doi:10.3969/j.issn.1000-436x.2014.z2.003

支持动态更新的多副本持有性证明方案

陈龙，罗玉柱
（重庆邮电大学 计算机取证研究所，重庆 400065）

摘 要：考虑多副本数据安全和数据动态更新的应用需求，提出一个支持数据动态更新的多副本数据持有性证明方案。本方案中原数据文件采用动态认证结构进行动态更新与管理，其他多个副本采用追加日志记录的方式记录数据的动态更新，支持公开聚合验证。若原数据文件或副本数据损坏或丢失，可恢复到最新状态。由此分析了方案的安全性、通信性能、存储性能，结果表明新方案是高效的、安全的。
关键词：云存储；动态更新；数据完整性；多副本
中图分类号：TP309.2 **文献标识码**：A **文章编号**：1000-436X(2014)Z2-0014-06

Multi-replica provable data possession scheme supporting data dynamic updating

CHEN Long, LUO Yu-zhu
(Institute of Computer Forensics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Considering the application demand of multiple replicas data security and data dynamic update, a multi-replica provable data possession scheme was proposed to supporting data dynamic updating. The original data file can be dynamically updated using dynamic verification structure. The other multiple-replica data files are just append with a log-record for each dynamic update operation. The multi-replica files support public auditability and batch verification. If the original data or any replica is corrupted or missing, it can be recovered to the latest status with another copy. The performances of the communication and storage as well as the security are analyzed. The results show that the scheme is efficient and secure.
Key words: cloud storage; dynamic update; data integrity; multiple-replica

1 引言

越来越多的用户将他们的数据迁移到云中，但云服务提供商(CSP)不完全可信，对于用户很少访问的数据，CSP 可能将其转移到非在线存储设备上，甚至将这些数据删除以节省存储开销，同时若用户数据损坏或丢失，CSP 可能向用户隐瞒事实，以逃避赔偿和维护自身的信誉。为此，研究人员提出了数据持有性证明协议(PDP, provable data possession)来检查云存储中数据的正确性和完整性。关于数据持有证明的研究，Zhu 等人[1]提出了交互式可证明数据持有协议(IPDP)，并为私有云构建了一个零知识 IPDP 协议，能够支持数据动态更新、公开可验证性和数据隐私性。他们也提出了一个高效的协作可证明数据持有协议，其适用于混合云。Wang 等人[2,3]利用 Merkle 散列树支持数据动态更新和使用 BLS 签名[4]支持公开可验证性，同时保护数据隐私性。Hao 等人[5]通过使用 RSA 同态验证标识构建了一种远程数据完整性验证协议，支持数据动态更新、公开可验证性和数据隐私性，不需要使用第三方审计者。

为提高数据的可靠性和可用性，用户可要求

收稿日期：2014-10-23
基金项目：重庆市自然科学基金资助项目(cstc2011jjA40031)；国家社会科学基金资助项目(14BFX156)
Foundation Items: The Natural Science Foundation Project of Chongqing (cstc2011jjA40031); The National Social Science Foundation Project of China (14BFX156)

CSP 对其重要数据进行多副本存储，即在位于不同地理位置的多个服务器上存储数据，以提高容灾抗毁和数据可恢复的能力。为此，数据持有性证明对所有副本进行检查，以确认各数据副本均被正确持有，并可以在发现某副本失效时，利用其他副本及时恢复。Curtmola 等人[6]提出了一种 MR-PDP 协议，但只有数据所有者才能对数据完整性进行验证。Hao 和 Yu[7]提出了一种基于 BLS 验证标识，适用于多副本存储的远程数据持有性验证协议，其能够支持公开可验证性和数据隐私性。Barsoum 等人[8]提出一种 PB-PMDP 协议，可以公开验证用户所有外包存储数据的完整性。现有的多副本持有性证明方案都未考虑实际应用中，数据动态更新时多副本支持更新的重要需求，实际上也不能支持动态更新条件下的数据容错安全及数据恢复。

本文借鉴数据库事务日志的思想，将动态数据更新安全、静态数据多备份安全结合起来，提出一种云环境下支持数据动态更新的多副本持有性证明方案。原数据采用原有动态数据更新方案，另生成若干副本采用静态数据方案，并且结合日志思想将动态更新日志数据追加到副本文件的后面。可单独验证原文件及副本文件的安全，也可一次性聚合验证。本文方案同时满足数据动态更新需求和多副本数据安全需求，支持第三方公开验证，如果原始数据丢失或损坏，可将其恢复到最新状态。

2　系统框架

2.1　预备知识

相关远程数据持有性证明协议中所用的同态认证标签都基于 BLS 签名[4] ,其中用到双线性映射。

双线性映射 $e:G\times G\to G_T$，其中，G 为 Gap Diffie-Hellman (GDH)群，G_T 为素数阶 p 的乘法循环群，映射 e 具备以下特点：1)高效可计算性；2)双线性：对任意的 $h_1,h_2\in G$ 和 $a,b\in Z_p$，有 $e(h_1^a,h_2^b)=e(h_1,h_2)^{ab}$；3)非退化性：$e(g,g)\neq 1$，$g$ 为 G 的一个生成元。

数据完整性验证方案中，可以采用默克尔散列树、认证跳表等动态认证结构来支持动态操作及其验证。本文以动态默克尔散列树作为示例来讨论相关问题。

2.2　网络架构模型

云存储具有 3 个不同的网络实体，分别是用户、云服务提供商和第三方审计者，具有代表性的云存储网络架构如图 1 所示。

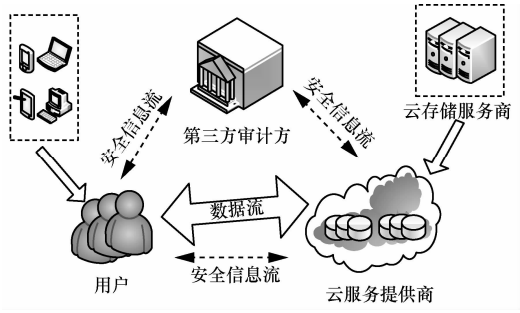

图 1　云存储网络架构

1) 用户(user)：数据文件的拥有者，有大量的数据文件需要存储在云端，并将数据的维护、管理和计算等任务委托给云服务器，用户可以是个体消费者或者公司、组织；

2) 云服务提供商 CSP(cloud storage provider)：管理云存储服务器，存储用户的数据文件，拥有巨大的存储空间和计算资源以管理用户的数据文件；

3) 第三方审计者 TPA(third party auditor)：收到用户的请求后，代表用户评估云存储服务的安全性，拥有云用户不具备的专业知识和能力。

3　支持动态更新的多副本持有效证明方案

3.1　方案概述

本文方案的基本思路为: 用户用一种对称加密算法将原文件加密，然后利用流加密再次加密得到多个不同的副本，从而可抵抗服务器间的合谋攻击，将副本数据文件上传至多个不同的存储服务器。为数据块生成同态验证标签，可实现对各副本的聚合验证，并减小验证过程中的数据传输量，在数据块标签中加入块位置信息和副本编号，以抵抗 CSP 的替换和重放攻击。在利用动态默克尔散列树对上传至存储服务器的原数据文件进行动态更新操作后，生成日志记录，日志记录包括操作日期、操作类型、操作针对的数据块序号及操作针对的数据块更新后的内容。采用对称加密法以及流加密法分别对日志记录进行加密，生成对应于副本数据文件的多份不同的副本日志记录，将副本日志记录上传至对应的存储服务器，以使副本数据文件和副本日志记录组成更新后的副本数据文件。从而实现多副本数据的动态更新，且数据可公开验证。如果原始数据丢失或损坏，可将其恢复到最新状态。

3.2 符号定义

支持动态更新的多副本持有性证明方案中的主要参数及说明如表 1 所示。

表 1 参数说明

参数	说明
n	文件分块数
r	每个块的子块数
s	文件副本数
K	对称加密密钥
k_r	流加密密钥
c	挑战数据块数

方案中的函数包括(以k'表示密钥长度):

$f(\cdot):\{0,1\}^* \to Z_p$ 为伪随机函数(pseudo- random function，PRF) 函数，由用户选定，并告知服务方和第三方;

$\psi(\cdot):\{0,1\}^{k'} \times \{0,1\}^{\mathrm{lb}^{(s)}+\mathrm{lb}^{(n)}+\mathrm{lb}^{(r)}} \to Z_p$ 为伪随机函数，用于生成伪随机数;

$E(\cdot):\{0,1\}^{k'} \times \{0,1\}^* \to \{0,1\}^*$ 为对称加密算法，用于对文件加密;

$H(\cdot):\{0,1\}^* \to G$ 为随机预言模型散列函数。

3.3 主要算法

支持动态更新的多副本持有性证明方案由(KeyGen、ReplicaGen、TagBlock、LogGen、LogApp、ProofChal 、 ProofGen 、 ProofVeri 、 ReplicaRes 、 ReplicaUpd) 10 个算法组成。

1) KeyGen

参数初始化算法。用户选择一个随机数$\alpha \leftarrow Z_p$和$r+1$个随机的元素$u_j \leftarrow G$， $j=\{0,1,2,\cdots,r\}$，其中u_0用于日志记录数据块第 0 个子块的标签计算，并计算$v=g^{\alpha}$，用户选定文件对称加密密钥K，用于加密原文件。用户选择决定副本数量 s，随机选择一个数$k_r \leftarrow Z_p$作为流加密密钥，便于密钥管理。从而私钥$sk=(\alpha,K,k_r)$和公钥$pk=(g,v,\{u_j\}_{j=0,1,\cdots,r})$。用户将公钥发给 CSP 和 TPA，私钥自己保存。

2) ReplicaGen

副本生成算法。

① 用户利用对称加密密钥K加密原数据文件F得到$F'=E(K,F)$， F'与F文件长度相同。

② 用户将F'分为 n 个子块，$F'=(c_1,c_2,\cdots,c_n)$每个子块c_i再分为分成 r 个基本块$c_i=(c_{i,1},c_{i,2},\cdots,c_{i,r})$。

③ 用户计算生成随机数$\gamma_{i,j}^{(k)}=\psi(k_r,i\|j\|k)$，$k=\{1,2,\cdots,s\}$， $i=\{1,2,\cdots,n\}$， $j=\{1,2,\cdots,r\}$。

④ 用户加密生成 s 个副本$F_k=(m_{i,j}^{(k)})=(c_{i,j}\oplus\gamma_{i,j}^{(k)})$，$k=\{1,2,\cdots,s\}$，$i=\{1,2,\cdots,n\}$，$j=\{1,2,\cdots,r\}$。

⑤ 用户记录原文件和副本文件的数据块长度信息，分别用L_O和L_R记录，初始值都为n。

3) TagBlock

标签生成算法。用户根据式(1)计算多副本每个数据块的签名$\sigma_i^{(k)}$，其中，$\omega_i^{(k)}=f(i\|k)$， $i=\{1,2,\cdots,n\}$， $k=\{1,2,\cdots,s\}$。

$$\sigma_i^{(k)}=(H(\omega_i^{(k)})\cdot\prod_{j=1}^{r}u_j^{m_{i,j}^{(k)}})^{\alpha} \tag{1}$$

简要说明一下原文件生成签名的过程，详细过程可以参考文献[9]，本文中将原文件视为第 0 个备份$F_{k=0}=F$。用户将原文件分割为n块，每块分为r块，$F_0=\{m_{i,j}^{(0)}\}$， $i=\{1,2,\cdots,n\}$， $j=\{1,2,\cdots,r\}$。根据式(2)计算原文件每个数据块的签名$\sigma_i^{(0)}$。

$$\sigma_i^{(0)}=(H(m_i^{(0)})\cdot\prod_{j=1}^{r}u_j^{m_{i,j}^{(0)}})^{\alpha} \tag{2}$$

用$\boldsymbol{\Phi}^{(k)}=\{\sigma_i^{(k)}\}_{i=1,2,\cdots,n}$表示签名集合。用户将$F_k$($k=\{0,1,\cdots,s\}$)和签名集合$\boldsymbol{\Phi}^{(k)},k=\{0,1,\cdots,s\}$上传到位于不同地理位置的 s+1 个服务器，接收到 CSP 的肯定回答后，用户删除所有相关信息。

4) LogGen

日志记录生成算法。日志记录数据块分为 4 个字段（操作时间、操作类型、数据块位置编号、数据块内容）：①操作时间，即动态操作执行具体日期及时间。②操作类型。根据动态操作的不同分为 3 种：插入、删除和更新，对应的代号分别为：I（Insert）、D（Delete）和 U（Update）。③执行操作的数据块位置编号，相对于动态更新的原数据文件 F，确定每次动态操作针对的数据块位置编号。④数据块内容。当执行插入操作时，记录新的数据块内容，当执行更新操作时，则记录更新的数据块内容，执行删除操作，则记录数据为全 0 的数据块。

a. 每次原数据文件进行动态更新操作后，这部分数据处理操作细节可以查看文献[9]，用户生成一条日志记录d，根据副本文件的数据块数量

信息 L_R，这条日志记录将追加为副本文件第 L_R+1 块数据。

b. 用户利用 K 加密得 $d'=E(K,d)$，d' 与 d 长度相同。

c. 将每条日志记录 d' 前 3 项内容组合为新增日志记录数据块的第 0 个子块，第 4 项数据块内容分为 r 个子块，即 $d'=\{c_{i,0},c_{i,1},\cdots,c_{i,r}\}$，$i=L_R+1$。

d. 用户加密生成 s 个副本对应的日志记录 $D_k=(m_{i,j}^{(k)})=(c_{i,j}\oplus\gamma_{i,j}^{(k)})$，其中，$\gamma_{i,j}^{(k)}=\psi(k_r,i\|j\|k)$，$k=\{1,2,\cdots,s\}$，$i=L_R+1$，$j=\{0,1,\cdots,r\}$。每条日志记录对于服务器属于固定长度的数据块，无实质区别。

5) LogApp

日志记录追加算法。用户将日志记录作为操作记录数据块追加到 s 个副本后，组成新的副本数据文件。

① 用户根据式(3)，计算每个副本日志记录数据块 D_k 的签名，其中，$\omega^{(k)}=f(i\|k)$，$k=\{1,2,\cdots,s\}$，$i=L_R+1$，$j=\{0,1,\cdots,r\}$。

$$\sigma^{(k)}=(H(\omega^{(k)})\prod_{j=0}^{r}u_j^{m_{i,j}^{(k)}})^{\alpha} \tag{3}$$

② 向 s 个服务器上传相应副本的追加新增数据块（对应的日志记录）和签名。

③ 上传完成后，进行数据完整性验证。验证通过说明用户追加日志记录数据块和签名成功，用户更新原文件和副本文件的数据块长度信息，根据用户执行的动态操作类型，插入、删除或更新，原文件数据块长度信息 L_0 对应操作为加一、减一或不变，副本文件的数据块数量信息 L_R 都执行加一操作，用户删除日志记录信息。

6) ProofChal

挑战生成算法。本方案可以通过聚合验证信息一次性验证原数据和多个副本的完整性，验证方 TPA 结合原文件和副本文件的数据块长度信息，随机生成数据块长度信息的子集，即 c 个元素的集合 $\{s_1,s_2,\cdots,s_c\}$，其中元素代表每个文件中被挑战数据块的位置，假设 $s_1\leqslant\cdots\leqslant s_c$，同时随机生成对应的挑战集合 $v_i\leftarrow B\subseteq Z_p$。发送挑战请求 $chal\{(i,v_i)\}_{s_1\leqslant i\leqslant s_c}$ 给证明方 CSP。

7) ProofGen

证据生成算法。当收到挑战 $chal=\{(i,v_i)\}_{s_1\leqslant i\leqslant s_c}$，CSP 根据式(4)和式(5)计算 $\mu_j^{(k)}$ 和 σ，其中针对原数据数据块，$j=\{1,2,\cdots,r\}$，针对每个副本数据块，$j=\{0,1,2,\cdots,r\}$。对应计算结果 $\{\sigma,\mu_j^{(k)}\}_{k=0,1,\cdots,s}$ 和一些关于原文件的辅助验证证据，这部分请参考文献[9]。CSP 将其作为证据发送 TPA 进行数据完整性验证。

$$\mu_j^{(k)}=\sum_{i=s_1}^{s_c}v_i m_{i,j}^{(k)} \tag{4}$$

$$\sigma=\prod_{k=0}^{s}(\prod_{i=s_1}^{s_c}(\sigma_i^{(k)})^{v_i}) \tag{5}$$

8) ProofVeri

证据验证算法。验证方 TPA 通过验证等式(6)是否成立，验证通过，说明 CSP 正确持有原文件和所有副本。

$$e(\sigma,g)=e(\prod_{i=s_1}^{s_c}H(m_i^{(0)})^{v_i}\cdot\prod_{j=1}^{r}u_j^{\mu_j^{(0)}},v)\cdot \prod_{k=1}^{s}e(\prod_{i=s_1}^{s_c}H(\omega_i^{(k)})^{v_i}\cdot\prod_{j=0}^{r}u_j^{\mu_j^{(k)}},v) \tag{6}$$

式(6)的正确性证明如下

$$\begin{aligned}
e(\sigma,g)&=e(\prod_{k=0}^{s}(\prod_{i=s_1}^{s_c}(\sigma_i^{(k)})^{v_i}),g)\\
&=e(\prod_{i=s_1}^{s_c}(\sigma_i^{(0)})^{v_i}\cdot\prod_{k=1}^{s}(\prod_{i=s_1}^{s_c}(\sigma_i^{(k)})^{v_i}),g)\\
&=e(\prod_{i=s_1}^{s_c}(\sigma_i^{(0)})^{v_i},g)e(\prod_{k=1}^{s}(\prod_{i=s_1}^{s_c}(\sigma_i^{(k)})^{v_i}),g)\\
&=e(\prod_{i=s_1}^{s_c}(H(m_i^{(0)})\cdot\prod_{j=1}^{r}u_j^{m_{i,j}^{(0)}})^{\alpha v_i},g)\cdot\\
&\quad e(\prod_{k=1}^{s}(\prod_{i=s_1}^{s_c}(H(\omega_i^{(k)})\cdot\prod_{j=0}^{r}u_j^{m_{i,j}^{(k)}})^{\alpha v_i}),g)\\
&=e(\prod_{i=s_1}^{s_c}H(m_i^{(0)})^{v_i}\cdot\prod_{j=1}^{r}u_j^{\sum_{i=s_1}^{s_c}v_i m_{i,j}^{(0)}},g^{\alpha})\cdot\\
&\quad e(\prod_{k=1}^{s}(\prod_{i=s_1}^{s_c}H(\omega_i^{(k)})^{v_i}\cdot\prod_{j=0}^{r}u_j^{\sum_{i=s_1}^{s_c}v_i m_{i,j}^{(k)}}),g^{\alpha})\\
&=e(\prod_{i=s_1}^{s_c}H(m_i^{(0)})^{v_i}\cdot\prod_{j=1}^{r}u_j^{\mu_j^{(0)}},v)\cdot\\
&\quad\prod_{k=1}^{s}e(\prod_{i=s_1}^{s_c}H(\omega_i^{(k)})^{v_i}\cdot\prod_{j=0}^{r}u_j^{\mu_j^{(k)}},v)
\end{aligned}$$

9) ReplicaRes

数据恢复算法。当云端的原数据发生不可恢复

的损坏或丢失时，这时用户可以利用任何一份副本数据，将其恢复到最新状态。恢复操作流程如下：

① 进行数据完整性验证，用户下载一份通过验证的副本数据文件。

② 用户根据文件对称加密密钥 K 和流加密密钥 k_r，将副本数据解密，恢复出原数据块和日志记录数据块。

③ 根据日志记录格式，按顺序读取日志记录，根据日志记录针对原数据顺序执行完所有日志记录的动态更新操作，即可得到此前最新状态的数据文件。

10) ReplicaUpd

副本更新算法。当云端的任何一个副本数据发生不可恢复的损坏或需要更新时，用户可以下载一份完好的副本数据，按日志将数据恢复到最新状态后，重新采用原方法处理后上传到服务器端，删除原始副本数据，完成副本更新。

4 安全性分析

4.1 安全分析

安全性证明主要包括2个方面。第一，如果CSP没有存储全部副本，则不能产生有效的验证证据。第二，如果CSP存储的副本数据出错，通过完整性验证的概率是可以忽略的。

定理1 若CSP能够生成有效的验证证据通过TPA的ProofVeri过程，则它必须存储所有副本数据块。

由于在数据块标签中加入块位置信息和副本编号，CSP无法伪造标签，进而无法进行替换和重放攻击。此数据完整性验证过程的安全性证明可参考文献[3]，本文省略。原文件加密并不影响其验证安全性。

定理2 若任意服务器所存储的数据损坏或丢失，则证明方通过完整性验证的概率是可以忽略的。

文献[10]已论证，当存储在服务器上的数据块损坏或丢失率为1%时，则验证方随机挑战 $c=460$ 个数据块就能以高于99%的概率发现数据损坏或丢失行为。为此，本文认为支持动态更新的多副本安全可验证方案在抽样检查中是高置信概率的，在数据块被损坏或丢失的情况下证明方仍能通过检查的概率是可以忽略的。

定理3 在不知道对称加密、流加密密钥的情况下，无法从一个备份推导出另一备份，数据安全强度不会降低。

本方案和现有多副本持有性证明方案比较，本方案通过记录每次动态操作生成日志记录，追加到多副本数据中，从而保证多副本数据支持实时动态更新，在云端原数据丢失后可以直接利用副本数据将数据恢复到最新状态。现有方案中多副本不支持动态更新，本方案弥补了以上不足。

4.2 性能分析

1) 公开验证性

本方案中，原数据文件和多副本数据可通过同态标签进行聚合验证，验证过程支持第三方公开验证，这部分请参考文献[9]。

2) 计算开销

在本方案的10个算法中，KeyGen、ReplicaGen和TagBlock都只在用户端执行一次，在数据动态更新操作时，LogGen、LogApp、ProofChal、ProofGen和ProofVeri验证方和证明方执行，ReplicaRes和ReplicaUpd只在数据损坏或出错时执行。

通过分析，以上算法中主要的计算开销为群 G 上的模乘运算（计算开销记为 T_M）、模幂运算（计算开销记为 T_E）和双线性对运算（计算开销记为 T_P）。因此，用户进行系统初始化时的主要开销为 $(r+1)T_E+(r+1)nKT_E+nKT_M$，与副本数、分块数和子块数分别成正线性关系；CSP生成验证证据的主要开销为 $(c+1)KT_E+(c+1)KT_M$，与副本数和被挑战块数成正线性关系；当用户或TPA验证证据的主要开销为 $(rK+r+c)T_E+2T_M+(K+1)T_P$，与副本数、子块数和被挑战块数成正线性关系；当原文件数据块动态更新时，用户追加日记记录的主要开销为 $(r+1)KT_E+KT_M$，与副本数和子块数分别成正线性关系。

3) 存储开销

在本方案中，用户将多副本数据及标签发送给CSP后，就可以删除本地所有副本数据和标签，只保留私钥 $sk=(\alpha,K,k_r)$、原文件和副本文件的数据块长度信息 L_O、L_R 及随机函数 $H(\cdot)$ 和 $f(\cdot)$。若完整性验证由TPA完成，则TPA只需保存公钥 $pk=(g,v,\{u_j\}_{j=0,1,\cdots,r})$ 以及函数 $H(\cdot)$ 和 $f(\cdot)$。CSP的各服务器只需要存储一个副本数据文件和该副本文件的标签集。

4) 通信开销

本方案利用双线性签名聚合属性，在基于BLS

方案中，可以将任意长度的信息聚合签名，数据完整性验证过程中，验证方和证明方之间 2 次通信的主要内容是 $chal\{(i,v_i)\}_{s_1\leqslant i\leqslant s_c}$、$\{\sigma,\mu_j^{(k)}\}_{k=0,1,\cdots,s}$ 和一些关于原文件的辅助验证证据，将这多个签名聚合成一个单独的短签名，一次将所有副本数据文件全部验证，从而大大降低了通信代价，通信数据量只比一份数据的动态更新方案增加了 $\{\mu_j^{(k)}\}_{k=1,2,\cdots,s}$，$\mu_j^{(k)}\in Z_p$，二者通信开销基本相当。

5　结束语

本文将多副本数据安全和数据动态更新的需求结合，提出一种云环境下支持动态更新的多副本持有性证明方案。该方案在支持实际应用的动态更新需求的基础上保证多副本数据安全。出现数据损坏或丢失时，可利用任意副本恢复数据到最新状态，保护用户数据安全。数据完整性验证方面，原数据采用动态数据完整性验证方案，副本数据采用静态数据完整性验证方案，把动态方案和静态方案有机地结合起来，同时支持公开验证和用户数据隐私保护。支持原数据和多副本数据聚合验证，方便用户管理多副本数据，降低了通信代价。分析了方案的安全性、通信性能、存储性能，结果表明新方案是高效的、安全的。

参考文献：

[1] ZHU Y, WANG H, HU Z, *et al*. Efficient provable data possession for hybrid clouds[A]. Proceedings of the 17th ACM Conference on Computer and Communications Security[C]. 2010.756-758.

[2] WANG Q, WANG C, REN K, *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.

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

[4] BONEH D, LYNN B, SHACHAM H. Short signatures from the weil pairing[A]. Advances in Cryptology—ASIACRYPT 2001[C]. Springer Berlin Heidelberg, 2001: 514-532.

[5] HAO Z, ZHONG S, YU N. A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability[J]. IEEE transactions on Knowledge and Data Engineering, 2011, 23(9): 1432-1437.

[6] CURTMOLA R, KHAN O, BURNS R, *et al*. MR-PDP: Multiple-replica provable data possession[A]. The 28th International Conference on Distributed Computing Systems[C]. 2008.411-420.

[7] HAO Z, YU N. A multiple-replica remote data possession checking protocol with public verifiability[A]. 2010 Second International Symposium on Data Privacy and E-Commerce (ISDPE)[C]. 2010. 84-89.

[8] BARSOUM A F, HASAN M A. Integrity verification of multiple data copies over untrusted cloud servers[A]. Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)[C]. IEEE Computer Society, 2012.829-834.

[9] CHEN L, SONG W, YIU S, *et al*. Ensuring dynamic data integrity based on variable block-size BLS in untrusted environment[J]. JDCTA, 2013, 7(5):837-846.

[10] ATENIESE G, BURNS R, CURTMOLA R, *et al*. Provable data possession at untrusted stores[A]. CCS '07[C]. 2007.598-609.

作者简介：

陈龙（1970-），男，重庆人，重庆邮电大学教授，主要研究方向为计算机取证、网络安全、云计算、智能信息处理等。

罗玉柱（1991-），男，山西霍州人，重庆邮电大学硕士生，主要研究方向为云计算安全。