计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191000111

云数据存储安全审计研究及进展

白利芳[1,2]　祝跃飞[1]　芦　斌[1]
1 信息工程大学网络空间安全学院　郑州 450000
2 中国软件评测中心网络安全测评工程技术中心　北京 100048
(bailifang@cstc.org.cn)

摘　要　云存储相比传统存储方式可避免存储平台重复建设及维护,其存储容量和性能的可扩展性、地理位置的无约束性及按需付费的服务模式有效优化了存储及社会资源配置。然而,云存储服务中数据所有权和管理权分离的特点,使得用户对保存在云端数据安全性及可控性的关注日益增长,国内外学者对此进行了大量的研究。文中论述了云数据在其生命周期各阶段的安全风险及其安全审计需求;构建了云存储数据安全审计机制的框架结构,并提出了审计机制的主要评价指标;综述了云数据存储安全审计现有机制,包括数据持有性证明机制、数据可恢复性证明机制、外包存储安全备份审计机制和存储位置审计机制;最后,从不同角度指出现有云数据存储安全审计研究存在的不足及下一步可研究的方向。

关键词:云存储;存储安全审计;审计框架;数据持有性证明;数据可恢复性证明;外包存储合规性
中图法分类号　TP309.2

Research and Development of Data Storage Security Audit in Cloud

BAI Li-fang[1,2],ZHU Yue-fei[1] and LU Bin[1]
1 School of Cyberspace Security,Information Engineering University,Zhengzhou 450000,China
2 Cybersecurity Testing Engineering Technology Center,China Software Testing Center,Beijing 100048,China

Abstract　Compared with traditional storage,cloud storage can avoid repeated construction and maintenance of storage platform. Its storage capacity and performance scalability,non-binding geographical location and fee-on-demand service mode effectively optimize storage and social resource allocation. However,due to the separation of data ownership and management rights in cloud storage services,users pay more and more attention to the security and controllability of cloud data. Researchers at home and abroad have conducted a lot of studies on this. The security risks and security audit requirements of cloud data in each stage of its life cycle are discussed. The framework structure of mechanisms of cloud data storage security audit is constructed and the main evaluation index of the audit mechanism is proposed. This paper reviews the existing mechanisms of cloud data storage security audit,including data provable data possession mechanism,provable data retrievability mechanism,outsourcing storage regularity audit mechanism and storage location audit mechanism. Finally,the shortcomings of the existing cloud data storage security audit research from different perspectives and the direction for further research are pointed out.

Keywords　Cloud storage,Storage security auditing,Auditing framework,Provable data possession,Provable data retrievability,Outsourcing storage regularity

1　引言

云存储以数据的安全存储和管理为核心,可实现任意地点、任意时间对任意持有数据的访问。对于存储需求不明确的用户,云存储容量和性能的可扩展性及按需付费的服务模式,可避免其存储平台重复建设及后期维护的成本和技术风险。云存储逐渐成为信息存储的趋势,但数据脱离了数据拥有者的物理管控。云存储服务中,数据在其生命周期的各个阶段均面临着各种各样的安全风险,有着不同的安全审计需求,云数据的安全审计面临巨大挑战,首先是被存储在云端的数据是否完整无误,其次是在数据完整性受到损坏后是否可以恢复。因此验证云存储服务提供商是否正确地持有数据,且当检测到数据损坏时是否可实现恢复尤为重要。此外,云服务商(Cloud Service Provider,CSP)可能并没有遵守云存储

到稿日期:2019-10-17　返修日期:2020-01-17　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家重点研发计划(2016YF0801601);国家自然科学基金青年科学基金(61601517)
This work was supported by the National Key R&D Program of China (2016YF0801601) and Young Scientists Fund Program of the National Natural Science Foundation of China(61601517).
通信作者:祝跃飞(mompidan@163.com)

服务等级协议(Cloud Storage SLA)，存储策略、保留的副本数量和存储位置等的不确定性同样困扰着用户。

为此，国内外研究者对云环境下的数据安全存储问题进行了大量研究，其中云数据存储安全审计是当前云计算领域的重要研究内容，其研究成果主要集中在数据完整性审计和数据外包存储合规性审计方面。本文首先讨论云数据各生命周期的安全风险及安全审计需求，构建云数据存储安全审计机制的框架结构；其次，提出审计机制的主要评价指标，然后在此基础上综述云数据存储安全审计现有机制并对几种经典机制进行详细分析；最后，讨论云存储中数据安全审计面临的挑战及下一步可研究的方向。

2 云数据安全审计需求及框架结构

2.1 云数据安全风险及审计需求

一般来讲，托管在云端的数据生命周期可分为 5 个阶段，即生成、存储、使用、存档和销毁(回收)。在各个阶段，云数据均面临不同种类、不同程度的安全风险[1]。

(1)生成阶段：即数据被创建但尚未迁移至云端的阶段，除了要为数据添加必要的属性(如安全级别信息等)外，还需制定安全审计策略、进行数据预处理等。

数据安全级别划分：不同性质的用户或同一性质的用户均有可能采取不同的安全级别划分策略。在多用户共享存储、网络和计算等资源的情况下，混乱的数据安全级别划分策略将无法保证有效防护数据安全。

数据审计策略制定：即明确审计对象、审计规则及响应方式，并组合关联调整策略优先级等。在传统存储架构下，审计人员制定有效的审计策略已很困难；在复杂的云环境下，数据的跟踪审计将更加艰难。

数据预处理：一方面，考虑到传输、存储、计算的开销，数据在迁移至云端前可能需要进行一些过滤或去重处理，甚至是格式的转换；另一方面，由于审计策略的需要，可能需要提前生成一些审计过程中用到的参数或数据，如验证元数据等。而这些工作的开展必须同时考虑安全性、成本与便捷性等因素间的平衡。

(2)存储阶段：数据安全迁移至云端，意味着数据物理控制权的转移。

外包存储合规性：数据上云，即将存储服务外包给 CSP。由于云环境用户共享存储资源的特点，CSP 若未采取有效的数据隔离策略，则可能导致用户数据泄露；特殊数据可能需被存储在特殊的地理位置或制定界限的地理范围内；CSP 是否按照用户意愿对数据进行备份，如备份数量、备份方式、备份期限等。

数据基本属性：自然灾害等不可抗因素；CSP 服务器遭受攻击、管理不当、违规操作、不正当使用或共享数据，尤其在大数据及人工智能盛行的当下，即便数据加密，攻击者依然可能通过大数据分析、机器学习等技术获取到有用信息，而用户并不知情。以上原因均会威胁到数据的基本属性，即保密性、完整性和可用性。

(3)使用阶段：用户访问存储在云端的数据，并对其做增删改查等操作。

访问控制：若 CSP 制定的访问控制策略不合理、不完善，则可能造成合法用户无法正常访问自己的数据或进行合规的操作，而未授权用户却能非法访问甚至窃取、篡改数据。

数据传输：用户通过网络使用云数据，若传输信道不安全，数据在传输过程中可能遭受中断、拦截、伪造、篡改、窃听和监视等安全威胁；传输时的其他不当操作可能造成数据不可用或完整性受损。

数据共享：不同软硬件在数据运算时，因数据的内容、格式和质量不同，可能会在进行兼容性处理时造成数据丢失；若实现数据共享的应用本身有安全漏洞，则基于该应用的共享可能存在数据泄露、丢失或被篡改等风险。

(4)存档阶段：随着使用频率的降低，某些云数据会被转移到单独的存储设备进行长期保存以备日后使用，即进入存档阶段。存档必须遵从规则，有些特殊的数据对归档的介质、方式和时间期限会有专门要求或规定，而 CSP 可能并未遵循，导致云数据面临合规性甚至法律问题。

(5)销毁(回收)阶段：当数据被回收或销毁时，数据拥有者向 CSP 发送命令并依赖 CSP 销毁或返还相应数据，同时意味着该数据在此次云存储服务的生命周期结束。

数据恶意恢复：在数据被销毁或返回后，可能留有剩余信息，如已销毁数据的残余数据或介质上的电磁残余，这些可能会被用于数据的恶意恢复。

CSP 不可信：用户无法保证 CSP 是否真的销毁或返还了全部数据，可能 CSP 之前保存有这些数据的多个备份，而这些备份数据并未被销毁或返还。

针对云环境下各阶段数据面临的安全风险，除了对其进行事前防范(如数据安全隔离、安全迁移、安全传输、安全访问等)和事中保护(如入侵防范、恶意代码防范等)，事后跟踪与追溯同样重要，尤其是站在数据拥有者的角度，针对已经迁移至云端的数据面临的完整性问题、存储合规性问题以及安全销毁等问题，其审计需求更为紧迫。

2.2 云数据存储安全审计框架结构

2.2.1 审计内容框架

基于前文涉及的主要安全风险及审计需求，本文提出的云数据存储安全审计的主要内容框架如图 1 所示。本文将根据此内容框架综述云数据存储安全审计机制及其相关工作的研究进展。

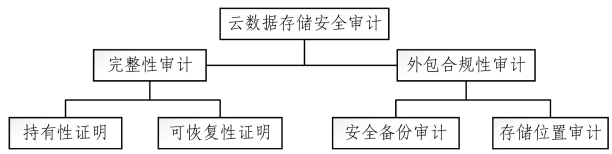

图 1 云数据存储安全审计内容框架

Fig. 1 Content framework of data storage security audit in cloud

2.2.2 审计流程框架

审计流程如图 2 所示。其中，User 为数据拥有者，同时也是云服务商 CSP 的用户，主要负责数据预处理及审计策略的制定，如果不支持第三方审计，那么他就是审计方；TPA(Third Party Auditor)为第三方审计，主要负责发起挑战及对

CSP 提供的应答消息进行验证；CSP 在审计过程中主要负责对 TPA 发起的挑战进行应答，以证明用户数据按约定安全存储。

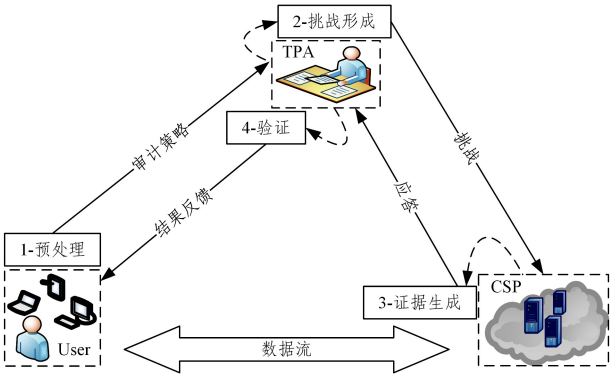

图 2 云数据存储安全审计流程框架

Fig. 2 Process framework of data storage security audit in cloud

2.2.3 审计机制评价指标

云数据存储安全审计机制的评价指标[2]主要围绕精确性、复杂性、适应性和安全性等方面。本文主要根据以下评价指标对各类审计机制进行分析：

1)精确性，即审计发现数据受损或其他问题的概率；

2)存储成本，即审计参与各方额外需要的存储开销，如秘钥、验证元数据的存储、状态变量等；

3)计算成本，包括用户数据预处理、被审计方证据生成、审计方验证等的计算开销；

4)传输成本，即审计参与各方之间的通信开销；

5)验证次数，若只支持有限次验证，意味着有限次后需要重新生成验证元等数据；

6)支持的动态操作，即数据被用户合法修改、删除、添加和插入后该机制的适应能力；

7)是否支持公开验证，即考虑数据隐私保护的前提下审计是否可由第三方完成；

8)可恢复性，即数据受损后是否一定程度可恢复；

9)安全性，即审计机制本身安全可靠，其证明方案一般采取标准模型和随机预言模型；

10)可扩展性，即是否支持分布式云存储环境或多云环境等。

3 云数据存储安全审计机制

3.1 数据持有性证明机制

数据持有性证明(Provable Data Possession，PDP)机制主要用于云数据完整性的检测，而可恢复性证明(Proofs of Retrievability，PoR)机制则用于完整性的确保，本节将基于现有 PDP 机制的实现原理对其进行分类，并进行介绍分析。

3.1.1 基于 HMAC 的 PDP 机制

最简捷的 PDP 机制[3]是审计方 A 将文件 F 发给 CSP 前计算其校验值，验证时向证明方 P 发送请求，P 计算被请求文件对应的校验值并返回给 A 进行对比。但该机制存在固有安全隐患：恶意用户破坏完整性之前预计算每个文件的校验值并保存(或直接保存每个文件首次验证后的正确响应即校验值)，修改验证程序使其在每次验证过程中返回该校验数据，绕过审计。为此，文献[3]最先提出远程数据的完整性检测方案：用户在本地为每个需验证的 F 随机生成 N 个挑战 $C_i(i=1,2,\cdots,N)$，将 C_i，F 作为 HMAC 函数的输入，得到 $HMAC(C_i,F)$作为预期响应 R_i，将$\{C_i,R_i\}$作为验证元数据在本地存储，为降低本地存储成本，考虑 $C_i(i=1,2,\cdots,N-1)$由 C_N 基于 Hash 生成，本地只需存储$\{\{R_i\},C_N,N\}$。验证时，A 向 P 发起挑战 C_i，P 计算 $HMAC(C_i,F')$生成 R_i'并发送给 A，A 通过对比 R_i'和 R_i 判断数据持有性。由于挑战的随机性，恶意用户无法预计算响应，需要绕过审计必须同时留存原文件 F 和 F'，而此行为很容易被发现。但该机制只适用于对静态数据的有限次验证，且存储成本和计算成本随挑战次数和文件数量线性增加。

惠普实验室 Shah 等[4]引入第三方审计，用户将秘钥 K 及 $E_K(F)$发送至 P，P 计算 $a=g^K \bmod p$(g 为 Z_P^* 的生成元，p 为一个大素数)和 $H(E_K(F))$(H 为某哈希函数)并将结果发送给 TPA 保存，F 具有完整性等价于 K 及 $E_K(F)$均完整。验证 $E_K(F)$完整性的思路与文献[3]相同，只是将 $HMAC(C_i,F)$更换为 $HMAC(C_i,E_K(F))$；而 K 的完整性验证利用模幂运算的同态性质实现。为解决有限次验证的问题，文献[4]提出 TPA 消耗完验证元数据后可向 P 请求加密文件以再次生成验证元数据，因 P 反馈给 TPA 的加密数据的完整性可能已被破坏，所以在再次生成前，TPA 采用 $H(E_K(F))$对其进行验证。该机制支持公开验证，代价是计算成本增加，即在给 A 返回 F 前须对秘钥 K 进行提取，而后解密 $E_K(F)$，且要求 TPA 维护长期状态信息。该机制验证次数可扩充，但验证元数据的再生成增加了计算和传输成本，代价较大。由于 $H(E_K(F))$一般远小于 $E_K(F)$，因此针对该机制，本文考虑可将 $H(E_K(F))$替换为 $E_K(F)$，即将 $HMAC(C_i,H(E_K(F)))$作为预期响应，既可减少计算成本又降低了传输成本。然而，每次验证 S 时须先计算加密文件的哈希值，并将其作为 HMAC 运算的输入以生成响应，其计算成本有待考察。随后，同一团队提出基于隐私保护的第三方审计机制[5]，其原理与文献[4]相同，只是对审计过程做了更严谨、更全面的论述。

综上，基于 HMAC 的 PDP 机制存在以下不足：1)验证次数受限；2)审计方存储成本大，需维持大量验证元数据、状态信息和秘钥信息等；3)公开验证问题突出；4)未考虑动态操作，只适用于静态数据完整性验证。

3.1.2 基于同态签名的 PDP 机制

(1)基于 RSA 同态的 PDP 机制

针对基于 HMAC 的 PDP 机制的不足，文献[3]考虑利用 RSA 签名机制的同态特性$(a^m)^r \equiv a^{mr} \equiv (a^r)^m \pmod N$构造 PDP 机制。其中，$N=pq$ 是 RSA 模数；p,q 为大素数。其原理为：审计方预计算 $a^m \bmod N=M$ 作为验证元数据，并随机生成 r，计算 $a^r \bmod N=C$ 作为挑战发送给 P(a,r 为 $2\sim N-2$ 的整数)，P 计算 $C^m \bmod N=R$ 作为应答，审计方计算 $M^r \bmod N=V$ 并比较 V 和 R。其中，N 为公开的 RSA 模数(1 024 bit)，a 为公开的 $2\sim N-2$ 的随机整数，m 为代表文件

F 的大整数。显然，该机制支持无限次验证且支持公开验证，但该机制将整个数据 F 用一个大整数 m 来表示以参与计算，计算成本较大，尽管在计算验证元数据 M 时文中考虑用欧拉函数来简化计算，即用 1 024 bit 的 $h(m)=m \bmod \varphi(N)$（关于 N 的欧拉函数）来替代 m，但每次应答 R 仍需大整数 m 参与计算得出，因此该机制不适用于大文件的完整性验证。文献[6]利用欧拉函数的性质对验证元数据的生成进行优化，即用 $h(m)$ 代替 M 作为元数据，$R=H(m)=r^m \bmod N$ 作为应答，由于函数 H 满足 $H(m+m')=H(m)H(m')$，因此关于整数加运算具有同态性，即 $H(m+m')\equiv H(m)H(m')(\bmod N)$。再结合欧拉定理，验证时只需比较应答 $R=H(m)=r^m \bmod N$ 和 $r^{h(m)} \bmod N$ 即可。但该方案的安全性和性能等方面的局限性还有待讨论（如 r 的选取（首先应是一个素数）），虽然其一定程度地减轻了验证元数据的计算和存储成本，但依然不适用于大规模数据。随后，文献[7]提出采用分块的思想进一步优化元数据集，支持无限次验证，但采用确定性的验证策略依然不能很好地满足大文件的验证。

文献[8]提出采用抽样的概率性验证策略来实现验证，首次形式化定义 PDP 机制，包含 4 个多项式时间算法{*KeyGen*，*TagBlock*，*GenProof*，*CheckProof*}；提出同态标签（Homomorphic Verifiable Tags，HVTs）的概念，并结合抽样策略提出两个基于 RSA 同态签名的 PDP 机制，即 S-PDP 和 E-PDP，实现步骤如下。

S-PDP：定义 $N=pq$ 是 RSA 模数（$p=2p'+1$ 和 $q=2q'+1$ 为安全的素数），g 是模 N 的二次剩余群 QR_N（Z^*_N 的 $p'q'$ 阶循环子群）的生成元。k，l，λ 是安全参数，n 为文件 F 加密且分块后的数据块数量，c 是每次挑战从 n 个数据块中抽样出来的数据块数量，h 是映射到 QR_N 给定且安全的哈希编码函数，H 是哈希函数。设 $f:\{0,1\}^k\times\{0,1\}^{\log_2(n)}\rightarrow\{0,1\}^l$ 为伪随机函数 PRF（Pseudo Random Function），$\pi:\{0,1\}^k\times\{0,1\}^{\log_2(n)}\rightarrow\{0,1\}^{\log_2(n)}$ 为伪随机置换函数。

Setup 阶段：

1）生成秘钥对：客户端 C 运行 $(p_k,s_k)\leftarrow KeyGen(1^k)$ 生成 $p_k(N,g)$，$sk=(e,d,v)$ 并存储，其中 $ed\equiv 1(\bmod p'q')$，e 是一个大的秘密素数，且 $e>\lambda$，$d>\lambda$，$v\xleftarrow{R}\{0,1\}^k$。

2）生成数据块标签：客户端 C 对所有 $1\leqslant i\leqslant n$ 运行 $(T_i)\leftarrow TagBlock(p_k,s_k,m_i,i)$，生成 $W_i=v\parallel i$，生成标签 $T_i=(h(W_i)\cdot g^{m_i})^d \bmod N$，并将 p_k，$F=(m_1,m_2,\cdots,m_n)$ 和 $\Sigma=(T_1,T_2,\cdots,T_n)$ 发送给 P 存储，同时将本地的 F 和 Σ 删除。

Challenge 阶段：

1）发起挑战：客户端 C 生成挑战 $ch=(c,k_1,k_2,g_s)$，其中 $k_1\xleftarrow{R}\{0,1\}^k$，$k_2\xleftarrow{R}\{0,1\}^k$，$g_s=g^s \bmod N$，$s\xleftarrow{R}Z^*_N$，输入参数的随机性使得攻击者无法再利用前期挑战阶段的任何信息来绕过审计，保证证据 ρ 的不可伪造性，最后将 ch 发送给 P。

2）证据生成：证明方 P 根据 ch 计算验证方 C 想要挑战的 c 个数据块的下标 $i_j=\pi_{k_1}(j)$，其中 $1\leqslant j\leqslant c$（此处保证了被请求验证数据块选取的随机性），同时计算系数 $a_j=f_{k_2}(j)$，运行 $V\leftarrow GenProof(p_k,F,ch,\Sigma)$ 生成证据 $V=(T,\rho)$，其中 $T=T_{i_1}^{a_1}\cdot\cdots\cdot T_{i_c}^{a_c}=(h(W_{i_1}^{a_1})\cdot\cdots\cdot h(W_{i_c})^{a_c}\cdot g^{a_1m_{i_1}+\cdots+a_cm_{i_c}})^d \bmod N$，$\rho=H(g_s^{a_1m_{i_1}+\cdots+a_cm_{i_c}} \bmod N)$。

3）验证：客户端运行 $CheckProof(p_k,(e,v),ch,V)$，即令 $\tau=T^e$，对 $1\leqslant j\leqslant c$ 循环计算 $i_j=\pi_{k_1}(j)$，$W_{i_j}=v\parallel i_j$，$a_j=f_{k_2}(j)$，$\tau=(\tau/h(W_{i_j})^{a_j}) \bmod N$。若 $H(\tau^s \bmod N)=\rho$，则验证通过，否则不通过。

E-PDP：该机制与 S-PDP 的区别在于审计过程中的系数 $a_j=1$。S-PDP 机制在指数知识假定（KEA-r）下是安全的，E-PDP 是一种弱安全的 PDP 机制[8-9]。这两种机制考虑在客户端分别计算 F 被分块后每个数据块的验证元数据，以降低验证数据的计算成本，采取随机抽样的概率策略选取被验证数据块，且证据无法伪造。由于同态性，P 可将多个数据块标签聚集成一个标签值响应挑战，从而降低传输成本。机制只需要用户维护常量的验证元数据，服务器的计算成本也近似为常量，可无限次验证，但不支持公开验证及动态操作。为了使得该机制支持公开验证，文中对该机制进行了调整：在 Setup 阶段将 e 变为公开，即 $p_k=(N,g,e)$，$s_k=(d,v)$，同时令 $W_i=f_v(i)$，并在该阶段结束后公开 v；在 Challenge 阶段令 $\rho=a_1m_{i_1}+\cdots+a_cm_{i_c}$，最后验证 $g^M\equiv\tau(\bmod N)$，除此之外并未就公开验证进行进一步的深入讨论。

文献[10]将 E-PDP 机制扩展到数据的多副本存储，提出一个针对数据多副本完整性保护的 MR-PDP 机制。该机制将 t 个数据副本采用随机掩码技术加以区分后，与块签名集合一并存储在 t 个远程服务器上，即在第 u 个服务器上存储的副本可表示为 $F_u(m_{u_1},m_{u_2},\cdots,m_{u_n})$，其中 $1\leqslant u\leqslant t$，$m_{u_i}=m_i+r_{u_i}$（$1\leqslant i\leqslant n$），r_{u_i} 由伪随机函数基于 $v\parallel i$ 生成，挑战阶段验证者在随机 t 个副本中选择其中一个副本 F_u 进行挑战，挑战过程与 E-PDP 相同，区别只在于证据元素 ρ 的生成，即 $\rho=g_s^{\sum b_i+r_{u_i}}$（$s_1\leqslant i\leqslant s_c$）。相应地，验证阶段改为判断 $(T^e\cdot g^{r_{ch}}/h_{i_1}\cdot\cdots\cdot h_{i_c})^s$ 是否等于 ρ，其中 $r_{ch}=\sum_{j=1}^{c}r_{i_j}$。显然，该机制只支持私有验证，其贡献在于对数据进行完整性验证时，不必对每一个副本文件都当作独立的数据文件分别进行验证，而是对所有副本数据进行随机验证，而且每次验证的开销与对单个文件验证的开销大致相同。

文献[11]根据满足同态属性的鉴定协议提出了构造公钥同态线性认证器（Homomorphic Linear Authenticator，HLA）的通用机制，并论述了如何将公钥 HLA 转化为支持公开验证方案，且方案通信复杂度与文件大小无关并支持无限次验证。文献[12]针对不受信任的 TPA 正式提出将“针对第三方验证者的隐私保护”定义为机制的安全要求之一，并基于文献[7]实现有关数据动态更新及数据隐私性的部分，基于文献[8]中的 RSAHVTs 技术实现公开验证的部分。

（2）基于 BLS 同态的 PDP 机制

文献[13-14]借鉴文献[8]中 HVTs 的思想，提出基于 BLS 签名的 PDP 机制。BLS 签名较 RSA 签名具有更短的签名位数[15]，且同样具有同态特性（签名可聚集），因此基于 BLS 签名的 PDP 机制可实现更低的存储成本和通信成本，实现步骤如下。

定义 $H:\{0,1\}^* \rightarrow G$ 是 BLS 哈希函数，是一种基于椭圆曲线的新型散列函数；$e:G\times G\rightarrow G_T$ 是具有同态性质的非退化双线性映射(对于双线性椭圆曲线，存在特殊函数 e 使其满足同态性质，即 $e(a^{\wedge}p,b^{\wedge}q)=e(a,b^{\wedge}pq)=e(a,b)^{\wedge}pq$，$g$ 是素数阶循环群 G 的生成元)。

Setup 阶段：

1)生成秘钥对：客户端 C 运行 $(p_k,s_k)\leftarrow Pub.Kg$ 生成 $p_k=(v,spk)$，$s_k=(\alpha,ssk)$ 并存储，其中 $(spk,ssk)\xleftarrow{R}SKg$ 为签名秘钥对，$\alpha\xleftarrow{R}Z_p$，$V\leftarrow g^{\alpha}$。

2)生成验证元数据：客户端 C 运行 $(\Phi,t)\leftarrow Pub.St(sk,F)$，将文件 F 分为 n 块，每个数据块的长度为 s，即 $\{m_{ij}\}(1\leqslant i\leqslant n,1\leqslant j\leqslant s)$，在足够大的域(如 Z_p)中随机取一个文件名 $name$，并随机选取 s 个辅助元素 $\{u_j\}\in G$，令 $t_0=name\parallel n\parallel u_1\parallel u_2\parallel\cdots\parallel u_s$，文件标签 $t=t_0\parallel SSIG_{ssk}(t_0)$。对所有 $1\leqslant i\leqslant n$ 计算验证元数据 $\Phi=\{\sigma_i\}$，其中 $\sigma_i\leftarrow(H(name\parallel i)\cdot\prod_{j=1}^{s}u_j^{m_{ij}})^{\alpha}$。将 $p_k,t,F=\{m_{ij}\}$ 和 Φ 发送给 P 存储，同时删除本地的 F 和 Φ。

Challenge 阶段：

1)发起挑战：审计方 A(客户端 C 或第三方审计)从集合 $[1,n]$ 中随机选择一个包含 c 个元素的子集 I，对每个 $i\in I$，随机生成 $v_i\xleftarrow{R}B$，其中 B 为 Z_p 的子集。则挑战 $ch=\{(i,v_i)\}$，A 将 ch 发送给 P。

2)证据生成：对于 c 个不同的 i，证明方 P 计算证据 $\{\mu_j\}$ 和 σ，其中 $\mu_j\leftarrow\sum_{(i,v_i)\in Q}v_im_{ij}\in Z_p(1\leqslant j\leqslant s)$，再将 $t,\{\mu_j\}$ 和 σ 发送给 A。

3)验证：A 用 spk 验证 t 上的签名，若签名不可靠，则输出 0 并终止，否则恢复文件名 $name$，n 以及 $\{u_j\}$；对比 $e(\sigma,g)$ 与 $e(\prod_{(i,v_i)\in Q}H(name\parallel i)^{v_i}\cdot\prod_{j=1}^{s}u_j^{\mu_j},v)$ 是否成立，若成立，则验证通过，否则不通过。

综上，该机制支持公开验证且可无限次验证，采用更短的 BLS 签名降低了计算和存储成本。P 同样采用了一定程度的签名聚集策略响应挑战，但对 c 个数据块的验证请求仍需对应 s 个证据(即 $\{\mu_j\}$)，其传输成本和验证阶段的计算成本不可忽视，且不支持动态操作。文献[16-18]均是基于上述 BLS 同态签名技术实现的。考虑到 BLS 同态机制可能泄露用户数据隐私，即多次挑战后 TPA 可能根据证据生成过程中的线性组合 μ_j 构成 m_{ij} 的值，从而造成数据泄露，文献[17-18]在文献[16](支持动态操作)的基础上提出通过随机掩码技术来实现用户数据隐私保护(引入两个参数 r 和 γ 来掩藏 u_j)，同时支持批量审计，即 TPA 可同时处理来自多个不同用户的审计请求。

3.1.3 基于身份的 PDP 机制

以上 PDP 机制大多基于 PKI，性能受限于传统数字签名中负责的证书管理、频繁的验证等问题。而基于身份密码体制[19]的 PDP 机制可有效解决此问题，且在用户注销或新注册时有明显的优势。在基于身份的密码体制中，用户的公钥为标识其身份的唯一信息(如用户的 IP 地址)，私钥由可信的私钥生成中心(Private Key Generator，PKG)利用用户身份信息和 PKG 的主密钥对计算得到。文献[20]在文献[21]提出的基于身份的聚合签名的基础上首次提出云环境下基于身份且考虑隐私保护的数据公开验证方案，通过最小化验证消息和 TPA 获取或存储的信息来简化密钥管理，降低了通信和计算的开销。与现有基于公钥基础设施的机制不同，该方案只有私钥生成中心拥有传统公钥，而用户只保留其身份而不绑定证书。文献[22]首次提出被证明是安全(基于标准模型下 CDH 问题假定)的基于身份的远程数据持有性验证(Remote Data Possession Checking，RDPC)协议，除了消除证书管理和验证的结构优势外，该协议在计算和通信方面也优于基于 PKI 现有的 RDPC 协议。文献[23]提出了多云存储环境下基于身份的 PDP 机制。文献[24]指出以上机制不允许敌手访问数据块标签不符合云存储的一般情况，且文献[23]仅利用数据块的哈希值即可生成挑战的应答，这显然不安全。为此，文献[24]提出了一种新的基于身份的 PDP 机制，利用同态密钥加密原语，降低了在基于 PKI 的 PDP 机制中建立和管理公钥认证的成本，并将 TPA 零知识隐私的安全模型形式化，考虑了数据隐私问题且对该机制进行了安全性证明。文献[25]在标准模型中提出了一个基于身份的公共验证方案。该方案采用双线性映射运算，比 RSA 算法中的模幂运算计算成本更高。为了提高验证效率，文献[26]利用 RSA 算法提出基于身份的高效远程数据完整性验证方案。文献[27-28]分别在文献[26]的基础上考虑了密钥泄露和数据隐私问题。文献[29-30]提出的基于身份的外包数据审计方案允许用户通过可识别身份对代理进行鉴别，并授权专用代理代表用户将数据上传到云存储服务器，无需复杂的证书管理，可定期进行完整性审计，还允许审计外包文件的数据来源、类型和一致性信息。文献[31-32]基于身份的 PDP 机制，考虑采用区块链中的新鲜值构造不可预测且易被验证的挑战消息，以预防恶意 TPA 伪造审计结果。文献[33]分析指出文献[26]提出的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用户的私钥；而文献[28]中的服务器可以伪造证据，使其在不利用存储数据的情况下依然能通过 TPA 的验证。随后，文献[33]针对文献[28]的方案进行了改进，保持了相同的通信和计算开销。文献[34]利用理想格上的小整数解问题，设计了一种基于身份的 PDP 方案，在随机预言模型下证明了该方案可以抵抗云服务器的适应性选择身份攻击，并验证了该方案良好的实现性能。文献[35]提出基于模糊身份的数据完整性审计机制，通过属性集合表示用户身份信息，并形式化云存储系统模型和安全模型。该机制具有一定的容错性，即可以利用与用户真实身份 A 足够“相似”的另一身份 A′来验证用户签名，因此该机制可应用于类似于生物特征认证等用户身份信息不能被完全正确提取的场景。

3.1.4 基于代数签名的 PDP 机制

基于代数签名的 PDP 机制是概率性策略安全方案，具有低网络带宽、合理的计算载荷及抗恶意修改的优势。代数签名即具有某种代数性质的哈希函数，即将较大的数据块压缩

成很小的比特串参与运算、传输。在 $GF(2^l)$ 域中，代数签名的计算大多为异或操作，由于基于代数签名的 PDP 机制在验证阶段不需要与原始数据进行比较，只需存储站点返回的响应即可验证，因此有着较低的计算和传输开销与较高的效率。文献[36]定义的代数签名具有“数据块签名的校验码与数据校验码的签名相同”的性质，即 $parity(sig(F_1),\cdots,sig(F_m))=sig(parity(F_1,\cdots,F_m))$，P 计算被挑战数据块的代数签名并返回，验证时对比签名数据的校验码是否与相应数据校验码的签名相等即可。为了防止攻击者预计算所有可能的挑战应答，文中将 $\langle x,n,s\rangle$（x 为起始偏移量，n 为样本数量，s 为步长）作为挑战请求，从而提高机制安全性。此外，该机制由于使用了线性纠删码来编码数据，可一定程度检测出数据损坏位置并对其进行恢复。文献[37]定义的代数签名具有数据块之和的签名等于数据块签名之和的性质。审计方 A 在数据外包之前计算 t（挑战次数）个验证标签并加密，然后发送至 P，且只需存储两个秘钥和几个随机数；P 计算被挑战数据块之和 S，将 S 与对应的验证标签 T 一并返回给 A；A 对 S 进行代数签名，对 T 进行解密，若二者相等即验证通过。该机制的性能优越，但受限于磁盘 I/O 而非代数签名或密码运算，支持有限次验证，为此该文随后提出了一种改进的方案，即在 P 上存储每个数据块的代数签名以用于后续验证标签的再生成。文献[38]与文献[37]的机制完全相同，只是增加了部分实验数据及分析。

3.1.5 支持动态操作的 PDP 机制

考虑到云环境下数据存储安全审计在支持动态操作方面的迫切需求，本文将支持动态操作的 PDP 机制单独讨论分析。文献[39]最先考虑支持动态操作，对文献[8]提出的 PDP 机制进行简单地改进后支持数据更新、删除、追加等操作，但不支持数据插入操作。该方案预计算所有挑战的验证元数据，因此只支持有限次验证，且每次更新都将重新创建剩余挑战及其元数据，故应用受限。文献[40]利用纠错码的线性特征来支持部分动态操作（不支持插入操作）及数据恢复，只支持私有的有限次验证。文献[18]引入 TPA，基于 BLS 同态和 MHT(Merkle Hash Tree)数据结构构造了一个支持公开验证的 DPDP 机制，TPA 通过服务器响应的认证路径及辅助认证信息来验证数据是否完整。文献[41]分别基于树和映射提出两种动态的 PDP 机制——TB-DMCPDP（Tree-based Multi-copy PDP）和 MB-DMCPDP（Map-based Dynamic Multi-copy PDP），且支持公开审计及备份数量的审计。其中，TB-DMCPDP 主要采用 MHT 数据结构，该文指出其也可采用其他认证数据结构（如跳表），该方案中 CSP 的存储成本及计算成本和验证方的计算成本均很高；而 MB-DMCPDP 主要基于映射-版本表（Map-Version Table），支持数据的动态更新，即当某个数据块被更新则其版本号将以 1 为步长增加，如此可同时保证用户在多个副本之间只维护一个版本表，整体性能优于 TB-DMCPDP。文献[42]针对无线传感网络中的分布式存储数据，利用代数签名及纠删码技术实现了用于验证多用户分布式数据的动态 PDP 机制，该机制能保护用户数据和身份的双重隐私。文献[12]在文献[7]的基础上进行了拓展以支持数据动态更新和公开验证，同时考虑了数据隐私问题。但文献[43]指出，所有支持数据动态更新的现有公开验证 PDP 机制都要求数据所有者发布一些与存储数据相关的元数据，包括文献[12]提出的机制。因此，审计方虽然不能在挑战/应答过程中得到被存储文件的任何信息，但可判断客户是否存有特定数据（元数据），并根据发布的元数据与文件的各个部分关联，即“针对第三方验证者的隐私保护”并不完全有效。为此，文献[43]对文献[12]的机制进行了改进，引入“零知识隐私”的概念，以保证第三方审计者无法从所有可用信息中得到有关客户的任何信息。文献[44]采用一种被称为平衡更新树的新数据结构，实现了支持动态操作的 PDP 机制。该机制可同时支持修订控制及多用户访问共享数据，且更新树始终保持平衡，其大小与整个外包存储数据的大小无关，而是随着数据块的动态更新次数的增多而增长，必要时可通过一个命令将其大小控制在一个阈值范围内。文献[45]提出支持全动态操作的 PDP 形式化框架，分别基于认证跳表和 RSA 树实现全动态 PDP 机制（该机制在 2009 年 ACM 的 CCS 会议上被首次提出[46]）：首先，基于文献[47]提出了一种改进的数据结构，即基于等级的认证跳表（Rank-based Authenticated Skip List），构造支持全动态操作的 PDP 机制，该机制的计算复杂度和通信复杂度均为 $O(\log n)$，由于采用概率策略，其检错率与文献[8,39]相当；其次，还提到该机制中的数据结构可用动态 Merkle 树实现相同的渐进性能，但 Merkle 树的劣势在于一系列更新操作后，它可能变得非常不平衡，但早期几乎没有在维护和更新验证信息的同时重新平衡 Merkle 树的算法，而针对认证跳表的数据结构却有着广泛的研究且实现简单；最后，本文基于 RSA 树提出了另一种全动态实现机制，该机制有着更高的检错率和更低的服务器计算开销，同时讨论了几种可替代的 HVTs 方法和动态数据结构，同文献[18]，其通过增加访存复杂度来获得对动态更新的支持，存在认证路径过长及辅助认证信息过多的缺陷。文献[48]基于 Paillier 同态密码体制提出了一种支持 TPA 同步审计的动态 PDP 机制，性能好且不需要引入新的数据结构，其局限性在于假设云服务提供者是可信赖的。

3.1.6 其他 PDP 机制

文献[49]提出一种基于椭圆曲线加密的交互式 PDP 机制，并提出利用概率查询和定期验证机制来降低每次验证的审计成本和实现及时异常检测；同时，提出最优化参数的方法以最小化云审计服务的计算成本。文献[50]提出在多云存储环境下支持数据迁移的 PDP 机制。Wang 等利用双线性函数的性质，设计实现了支持批处理审计任务的 PDP 机制，并优化了审计性能[51-52]。文献[52]基于环签名构造同态验证器，针对不可信云中的共享数据，提出了一种保护隐私的云共享数据公开验证机制，但不支持数据的动态操作，且未考虑在共享组内成员变动（如注销）时如何实现数据完整性验证和隐私保护。文献[53]针对云共享数据（不同的共享数据块由共享组内不同成员签名，附加签名 ID），引入了云代理服务器替代共享组内成员，来对注销用户注销前签名的数据块重新签名（同态签名），以降低现有用户在有成员注销时的通信和计算

压力;其还指出若引入 shamir 秘钥共享机制,该方案还可以拓展到多代理模型中,以适应重命名数据块数量巨大或组内成员变动较大的情形。该方案支持公开验证,且可拓展到支持动态操作(引入索引哈希表)及批量审计,其局限性在于无法应对合谋攻击,即云代理可能将新签名秘钥共享给已注销用户。文献[54]提出基于变色龙哈希保护用户身份隐私的公开审计方案,在可验证数据完整性的基础上,确保用户身份隐私不泄露给 CSP 和 TPA,并证明了该方案在随机 Oracle 模型中是安全的。

3.2 POR 机制

现有 POR 机制一般是在现有 PDP 机制的基础上或特定存储策略下,采用数据冗余机制(如多副本、冗余编码等)来提高存储可靠性。相同容错级别下,冗余编码能够以更小的数据冗余度来获得更高的数据可靠性,但编码方式较复杂,计算成本相对较大。

3.2.1 基于多副本的 POR 机制

该类机制通过牺牲存储成本,即采用冗余备份的思想来提高被保护数据的完整性和可用性。除了一般的备份方式外,文献[10]提出的经过随机掩码处理的备份方式的安全性更高,其中一个副本,对其进行的完整性被破坏后,可抽取其他任一完好副本,对其进行数据解码后采用同样的掩码技术生成新的副本,整个过程只需轻量级的运算,验证者可根据需求动态调整副本数量,同时可确认云服务商确实存储了约定数量的副本。该备份方式完整性保护完成度高,但缺点在于存储成本过高、不支持数据动态更新和公开验证。

3.2.2 基于数据纠错编码的 POR 机制

(1)基于"哨兵"的 POR 机制

文献[55]首次提出了 POR 的概念,并提出基于"哨兵"的 POR 机制,其基本思想是将随机赋值的"哨兵"随机嵌入编码并加密后的文件 F 中,当验证者 V 随机指定 p 个"哨兵"的位置并向证明者 P 发起挑战时,若 P"应答"被指定位置"哨兵"的正确值,则此次验证通过,否则认为数据被损坏并采用 RS 纠错码技术恢复数据。

Setup 阶段:

编码、加密:设 l-bit 为数据的存储单元,选取本原多项式 $P(x)$ 构造 $GF(2^l)$ 域。将 F(由 b 个块组成)以 k 为单位划组,纠错码 $RS(n,k,d)$ 对每个组编码得到 F'(由 $b'=bn/k$ 个块组成),每个新组包括 k 个数据块和 d 个校验码块 $\{Q_t\}_{t=0}^{d-1}$(由下列方程组计算而得,其中 α 为 $P(x)$ 的一个解,$\{m_r\}_{r=0}^{k-1}$ 为某数据组 k 个数据块在 $GF(2^l)$ 域的值),然后分别对称加密每个块,得到 F''。

$$\begin{cases} H_Q \times V_Q = 0 \\ H_Q = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha^1 & \cdots & \alpha^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha^{d-1} & \cdots & \alpha^{(n-1)(d-1)} \end{bmatrix} \\ V_Q = [m_0 \quad m_1 \quad \cdots \quad m_{k-1} \quad Q_1 \quad Q_2 \quad \cdots \quad Q_d]^{\mathrm{T}} \end{cases}$$

1)哨兵生成:令 $f:\{0,1\}^j \times \{0,1\}^* \to \{0,1\}^l$,$\{0,1\}^l$ 为单向函数,$a_w = f(\kappa, w)$,$w=1,2,\cdots,s$ 为哨兵,s 为哨兵数。将 a_w 追加到文件 F'' 得到 F''',即有 $F'''[b'+w]=a_w$。将哨兵值存储于本地。

2)置换:令 $g:\{0,1\}^j \times \{1,2,\cdots,b'+s\} \to \{1,2,\cdots,b'+s\}$ 为伪 PRF,$\widetilde{F}[i]=F'''[g(\kappa,i)]$,$i=1,2,\cdots,b'+s$。即将 $b'+s$ 个数据块位置随机置换,将 $\widetilde{F}$ 存储于远程服务器。

Challenge 阶段:

1)挑战:V 发起第 σ(初始值为 1)次挑战,随机指定未消耗的 q 个哨兵 $\{a_{\sigma m}\}_{m=1}^{q}$ 的位置 $\{p_{\sigma m}\}_{m=1}^{q}$,$\sigma$ 加 1。因为此次挑战暴露了 q 哨兵在 $\widetilde{F}$ 中的位置,其不能再被重复使用,所以 $\sigma \leqslant s/q$。

2)验证:P 返回被请求哨兵值 $\{\widetilde{F}[p_{\sigma m}]\}_{m=1}^{q}$,若 $\{\alpha_{\sigma m}=\widetilde{F}[p_{\sigma m}]\}_{m=1}^{q}$ 为真,则验证通过;否则进入 Recover 阶段。

Recover 阶段:

1)计算校正子:根据被损坏哨兵和 g^{-1} 运算确定拟恢复数据块所在数据组,解密该数据组后得到 $V_Q{}^*$。若校正子 S^* 满足 $H_Q \times V_Q{}^* = S^* = H_Q \times V_Q = 0$,则数据块无损(意味着被损坏的恰巧是哨兵),否则执行步骤 3)。

2)计算错误位置及错误值:由 RS 纠错算法可知,求解 u 个被损坏块的位置和错误值,即求解 $GF(2^l)$ 域中的一个 $2u$ 元一次方程组。因 $\{\alpha^v\}_{v=0}^{n-1}$ 的元素可与 n 个数据块一一对应,可将其用来标识 n 个数据块的位置。求解方程组,可得到错误位置标识为 $\{\alpha^v\}_{v=0}^{n-1}$ 中的某个元素,即求得数据组对应出错位置 m_{r_1},同时求得对应错误值为 α^{v_y}。

3)矫正错误值:令 $m_{r_0}=m_{r_0}^*+\alpha^{v_y}$,即可恢复第 r_0 个数据块。

综上,该机制只支持有限次验证,数据恢复能力有限且秘钥不可共享,对 F 的每次合法操作都需重新编码,生成哨兵。本地存储成本及数据编码、哨兵生成、数据恢复等计算成本均不容忽视。此外,在 Setup 阶段的步骤 2)中,若将置换 $\widetilde{F}[i]=F'''[g(\kappa,i)]$ 改为 $\widetilde{F}[g(\kappa,i)]=F'''(i)$,则效果更好,在 Challenge 阶段随机指定哨兵位置时,后者计算更为简明:选定第 w_0 个哨兵,则其在 $\widetilde{F}$ 中的位置即为 $g(\kappa,w_0)$,在验证过程中直接对比 $\widetilde{F}[g(\kappa,w_0)]$ 和 α_{w_0} 即可。

(2)基于签名的 POR 机制

文献[13]提出两个基于同态性质的 POR 机制,其中一个为在标准模型下安全的基于 PRF 的私有验证方案,另一个为在随机模型下安全的基于 BLS 签名的公开验证机制。两种机制均采用了将多个被挑战数据块的 HVTs 聚集成一个标签值的思想来提高验证效率,完整性的验证思路同 3.1.2 节中的内容,为了实现数据恢复功能,其初始设置阶段均对数据进行 RS 纠错编码预处理。两种机制都是无状态验证且支持无限次验证,但不支持动态更新。文献[9]在文献[8]提出的 PDP 机制的基础上,采用前向 ECC 对数据块进行编码以实现数据恢复功能,同基于"哨兵"的机制一样只支持少量的数据恢复,挑战次数受限,且在证据生成过程中要求服务器具有标签聚合编码能力。文献[36]提出利用具有同态性质的代数签名来实现数据完整性验证,使用线性纠删码来编码数据以实

现一定程度的数据恢复。

3.2.3 基于再生编码的 POR 机制

(1)基于线性网络编码的 POR 机制

文献[56]提出一种适用于基于网络编码的分布式存储系统的 POR 机制，网络编码即对原始数据块进行线性组合运算，生成与原数据块相同大小的编码块。这些编码块被分布式存储在 n 个服务器上，使得任意 k 个服务器共同存储至少 m 个编码块，其中编码系数(有限域中)是随机生成的。该机制的挑战及验证思路基于文献[13]中基于 PRF 的私有验证机制(基于 BLS 签名的公开验证机制也同样适用)，若检测出数据损坏，只要 n 个服务器中至少 k 个数据未被损坏，就可以恢复原始数据。文献[56]指出基于网络编码的分布式存储系统较基于 ECC 的系统更易受到额外攻击(如重放攻击和污染攻击)，二者分别通过加密随机生成的系数集和附加恢复验证标识来改善其安全性。相比基于纠错编码的 POR 机制，该机制的数据恢复的通信成本显著降低，且实验表明验证双方的计算成本都很低，但要求存储服务器具有数据随机线性组合的编码能力。

文献[57]为实现永久的单云故障修复，提出了一种基于代理的多云存储系统 NCCloud，即在分布式多云存储环境下，代理充当验证方应用程序和多个云之间的接口，若其中某个云的存储服务发生永久性故障，代理将激活修复程序，从其他存活的云中读取必要的数据块，重新构造新的编码数据块并存储在新的云中，整个修复过程不涉及云与云之间的直接交互。该系统基于再生编码的网络编码存储方案，设计了一种新的基于线性网络编码的 FMSR(Functional Minimum Storage Regenerating)编码，该编码并不只是简单地利用随机编码系数生成编码块，而是要确保生成的线性组合始终满足 MDS(Maximum Distance Separable code)属性。该编码只保留编码块，可实现与传统纠错编码相同级别的存储成本和更小的恢复带宽，且不要求存储节点具备一定编码能力。文献[57]在真实的多云环境中对该编码进行了验证评估，其中在 4 朵云存储环境中，相比基于 RS 纠错编码的 RAID-6 方式，其可节省近 50%的恢复流量，但数据恢复需要访问所需数据块的整个文件，适用于访问频率较小的静态存储数据。

(2)基于混合编码的 POR 机制

文献[58]提出了一个分布式加密系统 HAIL (High-Availability and Integrity Layer)，将云数据的完整性审计研究拓展到多服务器的分布式存储中，采用 MAC 和 ECC 对数据块进行混合编码，以提高云数据的安全性和完整性。首先，该文献首次在 POR 方案中考虑动态对抗，提出了一个形式化的动态对抗模型，并进行了严格的分析和参数选择，指出该系统对于一个活跃且移动的敌手(即可能逐步破坏整个服务器集群)是健壮的，且每台服务器的计算和带宽成本与前期的 POR 机制相当；其次，文中结合 PRFs，ECCs 及 UHFs(Universal Hash Function)构造了一种新型的纠错编码 IP-ECC (Integrity-Protected Error-Correcting Code)，由于 MAC 和奇偶校验码可以基于同一个 UHFs，因此可构建既是 MAC 又是奇偶校验码的块嵌入到 IP-ECC 中。因此，IP-ECC 同时也是一个抗破坏的 MAC。此时，服务器收到挑战，HAIL 可将 IP-ECC 中的多个 MAC 聚合成一个“复合 MAC“”作为响应，因此 HAIL 无需像绝大部分机制那样在服务器上额外存储验证元数据，且“复合 MAC”根据需要来计算，并不会增加存储成本或计算成本。此外，系统在针对单个服务器构建 POR 的基础上，同时利用服务器内部冗余和跨服务器冗余，将 POR 作为系统的一个构建块来验证单个服务器数据的可恢复性，若失败则通过跨服务器查询操作从跨服务器冗余(即 IP-ECC 数据块)中恢复损坏的数据块。该机制验证方的存储成本为常量级，只支持静态存储数据(如备份文件和归档文件)，不支持公开验证。

文献[59]在文献[57]的基础上结合对抗纠错编码(Adversarial Error-Correcting Code，AECC)[60]提出一种基于再生编码的 POR 机制 FMSR-DIP，可实现验证方在多服务器分布式存储环境下远程验证外包数据随机子集的完整性。其中，FMSR 与 AECC 均具有容错功能，只是适用对象不同，FMSR 适用于跨服务器分布式存储的文件，而 AECC 适用于存储在服务器的单个数据块。FMSR-DIP 与 HAIL 不同的是，在恢复过程中无需读取和重建整个文件，而是从其他幸存的服务器中读取一组比原始文件小的数据块，只重建丢失(或损坏)的数据块，降低了恢复成本。文献[59]在真实的云存储测试环境中对所提机制进行了验证评估，结果表明所提机制不需要存储服务器具备标签聚合或数据随机线性组合等编码能力，只需支持标准的读/写功能，但其只适用于读取频率较低的长期静态存储的外包数据完整性保护，应用场景受限。

3.2.4 其他 POR 机制

文献[61]提出了一个纯信息论的概念 POR 码(包含 3 个函数分别用来实现初始化、挑战信息读取和响应生成)，并给出将其转换为 POR 机制的方案，讨论了其安全性与某些参数之间的平衡问题。文献[60]提出了一套 POR 机制的理论框架，以改进现有方案，从而实现更低的存储成本和更高的检错率。上述 POR 机制的研究均不支持数据动态操作。文献[39]提出利用纠错码的线性特征来支持部分动态操作，但无法支持插入操作。文献[62]对该机制进行优化，考虑采用 Cauchy Reed-Solomon 线性编码来进行数据预处理，该方法有效地提高了恢复错误的效率，但更新操作需要云服务器重新生成所有的辅助容错信息，导致计算代价较高。文献[63]首次考虑在 POR 方案中加入公平性属性，在维护数据持有者利益的同时维护 CSP 的合法利益，即同时考虑了不诚实的客户指控诚实 CSP 的情形，并提出了一个新的增量签名方案来实现 POR 机制的公平性，采用基于范围的 2-3 树数据结构实现其动态操作，采用 ECC 码实现可恢复验证。但该方案不支持公开验证且性能还有待提升。文献[64]采用固定大小的多项式承诺(Polynomial Commitment)方案，结合同态线性认证器，首次提出了一种支持公开验证且通信成本为常数级的 POR 机制，不支持动态更新操作。文献[65]通过采用 ORAM (Oblivious Random Access Machine)方案[66](即可以用来隐藏 IO 操作的数据访问模式的加密方案)实现了支持动态操作的 POR 机制。ORAM 方案可以模糊化用户访问文件的顺

序、频率、模式等信息，避免了不可信的第三方通过收集用户访问操作信息推断出用户隐私。该机制依赖于 ORAM 方案的研究，目前在存储成本方面不太理想。

3.3 外包存储合规性审计

3.3.1 安全备份审计

采用冗余备份的方式来存储数据时，CSP 可能并未遵循 SLAs 协议，按照约定的备份策略存储数据，如只存储一份或几份数据而对外宣称按约定存储了多份数据。文献[10]提出的 MR-PDP 机制可以确认 CSP 是否提供约定数量的备份，但不支持公开审计。文献[67]基于 BLSHVTs 提出了一种支持公开验证的审计方案。文献[41]中的动态方案只有在所有副本均未被损坏且未更新时才能对挑战做出有效的响应，即 CSP 不能谎报副本数量。文献[68]指出现有备份存储的证明方案均依赖于时间假设，即只要 CSP 在一定时间范围内未响应，验证者将拒绝其回复的证明，并首次针对该问题提出了一个不依赖任何时间假设的构造方案。

3.3.2 存储位置审计

一些特殊的机构要求将自己的数据存储在特定的地理位置或地理范围(如不允许跨境存储等)，然后有些 CSP 可能有意或无意地将这些数据存储在指定位置之外的其他位置以寻求更低的成本。为此，文献[69]提出将 POS 协议(Proof of Storage Protocol)和地理位置相结合，以解决云存储中的位置合规性问题，但未给出具体方案。其中，POS 协议是允许客户在不下载完整数据的情况下验证数据的一种交互式密码协议。文献[70]结合 POS 协议和距离边界协议(Distance-bounding Protocol)提出了一种具体的体系结构 GeoProof，前者主要基于文献[55]方案的变体，后者是客户与 CSP 之间基于 RTT(Round-Trip Time)的一种认证协议，两种协议的结合可使数据拥有者不依赖 CSP 检查外包数据存储位置的合规性。文献[71]指出文献[70]在使用典型的 POS 协议时由于服务器端的计算开销问题涉及不必要的延迟，并对其进行了改进。

结束语 本文对云数据存储安全审计进行了全面介绍，概述了云数据的安全审计需求及其存储安全审计框架内容和指标，并根据其实现原理和功能对现有不同类型审计机制进行了论述，主要包括 PDP 机制、POR 机制和存储合规性审计。值得指出的是，现有审计机制在性能上均未实现实际应用场景中理想的轻量级，在功能方面各有侧重但不能顾全，例如有些机制支持数据恢复但不支持动态更新，支持动态更新后又不支持公开验证或未考虑隐私问题等；从应用实践的角度来讲，对更广泛云环境应用场景的综合考虑尚有努力空间，如分布式协同审计、跨云协同审计、批量审计等；从安全角度来讲，对动态敌手及其博弈有待进一步的考虑。

参 考 文 献

[1] YU X, WEN Q. A View about Cloud Data Security from Data Life Cycle[C]// International Conference on Computational Intelligence and Software Engineering. IEEE, 2010: 1-4.

[2] CHEN L X, XU L. Research on Provable Data Holding and Recovery Technologies in Cloud Storage Services [J]. Computer Research and Development, 2012, 49(S1): 19-25.

[3] DESWARTE Y, QUISQUATER J J, SAÏDANE A. Remote Integrity Checking[C]// Sixth Working Conference on Integrity and Internal Control in Information Systems. Springer, 2003: 1-11.

[4] SHAH M A, BAKER M, MOGUL J C, et al. Auditing to Keep Online Storage Services Honest[C]// USENIX Workshop on Hot Topics inOperating Systems. Usenix Association, 2007: 1-6.

[5] SHAH M A, SWAMINATHAN R, BAKER M. Privacy-Preserving Audit and Extraction of Digital Contents, HPL-2008-32R1 [R]. HP Laboratories, 2008.

[6] FILHO D, BARRETO P S. Demonstrating data possession and uncheatable data transfer[J]. Cryptology Eprint Archive, 2006, 1(1): 150-159.

[7] FRANCESC S, DOMINGO-FERRER J, MARTINEZ-BALLESTE A, et al. Efficient Remote Data Possession Checking in Critical Information Infrastructures [J]. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(8): 1034-1038.

[8] GIUSEPPE A, RANDAL B, REZA C, et al. Provable Data Possession at Untrusted Stores[C]// ACM Conference on Computer and Communications Security. ACM, 2007: 598-610.

[9] GIUSEPPE A, RANDAL B, REZA C, et al. Remote Data CheckingUsing Provable Data Possession [J]. ACM Transactions on Information and System Security, 2011, 14(1): 1-34.

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

[11] GIUSEPPE A, KAMARA S, KATZ J. Proofs of Storage from Homomorphic Identification Protocols[C]// International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2009: 319-333.

[12] 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 & Data Engineering, 2011, 23(9): 1432-1437.

[13] SHACHAM H, WATERS B. Compact Proofs of Retrievability [C]// International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2008: 90-107.

[14] SHACHAM H, WATERS B. Compact Proofs of Retrievability [J]. Journal of Cryptology, 2013, 26(3): 442-483.

[15] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing[C]// International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2001: 514-532.

[16] WANG Q, WANG C, LI J, et al. Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing [C]// European Conference on Research in Computer Security. Springer, 2009: 355-370.

[17] WANG C, WANG Q, REN K, et al. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing[C]// Proceedings of the 29thConference on Information Communications. IEEE Press, 2010: 525-533.

[18] WANG C,CHOW S M,WANG Q,et al. Privacy-Preserving Public Auditing for Secure Cloud Storage[J]. IEEE Transactions on Computers,2013,62(2):362-375.

[19] SHAMIR A. Identity based cryptosystems and signature schemes[J]. In Proceedings of Crypto 84 on Advances in Cryptology,1985:47-53.

[20] ZHAO J,XU C,LI F,et al. Identity-Based Public Verification with Privacy-Preserving for Data Storage Security in Cloud Computing[J]. Ieice Transactions on Fundamentals of Electronics,Communications and Computer Sciences,2013,96(12):2709-2716.

[21] GENTRY C,RAMZAN Z. Identity-Based aggregate signatures [C]// International Conference on Theory and Practice of Public-Key Cryptography. Springer,2006:257-273.

[22] DOMINGO-FERRER J,QIN B,WU Q,et al. Identity-Based Remote Data Possession Checking in Public Clouds[J]. IET Information Security,2014,8(2):114-121.

[23] PENG S,ZHOU F,XU J,et al. Identity-Based Distributed Provable Data Possession in Multicloud Storage [J]. IEEE Transactions on Services Computing,2016,9(6):996-998.

[24] YU Y,AU M H,ATENIESE G,et al. Identity-based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage[J]. IEEE Transactions on Information Forensics and Security,2017,12(4):767-778.

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

[26] YU Y,XUE L,AU M H,et al. Cloud data integrity checking with an identity-based auditing mechanism from RSA[J]. Future GenerationComputer Systems,2016,C(62):85-91.

[27] 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. Springer: Springer,2016:83-100.

[28] XU Z,WU L,KHAN M K,et al. A secure and efficient public auditing scheme using RSA algorithm for cloud storage[J]. The Journal of Supercomputing,2017,73(12):5285-5309.

[29] LIU Z,LIAO Y,YANG X,et al. Identity-Based Remote Data Integrity Checking of Cloud Storage From Lattices[C]// International Conference on Big Data Computing & Communications. IEEE Computer Society,2017:128-135.

[30] WANG Y,WU Q,QIN B,et al. Identity-based data outsourcing with comprehensive auditing in clouds[J]. IEEE Transactions on Information Forensics and Security,2017,12(4):940-952.

[31] TIAN M,YE S B,HONG Z,et al. Identity-based proofs of storage with enhanced privacy[C]// International Conference on Algorithms and Architectures for Parallel Processing. Springer,2018:461-480.

[32] XUE J,XU C,ZHAO J,et al. Identity-based public auditing for cloud storage systems against malicious auditors via blockchain [J]. Science China Information Sciences,2019,62(3):1-16.

[33] WANG S H,PAN X X,WANG Z W,et al. Analysis and improvement on identity-basedcloud data integrityverification scheme [J]. Journal on Communications,2018(11):98-105.

[34] TIAN M M,GAO C,CHEN J. Identity-based cloud storage integrity checking from lattices[J]. Journal on Communications,2019,40(4):128-139.

[35] LI Y,YU Y,MIN G,et al. Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems[J]. IEEE Transactions on Dependable and Secure Computing,2017,1(16):72-83.

[36] SCHWARZ T J,MILLER E L. Store,Forget,and Check:Using Algebraic Signatures to Check Remotely Administered Storage [C]// IEEE International Conference on Distributed Computing Systems. IEEE Computer Society,2006:1-12.

[37] CHEN L X. Using algebraic signatures for remote data possession checking[C]// 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE,2011:289-294.

[38] CHEN L X. Using algebraic signatures to check data possession in cloud storage [J]. Future Generation Computer Systems,2013,29(7):1709-1715.

[39] GIUSEPPE A,DI PIETRO R,MANCINI L V,et al. Scalable and Efficient Provable Data Possession[C]// International Conference on Security and Privacy in Communication Networks. ACM,2008:1-10.

[40] WANG C,WANG Q,REN K,et al. Ensuring Data Storage Security in Cloud Computing[C]// International Conference onAdvanced Computing,Networking and Security. IEEE Computer Society,2013:214-219.

[41] BARSOUM A F,ANWAR H M. On Verifying Dynamic Multiple Data Copies over Cloud Servers [EB/OL]. [2019-10-17]. https://eprint.iacr.org/2011/447.pdf.

[42] WANG Q,REN K,YU S,et al. Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance [J]. ACM Transactions on Sensor Networks,2011,8(1):1-24.

[43] YONG Y,MAN H A,YI M,et al. Enhanced privacy of a remote data integrity checking protocol for secure cloud storage[J]. International Journal of Information Security,2015,14(4):307-318.

[44] ZHANG Y B M. Efficient dynamic provable possession of remote data via balanced update trees[C]// ACM SIGSAC Symposium on Information,Computer and Communications Security. ACM,2013:183-194.

[45] ERWAY C C,KÜPÇÜ A,CHARALAMPOS P,et al. Dynamic Provable Data Possession[J]. ACM Transactions on Information and System Security,2015,17(4):1-29.

[46] ERWAY C C,KÜPÇÜ A,CHARALAMPOS P,et al. Dynamic Provable Data Possession[C]// ACM Conference on Computer and Communications Security. 2009:213-222.

[47] GOODRICH M T,TAMASSIA R,SCHWERIN A. Implementation of an Authenticated Dictionary with Skip Lists and Commutative Hashing[C]// Darpa Information Survivability Conference & Exposition II. IEEE,2001:68-82.

[48] SAXENA R,DEY S. Cloud Audit:A Data Integrity Verification Approach for Cloud Computing[J]. Procedia Computer Science,2016,89:142-151.

[49] YAN Z, HU H X, GAIL-JOON A, et al. Efficient audit service outsourcing for data integrity in clouds[J]. Journal of Systems and Software, 2012, 85(5): 1083-1095.

[50] ZHU Y, HU H, AHN G J, et al. Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(12): 2231-2244.

[51] TIAN M, YE S B, HONG Z, et al. Identity-based proofs of storage with enhanced privacy[C]// International Conference on Algorithms and Architectures for Parallel Processing. Springer, 2018: 461-480.

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

[53] WANG B, LI B, 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.

[54] ZHANG J H, ZHAO X B. Efficient chameleon hashing-basedprivacy-preserving auditing in cloud storage[J]. Cluster Computing, 2016, 19(1): 47-56.

[55] JUELS A, KALISKI J S. Pors: Proofs of Retrievability for Large Files[C]// ACM Conference on Computer and Communications Security. ACM, 2007: 584-597.

[56] CHEN B, REZA C, GIUSEPPE A, et al. Remote Data Checking for Network Coding-based Distributed Storage Systems[C]// ACM Workshop on Cloud Computing Security Workshop. ACM, 2010: 31-42.

[57] HU Y C, CHEN H, LEE P C, et al. NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-clouds[C]// USENIX Conference on File and Storage Technologies. Usenix Association. 2012: 1-8.

[58] BOWERS K D, JUELS A, OPREA A. HAIL: A High-availability and Integrity Layer for Cloud Storage[C]// ACM Conference on Computer and Communications Security. ACM, 2009: 187-198.

[59] CHEN H, LEE P C. Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage[C]// 2012 IEEE 31st Symposium on Reliable Distributed Systems. IEEE, 2012: 51-60.

[60] BOWERS K D, JUELS A, OPREA A. Proofs of Retrievability: Theory and Implementation[C]// ACM Workshop on Cloud Computing Security. ACM, 2009: 43-54.

[61] DODIS Y, SALIL V, DANIEL W. Proofs of Retrievability via Hardness Amplification[C]// Theory of Cryptography Conference on Theory of Cryptography. Springer, 2009: 109-127.

[62] CHEN B C R. Robust dynamic remote data checking for public clouds[C]// ACM Conference on Computer and Communications Security. ACM, 2012: 1043-1045.

[63] ZHENG Q J, XU S H. Fair and Dynamic Proofs of Retrievability[C]// ACM Conference on Data and Application Security and Privacy. ACM, 2011: 237- 248.

[64] YUAN J W, YU S C. Proofs of Retrievability with Public Verifiability and Constant Communication Cost in Cloud[C]// Proceedings of the 2013 International Workshop on Security in Cloud. Computing: ACM, 2013: 19-26.

[65] DAVID C, ALPTEKIN K, DANIEL W. Dynamic Proofs of Retrievability Via Oblivious RAM[J]. Journal of Cryptology, 2017, 30(1): 22-57.

[66] GOLDREICH O O R. Software protection and simulation on oblivious RAMs[J]. Journal of the Acm, 1996, 43(3): 431-473.

[67] HAO Z, YU N H. A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability[C]// International Symposium on Data. IEEE, 2010: 84-89.

[68] DAMGÅRD I, GANESH C, ORLANDI C, et al. Proofs of Replicated Storage Without Timing Assumptions[C]// Advances in Cryptology(CRYPTO 2019). Springer, 2019: 355-380.

[69] PETERSON Z J, GONDREE M, BEVERLY R. A Position Paper on Data Sovereignty: The Importance of Geolocating Data in the Cloud[C]// USENIX Conference on Hot Topics in Cloud Computing. Usenix Association, 2011: 9.

[70] ALBESHRI A, BOYD C, NIETO J G. GeoProof: Proofs of Geographic Location for Cloud Computing Environment[C]// International Conference on Distributed Computing Systems Workshops. IEEE Computer Society, 2012: 506-514.

[71] ALBESHRI A, BOYD C, NIETO J. Enhanced GeoProof: improved geographic assurance for data in the cloud[J]. International Journal of Information Security, 2014, 13(2): 191-198.

BAI Li-fang, born in 1990, doctorial student, is a member of China Computer Federation. Her main research interests include cloud storage security and network security protocol.

ZHU Yue-fei, born in 1964, Ph. D, professor, Ph. D supervisor. His main research interests include cryptography, data security and network security protocol.