)T_{\text{exp}}$
Challenge	$T_{\text{mul}}+T_{\text{inv}}+3T_{\text{hash}}+2T_{\text{exp}}$	$T_{\text{mul}}+T_{\text{inv}}+3T_{\text{hash}}+2T_{\text{exp}}$
ProofGen	$(c-1)T_{\text{add}}+(2c-1)T_{\text{mul}}+cT_{\text{exp}}$	$cT_{\text{add}}+(2c-1)T_{\text{mul}}+1T_{\text{hash}}+cT_{\text{exp}}$
Verify	$(c+1)T_{\text{mul}}+T_{\text{inv}}+cT_{\text{hash}}+(c+2)T_{\text{exp}}$	$(2c+2)T_{\text{mul}}+T_{\text{inv}}+(c+1)T_{\text{hash}}+(c+2)T_{\text{exp}}$

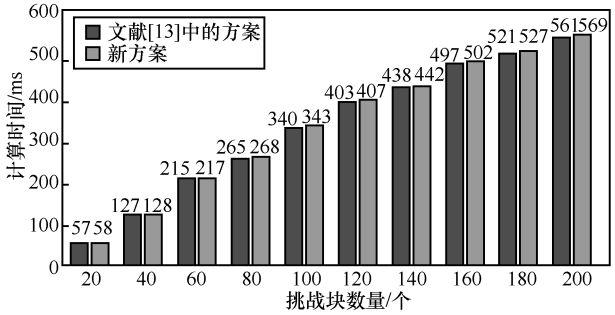

图 3　Verify 阶段的计算时间对比

5　结束语

在云存储应用环境下，用户无法确保存储在远端云服务器中的数据是完整的，如何验证远程数据的完整性是云安全领域研究的一个热点问题。本文分析了两种基于身份的云存储数据完整性验证方案的安全性，分析结果指出文献[12]的方案易遭受密钥恢复攻击，云服务器可以利用存储的数据信息计算得到用户的私钥；而文献[13]不能提供健壮性的安全需求，云服务器即使在丢失数据的情况下依然可以伪造证据通过 TPA 的验证。基于文献[13]的设计方法，本文提出一个新的改进方案，分析表明新方案在提供与文献[13]相同通信和计算开销的条件下，能提供更高的安全需求。针对基于身份密码系统下，云存储数据完整性验证方案存在的密钥泄露问题，外包数据的动态更新等问题将是我们下一步的研究重点。

参考文献：

[1] DESWARTE Y, QUISQUATER J, SAIDANE A. Remote integrity checking[M]//Integrity and Internal Control in Information Systems VI. 2004: 1-11.

[2] OPREA A, REITER M K, YANG K. Space-efficient block storage integrity[C]// Internet Society, Network and Distributed System Security Symposium. 2005: 1-12.

[3] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//ACM Conference on Computer and Communications Security. 2007: 598-609.

[4] ATENIESE G, BURNS R, CURTMOLA R, et al. Remote data checking using Provable data possession[J]. ACM Transactions on Information & System Security(TISSEC), 2011, 14(1): 1-34.

[5] JUELS A. Pors: proofs of retrievability for large files[C]// ACM Conference on Computer and Communications Security. 2007: 584-597.

[6] SHACHAM H, WATERS B. Compact proofs of retrievability[C]// Asiacrypt LNCS 5350. 2008: 90-107.

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

[8] SHAMIR A. Identity-based cryptosystems and signature schemes[C]// Crypto. 1984: 47-53.

[9] BONEH D, FRANKLIN M K. Identity based encryption from the weil pairing[J]. Siam Journal on Computing, 2001, 32(3): 213-229.

[10] ZHANG J, DONG Q. Efficient ID-based public auditing for the outsourced data in cloud storage[J]. Information Sciences, 2016, 343-344(C):1-14.

[11] YU Y, XUE L, MAN H A, et al. Cloud data integrity checking with an identity-based auditing mechanism from RSA[J]. Future Generation Computer Systems, 2016, 62(C):85-91.

[12] ZHANG J, LI P, SUN Z, et al. ID-based data integrity auditing scheme from RSA with resisting key exposure[C]// International Conference on Provable Security. 2016: 83-100.

[13] XU Z, WU L, KHAN M K, et al. A secure and efficient public auditing scheme using RSA algorithm for cloud storage[J]. Journal of Supercomputing, 2017(4):1-25.

[作者简介]

王少辉（1977−），男，山东潍坊人，博士，南京邮电大学副教授，主要研究方向为密码学、信息安全。

潘笑笑（1993−），男，安徽宿州人，南京邮电大学硕士生，主要研究方向为云安全。

王志伟（1976−），男，江苏扬州人，博士，南京邮电大学教授，主要研究方向为信息安全、密码学。

肖甫（1980−），男，湖南邵阳人，博士，南京邮电大学副教授，主要研究方向为无线传感器网络、信息安全等。

王汝传（1943−），男，安徽合肥人，博士，南京邮电大学教授，主要研究方向为信息安全、无线传感器网络等。