云数据持有性审计研究与进展

田 晖[1] 陈羽翔[1] 黄永峰[2] 卢 璈[3]
(华侨大学计算机科学与技术学院 厦门 361021)[1] (清华大学电子工程系 北京 100084)[2]
(华侨大学网络技术中心 厦门 361021)[3]

摘 要 作为云计算的重要分支,云存储以高性能和低成本等优势吸引了越来越多的组织和个人将大规模数据托管于其上。然而,云数据的外包特性和近年来频繁爆出的安全事件,使得用户对云存储服务的信心不足,其关键问题是如何确保存储在云端的数据的完整性。为应对该挑战,云数据持有性审计在最近几年被提出并受到了广泛的关注,文中对此进行了综述。首先,回顾了云数据持有性审计的一般模型和审计系统的设计目标;其次,按照实现的审计功能,对近年来的研究成果进行了分类介绍及对比分析;最后,指出了云数据持有性审计研究中存在的开放问题及发展趋势。

关键词 云存储,数据持有性,公开审计,云安全
中图法分类号 TP309.7 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.06.002

Research and Development of Auditing Techniques for Cloud Data Possession

TIAN Hui[1] CHEN Yu-xiang[1] HUANG Yong-feng[2] LU Jing[3]
(College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)[1]
(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)[2]
(Network Technology Center, Huaqiao University, Xiamen 361021, China)[3]

Abstract As an important branch of cloud computing, cloud storage possesses the advantages of low-cost and high-performance, and has attracted a growing number of organizations and individuals to outsource their data. However, due to the outsourcing characteristic of cloud data and frequent security accidents for cloud storage providers, users are lacking of confidence in the cloud storage server, of which the main problem is how to effectively check the integrity of cloud data. To overcome this challenge, the auditing for cloud data possession has been proposed and gotten widespread attention in recent years, and a comprehensive survey was provided in this paper. First, general models and design goals of the auditing for cloud data possession were reviewed. Second, the existing auditing schemes for cloud data possession were classified according to their auditing functions, their principles were analyzed, and their performances were compared. Finally, the open problems in the auditing for cloud data possession were identified, and the trends of future development were discussed.

Keywords Cloud storage, Provable data possession, Public auditing, Cloud security

1 引言

云存储是在云计算概念上延伸和发展出来的一个重要分支,其目标是利用云计算技术将大量不同类型的存储设备协同起来,提供可靠、弹性、高性能的数据存储服务[1]。由于云存储具有容量大和成本低的优势,越来越多的组织和个人都倾向于将大规模的数据托管于云服务提供商(Cloud Service Provider,CSP)。然而,长期以来,用户对云存储服务的安全性和可靠性总是存在不同程度的怀疑[2]。这一问题的根源在于云存储的数据外包特性使得用户无法以常规的方式去控制数据的访问与使用。此外,近年来频繁爆出的云数据安全事故(如亚马逊云服务器宕机事故[3]、Gmail 邮件丢失[4]、iCloud 中的数据被黑客删除[5]等)也使得用户对 CSP 的不信任程度进一步加剧。如何增强 CSP 的可信性以及用户对云存储服

到稿日期:2016-11-11 返修日期:2017-01-02 本文受国家自然科学基金项目(U1405254,U1536115,61302094),福建省高校新世纪人才支持计划(MJK2016-23),福建省高校杰出青年科研人才培育计划(MJK2015-54),福建省自然科学基金项目(2014J01238),国家留学基金(201507540001),华侨大学中青年教师科研提升资助计划(ZQN-PY115),华侨大学科技创新团队和领军人才支持计划(2014KJTD13)资助。
田 晖(1982—),男,博士,副教授,主要研究方向为网络与信息安全、云存储安全及多媒体内容安全等,E-mail:htian@hqu.edu.cn;陈羽翔(1992—),女,硕士生,主要研究方向为云存储安全;黄永峰(1967—),男,博士,教授,博士生导师,主要研究方向为网络与信息安全、大数据安全与隐私保护、云计算安全及多媒体内容安全等;卢 璈(1984—),女,硕士,工程师,主要研究方向为网络与信息安全。

务的信心已成为云存储技术和产业发展亟待解决的重要问题之一。为应对该挑战，云数据安全审计技术在最近几年被提出并受到了广泛的关注，其核心内容是如何有效验证云端数据的安全性和完整性[6-13]。

在云存储环境下，由于数据未被保存在本地，若使用传统的完整性验证方法则需将数据全部取回，这在当前大数据背景下显然是低效且不安全的。因而，以相关密码学理论为支撑、以云数据完整性（持有性）的远程验证为目标的安全审计技术应运而生[14-47]，并在近年得到了长足的发展。本文首先回顾了已有的云数据持有性审计模型及其目标，进而对不同类型的云数据（如归档数据、动态数据、多副本数据和共享数据等）持有性审计方法的研究现状进行了介绍和分析，最后对云数据持有性审计未来的发展趋势和挑战进行了总结和展望。

2　云数据持有性审计模型及目标

从已有文献来看，云数据持有性审计模型可分为两类：私有审计模型与公有审计模型。前者主要由用户承担审计工作，而后者则可引入独立的第三方作为审计者。

私有审计模型（见图 1）是最早出现的云数据持有性审计架构，主要包含两个实体[14-17]：CSP 与用户。CSP 管理并协同大规模的云服务器，使之提供稳定且高效的数据外包服务。用户是云存储服务的使用者，他们可将数据托管于 CSP 以最小化本地的存储和计算开销，但同时也希望能够定期地对外包数据的完整性进行验证。此种审计模型中，用户自身充当审计者的角色，审计过程由用户和 CSP 通过多次交互完成，审计架构相对简单。然而，该模型存在明显的不足：1）作为交易的双方，用户和 CSP 存在利益关系，任何利益方作为审计角色都将影响其结果的公正性和权威性[19]；2）审计工作需要频繁、定期地进行，这会给用户带来较大的额外开销，特别是使用移动智能终端（如手机、平板电脑等）时，其开销将会给用户带来极大的负担[23]。

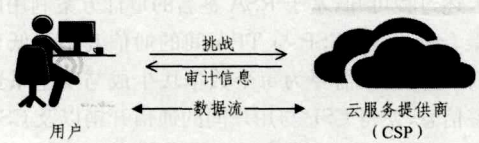


图 1　云数据持有性私有审计模型

鉴于私有审计模型存在的上述问题，近年来的研究普遍采用基于第三方的公开审计模型，如图 2 所示[18-25]。

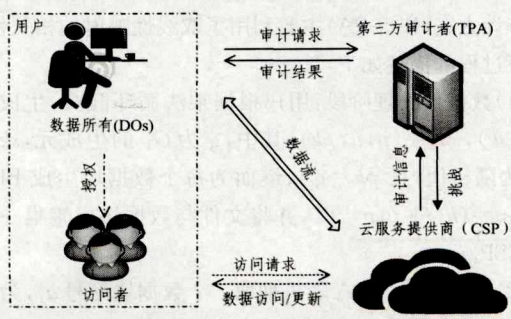


图 2　云数据持有性公开审计模型

该模型引入可信第三方（Third Party Auditor，TPA）来承担审计工作，以增强审计结果的可信性和权威性，并同时减轻用户负担。在公开审计模型中，TPA 根据用户的请求定期对外包数据的完整性进行审计并向用户反馈审计结果。然而，为保护用户隐私，审计方案应确保 TPA 在审计过程中不能获得关于数据内容的任何信息[24-25]。

此外，在云数据持有性审计方案的设计中，CSP 通常被视作是不可信的，尤其是当数据完整性因各种事故遭到破坏时，CSP 为通过审计可能会发起如下攻击。1）伪造攻击：CSP 为了隐瞒数据的丢失或破坏，伪造 TPA 需要的审计信息；2）取代攻击：CSP 使用数据文件中的正确部分取代出错部分，从而生成相关审计信息，以便通过验证；3）重放攻击：当数据出错时，CSP 企图将过去生成的审计信息反馈给 TPA 以通过审计；4）合谋攻击：CSP 中的多台服务器合谋或多用户共享情形下，CSP 与被撤销访问权限的用户合谋对数据或审计信息进行伪造。显然，为了实现安全的审计，云数据持有性审计方案应能有效地抵抗上述攻击。此外，从审计功能和效率上来说，理想的审计方案还需实现如下几个目标。

1）隐私保护：审计方案需保证 TPA 无法获知任何涉及隐私的数据信息。

2）支持批量审计：为提高审计效率，TPA 应能同时响应来自不同用户、不同云的多个审计需求。

3）支持多副本审计：在多副本存储的情形下，审计方案应能保证存储在 CSP 中的所有副本都是完整和正确的。

4）支持动态数据审计：审计方案应支持数据的动态更新，并能够对数据的完整性和新鲜度进行有效验证。

5）支持共享数据审计：审计方案应支持多用户数据共享情形下的数据更新、用户管理及数据完整性审计。

6）开销最小化：在确保审计结果正确性的同时，审计方案应尽可能减少计算和通信开销。

3　云数据持有性审计技术

如上所述，基于第三方的公开审计已成为云数据持有性审计的发展趋势，并涌现了许多富有成效的审计方案。一般而言，云数据持有性审计通常包含两个阶段：预处理（Setup）和挑战（Challenge）。预处理是用户在数据上传至云端前的初始化工作，其过程可描述为：用户生成公钥 PK 与密钥 SK，将待上传的文件 F 分块，并为每个数据块生成标签作为审计证据，而后将数据块及其对应标签上传至 CSP，并删除本地除密钥外的所有数据。挑战是指 TPA 应用户要求或定期对存储在 CSP 上的云数据进行审计，其过程为：TPA 随机选取若干数据块作为审计对象，并将挑战信息发送给 CSP；CSP 收到挑战请求后，生成指定数据块和对应标签的审计证据，并将其反馈给 TPA；TPA 通过验证所收到的审计证据来判断云数据的完整性。然而，已提出的方案在审计特性或审计功能方面各有侧重（见表 1）。本节将从公开审计和隐私保护[24-25]、批量审计[26-27]、多副本审计[28-32]、动态数据审计[33-40]及共享数据审计[41-47]等角度对研究进展进行回顾和分析。

表 1 云数据持有性审计方案的功能比较

审计方案	公开审计	动态数据	批量审计	多副本	可共享性	数据隐私保护	身份隐私保护	安全假设
CPOR[16]	√	×	×	×	×	×	—	RSA
SPDP[17]	×	√	×	×	×	×	—	RSA
PDP[18]	√	×	×	×	×	×	—	RSA
PPDP[19]	√	×	×	×	×	√	—	DLP
CL-PDP[20]	√	×	×	×	×	×	—	DLP
ID-RDP[21]	√	×	×	×	×	×	—	DLP
3P-PDP[24]	√	×	×	×	×	√	—	DLP
DAP[27]	√	√	√	×	×	√	—	DLP
MR-PDP[28]	×	×	×	√	×	√	—	RSA
BLS-PDP[29]	√	×	×	√	×	√	—	DLP
MF-RDC[30]	√	√	×	√	×	√	—	DLP
DM-DC[31]	√	√	×	√	×	√	—	DLP
2M-PDP[32]	√	×	√	√	×	√	—	DLP
DPDP[33]	×	√	×	×	×	×	—	RSA
MHT-PA[34]	√	√	√	×	×	√	—	DLP
FU-DPA[35]	√	√	×	×	×	√	—	DLP
IHT-PA[36]	√	√	—	×	×	√	—	DLP
DPA-FA[37]	√	√	√	×	×	√	—	DLP
DHT-PA[38]	√	√	√	×	×	√	—	DLP
MuR-DPA[39]	√	√	×	√	×	√	—	DLP
TB-PMDDP[40]	√	√	×	√	×	√	—	DLP
3P-ASD[41]	√	×	×	×	√	×	√	DLP
SM-PDP[42]	√	×	×	×	√	×	√	DLP
Knox[43]	√	×	×	×	√	√	√	DLP
Oruta[44]	√	√	√	×	√	√	√	DLP
Panda[45]	√	√	√	×	√	×	×	DLP
PBA-PDP[46]	√	×	√	×	√	√	×	DLP

注："√"表示支持；"×"表示不支持；"—"表示未提及或未涉及；RSA 指 Rivest-Shamir-Adleman 公钥加密系统，DLP 指离散对数问题(Discrete Logarithm Problem)。

3.1 公开审计与隐私保护

在海量数据存储的背景下，将数据取回到审计方后进行审计的方式显然是极其低效和不安全的。因而，实现无需数据取回的审计是云数据审计方案的基本要求之一[18-19]。最初的无取回审计方案[15-16]的主要思路是：用户为每个数据块生成消息验证码(Message Authentication Code，MAC)，在审计时只需传递密钥和 MAC。然而，由于 MAC 的验证需要用户密钥，使得该方案只能由用户作为审计者，难以拓展到公开审计中；而且，方案中事先选择的 MAC 密钥的数目是有限的，使用完毕后密钥的重生成过程将会给用户带来极大的计算开销。

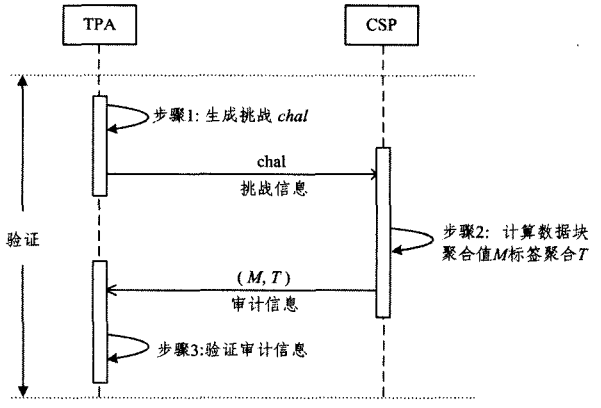


图 3 基于同态签名技术的公开审计过程

鉴于此，有研究者提出基于同态认证技术[48]的审计方案[18-42,45]，其思想是：用户为每个数据块生成不可伪造的同态标签，并随数据块一同传送到 CSP 上存储；在审计过程中(见图 3)，CSP 根据挑战需求反馈相应的数据块和标签的聚合值给 TPA；TPA 通过验证收到的审计证据来判断数据的完整性。相较于基于 MAC 的审计方案，基于同态认证的审计方案无需用户参与审计过程，减轻了用户的负担，并有效降低了审计过程中的通信开销(从 $O(n)$ 降低至 $O(1)$)，从而成为目前实现公开审计的最有效的途径。从已有文献看，根据采用的签名机制的不同，该类方案可细分为基于 RSA 签名的持有性审计[18,28,33]与基于 BLS 签名的持有性审计[19-27,29-32,34-42,45]。

基于 RSA 的持有性审计方案主要是利用了 RSA 签名的同态特性，具体过程如下。

(1)预处理阶段：用户生成密钥对 $PK=(N,g,pk)$，$SK=(sk)$，其中 N 为两个大素数 p 和 q 的 RSA 模数，g 为模 N 二次剩余集的生成元，随机数 pk 和 sk 满足 $pk \cdot sk \equiv 1 \bmod (p-1)(q-1)$；而后将文件 F 分块，即 $F=\{m_i \mid 0<I<n\}$，并为每个数据块生成 RSA 签名作为其对应标签，即 $\sigma_i=(h(i \parallel v)g^{mi})^{sk}$，其中 i 代表数据块标号，n 为数据块数目，v 为文件标识符，h 为哈希函数；最后将文件 F 以及数据块标签集 Ω 一同上传至 CSP。

(2)挑战阶段：通常采用抽样审计的方式。文献[18]已证明了若假设数据块出错概率为 1%，那么随机抽取 460(320)个数据块进行验证即可确保高达 99%(95%)的准确率。TPA 随机选择两个密钥 $k1$ 和 $k2$ 生成挑战信息 $chall=(c,k1,k2,gs)$，并将其发送至 CSP，其中 c 为抽取数据块数目，$gs=g^s$，s 为随机值。CSP 收到挑战信息后，首先计算 $a_i=f_{1k1}(i)$，$b_i=f_{2k2}(i)(0<i<c)$，其中 f_1，f_2 均为随机数生成函数，生成的 a_i 表示被抽样的数据块序号，b_i 表示每个数据块对应的随机值；继而计算数据块证据信息 $M=H(gs^D)$，$D=b_1m_{a1}+b_2m_{a2}+\cdots+b_cm_{ac}$，标签证据信息 $T=\prod_{0<i<c}\sigma_{ai}^{bi}$；最后将审计证据 (M,T) 发送至 TPA。TPA 收到证据后，首先计算 $t=t/h(a_i \parallel v)^{bi}(0<i<c)$，其中 $t=T^{sk}$，接着判断 $H(t')$ 与 M 是否相等。若两者相等，则验证通过；反之验证不通过。

从上述过程可知，基于 RSA 签名的审计方案利用同态签名的可聚合特性，将 CSP 与 TPA 间的通信开销降低至常数级。然而，该类方案需要为每个数据块生成与安全系数呈正比的标签信息，使得 CSP 与用户间的通信开销以及 CSP 对于标签的存储开销较大。BLS 是 Boneh 等人提出的一种新的签名技术[48]，在同等安全强度下，其签名长度较 RSA 签名更短，因而，有研究者提出用 BLS 签名代替 RSA 签名[24-25,34]，以降低通信和存储开销，并提高审计效率。基于 BLS 签名的公开审计方案(BLS-PA)主要利用了双线性映射的相关性质，其一般过程的描述如下[24-25,27]。

(1)数据预处理阶段：用户根据乘法循环群 G_1 生成密钥 $SK=(sk)$，$PK=(g,u,pk)$，其中，g 为 G_1 的生成元，$sk,u \in Z_p$ 均为随机值，且 $pk=g^{sk}$；继而为每个数据块生成 BLS 标签，即 $\sigma_i=(h(i \parallel v)g^{mi})^{sk}$，并将文件与数据块标签集一同存储至 CSP。

(2)挑战阶段：TPA 随机抽取 c 个数据块序号 a_i，为每个序号选取一个随机值 b_i，并将它们作为挑战 $chall=\{(a_i,b_i) \mid 0<i<c\}$ 发送到 CSP。收到挑战后，CSP 按照抽样序列分别

聚合数据块和标签，即 $T=\prod_{0<i<c}\sigma_{ai}^{b_i}$，$M=\sum_{0<i<c}b_i m_{ai}$，并将得到的聚合值$(T,M)$作为审计证据发送到 TPA。TPA 收到证据后，通过判断等式 $e(T,pk)=e(\prod_{0<i<c}h(a_i\parallel v)^{b_i}\cdot u^M,g)$ 是否成立来对数据完整性进行验证。若等式成立，则验证通过；否则验证不通过。

在公开审计方案中，由于 TPA 的引入，如何保护用户隐私在审计过程中不被泄露成为了一个需要重点解决的问题。虽然在上述基于同态认证技术的审计方案中 TPA 没有直接接触用户数据，但理论上 TPA 完全有可能通过求解线性方程组的方式从其收到的数据块聚合值中分析出用户原始数据的相关信息，从而使得用户隐私存在被泄露的风险。为应对这一挑战，Wang 等[24-25]提出将随机掩码植入到数据块聚合值中以防止 TPA 的逆向解析。具体来说，CSP 植入随机掩码的过程可表述为：$M'=M+rH(u^r)$，其中，u 为事先协商好的全局参量(global parameter)，$H(x)$为哈希函数。随后，CSP 将$\{M',r\}$作为审计信息发送给 TPA。随机掩码的引入不会影响数据完整性的验证，但 TPA 已无法通过求解线性方程组的方式获知任何的数据信息。此种保护用户隐私的机制也在其后的审计方案[27,36,38]中得到了广泛的应用。

3.2　批量审计与多副本数据审计

在公开审计中，TPA 经常同时收到来自多个用户的审计请求。若 TPA 将任务进行排队再逐一审计，其效率显然是不高的。因此，审计过程常采用批量审计的方式[26-27]，即利用同态标签的可聚合特性将不同审计请求产生的审计证据聚合后再一次性完成验证，其过程如图 4 所示。

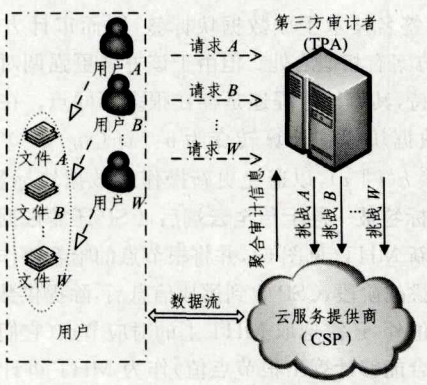


图 4　批量审计原理图

在基于 BLS 签名的审计方案中，对于存储在云端的 w 个不同用户的 w 个文件而言，批量审计过程一般可表述如下。

(1)数据预处理阶段：w 个用户分别产生他们的密钥$\{SK_i=(sk_i),PK_i=(u_i,g_i,pk_i)\mid 0<i<w\}$后，将文件分块并计算每个数据块的 BLS 标签；随后用户将所有数据块 $F=\{m_{ij}\mid 0<i<w,0<j<n\}$($n$ 为数据块数目)以及数据块对应标签$\Omega=\{\sigma_{ij}\mid 0<i<w,0<j<n\}$一同存储于 CSP。

(2)挑战阶段：TPA 同时收到来自 w 个用户的审计请求，即 $R=\{req_i\mid 0<i<w\}$，并依照前文所述生成挑战信息$chall=\{(a_i,b_i)\mid 0<i<c\}$，同时将其发送给 CSP 上存储了 w 个文件的服务器。CSP 对所有服务器返回的数据块和标签信息$\{(\sigma_{ij},m_{ij})\mid 0<i<w,0<j<c\}$分别计算标签证据 $\Phi=\prod_{0<i<w}(\prod_{0<j<c}\sigma_{i,aj}^{bj})$和数据块证据 $M_i=\sum_{0<j<c}b_j m_{i,aj}\ (0<i<w)$，并将证据信息$(\Phi,\{M_i\mid 0<i<w\})$发送到 TPA。TPA 收到证据信息后，判断等式 $e(\Phi,g)=\prod_{0<i<w}e(\prod_{0<j<c}h(v\parallel j)^{b_j}u^{M_i},pk)$是否成立。若成立则审计通过，反之审计不通过。

从上述过程不难看出，相比于逐一审计的方式，批量审计有如下优势：1)所有标签信息在传递给 TPA 之前就被聚合，有效地减少了通信开销；2)由于审计证据是在聚合后再进行一次性验证，因此减少了 TPA 做双线性映射运算的次数。简言之，批量审计不仅可有效提高 TPA 的审计效率，而且可减少 CSP 与 TPA 间的通信开销。然而，值得注意的是：在批量审计中，只有当所有用户数据均正确且完整时，“打包”处理的高效性才能体现；一旦有数据出错，审计将无法通过，此时定位出错数据将成为需要解决的一个新问题[9]。当然，最直接的解决措施是对各数据块逐一进行审计以找出错误，但该方式的处理效率显然是不高的。因此，如何快速定位出错数据仍是亟待解决的重要问题。

此外，云存储应用中，用户通常会要求采用多副本备份的方式提高其数据的可靠性[28]。不同于前述方案，多副本数据的审计既需要保证各副本的完整性，还需保证副本数目的正确性。由于所有副本数据的内容是一致的，如果用户将其直接存储在云端，不诚信的 CSP 只需持有少量甚至单个正确的副本即可通过审计。因此，在数据初始化阶段，需对多副本数据进行差别化处理。Curtmola 等[28]通过改进基于 RSA 签名的审计方案，首先提出了一种多副本数据审计的方案(MR-PDP)。在数据预处理阶段，用户秘钥和数据块对应标签的生成方式与前述基于 RSA 签名的审计方案相同。但为实现多副本数据的差别化，用户先使用秘钥 sk 将文件加密成 $F'=\{m_i'\mid 0<i<n\}$，然后利用随机掩码为之生成多个不同的副本数据块，即 $F_i'=\{b_{ij}\mid 0<i<w,0<j<n\}$，$b_{ij}=m_i'+r_{ij}$，其中 w 为副本数目，n 为数据块数目，r_{ij} 为随机数生成函数和用户秘钥共同作用生成的随机掩码。在挑战阶段，审计者依次验证每一个副本的完整性。其挑战 $chall$ 和证据(T,M)的生成过程均与上述基于 RSA 签名的审计方案一致；所不同的是，由于引入了随机掩码，审计者收到证据后需要先对标签聚合值进行处理：$T=T\cdot g^{r_{chall}}$，$r_{chall}=\sum_{0<i<c}r_{ai}$，再做验证。该方案初步解决了多副本数据审计的问题，但仍存在如下不足：1)审计阶段所要用到的信息 r_{chall} 是用户密钥生成的掩码累加值，因而审计工作不能交由除用户外的其他实体完成，即不支持公开审计；2)对多个副本文件需逐一审计，其效率显然是不高的。

随后，Barsoum 等[31]提出了一种基于 BLS 签名的多副本公开审计方案。该方案通过加密的方式实现了副本数据的差别化，并采用类似批量审计的方式通过单次交互验证多副本数据的持有性。在数据预处理阶段，用户需要为给定文件 F 生成指定个数的副本$\{F_i'\mid 0<i<w\}$，其中每个副本由用户将 F 与其副本序号拼接并加密得到，即 $F_i'=E_{sk}(F\parallel i)$，$i$ 为副本序列号，sk 为用户秘钥，E 为加密方法。此处秘钥生成、标签生成等过程与前述 BLS-PA 相同。在挑战阶段，挑战信息将发送到所有存储有副本的服务器；CSP 将所有副本的数据块与标签分别聚合，其过程为：$\Phi=\prod_{0<i<w}(\prod_{0<j<c}\sigma_{i,aj}^{bj})$，$M_i=\sum_{0<j<c}$

$b_j m_{i,aj}$ $(0<i<w)$，其中 $m_{i,aj}$ 表示第 i 个副本的第 a_i 个数据块，其他变量与前述 BLS-PA 一致。CSP 最后将 $(\Phi,\{M_i \mid 0<i<w\})$ 作为审计证据发送给 TPA。TPA 收到审计信息后，通过判断等式 $e(\Phi,g)=e((\prod_{0<j<c} h(v\|j)^{bj})^w, pk)$ 是否成立来对多副本持有性进行验证。若成立则审计通过，否则审计不通过。与前述 MR-PDP 方案[26]相比，该方案具有如下优势：1)审计过程无需用户参与，从而可支持公开审计；2)审计过程通过 TPA 与 CSP 的一次交互完成，相较于 MR-PDP 的逐一审计，有效地降低了通信开销和计算开销。然而，该方案中实现副本区别化的加密、方式开销较大，特别是对于频繁更新的动态数据，反复地加密、解密显然不是一个理想的选择。而且，上述两种方案均不支持动态多副本数据的审计，本文将在下节继续讨论此问题。此外，与批量审计类似，当所有副本数据都正确且完整时，现有方案所采用的“先聚合证据再审计”策略能显著提高审计效率；而一旦有副本出错，如何快速定位出错副本将成为一个新的值得深入研究的重要问题[31]。

3.3 动态数据审计

云存储环境中存在大量需频繁更新(需进行增、删和改操作)的数据，称为动态数据。传统的基于静态数据(或称归档数据)的审计方案不能直接应用此类数据，其原因在于：1)传统审计方案中数据块标签的计算过程 $\sigma=(h(i\|v)g^{mi})^{sk}$ 涉及数据块的序号值 i，而数据块的增、删操作会引起序号值的变化，并最终导致相关数据块标签需要重新生成，从而给用户带来较大的额外开销；2)频繁更新操作使得数据块的版本信息不断变化，审计过程不但要验证数据的完整性，还需确保数据的新鲜度(即最新版本)。鉴于此，需设计支持数据动态性的云数据持有性审计方案[33-40]。表 2 列出了几种代表性的动态数据持有性审计方案的性能比较。

表 2 动态数据持有性审计方案的性能比较

审计方案	通信开销	计算开销				检测率
		验证		更新		
		CSP	审计者	CSP	DO/TPA	
DPDP[33]	cO(log$n$)	cO(log$n$)	cO(log$n$)	tO(log$n$)	tO(log$n$)	$1-(1-v)^c$
MHT-PA[34]	cO(log$n$)	cO(log$n$)	cO(log$n$)	tO(log$n$)	tO(log$n$)	$1-(1-v)^c$
FU-DPA[35]	cO(log$n$)	cO(log$n$)	cO(log$n$)	tO(log$n$)	tO(log$n$)	$1-(1-v)^{c\cdot s}$
IHT-PA[36]	O(c+s)	O(c+s)	O(c+s)	O(t)	O($t\cdot n$)	$1-(1-v)^{c\cdot s}$
DAP[27]	O(c)	O(c)	O($c\cdot$s)	O(t)	O($t\cdot n$)	$1-(1-v)^{c\cdot s}$
DHT-PA[38]	O(c)	O(c)	O($c\cdot$s)	O(t)	O($t\cdot n$)	$1-(1-v)^c$
MuR-DPA[39]	cO(log$w\cdot n$)	cO(log$w\cdot n$)	cO(log$w\cdot n$)	tO(log$w\cdot n$)	tO(log$w\cdot n$)	$1-(1-v)^c$

注：n 为文件的数据块数目；s 为每个数据块的分段数；c 是审计的数据块数目；v 是文件错误率；t 为更新数据块数目。对于错误率为 v 的文件，抽样审计 c 个数据块($c\cdot s$ 个数据段)，至少一个数据块(段)被检测到的概率为 $1-(1-v)^c$ $(1-(1-v)^{c\cdot s})$。

从已有文献[33-40]来看，将认证数据结构与审计算法相结合是实现动态数据审计的有效途径。Erway 等人[33]首先引入了跳表(skip list)机制(见图 5)，设计了一种支持数据动态更新的云数据持有性审计方案(DPDP)。该方案的数据预处理过程与基于 RSA 签名的审计方案类似。在审计阶段，CSP 接受挑战后先计算数据块聚合值 M，而后根据跳表生成数据块验证路径 $\gamma=\{\sigma_{ai},\varsigma_{ai} \mid 0<i<c\}$(包含被审计数据块的标签以及对应跳表中的路径)，并将上述证据信息 (M,γ) 发送给审计者。收到审计证据后，审计者先验证数据块与标签是否匹配，并进一步通过路径和标签信息对数据块的新鲜度进行验证。DPDP 方案是第一种支持全部数据更新操作的审计方案，但仍存在如下不足：1)数据块对应的验证路径过长，涉及的辅助变量很多，再加上每个被审计数据块都需要定义验证路径，使得 CSP 与审计者间通信的开销较大；2)由于 DPDP 在验证跳表中对应数据块路径时需使用数据块标签的生成密钥，使得该方案不能扩展到公开审计。

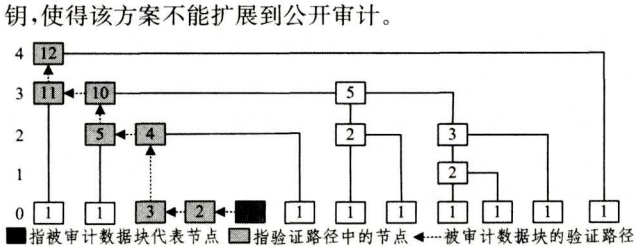


图 5 基于跳表的审计过程图

随后，Wang 等人[34]提出了一种基于 MHT (Merkle Hash Tree)的动态数据公开审计方案(MHT-PA)。该方案采用 BLS 签名技术生成数据块标签，因而审计方案与前述 BLS-PA 方案有相似之处。但由于该方案更强调对数据更新操作的支持，其审计过程还是存在很多不同点。在数据预处理阶段，数据块标签计算转变为 $\sigma=(h(m_i)g^{mi})^{sk}$，即使用 $h(m_i)$ 替换 $h(i\|v)$，以避免更新操作对数据块标签的影响。数据块与标签被一同上传至云端后，CSP 还需根据数据块的哈希值更新 MHT(见图 6)，并将根节点的哈希值作为元数据保存。在挑战阶段，CSP 收到挑战信息后，除提供数据块及标签的聚合值外，还需选取 MHT 上的对应节点(它们与被抽取数据块配合能够计算出根节点值)作为 MHT 审计路径信息 Ψ，一并发送给 TPA。

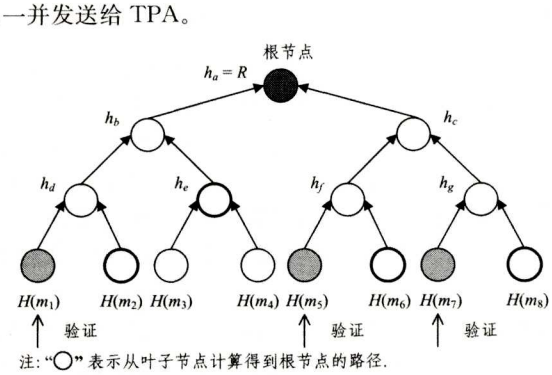


图 6 基于 MHT 的审计过程

如图 6 所示，假设被审计数据块为 $\{m_1,m_5,m_7\}$，那么路径信息 Ψ 应包括 MHT 中的节点 $\{H(m_2),H(m_6),H(m_8),h_e\}$。收到审计证据后，TPA 根据路径信息 Ψ 计算 MHT 的

根节点，若结果与 CSP 提供的元数据 R 一致，则继续按照与前述 BLS-PA 方案中相同的方式验证数据完整性，否则审计不通过。在此基础上，Liu 等人[35]通过在 MHT 上增加等级(Rank)信息的方式实现了支持细粒度数据更新的审计方案。基于 MHT 的审计方案尽管能够实现动态数据的公开审计，但在审计和更新效率上依然存在一些问题，如审计过程传递的 MHT 审计路径信息增加了 CSP 与 TPA 间的通信开销，更新过程中对 MHT 的维护会同时增加 CSP 与用户的负担等。

为进一步提高动态数据的审计和更新效率，Zhu 等人[36]提出了一种基于 IHT(Index Hash Table)的审计方案(IHT-PA)。IHT(见图 7)是一个存储于 TPA 上用于记录数据块新鲜度信息的一维表，数据块 m_i 的新鲜度信息 χ_i 包括数据块序号 B_i、版本信息 V_i 以及随机值 R_i，它们共同参与数据块标签的生成。具体而言，在数据预处理阶段，用户首先生成数据块 m_i 的新鲜度信息 $\chi_i=\{B_i,V_i,R_i\}$，并进一步计算其标签 $\sigma_i=(h(B_i\|V_i\|R_i)u^{m_i})^{sk}$。与传统方案一样，将数据块及其对应标签上传至 CSP，而新鲜度信息则存储到位于 TPA 上的 IHT 中。每次有数据块被更新后，其新鲜度和对应的标签都将被重新生成。该方案的挑战阶段涉及的相关操作与前述 BLS-PA 方案相似。TPA 接收审计证据后，先按照挑战信息从 IHT 中选取抽样数据块的新鲜度信息 $\Lambda=\{\chi_{ai}=(B_{ai},V_{ai},R_{ai})\mid 0<i<c\}$，并通过验证等式 $e(T,g)=e(\prod_{0<i<c}h(B_{ai}\|V_{ai}\|R_{ai})^{b_i}\cdot u^M,pk)$ 是否成立来进行审计。如果等式成立，则审计通过，否则审计不通过。

No.	B_i	V_i	R_i	
0	0	0	0	← 表头
1	1	2	r_1'	← 更新
2	2	1	r_2	
3	4	1	r_3	← 删除
4	5	1	r_5	
5	5	2	r_5'	← 插入
⋮	⋮	⋮	⋮	
n	n	1	r_n	
$n+1$	$n+1$	1	r_{n+1}	← 追加

注：No. 指序号，B_i 是数据块的编号，V_i 是数据块的版本号，R_i 是随机数。

图 7　IHT 示意图

在 IHT-PA 方案中，用户直接将新鲜度信息保存在 TPA 的 IHT 中，因此，只要数据块及其标签未按要求更新，验证过程就会因为标签与 IHT 中的新鲜度信息不匹配而无法通过，从而同时保证了动态数据的完整性和新鲜度。此外，较上述 MHT-PA 方案，IHT-PA 将新鲜度信息直接存储于 TPA 的做法还大大减少了审计过程中的通信量和审计更新过程中的计算量。然而，该方案仍存在更新效率不高的问题：1)由于数据块标签与数据块序号相关，因此当数据块插入和删除操作引起序号改变时，将造成大量的数据块标签需重新生成，为用户带来较大的计算开销；2)由于 IHT 的顺序结构，其上的数据更新操作尤其是 IHT 上的插入和删除操作，将导致表中平均半数的元素被移动，因而更新效率不高。鉴于此，文献[38]设计了一种新的二维数据结构 DHT(Dynamic-Hash Table)，用于存储云数据的新鲜度信息。DHT(见图 8)将文件记录按类似数组的结构进行管理，而每个文件记录通过指针指向其包含的数据库记录链表。每个文件记录包括给定文件的索引号(NO_i)和文件标识符(ID_i)，而数据块记录包括对应数据块的版本号 $v_{i,j}$ 和更新的时间戳 $t_{i,j}$。相比于 IHT，DHT 的优势在于：1)在数据块的新鲜度信息中不包括数据块序号，从而避免了由于数据块的插入和删除操作而引起的大量数据块标签的重生成；2)采用链表结构对数据块新鲜度记录进行管理，从而能支持高效的数据更新操作。

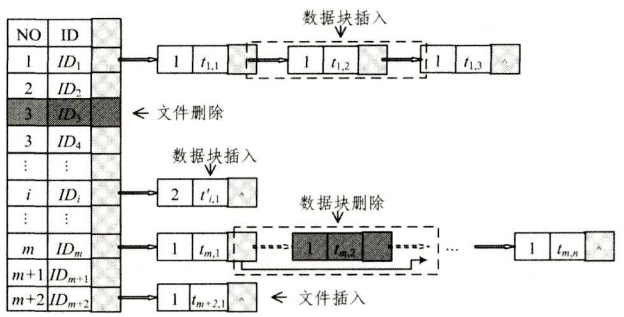
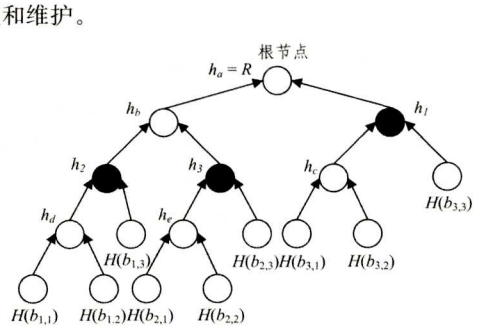


图 8　DHT 示意图

尽管上述方案主要以普通的动态数据为审计对象，但它们所涉及的审计数据结构也可扩展到动态多副本数据的审计。文献[39]和文献[40]分别提出了基于改进 MHT 和 IHT 的动态多副本数据审计方案。由于将 MHT 直接应用到多副本审计中会造成 CSP 需为每个副本维护一个 MHT，从而使得 TPA 的审计开销及 TPA 与 CSP 间的通信开销均与副本数目成正比，其效率显然是不高的。为此，文献[39]将多个 MHT 整合为一棵树，称为 MR-MHT(Multiple Replica-Merkle Hash Tree)，如图 9 所示，该树的叶子节点即为各副本数据块的哈希值，但同一副本的各数据块用同一个子树进行管理和维护。



注："●"指同一数据块的副本；"○"代表数据块的子树；$H(b_{i,j})$ 为第 i 个副本第 j 个数据块的哈希值；h 表示中间节点的哈希值。

图 9　MR-MHT 示意图

相较于多个单独 MHT，这种组织方式既可提高管理和维护的效率，又可降低 CSP 保存这些审计信息的存储开销。文献[40]中提出的改进 IHT 被称为 MVT(Map-version Table)。MVT 与 IHT 结构类似，但摒弃了 IHT 中每个数据块记录中包含的数据块序号，从而避免了更新操作可能导致的标签重生成。由于更新操作对所有副本相同，数据新鲜度信息可由多个副本共享，因此一个 MVT 即可实现对动态多副本数据审计的支持。与 MR-MHT 方案相比，基于 MVT 的审计方案用于存储新鲜度信息的存储空间时与副本数目无关，并且其审计过程中所涉及的通信开销和计算开销更小。

3.4 共享数据审计

随着部门内的协同工作开始在云上部署，多用户共享(即存储在云端的数据将支持被多个用户访问并且修改)已成为云数据的一大新特性[41]。由于共享数据的特殊性，针对它的审计方案不仅需要保证数据的完整性，还需考虑用户组的管理(用户组变动)[45-47]及用户的个人隐私保护[41-44]等问题。用户组变动的主要影响在于，一旦有用户退出，所有涉及其签名的数据块标签都需要重新生成。如何避免这些数据块标签的重生成给组内用户带来过大的额外开销，是需要解决的重要问题。共享数据中的隐私保护问题不仅涉及数据内容本身的隐私，而且还需保护用户的行为隐私，如避免 TPA 在审计过程中得知哪些用户频繁地修改数据块。

为应对上述挑战，Wang[41]等提出一种基于代理签名的共享数据审计方案——3P-ASD。该方案的主要特点在于：1)无论数据何时被何人修改，其标签都由同一密钥生成，并且使用同一公钥即可完成审计，因此 TPA 不能通过审计时用户公钥的使用情况得知用户的行为隐私；2)每当用户退出，数据块标签的重生成都交由 CSP 使用代理签名完成，从而避免了组内用户额外的计算开销。该方案的具体过程可概括为：在数据预处理阶段，数据拥有者(Data Owner，DO)也即用户组管理者生成钥对 $PK=(u,g,pk)$ 和 $SK=(sk)$，并广播给组内所有用户，同时利用密钥为各数据块生成标签并将其上传至 CSP。当数据块被修改时，修改者使用同一密钥重新计算标签。一旦有用户退出用户组，DO 将重新生成钥对 $PK'=(u',g',pk')$ 和 $SK'=(sk')$，并广播告知其他用户，同时 DO 将新旧密钥的比值 $q=sk'/sk$ 发送给 CSP，并要求其通过代理签名重新生成数据块标签，即 $\sigma'=(\sigma)^q=((h(i\|v)g^{mi})^{sk})^{sk'/sk}=(h(i\|v)g^{mi})^{sk'}$。该方案挑战阶段所做的工作与前述 BLS-PA 方案类似，在此不再赘述。值得指出的是，3P-ASD 方案尽管通过统一用户密钥实现了对用户个人隐私信息的保护，但仍存在如下问题：每当用户退出用户组，签名密钥和所有数据块标签将重新生成，而密钥的更替以及标签的重生成将增加用户与 CSP 的通信开销和计算开销。

鉴于此，Wang 等[45]进一步提出了一种共享数据审计方案 Panda。该方案与 3P-ASD 的不同之处在于：1)用户各自生成密钥而不是共用同一密钥；2)在 CSP 中将预先存储每两个用户密钥的比值，用于用户退出时标签的重生成；3)TPA 使用被审计数据块最后修改的公钥而不是使用同一公钥进行验证。具体来说，在数据预处理阶段，DO 和用户各自生成钥对，且 DO 为每个数据块生成初始标签随数据块一同上传到云端。当用户更新数据时，使用自身的密钥重新生成标签信息。一旦有用户退出，DO 选择其他用户接收退出用户处理过的数据块文件，并由 CSP 使用代理签名为之重新生成标签。举例来说，如果用户 A 要退出用户组并将其数据交由用户 B 管理，CSP 利用所存储的用户 B 与用户 A 的密钥的比值 $q=sk_B/sk_A$ 对用户 A 最后修改数据块的标签做幂运算，即 $\sigma_B=(\sigma_A)^q=((h(i\|v)g^{mi})^{skA})^{skB/skA}=(h(i\|v)g^{mi})^{skB}$，完成标签生成者由 A 至 B 的转变。从这一点来说，相比 3P-ASD 方案，Panda 在用户退出时不需要更新用户组密钥以及所有数据块标签，降低了 CSP 与用户间的通信开销和 CSP 上的计算开销。然而，该方法的安全性随后也受到质疑[46]：如果已退出的用户与 CSP 合谋，其密钥能与保存在 CSP 中的密钥通过比值计算出任意用户的密钥，那么数据标签将被轻易篡改或伪造。此外，由于被审计数据块的标签由不同用户密钥生成，TPA 能够通过审计时用户公钥的使用情况得知诸如哪些用户频繁地修改了数据块等隐私信息。

文献[46]提出了一种基于多项式认证机制的共享数据审计方案，其优势在于可聚合来自不同用户的审计信息以提高审计效率。该方案中，TPA 管理用户修改数据块的日志文件，并将被审计数据的修改记录与挑战一同发送到 CSP；CSP 按记录还原被审计数据块的标签，并将它们进行聚合后发送给 TPA。相比于基于同态签名机制的审计方案，该方案可减少审计时 TPA 的计算量以及一 CSP 的通信量，但在隐私保护方面有待改善，因为 TPA 能够很容易地从其管理的用户修改日志文件中获取用户隐私信息。

针对用户隐私保护的问题，Wang 等[44]提出了一种基于环签名的共享数据审计方案，称为 Oruta。与 Panda 方案类似，用户组中的用户仍然使用各自的密钥为修改后的数据块计算标签，但 Oruta 中的标签生成机制发生了改变，即由环签名代替了 Panda 中的 BLS 签名。基于环签名的标签生成过程可概括表述为：假设用户组成员人数为 d，每个用户都生成各自的钥对 $\{SK_j=(sk_j),PK_i=(u_j,g,pk_j)\mid 0<j<d\}$(所有用户的公钥 g 相同)，数据块 m_i 的环签名是一组拥有 d 个签名的集合 $\{\sigma_{i,j}\mid 0<j<d\}$，其中只有一个签名的生成方式与其他 $d-1$ 个不同。例如，如果数据块 m_i 被用户 s 修改，那么除去标签 $\sigma_{i,s}$ 外，其他标签的计算方法一致：$\sigma_{i,j}=g^{ki,j}$，$j\neq s(k_{i,j}\in Zp$是由用户生成的随机值)，而标签 $\sigma_{i,s}$ 的生成则需要用户 s 的密钥参与，即：

$$\sigma_{i,s}=(h(id)u^{m_i}/\varphi(\prod_{j=1,j\neq s}^{d}pk_j^{a_{i,j}}))^{1/sk_s}$$

其中，id 为数据块序列值。在挑战阶段，CSP 接收 TPA 发送的挑战 $chall$ 后，按照抽样序列分别聚合数据块以及标签，并将其作为证据传送到 TPA，其中数据块聚合值为 M，基于环签名的标签被分别聚合为 d 个：$T=\{T_j\mid 0<j<d\}$，其聚合方式与其他方案相似。收到证据信息后，TPA 通过判断等式 $\prod_{0<j<d}e(T_j,pk_j)=e(\prod_{0<i<c}h(id_{ai})^{bi}\cdot u^M,g)$ 是否成立给出审计结果。若等式成立，则审计通过；否则审计不通过。

从以上描述中可以看出，在 Oruta 中 TPA 使用全部用户的公钥即可完成验证，而不需对应最后修改这些数据块的用户公钥，从而使得 TPA 不能通过审计时用户公钥的使用情况获知用户的行为隐私。然而，当有用户退出用户组时，该方案中所有数据块的标签都将被重生成，因为每个数据块的标签都包含所有用户的公钥，任何用户钥对的变化都将造成所有标签的重生成。从这个意义上讲，Oruta 方案没能有效解决用户退出导致的数据块标签重生成开销较大的问题。表 3 列出了上述几种代表性的多用户共享数据持有性审计方案的功能和性能的比较。

表 3　共享数据持有性审计方案的性能比较

审计方案	用户组变动	个人隐私保护	通信开销	计算开销				检测率
				验证		用户退出		
				CSP	TPA	CSP\代理	DO	
3P-ASD[41]	√	√	$O(c+s)$	$O(c)$	$O(c+s)$	$O(m\cdot n)$	—	$1-(1-v)^c$
SM-PDP[42]	√	√	$O(c+s)$	$O(c)$	$O(c+s)$	$O(m\cdot n)$	—	$1-(1-v)^{c*s}$
Knox[43]	√	√	$O(c+s)$	$O(c)$	$O(c+s)$	—	$O(m\cdot n)$	$1-(1-v)^c$
Oruta[44]	√	√	$O(c+s+d)$	$O(c+s+d)$	$O(c+s+d)$	—	$O(m\cdot n)$	$1-(1-v)^{c*s}$
Panda[45]	√	×	$O(c+d)$	$O(c)$	$O(d\cdot c)$	$O(m^*)$	—	$1-(1-v)^c$
PBA-PDP[46]	√	×	$O(c+m^{**})$	$O(c\cdot s)$	$O(c)$	$O(m^*)$	—	$1-(1-v)^{c*s}$

注：d 是用户组用户数目；m 是存储的文件数目；m^* 指被退出用户修改的数据块数目；m^{**} 指上传后被修改的数据块数目。

4　未来的工作与挑战

经过近些年的长足发展，云数据持有性审计在审计模型和针对不同类型的云数据审计技术方面均取得了富有成效的研究成果。然而，随着云存储技术的发展和研究的不断深入，未来云数据持有性审计研究仍存在着如下挑战和有待进一步探索的问题。

4.1　细粒度的动态数据审计

现有的动态数据审计方案都是以数据块为更新粒度的，即其所有的增加、删除操作都必须以数据块为最小单位。然而，实际应用中（如微博数据）存在许多频繁而数据量很小的更新。此种情况下若使用现有的动态数据审计方法，将会带来非常大的存储开销。例如，对分块大小为 1MB 的文件做 n 次数据增加操作，且每次操作所添数据大小都为 1kB，但由于数据块为最小更新粒度，增加 nkB 的数据量将需要插入 nMB 的数据，这显然是极其低效的。因此，未来还需进一步研究支持细粒度更新的数据完整性审计方案。

4.2　多副本/批量审计的完备性

对多个用户审计请求和多个副本审计的批量操作是提高审计效率的有效方式。然而，此种操作方式的优势仅在所有用户数据或多副本数据都正确且完整的情况下才能体现；一旦审计不通过，即用户数据或副本数据出错时，这种操作方式将无法定位出错的用户文件或副本文件，使得审计方案不再完备。当然，转而对各用户的请求或多副本文件进行逐一审计是最简单和直接的方式，但是其效率显然是相当低下的。此外，文献[31]曾设想通过“二分查找”的方式进行定位，虽未实现，但是不难想象该方式的查找过程将涉及大量审计信息的多次聚合和验证操作，仍会给 CSP 和 TPA 带来较大的通信和计算开销。因此，如何快速、准确地定位出错的用户文件（或副本文件）仍是批量审计（多副本审计）中一个尚待解决的开放问题。

4.3　共享数据的高效审计

随着企业或部门内的协同工作开始在云上部署，组织或部门内部的云数据共享成为了云存储服务的一种常态化应用。针对越来越多的多用户共享云数据，研究与之相适应的数据持有性审计方案是一项重要的研究课题。如 2.4 节所述，已有研究者注意到此问题，并开展了积极的研究，但已有成果各有优缺点。目前尚无方案能够同时实现审计高效性、审计安全性、用户组的高效管理和用户行为隐私保护等目标。因此，未来仍需研究和探索更完善和高效的多用户共享数据的持有性审计方案。

4.4　多媒体云数据的高效审计

多媒体数据（如图片、音频以及视频）占用空间较大，是被上传至云端的常见数据类型之一，并且其由于在一次生成后基本上不做修改，因此可视作静态数据。然而，多媒体数据大多规模较大，采用现有的静态数据审计方法生成并验证同态标签将消耗大量计算资源，并不是最“对症”的审计方法。因此，针对多媒体数据活跃度较小的特点，可利用可逆透明水印来实现高效审计。通过将水印嵌入图像、音频或视频中作为审计证据，代替现有的基于同态标签技术的审计方案，解决标签计算量、存储量过大的问题。当然，在不影响数据完整性的前提下，如何提取作为审计证据嵌入的水印并进行高效的验证，是需要深入研究的重要问题。

4.5　面向大数据的高效审计

在大数据时代，海量数据区别于传统数据，具有 4V（Volume，Variety，Value，Velocity）特性。在此背景下，如何实现海量数据的高效审计是又一个有待深入研究的重要问题，其关键是如何实现“审计的高效性”和“功能的完备性”。我们认为可从如下两方面入手。

（1）根据云存储服务和大数据应用的发展需要，细分数据类型，针对特定安全风险和审计需求，精细定制审计算法，进而提高审计效率。数据的分类方法可以有多种：1）对于不同的数据种类，审计方法不同。如动态数据更新频繁，更强调数据的新鲜度；而归档数据体量巨大，则更注重数据的完整性。因此对不同的数据种类采用更合适的审计方案，能够有效地提高审计速度。2）针对不同的价值密度，审计方法不同，海量数据的多样性使得数据的价值密度也不尽相同。例如，一篇 PDF 文档与一段视频之间的价值密度差异极大。因此，对于价值密度不同的大规模数据，区别其审计方法才能实现审计的高效性。

（2）从审计系统的实际应用出发，建立可扩展的大数据持有性审计模型，实现多种不同审计方法之间的信息互补和功能协作，解决特定算法的审计功能单一问题，增强整个审计系统的功能和审计目标的完备性。

结束语　为增强用户对云存储服务的信心，云数据持有性审计在最近几年被提出并受到了广泛的关注，现已涌现出许多富有成效的研究成果。然而，从整体上来看，目前在云数据持有性审计方面的研究尚不成熟，尚未建立完整的理论和技术体系，且已有方案与审计系统的实际应用还有较大的距离，还需不断深入和完善。本文首先回顾了云数据持有性审计的私有审计模型和公开审计模型及其审计目标，进而从不同审计功能和数据类型的角度对近年来云数据持有性研究领

域的主要成果进行了分类归纳和总结，最后在对研究现状进行深入分析的基础上，对该领域未来的工作和发展趋势进行了总结和展望。

参 考 文 献

[1] DEWAN H, HANSDAH R C. A Survey of Cloud Storage Facilities[C]// Proceedings of the 7th IEEE World Congress on Services. 2011: 224-231.

[2] WANG C, WANG Q, REN K, et al. Toward Secure and Dependable Storage Services in Cloud Computing [J]. IEEE Transactions on Services Computing, 2012, 5(2): 220-232.

[3] Digital in 2008. Amazon S3 Availability Event [OL]. http://status. aws. amazon. com/s3-20080720. html.

[4] Digital in 2006. Reports of Mass Email Deletions [OL]. http://www. techcrunch. com/2006/12/28/gmail-disaster-reports-of-massemail-deletions.

[5] Digital in 2014. iCloud 被黑客攻击而泄密 引发"公有云"恐慌 [OL]. http://www. chinacloud. cn/show. aspx? id = 17748&cid=29.

[6] FENG C S, QIN Z G, YUAN D. Techniques of Secure Storage for Cloud Data [J]. Chinese Journal of Computers, 2015, 38(1): 150-163 (in Chinese)
冯朝胜，秦志光，袁丁. 云数据安全存储技术[J]. 计算机学报，2015, 38(1): 150-163.

[7] TAN S, JIA Y, HAN W H. Research and Development of Provable Data Integrity of Cloud Storage[J]. Chinese Journal of Computers, 2015, 38(1): 164-177. (in Chinese)
谭霜，贾焰，韩伟红. 云存储中的数据完整性证明研究及进展 [J]. 计算机学报，2015, 38(1): 164-177.

[8] CHEN L X, XU L. Research on Provable Data Possession and Recovery Technology in Cloud Storage [J]. Journal of Computer Research and Development, 2012, 49(S1): 19-25. (in Chinese)
陈兰香，许力. 云存储服务中可证明数据持有及恢复技术研究 [J]. 计算机研究与发展，2012, 49(S1): 19-25.

[9] WANG C, REN K, LOU W J, et al. Toward Publicly Auditable Secure Cloud Data Storage Services [J]. IEEE Network, 2010, 24(4): 19-24.

[10] YANG K, JIA X H. Data Storage Auditing Service in Cloud Computing: Challenges, Methods and Opportunities [J]. World Wide Web-internet & Web Information Systems, 2012, 15(4): 409-428.

[11] SOOKHAK M, GANI A, TALEBAIN H, et al. Remote Data Auditing in Cloud Computing Environments: A Survey, Taxonomy, and Open Issues [J]. ACM Computing Surveys, 2015, 47(4): 65.

[12] SOOKHAK M, TALEBAIN H, AHMED E, et al. A Review on Remote Data Auditing in Single Cloud Server: Taxonomy and Open Issues [J]. Journal of Network & Computer Applications, 2014, 43(5): 121-141.

[13] RYOO J, RIZVI S, AIKEN W, et al. Cloud Security Auditing: Challenges and Emerging Approaches [J]. IEEE Security & Privacy, 2014, 12(6): 68-74.

[14] SEBÉ F, DOMINGO-FERRER J, MARTÍNEZ-BALLESTÉ A, et al. Efficient Remote Data Possession Checking in Critical Information Infrastructures [J]. IEEE Transactions on Knowledge and Data Engeering, 2008, 20(8): 1034-1038.

[15] JUELS A, KALISKI J R B S. PoRs: Proofs of Retrievability for Large Files [C] // Proceedings of the 14th ACM Conference Computer and Communications Security. 2007: 584-597.

[16] SHACHAM H, WATERS B. Compact Proofs of Retrievability [C]// Proceedings of the 14th Theory and Application of Cryptology and Information Security: Advances in Cryptology. 2008: 90-107.

[17] ATENIESE G, PIETRO R D, MANCINI L V, et al. Scalable and efficient provable data possession [C]// Proceedings of the 4th International Conference on Security and Privacy in Communication Networks. 2008: 1-10.

[18] ATENIESE G, JOHNS R B, CURTMOLA R, et al. Provable Data Possession at Untrusted Stores [C]// Proceedings of the 14th Computer and Communications Security. 2007: 598-609.

[19] WANG H. Proxy Provable Data Possession in Public Clouds [J]. IEEE Transactions on Services Computing, 2013, 6(4): 551-559.

[20] WANG B Y, LI B C, LI H, et al. Certificateless Public Auditing for Data Integrity in the Cloud [C]// Proceedings of the IEEE Conference on Communications and Network Security. 2013: 136-144.

[21] WANG H K, WU Q H, QIN Bo, et al. Identity-based Remote Data Possession Checking in Public Clouds [J]. IET Information Security, 2014, 8(2): 114-121.

[22] YU J, REN K, WANG C, et al. Enabling Cloud Storage Auditing with Key-Exposure Resistance [J]. IEEE Transactions on Information Forensics and Security, 2015, 10(6): 1167-1180.

[23] LIU C, RAJIV R, ZHANG X Y, et al. Public Auditing for Big Data Storage in Cloud Computing--A Survey [C]// Proceedings of the 16th IEEE International Conference on Computational Science and Engineering. 2013: 1128-1135.

[24] WANG C, WANG Q, REN K, et al. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing [C]// Proceedings of the 29th IEEE International Conference on Computer Communications. 2010: 1-9.

[25] 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.

[26] ZHU Y, HU H X, GAIL-JOON A, et al. Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(12): 2231-2244.

[27] YANG K, JIA X H. 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.

[28] CURTMOLA R, KHAN O, BURNS R C, et al. MR-PDP: Multiple-Replica Provable Data Possession [C]// Proceedings of the 28th IEEE International Conference on Distributed Computing Systems. 2008: 411-420.

[29] HAO Z, YU N H. A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability[C]// Proceedings of the 2nd IEEE International Symposium on Data, Privacy and E-Commerce. 2010: 84-89.

（下转第 50 页）

[30] WANG C, WONG W. Extending the lifetime of NAND Flash memory by salvaging bad blocks[C]// Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2012.

[31] JUNG H, YOON K, SHIM H, et al. LIRS-WSR: Integration of LIRS and Writes Sequence Reordering for Flash Memory[J]. IEEE Transactions on Consumer Electronics, 2008, 54(3): 1215-1223.

[32] JO H, KAMG J U, PARK S Y, et al. FAB: flash-aware buffer management policy for portable media players[J]. IEEE Transactions on Consumer Electronics, 2006, 52(2): 485-493.

[33] KIM H, AHN S. BPLRU: a buffer management scheme for improving random writes in NAND Flash storage[C]// USENIX Association. 2008: 1-14.

[34] OU Y, HÄRDER T, JIN P. CFDC: a NAND Flash-aware replacement policy for database buffer management[C]// ACM. 2009: 15-20.

[35] PARK S, et al. CFLRU: a replacement algorithm for NAND Flash memory[C]// ACM. 2006: 234-241.

[36] KIM J, SHIM H, PARK S Y, et al. NAND FlashLight: a lightweight NAND Flash file system for embedded systems[J]. ACM Transactions on Embedded Computing Systems (TECS), 2012, 11(1): 18.

[37] SIMMONDS C. Linux NAND Flash file systems JFFS2 vs UBIFS[C]// Embedded Systems Conference UK. 2009.

[38] HAN C X, CHEN X L, XI L I, et al. Impact of UBIFS Wear—leveling on System I/O Performance[J]. Computer Engineering, 2009, 35(6): 260-262.

[39] BROWN N. JFFS2, UBIFS, and the growth of NAND Flash storage[EB/OL]. https://lwn.net/Articles/528617.

[40] KANG E, JACKSON D. Formal modeling and analysis of a NAND Flash filesystem in Alloy[M]// Abstract state machines, B and Z. Springer Berlin Heidelberg, 2008: 294-308.

(上接第 16 页)

[30] XIAO D, YANG Y, YAO W B, et al. Multiple-File Remote Data Checking for Cloud Storage [J]. Computers & Security, 2012, 31(2): 192-205.

[31] BARSOUM A F, HASAN M A. On Verifying Dynamic Multiple Data Copies over Cloud Servers [OL]. http://cacr.uwaterloo.ca/techreports/2011/cacr2011-28.pdf.

[32] CHEN H F, LIN B G, YANG Y, et al. Public Batch Auditing for 2M-PDP Based on BLS in Cloud Storage [J]. Journal of Cryptologic Research, 2014, 1(4): 368-378.

[33] ERWAY C C, KÜPÇÜ A, PAPAMANTHOU C, et al. Dynamic Provable Data Possession [C]// Proceedings of the 16th ACM Conference on Computer and Communications Security. 2009: 213-222.

[34] 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.

[35] LIU C, ZHANG X, CHI Y, et al. Authorized Public Auditing of Dynamic Big Data Storage on Cloud with Efficient Verifiable Fine-Grained Updates [J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(9): 2234-2244.

[36] ZHU Y, AHN G J, HU H X, et al. Dynamic Audit Services for Outsourced Storage in Clouds [J]. IEEE Transactions on Services Computing, 2013, 6(2): 227-238.

[37] JIN H, JIANG H, ZHOU K. Dynamic and Public Auditing with Fair Arbitration for Cloud Data [C]// IEEE Transactions on Cloud Computing. 2016: 1.

[38] TIAN H, CHEN Y X, CHANG C C, et al. Dynamic-Hash-Table Based Public Auditing for Secure Cloud Storage [C]// IEEE Transactions on Services Computing. 2016: 1.

[39] LIU C, RAJIV R J, YANG C, et al. MuR-DPA: Top-down Leveled Multi-Replica Merkle Hash Tree Based Secure Public Auditing for Dynamic Big Data Storage on Cloud [J]. IEEE Transactions on Computers, 2015, 64(9): 2609-2622.

[40] BARSOUM A F, HASAN M A. Provable Multicopy Dynamic Data Possession in Cloud Computing Systems [J]. IEEE Transactions on Information Forensics & Security, 2015, 10(3): 485-496.

[41] WANG B Y, LI H, LI M. Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics [C]// Proceedings of the IEEE International Conference Communication. 2013: 539-543.

[42] WANG B Y, CHOW S, LI M, et al. Storing Shared Data on the Cloud via Security-Mediator [C]// Proceedings of the 33rd IEEE International Conference on Distributed Computing Systems. 2013: 124-133.

[43] WANG B Y, LI B C, LI H. Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud [C]// Proceedings of the 10th International Conference on Applied Cryptography and Network Security. 2012: 507-525.

[44] WANG B Y, LI B C, LI H. Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud [J]. IEEE Transactions on Cloud Computing, 2014, 2(1): 43-56.

[45] WANG B Y, LI B C, LI H. Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud [J]. IEEE Transactions on Services Computing, 2015, 8(1): 92-106.

[46] YU Y, LI Y N. Public Integrity Auditing for Dynamic Data Sharing with Multiuser Modification [J]. IEEE Transactions on Information Forensics & Security, 2015, 10(8): 1717-1726.

[47] LUO Y C. Efficient Integrity Auditing for Shared Data in the Cloud with Secure User Revocation [C]// Proceedings of the IEEE Trustcom /BigDataSE/ISPA. 2015: 434-442.

[48] JOHNSON R, MOLNAR D, SONG D, et al. Homomorphic Signature Schemes [C]// Proceedings of the Cryptographers' Track at the RSA Conference. 2002: 244-262.

[49] BONEH D, GENTRY C, LYNN B, et al. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps [C]// Proceedings of the 22nd Theory and Applications of Cryptographic Techniques. 2003: 416-423.