计 算 机 研 究 与 发 展
Journal of Computer Research and Development

https://crad.ict.ac.cn
62(9): 2283−2297, 2025

云存储数据加密重删攻击与防御技术研究进展

吴 健[1] 付印金[1] 方艳梅[1] 刘 垚[2] 付 伟[4] 操晓春[1] 肖 侬[3]

[1](中山大学网络空间安全学院 广东深圳 518107)

[2](中山大学计算机学院 广州 510006)

[3](中山大学微电子科学与技术学院 广东珠海 519082)

[4](海军工程大学信息安全系 武汉 430033)

(wjey0412@163.com)

Review on Encrypted Data Deduplication Attacks and Countermeasures in Cloud Storage

Wu Jian[1], Fu Yinjin[1], Fang Yanmei[1], Liu Yao[2], Fu Wei[4], Cao Xiaochun[1], and Xiao Nong[3]

[1](*School of Cyber Science and Technology*, *Sun Yat-sen University*, *Shenzhen*, *Guangdong* 518107)

[2](*School of Computer Science and Engineering*, *Sun Yat-sen University*, *Guangzhou* 510006)

[3](*School of Microelectronics Science and Technology*, *Sun Yat-sen University*, *Zhuhai*, *Guangdong* 519082)

[4](*Department of Information Security*, *Naval University of Engineering*, *Wuhan* 430033)

Abstract Data deduplication is a vital technology for efficiently managing big data, widely adopted in cloud storage systems to reduce redundancy and save space. To integrate deduplication with encryption, convergent encryption has become a common approach. This method allows for the encryption of data while still enabling deduplication to produce the same ciphertext for identical plaintexts. However, cloud service providers' outsourcing models and the deterministic nature of convergent encryption can introduce data security issues. The encryption patterns of data can become predictable, potentially exposing sensitive information to attackers, which may create serious security implications. As a result, encrypted data deduplication has emerged as an important research topic in cloud storage security. We firstly introduce the concept of data deduplication, encrypted deduplication algorithms, and discuss the security challenges associated with encrypting and deduplicating data in cloud storage. Then we review the current research status from both attack and defense perspectives, covering three main types of attacks: brute force attacks, which try to decrypt data through extensive guessing; frequency analysis attacks, which exploit frequency characteristics in ciphertexts; and side-channel attacks, which leverage information from response or traffic characteristics. For each attack type, representative defense strategies are analyzed along with their strengths and weaknesses. Finally, we highlight the challenges faced by existing encrypted data deduplication defenses and suggest future research directions aimed at improving these techniques.

Key words cloud storage; data deduplication; data encryption; means of attacking; defense measures

收稿日期: 2024-05-31; **修回日期**: 2025-02-07

基金项目: 国家重点研发计划项目(2022YFB4500304); 国家自然科学基金项目(62332021, 61832020, 62276273); CCF-华为胡杨林基金存储领域专项课题 (CCF-HuaweiST2021001)

This work is supported by the National Key Research and Development Program of China (2022YFB4500304), the National Natural Science Foundation of China (62332021, 61832020, 62276273), and CCF-The Huawei Populus Euphratica Fund-storage Special Project (CCF-HuaweiST2021001).

通信作者: 付印金(fuyj27@mail.sysu.edu.cn)

摘 要 重复数据删除作为一种面向大数据的高效缩减技术,已经被广泛应用于各种云存储系统和服务中,为了兼容数据重删和加密,通常采用收敛加密.然而,这种云服务商的外包存储方式以及确定性的加密方式会导致一系列数据安全问题.目前,数据加密重删技术已成为云存储领域的研究热点.首先介绍重复数据删除技术的概念、基础加密重删算法和云存储中数据加密重删的安全挑战.其次从攻击和防御的角度阐述当前云存储数据加密重删安全研究现状,攻击包括:蛮力攻击、频率攻击、侧信道攻击3种类型.围绕每种攻击类型,梳理对应的代表性防御方案,并总结各个方案的优势和缺陷.最后,针对当前数据加密重删防御方案存在的问题进行总结,并对未来的研究方向进行展望.

关键词 云存储;数据重删;数据加密;攻击手段;防御措施

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202440379 **CSTR:** 32373.14.issn1000-1239.202440379

随着数据爆炸式增长,云存储已成为大数据计算和分析的关键基础设施.根据国际数据公司(IDC)的预测,到2025年,全球数据量将从2018年的33 ZB增长到175 ZB[1].根据微软公司和EMC公司的云存储研究表明,云存储服务中分别有50%和85%的数据是重复的[2].相关研究也表明,备份、归档、虚拟机映像等存在80%~90%的重复数据[3].重复数据删除技术作为一种快速消除数据副本的高效缩减技术,对于云服务提供商和用户来说,不仅能节省云存储空间和通信开销,也能降低用户的云成本开销.该技术已被广泛应用于各种存储系统与云存储服务.

然而,随着用户对数据隐私保护以及网络空间安全诉求的提高,传统云存储服务并未将安全问题纳入系统设计范围,导致现有的不少云服务仍然采用明文形式存储外包数据,这对于用户隐私保护和维护网络安全极为不利.同时,传统密码学的数据安全加密与重复数据删除本质上是不兼容的,这给加密数据重删处理带来严重的安全风险.因此,针对加密数据的重复数据安全删除成为当今云存储领域内的一个重要研究方向.本文通过分析近年来云存储安全重删的研究进展,期望能为云存储安全重删技术的发展提供研究参考.

1 云存储安全重删概述

1.1 重复数据删除技术

重复数据删除(以下简称重删)是一种面向大数据的高效数据缩减技术,通过识别重复的数据对象(如文件或块),可以检测已存在重复数据对象并且只保留唯一的副本.重删处理的基本流程可以分为数据分块、指纹计算、重复检测和消除、数据存储[4]4个步骤.首先,系统对传入的文件按照相应的分块算法将文件切分成块,再通过加密哈希算法计算出每个块的指纹.其次,重删系统通过在块索引中查询每个块指纹比对是否存在相同匹配项.最后,系统选择不重复的唯一数据块进行存储,它是一种以时间换空间、以计算换存储的方法,可以使用先进的数据结构和系统架构优化来提升重删处理性能[5].

根据数据分块粒度不同,重删可以分为文件级重删和块级重删.在文件级重删中,文件是重复数据识别和存储的基本数据对象.而在块级重删中,系统按照定长或变长分块算法将文件数据流划分为块,以块为数据对象.数据分块粒度越小,去重的数据就越多,但相应的计算开销也会越大.

根据数据重删操作所处的I/O处理阶段不同,重删技术还可以划分为离线重删和在线重删.其中离线重删通过设置缓冲区,待所有数据缓存完毕后,统一进行重删.而在线重删技术在数据到达存储设备之前即时完成重删处理,只存储数据块的唯一副本.与离线重删相比,在线重删不需要事先准备大容量的数据缓冲区,但其由数据重删技术所导致的数据写回过程中产生的块碎片会降低存储系统的性能.

云存储重删系统根据重删操作发生位置的差异,可以分为客户端重删和服务器端重删.如图1所示,

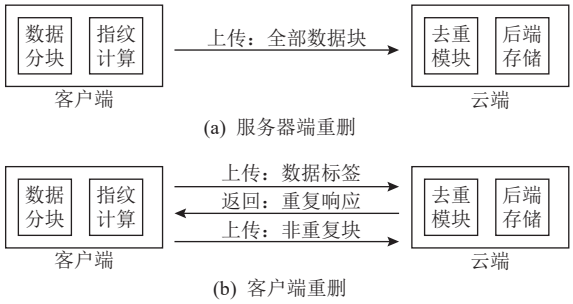


Fig. 1 Two deduplication modes at different positions

图1 2种不同位置的重删模式

在客户端重删中，用户先上传待发送数据的指纹，云服务器接收并进行判断该数据是否已经存储，随后向客户端返回重删响应. 当云服务器端响应为“1”时，表明数据已经存储，用户无需上传；当响应为“0”时，则表示该数据并未存储，需要上传. 在服务器端重删模式中，用户需要将数据全部上传至云服务器，云端根据收到的数据进行重复检测和删除处理. 与客户端重删模式相比，基于服务器端重删的用户无法得知数据的实际重删状态，需要上传全部数据，大大增加了网络带宽和通信开销. 随着物联网、边缘计算的兴起，出现了基于网关[6]或边缘计算[7]的重删模式. 相比于传统的云端重删，这种新型的重删模式不仅可以减少网络带宽和通信开销，还可以抵抗客户端重删模式中潜在的侧信道威胁.

1.2 收敛加密算法

随着云存储环境中产生海量重复数据的场景增多，安全重删正在成为云存储服务最重要的需求之一. 然而，在云存储多用户场景中，每个用户采用自己的私钥对文件进行加密，这就导致相同的文件会被加密成不同的密文，因此无法对密文进行数据重删. 为了使数据加密和重删兼容，Douceur 等人[8]采用收敛加密（convergent encryption, CE）算法用于加密重复数据删除. 收敛加密使用明文的指纹 Fp 作为收敛加密密钥 Key，即 $Key \leftarrow Fp$，密文 $C \leftarrow Encrypt(M, Key)$，解密得到明文 $M = Decrypt(C, Key)$，故不同用户间相同的明文经过收敛加密后依旧相同.Bellare 等人[9]对收敛加密算法及其变种进行总结，并将其形式化为基于内容的消息锁加密 (message-locked encryption, MLE) 机制，通过引入系统安全参数 θ 提升加密密钥的安全性：明文密钥 $Key \leftarrow KeyGen(Fp, \theta)$，加密得到密文 $C \leftarrow Encrypt(M, Key)$. 其次对明文进行标准对称加密，即密文 $C \leftarrow Decrypt(M, Key)$，并生成文件标签 $T \leftarrow TagHash(M)$，实现对密文数据的重复对比.

1.3 云存储重删安全挑战与攻击威胁

针对云存储重删服务中密文数据主要存在的机密性与完整性安全挑战，本节主要阐述和总结蛮力攻击、侧信道攻击、频率攻击等 3 种攻击威胁模型.

1）机密性挑战

为了兼容数据加密与重删，通常采用收敛加密原语加密数据内容，然而这种确定性加密方式容易导致密文遭受蛮力攻击. 例如，外部攻击者在窃取密文序列后，可以尝试使用暴力破解的方式获得原始明文数据. 另一方面，确定性加密还会泄露底层的明文块分布频率，攻击者会利用块的局部性特征对密文块发动不同级别的频率分析[10]来加快暴力破解的速度. 因此，传统的收敛加密模式会给云存储的数据安全重删带来巨大的机密性挑战.

2）完整性挑战

在客户端进行跨用户重删过程中，恶意用户可以仅凭文件的指纹值获取云服务器的信任，以此发动侧信道攻击[11]. 攻击者通过文件指纹不仅可以学习目标文件的存在性信息和敏感内容，还可以非法获得目标文件的控制权[12]，并下载或替换原有正确的文件. 因此，如何提高客户端安全重删模式下的数据完整性是一项重要挑战.

表 1 总结了云存储加密重删面临的主要攻击威胁. 收敛加密的确定性加密导致数据容易遭受蛮力攻击和频率攻击，解决此类攻击的主要方法是部署可信的第三方密钥服务器，或设计更复杂的收敛加密原语. 而客户端重删模式的局限性使得数据容易遭受侧信道攻击，主要解决手段是在服务器端部署相应防御算法以及增加所有权证明步骤. 在后续的内容中我们会深入讨论解决各种攻击对应的防御机制.

Table 1 Attack and Defense Methods of Cloud Storage Encryption Deduplication

表 1 云存储加密重删的攻击方法与防御手段

攻击类型	攻击模型	主要解决手段
蛮力攻击	采用暴力破解的方式推断密钥	内容加密、密钥共享、可信执行环境
频率攻击	通过块频率分布和局部性信息推断明文	频率加密、顺序置乱
侧信道攻击	凭借文件指纹和确定性响应窃取数据隐私	随机阈值策略、所有权证明

2 蛮力攻击

2.1 攻击定义

蛮力攻击是指攻击者使用穷举方法，尝试所有可能的密钥组合，直到找到正确的密钥从而破解加密算法. Douceur 等人[8]定义的收敛加密原语，即用数据内容的指纹作为其加密密钥. 由于其密钥取决于数据内容本身，使得数据密文的加密密钥与数据明文内容存在一一对应的关系，这种确定性的映射关系导致加密密钥容易遭受蛮力攻击. 例如，对于已知明文空间来源和部分上下文信息的密文数据，攻击者可以在较短的时间内采用暴力穷举的方式计算所有候选明文块的内容并得出密钥，检查解密结果是否与预期明文相匹配，从而达到破解明文数据的目的.

2.2 防御方案

2.2.1 基于 MLE 的防御方案

Bellare 等人[13]基于MLE加密模型提出了DupLESS系统架构,通过引入额外的第三方密钥服务器实现安全重删.如图 2 所示,客户端通过与密钥服务器的交互完成 MLE 密钥的加解密操作,随后客户端与服务器进行通信,实现文件上传或下载.具体来说,DupLESS 采用 RSA 盲签名算法[14]实现不经意伪随机函数(OPRF)并生成文件密钥,该密钥由一个系统密钥和明文块内容共同计算得出.具体来说,客户端将明文块的“盲”指纹发送至密钥服务器,密钥服务器使用系统密钥生成该指纹的 RSA 哈希值,即为 MLE 密钥.因此,密钥管理器可以在不知道明文块指纹的情况下为具有相同指纹的明文块计算得到相同的密钥.此外,DupLESS 还可通过密钥服务器限制客户端的块上传请求速率,以抵御恶意客户端的在线蛮力攻击.

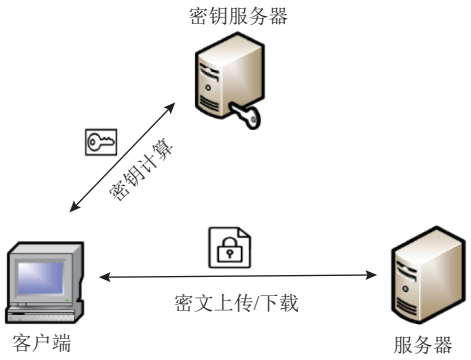


Fig. 2 Key server-aid architecture

图 2 密钥服务器辅助架构

Chen 等人[15]针对大容量文件提出了一种面向块级的消息锁定加密原语 BL-MLE,该方案使用文件密钥作为主密钥对块密钥进行双重加密,并将块密钥也上传至云端以减轻本地存储开销.在这种双重加密机制中,用户只需通过主密钥就可以获得数据的访问权限.为了支持频繁的密文更新,Zhao 等人[16]在此基础之上提出了一种支持块级更新机制的消息锁定加密原语 UMLE,通过实现块二叉树的递归来改变原有的静态 BL-MLE 加密算法.当更新一个大小为 M 的文件时,UMLE 算法的时间复杂度仅为 $O(\log M)$.Ha 等人[17]提出了一种基于 MLE 的可更新加密方案,其中数据所有者可以使用原始密钥和新密钥以及异或操作来生成更新令牌,然后将它们发送到云服务器来更新外包数据的密钥.密钥更新可以有效抵抗离线的蛮力攻击,提升密文的安全性.

Jiang 等人[18]针对相似数据实现了一种广义安全重删方案 FuzzyMLE,该方案主要利用数据的相似性哈希和基于纠删码的模糊提取器,并基于 DupLESS 架构设计了第三方密钥服务器辅助加密模式.相比传统的精确重删,FuzzyMLE 可实现更高的数据缩减率.Song 等人[19]则基于云共享媒体场景实现了相似图片的重复数据删除方案 SimLESS,上传者可以通过设置文件的相似阈值,从而控制是否执行重复数据删除操作,实现细粒度的存储效率优化.

2.2.2 基于秘密共享的防御方案

Li 等人[20]在此基础上提出了一种基于秘密共享的分布式密钥管理方案 Dekey.首先,每个用户管理各自的文件级主密钥.其次,Dekey 通过 Ramp 秘密共享方案[21]将块密钥拆分为多个密钥共享分量,并实现了收敛的 RSSS 秘密共享算法,将秘密分量分发给多个密钥服务器.基于分布式的多密钥服务器方案一定程度上规避了 DupLESS 中单点故障的风险,但随着文件数量的增加,密钥开销也比单服务器模式更大.

Li 等人[22]提出了无密钥模式的多云安全重删系统 CDStore,文件经过收敛的 CAONT-RS 秘密共享算法[23]形成若干秘密分量,并分发到多个云存储.同时保留秘密共享的可靠性和无密钥的安全特性.

高文静等人[24]提出了一种基于双层加密和密钥共享的云数据安全去重方案,摆脱了可信第三方的束缚.通过划分数据流行度,对隐私程度较高的非流行数据采用双层加密机制进行保护,内层为收敛加密,外层为对称加密.借助门限秘密共享机制[25],将外层加密使用的加密密钥保存到多个密钥管理服务器,实现不同用户间的密钥共享.

2.2.3 基于可信执行环境的防御方案

针对 DupLESS 中耗时的 OPRF 密钥计算协议,Ren 等人[26]提出 SGXDedup,通过引入软件防护扩展(software guarded extensions, SGX)硬件辅助机制[27]保护 MLE 密钥计算.SGX 通过创建受到保护的内存区域 Enclave,将机密性的代码和数据在安全容器中运行,并实现与其他应用程序的隔离.如图 3 所示,SGXDedup 建立在 DupLESS 的三方的架构上,在密钥服务器中部署安全容器 Enclave 以代替 OPRF 协议来保护 MLE 密钥的计算.在加密明文块时,客户端和密钥服务器的 Enclave 建立基于共享盲密钥的安全通道,由于 SGX 的屏蔽机制,密钥服务器或攻击者无法通过任何外部方式访问 Enclave 中的代码和数据,故无法学习任何明文块的内容.

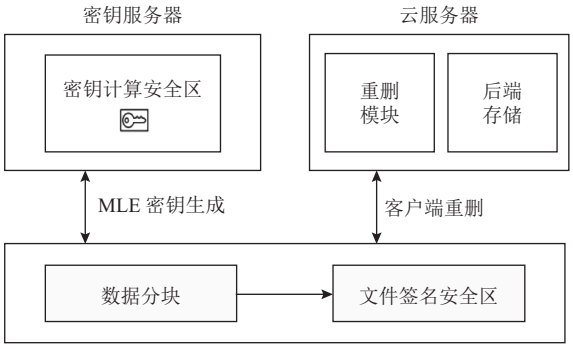


Fig. 3 SGXDedup architecture

图 3 SGXDedup 架构

SGXDedup 采用硬件计算来代替计算开销巨大的 OPRF 协议，通过在客户端与 Enclave 容器建立端到端的受保护的加密通道和设置多层密钥保护，不仅提升 MLE 密钥的计算速率，而且可以避免 DupLESS 中单点故障导致的密钥泄露.

Yang 等人[28]提出一种基于可信硬件机制 SGX 的防御方案 DEBE 作为一种新的加密重删模式：DEBE 先对明文块执行重删，然后对非冗余块使用语义安全的加密进行保护. DEBE 主要采用 2 种关键设计：1)在云中部署 SGX 的可信容器 Enclave. 客户端通过与云中的 Enclave 容器建立安全通道进行明文块的上传. 接着云中的 Enclave 先后执行重删、压缩和加密. 该过程利用 SGX 的屏蔽机制，云或攻击者将无法访问到明文块的任何内容. 2)针对 Enclave 的页缓存容量限制以及备份数据集中重复块的频率分布，根据数据块频率分 2 阶段进行重删处理. 在 Enclave 内外分别维护一个最小堆 Top-k 索引和一个全局索引，执行频繁块和非频繁块的重删.

2.2.4 基于密钥口令的防御方案

基于服务器辅助的防御方案通常需要在用户和云之间部署一个第三方密钥服务器，这在实际应用场景中通常无法实现，因此 Liu 等人[29]提出了无需第三方密钥服务器的加密重删方案，其核心思想是使用基于口令的认证密钥协议(password authenticated key exchange, PAKE)与同态加密方法相结合[30]，即双方在没有预先共享密钥的情况下，使用一个低熵的口令来协商出一个共享密钥. 在基于 PAKE 的无密钥服务器方案中，每个明文块的验证信息包含块指纹和一个短签名. 服务器通过检索具有相同签名的其他客户端并在客户端间执行 PAKE 协议. 若检索不到，则该客户端将使用一个私有密钥进行加密. 哈冠雄等人[31]在其基础上实现了基于数据流行度的加密去重方案，其主要思想是基于 Count-Min sketch 算法[32]和 Merkle Puzzles 协议[33]实现数据流行度的安全统计，并在用户间执行对称密钥口令交换协议[34](symmetric password authenticated key exchange, sPAKE)来实现不流行数据的加密去重.

然而，PAKE 协议的短哈希值会存在一定的冲突率，这就导致在发生哈希冲突时，客户端可能需要执行多轮验证协议，造成额外的通信开销. 其次，密钥交换过程需要其他用户在整个安全重删过程中始终保持在线状态才能够完成密钥的共享. 因此，该方案将会随着用户数的增加而造成巨大的通信开销.

咸鹤群等人[35]提出基于双线性映射的文件重复标识方案，并与群签名的数据签名方案相结合，实现了文件流行度的隐私查询. 该方案采用多层混合加密模式，利用广播加密技术[36]进行密钥的发放，可以抵抗针对收敛加密的蛮力攻击，其缺陷是基于双线性映射的冗余检测会造成较高的计算开销.

表 2 从是否引入第三方和加密开销的时间复杂度 2 个方面总结了 3 种类型的抗蛮力攻击的防御方案. 基于收敛加密的方案和基于密钥口令的方案具有较高的加密计算开销和密钥存储开销，而基于秘密分享的方案则实现了无密钥保护，可以有效防止单点故障，其缺陷是只适用于分布式密文存储场景. 基于可信执行环境方案的系统性能则要优于传统的密码学方案，这是因为基于硬件的保护机制取代了传统软件加密算法，极大提高了系统的加密性能.

Table 2 Comparison of Schemes Against BFA

表 2 抗蛮力攻击的方案对比

方案	算法	第三方	加密开销
文献 [13]	收敛加密	√	$O(m)(Hash+PRF+AES)$
文献 [16]		×	$O(m)(Hash+AES)+O(\log m)$
文献 [20]	秘密分享	√	$O(m)(Hash+RSSS)+O(1)(m)$
文献 [22]		×	$O(1)(Hash+AES)+O(m)(CAONT)$
文献 [24]		√	$O(n)(Hash)+O(m)(RSSS)$
文献 [29]	密钥口令	×	$O(u)(Hash+PAKE+AES)$
文献 [31]		×	$O(m)(Hash+AES)+O(u)(sPAKE)$
文献 [35]		×	$O(m)(Hash+Pair+AES)+O(u)(HIBE)$

注：*AES* 表示执行 1 次对称加密的时间开销；*PAKE* 表示执行 PAKE 密钥协议的开销；*PRF* 表示执行 1 次伪随机函数的开销；*RSSS* 表示执行 Ramp 秘密共享算法的开销；*HIBE* 表示执行广播加密的开销；m，u 分别表示数据大小和数据拥有者的数量. “√”表示引入第三方；“×”表示未引入第三方. BFA 代表蛮力攻击.

3 频率攻击

Li 等人[10]在 2020 年的研究表明，确定性加密不

仅会使密文遭受暴力破解，而且会泄露原始明文的底层频率分布，攻击者可以对窃取的密文进行频率分析并推断原始明文. 本节介绍频率攻击及其2种应对方法.

3.1 攻击定义

频率攻击起源于古典密码分析攻击[37]. 在重复数据删除的场景中，实际的备份数据集经常表现出倾斜的频率分布，即一小部分数据块出现频率非常高，大部分数据块出现频率非常少，这种频率分布允许攻击者在窃取未经重删的完整密文块序列时，通过已知的辅助信息来发动频率分析并推断原始明文块. Li 等人[10]从备份数据集中块频率分布和局部性的角度提出了针对加密重删的2种攻击模型，即基于频率的基础攻击模型和基于块局部性的位置攻击.

基础攻击模型如图4所示，假设M为某一最新版本备份文件的密文块序列，C为先前版本备份文件的明文块序列，2个块序列都保持原有的逻辑顺序且未被重删. 具体推断过程如下：攻击者分别基于C序列和M序列从高到低统计每个块的出现频率并排序，并按频率高低顺序依次匹配块M_i和C_i，即M_i和C_i在各自的序列中有相同的频率排名. 该攻击模型很大概率可以成功推断出备份版本间高频率分布的频繁块，对于大部分的低频率分布的频繁块则较难被推断出.

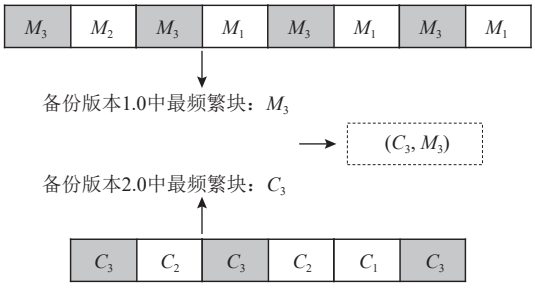


Fig. 4 Frequency inference based on adjacent versions

图4 基于相邻版本的频率推断

基于块局部性的增强频率攻击模型如图5所示，该模型基于备份版本文件的局部性分布规律：如果明文块M_i与密文块C_i的频率相对应，那么M_{i+1}，M_{i-1}与C_{i+1}，C_{i-1}也可能相对应，这因为在备份流中，数据块的局部性特征表明，块的顺序是相对固定的. 在此基础之上，还可以通过判断相邻块的大小来阐述更高的推断率.

2022年 Li 等人[38]在块局部性的基础上提出了一种基于相对频率分布的攻击. 相对频率分布表征了数据在逻辑顺序中共现的可能性，即相同数据的

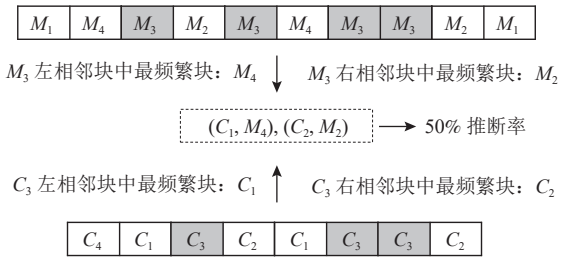


Fig. 5 Enhancement frequency inference based on block locality

图5 基于块局部性的增强频率推断

相对频率分布在不同的备份中是稳定的. 因此，这种攻击模型可以通过检查每对密文和明文的相对频率分布来过滤错误推断的匹配序列和确保推断密文对应明文的高置信度.

3.2 防御方案

3.2.1 基于频率改变的防御方案

Stanek 等人[39]提出了一种基于文件流行度的加密重删方案，其主要原理是：当文件重复数达到一定阈值的时候，将被判定为流行文件. 具体来说，服务器对不同流行程度的文件进行双层的加密，即内层的流行文件进行传统的收敛加密，以保证重删. 外层的不流行文件在收敛加密的基础上通过独立的密钥服务器辅助的非对称阈值加密方案获得. 当外层不流行文件的上传数量达到阈值时，服务器可以在外层对该文件进行解密，从而执行重删. 此外，Stanek 等人[40]还引入门限收敛密码系统以保护不流行数据加密密钥的安全性.

通过这种方式，基于流行度的混合加密通过牺牲部分存储空间以获取更高级别的安全性. 基于文件流行度的加密方案对文件从用户共享数的角度区分机密性，对于不流行的数据，其加密等级可以抵抗普通的蛮力攻击和频率攻击，而高频率的流行数据依然容易遭受频率攻击.

针对备份工作负载中频率高的数据块更容易产生信息泄露，Yang 等人[41]提出了一种存储效率与数据机密性可调的加密原语 TED. 其核心思想是明文块的密钥不仅取决于块内容，还取决于冗余副本数. TED 建立在 DupLESS 密钥服务器辅助的架构上，所有明文块仍采用确定性 MLE 加密. 当某个明文块的重复数量到达阈值时，则采用不同的 MLE 密钥进行加密，即 MLE 密钥由块内容、块频率以及一个可调节的平衡参数共同决定.

Xie 等人[42]在 TED 的基础之上实现了基于边缘计算的频率保护重删方案，该方案通过定义每个数

据块的安全级别来量化用户的安全需求，并设计了一种具有多个安全级别的混合加密方案. 该方案可以有效减少密文块的频率泄露和推断概率.

Miranda 等人[43] 基于可信执行环境 SGX 提出了具有频率保护的重删系统 S2Dedup. 该方案利用备份文件的重删局部性特征，实现了基于时间边界的 Epoch 频率加密，并利用可信执行环境技术保护数据传输和中间哈希计算，有效地抵抗频率攻击.

相比于基于文件流行度的加密重删防御方案，TED 和 S2Dedup 均实现了可调节的频率加密方式，前者基于全局的副本频率，后者则基于每个 Epoch 内的局部特征，使得云或用户可以在存储效率和数据机密性之间实现更细粒度的平衡. 然而，这 2 种加密重删方案以牺牲存储空间为代价，实现更安全的加密模式.

3.2.2 基于顺序改变的防御方案

Li 等人[10] 提出了最小指纹加密与顺序置乱的方法来抵抗基于块局部性的频率攻击. 具体来说，该方法将数据集划分为超块. 基于 Broder 定理[44] 提出了最小指纹 MinHash 加密. 该定理认为如果 2 个超块的指纹序列中的最小指纹是相同的，那么这 2 个超块大概率是相同的. 因此，对于每个超块，首先选取超块序列中的最小指纹作为加密密钥，并对超块中的每个块加密. 其次，在 MinHash 加密的基础上，对每个超块内的数据块分布进行置乱，破坏原有的逻辑顺序和局部性，而每个超块间的逻辑顺序不变. 在具体的存储设计中，重删后存储的容器单元大小大于超块大小，因此避免了顺序改变带来的恢复性能降低.MinHash 加密在一定程度上改变密文块的频率分布，同时也牺牲重删性能，增加存储开销. 数据块逻辑置乱和 MinHash 加密结合的防御方案大大降低基于块局部性的频率攻击的推断率，并获得较低的存储开销.

表 3 从是否引入第三方、加密开销、加密类型和安全级别 4 个方面总结了抗频率攻击的防御方案. 基于流行度的方案[39−40] 具有较低的安全级别，这是因为该类方案仅对低频率的块提供双层加密，依然存在高频率块推断的风险. 其次，基于可信执行环境的方案[43] 采用先去重后单密钥加密模式，因此在密钥开销方面要优于采用确定性加密的保护方案[10,39].

Table 3 Comparison of Schemes Against Frequency Attack

表 3 抗频率攻击的方案对比

方案	引入第三方	加密开销	加密类型	安全级别
文献 [10]	×	高	相似性加密	高
文献 [39]	√	高	双重加密	低
文献 [40]	√	中	双层加密	中
文献 [41]	√	中	概率加密	中
文献 [42]	√	高	多层加密	中
文献 [43]	×	低	单密钥加密	高

注: “√”表示引入第三方; “×”表示未引入第三方.

4 侧信道攻击

在客户端重复数据删除模式中，用户通过上传一个文件指纹便可获得云端的重删响应信号，这导致云服务器中的文件信息以一种非常规的方式通过侧信道泄露. 本节介绍侧信道攻击及其防御方案.

4.1 攻击定义

侧信道攻击指攻击者利用计算机不经意间释放出的信息或信号来非法窃取相关信息的攻击模式[45]，例如通过键盘打字的声音来判断用户密码. Harnik 等人[11] 在基于跨用户的重删研究中提出，攻击者可以使用客户端和云服务器的重删通信通道作为一个侧面通道，学习有关其他用户的文件内容. 具体来说，侧信道会导致 2 种攻击威胁：

1)证明文件存在(existence of file, EOF). 攻击者可以通过上传某一文件的指纹，以确认云存储中是否存在该文件. 如图 6 所示，在基于客户端的重删模型中，攻击者上传块的指纹，如果云服务器响应的是“1”，则表明云中没有该文件，需要用户全部上传. 如果云服务器响应的是“0”，攻击者便可以得知文件的存在性信息.

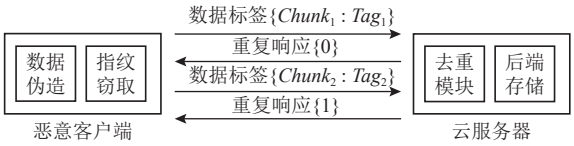


Fig. 6 Attack of EOF

图 6 证明文件存在攻击

2)学习文件内容(learning file content, LFC). LFC 是一种基于暴力推断云中文件存在性信息的攻击方法. 该攻击方法假设能够事先获得其他用户文件的大部分内容，并且剩余的敏感内容的消息空间是可预测的. 攻击者通过构建所有可能的伪造文件，并逐一上传至云端，通过观察云端对伪造文件的响应信号或网络中的流量变化作为侧信道，推断目标文件是否存在. 如果某一伪造文件收到云端的信号是不需要上传，或此时网络中文件上传的流量大小为 0,

则目标文件就被推测成功. 如图 6 所示, 攻击者通过枚举文件的所有可能内容上传至云端验证, 当云端响应已存在信号时, 表明攻击者推断成功.

3)非法访问(illegal access, IA). 除了隐蔽的侧通道攻击造成的信息泄露, 客户端重删还容易受到非法访问攻击.

Halevi 等人[12]指出在将文件上传到云存储之前, 客户端重删模式允许用户通过文件的指纹便可以获得云服务器的授权访问, 这种访问机制会导致云中的数据遭受严重的攻击. 首先, 如图 7 所示, 攻击者通过文件的指纹获得云服务器的授权后, 可以下载或删除该文件以破坏数据的隐私性和完整性. 其次, 在一些公有云服务中, 文件以加密形式外包存储, 攻击者利用公有云无法验证密文和标签的一致性这一缺点, 在通过文件指纹获得云服务器的授权后, 上传与正确文件不一致的伪造文件, 这就导致正确的文件被覆盖. 后续其他用户在下载该文件时, 只能获得攻击者的伪造文件. 除此之外, 在多用户的云存储场景中, 一个文件常常被许多用户共享, 而当某一用户删除该文件后, 存储系统应当撤销该用户对文件的授权访问, 避免隐私泄露, 即云存储还应采用灵活的访问控制机制来应对非法用户的未授权访问.

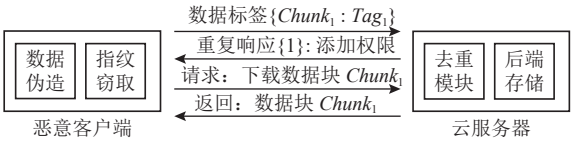


Fig. 7 Attack of illegal access

图 7 非法访问攻击

Mulazzani 等人[46]构造了一个 Dropship 的模拟攻击案例, 该案例利用 Dropbox 的跨用户重删实现非法访问攻击. 在该案例中, 用户文件的哈希值通过社交媒体或论坛等渠道被恶意泄露, 而 Dropbox 允许客户端仅凭单个指纹值便可访问云服务器中对应的文件.

4.2 防御方案

4.2.1 基于响应改变的防御方

Harnik 等人[11]提出了一种服务器端随机阈值对策 RTS, 其具体方案是云服务器为每个上传的文件设定一个阈值T_F. 阈值参数从范围 [2,d] 中随机选择, 其中d是一个系统参数. 当用户上传一个新文件时, 云服务器检查该文件的累计上传数C是否超过阈值T_F. 如果$d \leqslant T_F$, 客户端需要上传整个文件, 将在服务器端执行重删; 否则, 该文件不必上传, 只需要在客户端执行重删. Harnik 的对策可以看作是客户端重删和服务器端重删的折中方案, 在某个文件上传达到设定阈值前, 执行消耗高带宽的服务器端重删以换取安全性. 达到阈值后, 执行客户端重删.

Chia-mu 等人[47]提出了一种服务端随机响应的防御方案 RARE. 在该方案中, 云服务器不再对每个块单独返回直接的存在性响应, 而是以 2 个块组成的 1 对块序作为一个数据对象进行响应. 当 2 个块都是非重复时, 则要求客户端全部上传. 当 2 个块中有 1 个或 2 个都是重复块, 则要么上传全部块, 要么上传 2 个块的异或值. 通过该响应策略, 云服务端可以完整恢复需要存储的非重复块, 并且混淆块的实际重删状态.

Ha 等人[48]认为 RARE 方案依旧存在防御漏洞, 攻击者可以使用未存储的伪造块和存储的已知块来学习文件的存在性信息. 针对这种威胁, Ha 等人[48]的方案通过在异或操作之后上传客户端所有已经执行异或操作的块, 以防止攻击者进一步了解块的存在状态.

Vestergaard 等人[49]提出了一种改进的客户端重删响应编码 CIDER, 将 RARE 的原理推广到同时上传 2 个以上的数据块, 响应需要返回块的线性组合, 而不是异或值. 接收到请求后, 云服务器可以借助存储中的重复块恢复请求中的非重复块.Tang 等人[50]提出了一种广义的重删框架. 具体来说, 对每个文件分成“基”和“偏差”2 个部分, 对提取基的部分实现客户端重删, 对偏差支持实现服务端重删. 通过细分客户端与云端重删策略, 在此框架的帮助下, “0”和“1”的存在信号与重删响应之间的联系被打破, 保护了敏感数据的隐私.

上述随机化的防御方案的优势在于, 通过部署一些轻量级的响应设计, 可以降低隐私泄露的风险. 然而这些方案存在以下缺陷: 1)通过结合客户端重删和云端重删, 增加了网络带宽和通信开销, 造成网络拥塞, 达到设定阈值后, 依旧存在隐私泄露风险; 2)RARE 及其针对响应改变的方案依旧存在安全漏洞, 攻击者可以在达到阈值或构造不同分布的块来发起攻击; 3)在多用户的云存储场景下为不同流行度的文件选择合适的阈值是难以评估的, 并且当用户数和文件的上传量增大时, 相关数据结构会造成巨大的存储开销.

Li 等人[51]从攻击检测的角度提出一种防御方案 FeatureSpy, 基于 LFC 攻击场景假设, FeatureSpy 采用保留相似性的加密方式, 并非对整个明文块采用特征加密, 而是在块内部选取极小一部分字节级数据作为特征指示符进行特征加密, 其余大部分数据用 MLE 密钥进行加密. 因此, 基于相似的块具有相似的

特征指示符，相似的特征指示符也具有相似的特征密文，如果短时间内检测到大部分块具有相同的特征指示符，则向服务器报告攻击. 不同于随机化防御方案，FeatureSpy 提出基于特征加密的方式以检测 LFC 攻击，通过模拟多种工作负载和伪造文件，该方案实现了较高的检测率. 然而，FeatureSpy 对于攻击场景的假设存在较大的局限性，攻击者可以通过分批次地上传伪造文件，破坏假设场景中的集中式分布，进而继续发动 LFC 攻击.

4.2.2 基于流量冗余的防御方案

在客户端重删通信过程中，攻击者可以根据网络流量的侧通道进行 LRI 攻击，以感知重复数据删除是否发生，而不探测网络中传输的数据本身. 为了破坏端到端的响应隐私泄露和流量分析，Shin 等人[52]提出了一种基于差分隐私[53]的网关重删方案，其核心思想是在网关与云端的通信中添加虚拟的噪声数据，以混淆网络中的实际流量状态，防止攻击者通过分析流量发起侧信道攻击.

Zuo 等人[54]则提出了一种为每个上传文件添加随机重复块的方案 RRCS，通过改变网络中的流量大小来防止攻击者进行流量分析攻击. 在他们的方案中，一个文件被分成大小相等的块，其中一个应该包含具有最低熵的敏感信息. 为了混淆攻击者，无论敏感块是否重复，都需要数目无穷的随机选择块作为响应. 但是，由于云服务商（cloud service provider，CSP）不知道其位置，因此并不总是保证涉及敏感块. 一旦不需要，存在隐私就会立即泄露.

Tang 等人[55]针对 RRCS 方案提出了改进的附加块攻击防御策略，它是一种基于标记策略的跨用户客户端重删框架，该框架允许云服务器在返回响应之前将随机选择的一定数量的数据块标记为非重复，以抵御随机块生成攻击和侧信道攻击.

表 4 从算法总结、计算开销、通信开销、安全等级总结了抗侧信道攻击的服务器端防御方案. 由表 4 可知，基于响应变换的方案具备更高的安全性，但代价是消耗较高的计算和通信开销. 基于流量混淆的方案通常具有更低的通信开销，其缺陷是依旧具有响应信息泄露的风险.

4.2.3 基于边缘计算的防御方案

随着雾计算和边缘计算的兴起，Koo 等人[56]提出了一种基于云边端协同的存储模型的 2 阶段重删方案用于抵抗侧信道攻击. 与雾计算类似，边缘计算是一种分布式计算模型，通过对分布在数据源和终端设备的边缘网络节点上提供计算、存储和网络服务. 雾计算是云计算的一种扩展，目标是解决云计算中的延迟、带宽限制和数据隐私等问题. 如图 8 所示，第 1 阶段为用户到雾节点的服务器端重删，用户无法得知数据的重删状态. 第 2 阶段为分布式雾节点到云的客户端重删，防止网络核心出现拥塞，此阶段用户无法实现侧信道监听.

Table 4 Comparison of Defense Algorithms Against Side-Channel Attack

表 4 抗侧信道攻击的防御算法对比

方案	算法总结	计算开销	通信开销	安全等级
文献 [11]	阈值和随机响应	低	中	低
文献 [47]	块内容异或	高	中	高
文献 [48]	重复上传	中	高	高
文献 [49]	组合线性变换	中	高	高
文献 [50]	广义重删	高	中	高
文献 [51]	特征加密和检测	高	低	中
文献 [54]	添加冗余块噪声	低	低	中
文献 [55]	随机重复上传	中	高	高

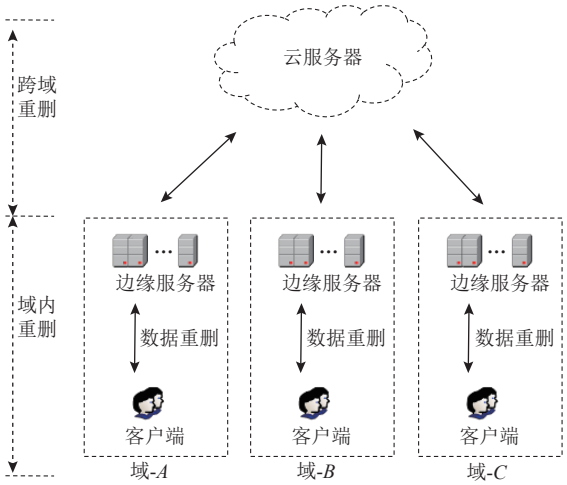


Fig. 8 Two-stage cloud deduplication architecture based on cloud and fog storage

图 8 基于云雾存储的 2 阶段云重删架构

云雾协同的 2 阶段重删模式与基于网关的重删模式都通过增加“中间件”来改变用户到云的端到端重删模式，从而防止恶意客户端发起侧信道攻击. 雾节点扩展了局域网网关的计算和存储能力，并且具备较强的灵活性，可以协同中心云以提高整体系统的性能和效率.

Shin 等人[57]提出了基于跨域的分布式密钥服务器方案，通过在边缘内和边缘间执行 2 阶段的重复数据删除. 此外，该方案还部署了基于双线性映射的密钥变换和共享机制. Xue 等人[58]在此基础上提出了

一种有效的安全跨域去重方案 EPCDD.该方案利用双线性映射实现指纹识别,并基于二叉搜索树进行存储和搜索文件标签.除了双线性映射,基于二叉树的索引结构具有较大的构造和查找开销.

与基于收敛加密的分布式密钥服务器方案不同的是,Song 等人[59]提出了一种云雾协同的 2 级重删方案 FCDedup,在该方案中拥有相同数据的用户可以在不同雾节点间共享加密密钥,无需经过第三方密钥服务器.同时,FCDedup 还实现了安全的访问控制和身份认证机制.但是该方案依旧采用双线性映射,具有高昂的计算开销.与 FCDedup 方案类似,Shin 等人[60]实现了基于边缘节点的密钥口令加密方案,同一数据的上传者和拥有者在边缘服务器间通过运行 PAKE 密钥口令协议和轻量级的加性同态加密以实现密钥共享.然而这 2 种方案都需要进行频繁的通信,降低了密文的上传和恢复效率.

Fu 等人[61]面向医疗数据管理提出了一种基于雾到多云和应用感知的安全重删系统 SafePHR.通过引入雾到多云协同存储模型,可以将低延迟、高安全的本地雾与无限容量云相结合,并在远程多云中内置灾难恢复功能,增强医疗数据的隐私保护.利用多云存储的灵活性和可靠性,不同应用类型的数据会被保存在不同的云服务中,保证可靠性的同时还防止攻击者从雾到云的网络侧信道中完整地学习隐私信息.SafePHR 还在多个云供应商之间部署容错编码方案,进一步增强了云存储的弹性和安全性.表 5 从是否引入第三方、主要算法、计算开销总结了 4 种基于密码学的云边协同的安全重删方案.

Table 5 Comparison of Edge-Secure Duplicate Data Deletion Schemes

表 5 边缘安全重复数据删除方案对比

方案	第三方	主要算法	计算开销
文献 [57]	√	双线性映射	$O(m)(Hash+3Pair+AES)$
文献 [58]	√	双线性映射 二叉搜索树	$O(m)(Hash+2Pair+AES)+$ $O(\log k)(Hash+Pair)$
文献 [59]	×	双线性映射	$O(m)(2Hash+8Pair+AES)$
文献 [60]	×	密钥口令 同态加密	$O(m)(AES+Hash)+O(u)(PAKE)$

注:*AES* 表示执行 1 次对称加密的时间开销;*Hash* 表示执行哈希函数的开销;*PAKE* 表示执行 PAKE 密钥协议的开销;*Pair* 表示执行 1 次双线性映射的开销;m、u、k 分别表示数据大小、数据拥有者的数量和系统参数.“√”表示引入第三方;“×”表示未引入第三方.

4.2.4 基于所有权证明的防御方案

为应对客户端重删模式导致的非法访问攻击,Halevi 等人[12]引入了所有权证明(proof of ownership, PoW)方案.PoW 方案是一种挑战-响应协议,主要包含客户端上传文件、服务器发起挑战、客户端响应、服务器验证 4 个阶段.首先,客户端向云中上传文件指纹,云服务器进行重复检测.如果是重复指纹向云服务器发起挑战,要求客户端证明其对文件的拥有权.其次,客户端根据文件内容计算对应的验证数组,并发送回云服务器.最后,云服务器对收到的数组信息进行验证.

Halevi 等人[12]实现了一种基于 Merkle 树[62]所有权证明方案 MHT-PoW.首先,MHT-PoW 将文件编码为固定大小的缓冲区并进行分块,每个块对应 Merkle 树的叶节点.在 MHT-PoW 的验证过程中,服务器随机选择叶节点的块索引作为挑战信息.客户端返回从该叶节点到根节点的路径信息,服务器最终验证根节点数值的正确性.Shai 等人的方案实现了对客户端重删文件的完整性验证过程.然而,该方案的文件编码和 MHT 构造会带来巨大的计算和 I/O 开销.Blasco 等人[63]则提出了一种基于布隆过滤器(Bloom Filter)的所有权证明方案 BF-PoW.在该方案中,服务器计算每个块对应的哈希令牌并插入布隆过滤器中.在挑战阶段,服务器要求客户端上传一定数量的哈希令牌来证明所有权.与 MHT-PoW 方案相比,BF-PoW 利用了布隆过滤器的查找特性,极大提高了对伪造用户的验证速度.然而,当用户数较少时,BF-PoW 方案会有较高的假阳性率.

然而,在实际的所有权验证场景中,数据权限被撤销的用户可以存储其先前访问的有效标签证明.为了抵抗这种潜在的重放攻击,Yuan 等人[64]实现了一种所有权动态管理和用户可扩展的重复数据删除方案 DedupDUM.该方案主要采用密钥更新技术实现用户权限的动态管理,并设计预先验证的访问控制机制,以防止未经授权的云用户下载数据.Jiang 等人[65]在该方案的基础上实现了基于布隆过滤器的 BF-DedupDUM 方案,并设计了一种惰性更新策略,保证数据机密性的同时降低更新频率,减少计算开销.Tian 等人[66]则提出了一种基于区块链的去中心化安全重复数据删除和共享审计方案,该方案利用基于双存储服务器架构实现双向共享审计机制,以防止用户遭受单点故障和重放攻击.此外,该方案还基于 BLS 签名实现一种轻量级认证生成算法和更新协议来减少元数据冗余.

Tian 等人[67]提出了一种基于公有云和私有云相结合的所有权证明方案,并采用代理重加密的方案实现数据所有权的动态更新,提升了系统的整体性能.为了实现更高的安全保证,Ma 等人[68]提出了一种随

机客户端重复数据删除方案，通过在块之间实现随机重复数据删除来防止外部攻击者的合谋攻击和蛮力攻击，并为每个数据生成2个标签以抵御重复攻击. 该方案通过牺牲存储效率来抵御侧信道攻击.

表6从标签一致性、动态所有权、数据共享和数据审计4个方面对比了5种所有权综合证明方案. 这些方案采用密钥更新、私有云或区块链审计机制实现用户数据的安全访问，可以抵抗不同类型的攻击，达到了更高的安全性. 然而，大规模密钥更新或双向审计会产生大量的计算开销.

Table 6 Comparison of PoW Schemes

表6 所有权证明方案对比

方案	标签一致性	动态所有权	数据共享	数据审计
文献[64]	√	√	×	×
文献[65]	√	√	×	×
文献[66]	√	×	√	√
文献[67]	√	√	√	×
文献[68]	√	√	×	√

注："√"表示引入第三方；"×"表示未引入第三方.

5 总结与未来研究展望

随着云存储系统中重删技术的广泛应用，其所面临的攻击威胁越来越大，云存储中安全重删的需求也越来越高. 本文从攻击与防御角度深入分析和总结了当前云中数据安全重删技术的攻击种类和防御方案. 通过分析，可以发现现有方案还存在3个问题：1) 基于内容加密的MLE加密原语不仅使得密文块容易遭受传统的暴力破解，一些古典的密码统计分析方法也被应用于推断MLE加密的密文，如频率攻击. 将密码分析与重删中数据块的分布特性相结合还会大大提升攻击的成功率，这表明MLE加密原语的缺陷会在未来的密码分析与加密重删相结合的研究中被进一步扩大. 2)现有针对客户端重删侧信道攻击的防御方案尚不成熟. 诸如Harnik等人[11]的随机化防御策略和RARE[47]等方案难以适应公有云的多用户场景，并且具有较高的存储空间和通信开销. 如何平衡系统安全性与网络带宽开销是应对安全重删中侧信道攻击的一个研究重点. 3)现有的所有权证明方法虽然能抵抗侧信道非法访问攻击和重放攻击，但仍存在较高的计算和空间开销，且无法达到用户和数据增长下的动态高扩展性. 如何实现安全性和性能开销的权衡还需深入研究.

随着云外包数据安全多元化的发展，许多新兴的存储与安全技术常常与重删相结合. 通过本文的分析，对未来云安全重删研究方向的展望如下：

1)传统的云存储重删架构只包含用户与云服务2个实体，而随着雾计算、边缘计算、多云等新型云计算架构的发展，许多研究者们提出基于多云融合[22,69]、云边协同[53−54,56,60]的数据安全重删技术. 多云存储环境利用多个云计算提供商的存储资源来构建高可用性和可扩展性分布式存储解决方案的系统，需要解决如何提高重删系统可扩展性、协调多云中不同加密方式和编码方式、跨云重删以及抵御不同攻击类型等挑战. 雾计算和边缘计算通过在边缘设备、边缘服务器和数据中心之间建立一个分层的计算和存储体系结构，实现更低的延迟和更高的带宽利用率，其难点在于建立复杂的网络架构和管理机制的同时保障数据的机密性、一致性和可靠性.

2)随着可信执行环境的发展，研究者们提出了许多基于英特尔可信硬件SGX的云安全重删方案[70−71]. 可信执行环境主要通过采用受保护的加密内存空间、特殊的处理器指令集等相关设计防止恶意软件和攻击者对计算系统进行未授权访问和篡改. 例如，在SGXDedup[25]中，客户端能够通过密码认证检查特定的程序是否在安全容器Enclave内执行，这种屏蔽机制使客户端与云建立端到端加密通道，从而保障用户数据加密过程的安全性. 比起传统的第三方密钥服务器辅助加密模式，基于硬件的保护机制不仅能提供更高的机密性，避免密钥服务器的单点故障或暴露给不可信的云服务商，同时大大减少计算和通信开销. Ren等人[26]已经将基于SGX的安全重删研究应用于代替传统的MLE加密以及抵抗侧信道攻击等相关领域. 然而，将可信执行环境与云存储重复数据删除安全相结合也面临2个关键挑战. 可信执行环境的安全区大小通常有限，其空间远小于重复数据删除中的指纹索引大小，而在安全区内过高的空间会产生显著的分页开销[72]，从而严重影响重删系统的性能. 其次，安全区内的数据与外部无保护的内存互相隔离，只通过特定的系统调用函数实现数据通信，而频繁地进出安全区会产生较高的CPU上下文切换开销，进而降低系统效率. 如何设计在有限的安全区大小内设计高性能的重删系统架构，降低性能损失会是未来安全重删研究的重点方向.

作者贡献声明：吴健提出了论文整体架构和思路，并负责论文撰写和修订；付印金审阅论文的整

体架构和思路，并提出指导意见；方艳梅负责论文部分内容的撰写和修订；刘垚负责论文图表制作、参考文献的整理，以及论文整体的审阅和修订；付伟负责论文数据分析总结；操晓春和肖侬负责论文审阅和修订.

参 考 文 献

[1] Statista. Data created [EB/OL]. [2024-11-21]. https://www.statista.com/statistics/871513/worldwide-data-created/

[2] Meyer D T, Bolosky W J. A study of practical deduplication[C]//Proc of the 9th USENIX Conf on File and Storage Technologies. Berkeley, CA: USENIX Association, 2011: 229−241

[3] Wallace G, Douglis F, Qian H, et al. Characteristics of backup workloads in production systems[C/OL]//Proc of the 10th USENIX Conf on File and Storage Technologies. Berkeley, CA: USENIX Association, 2012 [2025-01-14]. https://www.usenix.org/system/files/conference/fast12/wallace2-9-12.pdf

[4] Fu Yinjin, Xiao Nong, Liu Fang. Research and development on key techniques of data deduplication[J]. Journal of Computer Research and Development, 2012, 49(1): 12−20 (in Chinese)
(付印金, 肖侬, 刘芳. 重复数据删除关键技术研究进展[J]. 计算机研究与发展, 2012, 49(1): 12−20)

[5] Zhu B, Li Kai, Patterson R H. Avoiding the disk bottleneck in the data domain deduplication file system[C]//Proc of the 6th USENIX Conf on File and Storage Technologies. Berkeley, CA: USENIX Association, 2008: 269−282

[6] Heen O, Neumann C, Montalvo L, et al. Improving the resistance to side-channel attacks on cloud storage services[C/OL]//Proc of the 5th IEEE Int Conf on New Technologies, Mobility and Security. Piscataway, NJ: IEEE, 2012 [2025-01-14]. https://ieeexplore.ieee.org/docu ment/6208705

[7] Xie Qingyuan, Zhang Chen, Jia Xiaohua. Security-aware and efficient data deduplication for edge-assisted cloud storage systems[J]. IEEE Transactions on Services Computing, 2023, 16(3): 2191−2202

[8] Douceur J R, Adya A, Bolosky W J, et al. Reclaiming space from duplicate files in a serverless distributed file system[C]//Proc of the 22nd IEEE Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 2002: 617−624

[9] Bellare M, Keelveedhi S, Ristenpart T. Message-locked encryption and secure deduplication[C]//Proc of the 32nd Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2013: 296−312

[10] Li Jingwei, Lee P P C, Tan Chufeng, et al. Information leakage in encrypted deduplication via frequency analysis[J]. ACM Transactions on Storage, 2020, 16(1): 1−30

[11] Harnik D, Pinkas B, Shulman-Peleg A. Side channels in cloud services: Deduplication in cloud storage[J]. IEEE Security & Privacy, 2010, 8(6): 40−47

[12] Halevi S, Harnik D, Pinkas B, et al. Proofs of ownership in remote storage systems[C]//Proc of the 18th ACM Conf on Computer and Communications Security. New York: ACM, 2011: 491−500

[13] Bellare M, Keelveedhi S, Ristenpart T. DupLESS: Server-aided encryption for deduplicated storage[C]//Proc of the 22nd USENIX Conf on Security. Berkeley, CA: USENIX Association, 2013: 179−194

[14] Chien H Y, Jan J K, Tseng Y M. RSA-based partially blind signature with low computation[C]//Proc of the 8th Int Conf on Parallel and Distributed Systems. Piscataway, NJ: IEEE, 2001: 385−389

[15] Chen Rongmao, Mu Yi, Yang Guoming, et al. BL-MLE: Block-level message-locked encryption for secure large file deduplication[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(12): 2643−2652

[16] Zhao Yongjun, Chow S S M. Updatable block-level message-locked encryption[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(4): 1620−1631

[17] Ha Guanxiong, Jia Chunfu, Chen Yuchen, et al. A secure client-side deduplication scheme based on updatable server-aided encryption[J]. IEEE Transactions on Cloud Computing, 2023, 11(4): 3672−3684

[18] Jiang Tao, Yuan Xu, Yuan Chen, et al. FuzzyDedup: Secure fuzzy deduplication for cloud storage[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(3): 2466−2483

[19] Song Mingyang, Hua Zhongyun. Zheng Yifeng, et al. SimLESS: A secure deduplication system over similar data in cloud media sharing[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 4700−4715

[20] Li Jin, Chen Xiaofeng, Li Mingqiang, et al. Secure deduplication with efficient and reliable convergent key management[J]. IEEE Transactions on Paralled Distributed Systems, 2014, 25(6): 1615−1625

[21] Iwamoto M, Yamamoto H. Strongly secure ramp secret sharing schemes[C]// Proc of the 2nd Int Symp on Information Theory. Piscataway, NJ: IEEE, 2005: 1221−1225

[22] Li Mingqiang, Qin Chuan, Li Jingwei, et al. CDStore: Toward reliable, secure, and cost-efficient cloud storage via convergent dispersal[J]. IEEE Internet Computing, 2016, 20(3): 45−53

[23] Resch J, Plank J. AONT-RS: Blending security and performance indispersed storage systems[C/OL]//Proc of the 9th USENIX Conf on File and Storage Technologies. Berkeley, CA: USENIX Association, 2011 [2025-01-14]. https://www.usenix.org/legacy/event/fast11/tech/full_papers/Resch.pdf

[24] Gao Wenjing, Xian Hequn, Cheng Runhui. A cloud data deduplication method based on double-layered encryption and key sharing[J]. Chinese Journal of Computers, 2021, 44(11): 2203−2215 (in Chinese)
(高文静, 咸鹤群, 程润辉. 基于双层加密和密钥共享的云数据去重方法[J]. 计算机学报, 2021, 44(11): 2203−2215)

[25] Cao Tianjie, Lin Dongdai, Xue Rui. A randomized RSA-based partially blind signature scheme for electronic cash[J]. Computers & Security, 2005, 24(1): 44−49

[26] Ren Yanjing, Li Jingwei, Yang Zuoru, et al. Accelerating encrypted deduplication via SGX[C]//Proc of the 29th USENIX Annual

Technical Conf. Berkeley, CA: USENIX Association, 2021: 957−971

[27] Intel. Intel SGX [EB/OL]. [2025-01-14]. https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/software-guard-extensions.html

[28] Yang Zuoru, Li Jingwei, Lee P P C. Secure and lightweight deduplicated storage via shielded deduplication-before-encryption[C]//Proc of the 30th USENIX Annual Technical Conf. Berkeley, CA: USENIX Association, 2022: 37−52

[29] Liu Jian, Asokan N, Pinkas B. Secure deduplication of encrypted data without additional independent servers[C]//Proc of the 22nd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2015: 874−885

[30] Bellovin S M, Merritt M. Encrypted key exchange: Password-based protocols secure against dictionary attacks[C]//Proc of the 13th IEEE Computer Society Symp on Research in Security and Privacy. Piscataway, NJ: IEEE, 1992: 72−84

[31] Ha Guanxiong, Jia Qiaowen, Chen Hang, et al. Data popularity-based encrypted deduplication scheme without third-party servers[J]. Journal on Communications, 2022, 43(8): 17−29 (in Chinese)
(哈冠雄, 贾巧雯, 陈杭, 等. 无第三方服务器的基于数据流行度的加密去重方案[J]. 通信学报, 2022, 43(8): 17−29)

[32] Cormode G, Muthukrishnan S. An improved data stream summary: The Count-Min sketch and its applications[J]. Journal of Algorithms, 2005, 55(1): 58−75

[33] Merkle R C. Secure communications over insecure channels[J]. Communications of the ACM, 1978, 21(4): 294−299

[34] Mcquoid I, Rosulek M, Roy L. Minimal symmetric PAKE and 1-out-of-*N* OT from programmable-once public functions[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 425−442

[35] Xian Hequn, Liu Hongyan, Zhang Shuguang, et al. Verifiable secure data deduplication method in cloud storage[J]. Journal of Software, 2020, 31(2): 455−470 (in Chinese)
(咸鹤群, 刘红燕, 张曙光, 等. 可验证的云存储安全数据删重方法[J]. 软件学报, 2020, 31(2): 455−470)

[36] Du Xinjun, Wang Ying, Ge Jianhua, et al. An ID based broadcast encryption scheme for key distribution[J]. IEEE Transactions on Broadcasting. 2005, 51(2): 264−266

[37] Wikipedia. Frequency analysis [EB/OL]. [2024-04-08]. https://en.wiki pedia.org/wiki/Frequency_analysis

[38] Li Jingwei, Wei Guoli, Liang Jiacheng, et al. Revisiting frequency analysis against encrypted deduplication via statistical distribution[C]//Proc of the 41st IEEE Conf on Computer Communications. Piscataway, NJ: IEEE, 2022: 290−299

[39] Stanek J, Sorniotti A, Androulaki E, et al. A secure data deduplication scheme for cloud storage[C]//Proc of the 18th International Conf on Financial Cryptography and Data Security. Berlin: Springer, 2014: 99−118

[40] Stanek J, Kencl L. Enhanced secure thresholded data deduplication scheme for cloud storage[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(4): 694−707

[41] Yang Zuoru, Li Jingwei, Ren Yanjin, et al. Tunable encrypted deduplication with attack-resilient key management[J]. ACM Transactions on Storage, 2022, 18(4): 1−38

[42] Xie Qingyuan, Zhang Chen, Jia Xiaohua. Security-aware and efficient data deduplication for edge-Assisted cloud storage systems[J]. IEEE Transactions on Services Computing, 2023(16): 2191−2202

[43] Miranda M, Esteves T, Portela B, et al. S2Dedup: SGX-enabled secure deduplication[C/OL]//Proc of the 14th ACM Int Conf on Systems and Storage. New York: ACM, 2021 [2025-01-14]. https://doi.org/10.1145/3456727.3463773

[44] Luo Shengmei, Zhang Guangyan, Wu Chengwen, et al. Boafft: Distributed deduplication for big data storage in the cloud[J]. IEEE Transactions on Cloud Computing, 2015, 8(4): 1199−1211

[45] Chevallier-Mames B, Ciet M, Joye M. Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity[J]. IEEE Transactions on Computers, 2004, 53(6): 760−768

[46] Mulazzani M, Schrittwieser S, Leithner M, et al. Dark clouds on the horizon: Using cloud storage as attack vector and online slack space[C]//Proc of the 20th USENIX Security Symp. Berkeley, CA: USENIX Association, 2011: 5−16

[47] Chia-mu Y, Gochhayat S P, Conti M, et al. Privacy aware data deduplication for side channel in cloud storage[J]. IEEE Transactions on Cloud Computing, 2020, 8(2): 597−609

[48] Ha Guanxiong, Chen Hang, Jia Chunfu, et al. Threat model and defense scheme for side-channel attacks in client-side deduplication[J]. Tsinghua Science and Technology, 2022, 28(1): 1−12

[49] Vestergaard R, Zhang Qi, Lucani D E, et al. CIDER: A low overhead approach to privacy aware client-side deduplication[C/OL]//Proc of the 20th IEEE Global Communications Conf. Piscataway, NJ: IEEE, 2021 [2025-01-14]. https://ieeexplore.ieee.org/document/9348272

[50] Tang Xin, Liu Zhi, Shao Yan, et al. Side channel attack resistant cross-user generalized deduplication for cloud storage[C]//Proc of the 28th IEEE Int Conf on Communications. Piscataway, NJ: IEEE, 2022: 998−1003

[51] Li Jingwei, Ren Yanjing, Lee P P C, et al. FeatureSpy: Detecting learning-content attacks via feature inspection in secure deduplicated storage [C/OL]//Proc of the 42nd IEEE Int Conf on Computer Communications. Piscataway, NJ: IEEE, 2023 [2025-01-14]. https://ieeexplore.ieee.org/ document/10228971

[52] Shin Y, Kim K. Differentially private client-side data deduplication protocol for cloud storage services[J]. Security and Communication Networks, 2015, 8(12): 2114−2123

[53] Dwork C, Lei Jing. Differential privacy and robust statistics[C]//Proc of the 41st Annual ACM Symp on Theory of Computing. New York: ACM, 2009: 371−380

[54] Zuo Pengfei, Hua Yu, Wang Cong, et al. Mitigating traffic-based side channel attacks in bandwidth-efficient cloud storage[C]//Proc of the 32nd IEEE Int Parallel and Distributed Processing Symp. Piscataway, NJ: IEEE, 2018: 1153−1162

[55] Tang Xin, Chen Xiong, Zhou Ran, et al. Marking based obfuscation strategy to resist side channel attack in cross-user deduplication for cloud storage[C]//Proc of the 21st IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Piscataway, NJ: IEEE, 2022: 547−555

[56] Koo D, Shin Y, Yun J, et al. A hybrid deduplication for secure and efficient data outsourcing in fog computing[C]//Proc of the 8th IEEE Int Conf on Cloud Computing Technology and Science. Piscataway, NJ: IEEE, 2016: 285−293

[57] Shin Y, Koo D, Yun J, et al. Decentralized server-aided encryption for secure deduplication in cloud storage[J]. IEEE Transactions on Services Computing, 2020, 13(6): 1021−1033

[58] Xue Yang, Lu Rongxing, Choo K K R, et al. Achieving efficient and privacy-preserving cross-domain big data deduplication in cloud[J]. IEEE Transactions on Big Data, 2022, 8(1): 73−84.

[59] Song Mingyang, Hua Zhongyun, Zheng Yifeng, et al. FCDedup: A two-level deduplication system for encrypted data in fog computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(10): 2642−2656

[60] Shin H, Koo D, Hur J. Secure and efficient hybrid data deduplication in edge computing[J]. ACM Transactions on Internet Technology, 2022, 22(3): 1−25

[61] Fu Yinjin, Xiao Nong, Chen Tao, et al. Fog-to-multicloud cooperative ehealth data management with application-aware secure deduplication[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(5): 3136−3148

[62] Merkle R C. Protocols for public key cryptosystems[C]//Proc of the 1st IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 1980: 122−134

[63] Blasco J, Di P R, Orfila A, et al. A tunable proof of ownership scheme for deduplication using Bloom filters[C]//Proc of the 2nd IEEE Conf on Communications and Network Security. Piscataway, NJ: IEEE, 2014: 481−489

[64] Yuan Haoran, Chen Xiaofeng, Jiang Tao, et al. DedupDUM: Secure and scalable data deduplication with dynamic user management[J]. Information Sciences, 2018, 456: 159−173

[65] Jiang Shurong, Jiang Tao, Wang Liangmin. Secure and efficient cloud data deduplication with ownership management[J]. IEEE Transactions on Services Computing, 2020, 13(6): 1152−1165

[66] Tian Guohua, Hu Yuhan, Wei Jianghong, et al. Blockchain-based secure deduplication and shared auditing in decentralized storage[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(6): 3941−3954

[67] Tian Guohua, Ma Hua, Xie Ying, et al. Randomized deduplication with ownership anagement and data sharing in cloud storage[J]. Journal of Information Security and Applications, 2020, 51: 2214−2126

[68] Ma Xuewei, Yang Wenyuan, Zhu Yuesheng, et al. A secure and efficient data deduplication scheme with dynamic ownership management in cloud computing[C]//Proc of the 23rd IEEE Int Performance, Computing, and Communications Conf. Piscataway, NJ: IEEE, 2022: 194−201

[69] Zhang Di, Le Junqing, Mu Nankun, et al. Secure and efficient data deduplication in jointcloud storage[J]. IEEE Transactions on Cloud Computing, 2023, 11(1): 156−167

[70] Cui Helei, Duan Huayi, Qin Zhan, et al. SPEED: Accelerating enclave applications via secure deduplication[C]//Proc of the 39th IEEE Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 2019: 1072−1082

[71] Dang H, Chang E C. Privacy-preserving data deduplication on trusted processors[C]//Proc of the 10th IEEE Int Conf on Cloud Computing. Piscataway, NJ: IEEE, 2017: 66−73

[72] Li Mingyu, Xia Yubin, Chen Haibo. Memory optimization system for SGXv2 trusted execution environment[J]. Journal of Software, 2022, 33(6): 2012−2029 (in Chinese)
(李明煜, 夏虞斌, 陈海波. 面向 SGX2 代新型可信执行环境的内存优化系统[J]. 软件学报, 2022, 33(6): 2012−2029)



Wu Jian, born in 2000. Master. His main research interests include data deduplication and cloud security.

吴　健, 2000 年生. 硕士. 主要研究方向为数据重删、云安全.



Fu Yinjin, born in 1984. PhD, associate professor, master supervisor. Distinguished member of CCF. His main research interests include data reduction, cloud storage, big data security.

付印金, 1984 年生. 博士, 副教授, 硕士生导师. CCF 杰出会员. 主要研究方向为数据缩减、云存储、大数据安全.



Fang Yanmei, born in 1966. PhD, associate professor. Senior member of CCF. Her main research interests include machine learning & deep learning, multimedia information security, and artificial intelligence security.

方艳梅, 1966 年生. 博士, 副教授. CCF 高级会员. 主要研究方向为机器学习与深度学习、多媒体信息安全、人工智能安全.



Liu Yao, born in 1987. PhD, assistant professor. Member of CCF. His main research interest includes digital integrated circuit design, hardware security, and computer security.

刘　垚, 1987 年生. 博士, 助理教授. CCF 会员. 主要研究方向为数字集成电路设计、硬件安全、计算机安全.



Fu Wei, born in 1978. PhD, associate professor. Member of CCF. His main research interests include information security, network security, and cloud storage security.

付　伟, 1978 年生. 博士, 副教授. CCF 会员. 主要研究方向为信息安全、网络安全、云存储安全.



Cao Xiaochun, born in 1980. PhD, professor, PhD supervisor. Distinguished member of CCF. His main research interests include content security in cyberspace, artificial intelligence, and computer vision.

操晓春, 1980 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为网络空间内容安全、人工智能、计算机视觉.



Xiao Nong, born in 1969. PhD, professor, PhD supervisor. Fellow of CCF. His main research interests include computer architecture, network computing, and big data storage.

肖　侬, 1969 年生. 博士, 教授, 博士生导师. CCF 会士. 主要研究方向为计算机体系结构、网络计算、大数据存储.