|  |  |
| --- | --- |
| 学号 | 2017202110100 |
| 密级 |  |

**海量信息存储课程论文**

|  |
| --- |
| **云存储系统安全重复数据删除** |

|  |  |
| --- | --- |
| 院（系）名 称 ： | 计算机学院 |
| 专 业 名 称 ： | 信息安全 |
| 学 生 姓 名 ： | 黎佳玥 |
| 课 程 教 师 ： | 何水兵 教授 |

二〇一七年十二月

摘 要

重复数据删除已经吸引了许多云服务提供商（CSP）作为降低存储成本的一种方式。尽管一般的重复数据删除方法已经被越来越多的人接受，但是由于云存储的外包数据传输模式，它带来了许多安全和隐私问题。为了处理特定的安全和隐私问题，已经针对云数据提出了安全的重复数据删除技术，从而形成了多种解决方案和权衡。因此，在本文中，我们将讨论云数据安全重复数据删除技术的研究，以考虑在云存储领域最广泛使用的攻击场景。

在对重复数据删除系统进行分类的基础上，探索内外攻击者的安全风险和攻击场景。然后，我们描述了针对特定或组合威胁模型（包括密码和协议解决方案）下的不同安全问题的每种方法的最新安全重复数据删除技术。我们针对不同的安全目标对安全性和效率进行讨论和比较。最后，我们确定并讨论云存储中的安全重复数据删除未解决的问题和进一步的研究挑战。

关键词：重复数据删除，云计算，依赖于消息的加密，所有权证明，流量混淆，确定性信息分散

目 录

[1 绪论 1](#_Toc500165884)

[1.1 范围和贡献 2](#_Toc500165885)

[1.2 相关工作 2](#_Toc500165886)

[1.3 组织 3](#_Toc500165887)

[2 重复设计标准 4](#_Toc500165888)

[2.1 数据粒度 4](#_Toc500165889)

[2.2 重复数据删除位置 4](#_Toc500165890)

[2.3 重复检查边界 5](#_Toc500165891)

[2.4 系统架构 5](#_Toc500165892)

[3 使用重复数据删除云安全风险 6](#_Toc500165893)

[3.1 内部攻击者造成的威胁 6](#_Toc500165894)

[3.1.1数据泄露 6](#_Toc500165895)

[3.1.2数据丢失 7](#_Toc500165896)

[3.2 外部攻击者造成的威胁 7](#_Toc500165897)

[3.2.1信息泄漏通过旁边渠道 7](#_Toc500165898)

[3.2.2未经授权的数据访问 8](#_Toc500165899)

[3.2.3滥用云存储服务 8](#_Toc500165900)

[3.2.4数据中毒 9](#_Toc500165901)

[4 安全重复计划的分类和评估标准 10](#_Toc500165902)

[4.1 分类 10](#_Toc500165903)

[4.2 评价标准 11](#_Toc500165904)

[5 开放的安全问题和挑战 13](#_Toc500165905)

[5.1 数据所有权管理 13](#_Toc500165906)

[5.2 实现语义上的安全重复数据消除 13](#_Toc500165907)

[5.3 分散重复数据删除系统中的PoW 14](#_Toc500165908)

[5.4 重复数据删除对安全构成新的威胁 14](#_Toc500165909)

[6 （即时）安全性重用方案的实用性 16](#_Toc500165910)

[6.1 理论与实践之间的巨大差距 17](#_Toc500165911)

[6.2 缺乏足够数量的软件实现 18](#_Toc500165912)

[7 结论 20](#_Toc500165913)

[参考文献 21](#_Toc500165914)

# 绪论

云计算模式由于提高了可靠性，可扩展性，无处不在的网络访问，快速配置和按需安全控制，所以成本微不足道，大大改变了中小型企业和个人数据的数据管理。

思科全球云指数（2012-2017）[1]显示，到2017年，全球数据中心流量预计将增长三倍，达到每年7.7兆字节。到2019年，全球数据中心流量将达到每年10.4兆字节，所有数据中心流量的83％将来自云，五分之四的数据中心工作量将在云中进行处理。随着云大数据的空前增长，云存储中存在相同数据的多个副本，因为大量的用户独立地存储相同的文档，媒体或软件包。

重复数据删除是指消除冗余文件或更多细粒数据块的技术。重复数据删除标识常见的数据块或文件，只存储一个实例。通过这样做，重复数据删除可以大幅减少存储大型数据集所需的云空间。最近的研究表明，跨用户（或跨用户）重复数据删除技术可以将标准文件系统的存储成本降低50％以上，备份应用程序的存储成本降低90％到95％[2]。

但是，随着数据安全性正在成为云计算服务的最重要的要求之一，针对不同云应用场景中的特定安全性目标，安全重复数据删除的需求大大增加。然而，传统的密码方法，包括加密和重复数据删除，在很大程度上是相互对立的。重复数据删除利用数据相似性，而密码学的目标是语义安全性，使得密文与随机数据不可区分。在保护数据安全的同时，从重复数据删除中受益仍然是云存储系统中需要解决的关键和挑战性问题。因此，安全的重复数据删除的目的是为了既提供空间效率又提供数据安全性来对抗内部和外部的攻击者。

为此，在数据粒度，重复数据删除位置，重复检查边界和系统架构等多方面的设计决策方面，针对特定安全性和效率目标，安全重复数据删除技术已经在内部或外部攻击者的面前进行了深入研究。根据与关键设计决策相对应的这四个标准，现有的安全重复数据删除方案可以被分类为基于密码或基于安全协议的方法，包括基于消息的加密，所有权证明（PoW），流量混淆以及确定性信息分散他们的安全目标。

依赖于消息的加密主要关心云存储中外包数据的机密性。为了实现这个安全目标，它使独立的数据所有者能够从外包的数据中生成加密密钥。它允许云存储识别由不同用户加密的密文副本。PoW是数据上传器（或用户）和云存储服务器之间执行的质询-响应协议。它允许云服务器验证想要外包数据的用户是否确实拥有数据而不需要上传数据。流量模糊处理解决了客户端重复数据删除中的安全问题。在重复数据删除过程中，如果攻击者能够观察到网络上的流量模式，基于流量混淆的解决方案就会通过混淆网络流量的模式来削弱他们的能力，这使得他们无法从观察到的网络流量中获取任何信息。确定性信息分散方法解决了多重体系结构中服务器端重复数据删除系统中外包数据的机密性和可用性问题。将数据分散到多个云存储服务器上，通过调整秘密共享算法以及信息分散算法（IDA），提高了外包数据的可靠性。

## 范围和贡献

本论文对安全重复数据删除系统进行了广泛的介绍，并解释了其设计决策和主要挑战，主要关注云存储中的安全重复数据删除。

本次调查的第一个贡献是确定所有安全重复数据删除系统共有的四个关键设计决策：数据粒度，重复数据删除位置，存储架构和云服务模型。对于其中的每一个，我们扩展现有的分类，定义来自内部和外部攻击者的安全威胁，并描述为解决安全重复数据删除的主要挑战而采取的不同方法。作为第二个贡献，我们将用于安全重复数据删除的现有最先进解决方案分为四种不同的方法：基于消息的加密，PoW，流量混淆以及基于安全目标的确定性信息分散。然后，我们从安全性和效率的角度对这些问题进行广泛的分析，并从理论和实践的角度讨论各自的优缺点。最后，我们还深入了解未解决的开放安全问题以及安全重复数据删除的实际问题方案。

## 相关工作

一般重复数据删除技术已经有许多广泛的调查。 Mandagere等人[3]首先探讨了不同条件下各重复数据删除系统的有效性和效率的权衡。在他们的工作中，通过描述可用重复数据删除技术的分类，使用真实世界的备份数据集进行实验评估。

根据Mandagere等人[3]建立的分类标准， Paulo和Pereria [6]根据六个关键设计决策进一步对重复数据删除系统进行了分类：数据粒度，位置，时间，索引，算法和范围。然后，在设计决策的组合中，已经针对诸如归档/备份存储和主存储（例如，HDD，SSD和RAM）的不同存储环境中的性能和有效性分析了每个去重技术。

通过Mandagere等人[3]的扩展和Paulo和Pereira [6]，Fu等人[51]根据以下功能和操作方面提出了更多关于重复数据删除技术的细粒度分类：关键值，指纹预取，分段，采样，重写和还原。他们还提出了评估重复数据删除和探索备份性能和存储成本之间的权衡的通用框架。

作为另一项工作，Meyer和Bolosky [2]研究了使用近1000个Windows文件系统的重复数据删除对系统的实际影响。他们比较了全文件重复数据删除和块级别重复数据删除之间的存储成本效率。

以前的大部分调查工作都集中在重复数据删除方案的效率和有效性分析上，而没有任何安全考虑。与以往的调查不同，本文主要关注于确定数据机密性，完整性和可用性方面的各种安全威胁，以及对云存储中重复数据删除系统的显着攻击。本文还介绍了对现有的安全重复数据删除技术进行分类并分析它们以减轻这些攻击。这项工作的贡献可以从它的原创性中找到，它不同于以前的调查工作，这些调查工作的重点是在没有任何安全考虑的情况下探索一般重复数据删除技术的有效性和性能权衡。

## 组织

文章的其余部分安排如下。第2节提供了重复数据删除系统概述，包括数据粒度，位置，云系统体系结构和云服务模型等方面。第3节描述了内外攻击者对云存储的安全威胁模型。第4节定义了衡量安全重复数据删除技术的安全性和效率的标准。第5节和第6节分别讨论了目前的研究重点，分别讨论了针对安全重复数据删除的未解决安全问题及其实际限制。第7部分提出了关于调查的最后评论。

# 重复设计标准

在云存储系统中，重复数据删除通常由以下实体完成：作为数据所有者的客户端（或用户）和为外包数据提供存储空间的云服务提供商（CSP）。云存储的重复数据删除方案可能因安全目标和设计标准而异。在本节中，我们根据Mandagere等人[3] 和Paulo和Pereira [6]建立的标准对这些方案进行分类：数据粒度，重复数据删除位置，重复检查边界和系统架构。

## 数据粒度

有多种方法可以将数据划分为基本单位以消除重复。在文件级重复数据删除（这是最直接的方法之一）中，文件被视为重复数据删除的基本单位。哈希签名是从整个文件的内容计算出来的，然后通过比较哈希签名来查找云中的重复文件。块级重复数据删除是另一种常见的方法，其中将文件分成多个块，并为每个块检查重复项。有两种不同的方法将文件分成块：固定大小的分块和可变大小的分块。在固定大小的分块中，每个文件被分成固定大小的块，并且基于文件的重复数据删除中使用的哈希算法被应用于每个块。另一方面，可变大小的组块使用Rabin指纹[7 8]作为生成哈希签名的算法。 Rabin指纹算法将文件视为字节流，并在字节流中查找预先指定的分隔符。然后将由分隔符分隔的每个数据块视为重复数据删除的基本单位。

块级重复数据删除通常具有比文件级重复数据删除更高的重复数据删除率，但由于每个块生成大量元数据，因此会产生相当大的开销。相比之下，文件级重复数据删除在维护元数据方面的开销相对较低[2 10]。

## 重复数据删除位置

可以在服务器端，客户端和网关侧执行重复数据删除。

在服务器端重复数据删除中，希望将数据外包的客户端总是通过网络将其上传到云存储服务器。一旦CSP收到数据，它就会通过在存储的数据中找到重复项并删除它们来执行重复数据删除。

在客户端重复数据删除方法中，客户端首先计算要上传的数据的哈希值，并在上传数据的实际内容之前将其发送给CSP。然后，CSP检查存储器中是否存在相同的散列值。如果找到相同的哈希值，则服务器取消实际上传并将客户端标记为现有数据的所有者。否则，客户端继续将数据上传到服务器。与服务器端重复数据删除相比，这种方法在减少网络带宽消耗方面具有优势[11]。

在一些云计算环境中，特别是在混合云计算模型中，通常将某种类型的设备服务器（称为存储网关）部署在客户专用网络上。存储网关为客户提供对公共云存储服务的访问。网关侧重复数据删除利用存储网关执行重复数据删除。在接收到来自客户端的外包数据后，存储网关代表客户端与CSP一起执行重复数据删除。

## 重复检查边界

基于重复的检查边界，重复数据删除技术可以分为内部用户和用户间的方法。在上传数据中发现重复数据时，内部重复数据删除仅考虑已由相同数据所有者外包的云存储中的数据。另一方面，互联用户重复数据删除会考虑云存储中多个数据所有者之间的所有存储数据。

内部用户重复数据删除技术和内部用户重复数据删除技术有效地消除了云存储中的冗余数由于内部用户重复数据删除查找并删除了相同用户数据的重复副本，因此该技术对于备份系统更为有用，因为备份系统可能在单个用户进行的重复备份中找到重复项[12]。对于不同用户所拥有的数据（例如业务文件，虚拟机映像[13]和工作站文件系统快照[2]）之间存在大量重复数据的组织，存储系统内部的重复数据删除是有效的。

## 系统架构

安全重复数据删除方案可以建立在两种不同的云系统架构上：1）单个云架构，其中单个CSP构建并提供存储服务；以及（2）多云架构，也称为云云架构云存储服务构建在多个CSP上，用户数据分散在多个云存储中。

对于许多商业CSP（如Dropbox，Google Drive和Mozy），单一云架构是流行的方法，然而存在潜在的问题，例如由于系统故障和供应商锁定造成的数据丢失[4]。Multicloud架构试图避免数据复制的单点故障[4]，并分成多股[9 14]，然后将这些份额分散到多个独立的云存储中。多RAID架构的重复数据删除技术，如RAIDer [14]和CDStore[9]处理这些分散的股份，这些分散的股份通常由诸如Reed-Solomon代码的删除码处理，作为检查重复的基本单位。

# 使用重复数据删除云安全风险

CSP拥有云基础架构，包括存储磁盘和提供数据外包服务的网络，而外包数据由数据所有者持有。数据和基础设施之间的所有权分离导致数据拥有者（即客户端）在外包数据之后失去对其数据的实际控制，从而产生关于外包数据的安全性和隐私性的安全考虑。特别是采用重复数据删除技术的云存储系统面临着更为严重的各种安全风险，因为重复数据删除技术本身就成为攻击者攻击的手段。在云计算的一般威胁模型中，考虑了两种类型的攻击者：

内部攻击者：这是CSP内的攻击者。他们可能是CSP的恶意雇员，具有存储系统，网络或其他基础架构的管理权限，甚至是云服务提供者本身。外部黑客妥协并控制云存储系统也被认为是内部攻击者。这些内部攻击者有一个存储在云存储中的不感兴趣的数据，除了数据拥有者之外，其他访问都被禁止。要了解数据中的信息，他们可以发起复杂的攻击，包括利用重复数据删除作为杠杆作用。另外，为了拒绝服务（DoS）或其他目的，内部攻击者也可能有兴趣打破存储器中数据的完整性。

外部攻击者：这是CSP之外的攻击者。他们可能是恶意用户，他们无权访问其他用户拥有的外包数据，但有兴趣获取数据中的敏感信息。我们也认为任何黑客都可以攻击合法用户并伪装成他们作为外部攻击者发动攻击。与内部攻击者一样，外部攻击者的主要目标是从云存储中的外包数据中获取有用的信息。为了实现这种攻击，他们可能会利用重复数据删除协议。他们也对DoS感兴趣，通过破坏数据以及滥用云存储服务来实现其他恶意目的。

在下一节中，我们将介绍使用重复数据删除技术的云存储系统内部和外部攻击者所造成的各种安全风险。

## 内部攻击者造成的威胁

云存储中的外包数据暴露于内部攻击者的威胁之下，他们将利用其控制云服务平台和后端基础架构来访问数据的能力。

3.1.1数据泄露

CSP通常不愿意将用户提供的密钥的加密算法应用于存储的数据，因为即使对于相同的数据，重复数据删除也是不可能的。换句话说，使用不同的密钥，常规的加密算法（例如AES）将产生完全不同的密文用于相同的消息，这使得如果难以根据密文来检查这些明文是否相同。因此，为了实现重复数据删除和降低云存储成本，外包数据的处理不需要应用加密。这可能最终导致好奇的内部攻击者的数据泄露，这些攻击者可以访问云存储中的普通数据。为解决有关机密性和存储效率的冲突问题，大多数云存储服务（如Dropbox或AWS S3）都会根据自己选择的加密密钥对数据进行加密，而不是由数据所有者进行加密。这种方法可以对加密数据进行重复数据删除。然而，在我们的威胁模型中，由于恶意CSP能够访问云存储中的明文数据并从中获取隐私信息，因此它仍然被认为易受内部攻击。

3.1.2数据丢失

一旦数据上传到云存储，数据拥有者将失去对数据的控制权。外包数据的可靠性取决于云基础架构对系统故障的容忍程度。然而，尽管CSP为提供无缝和无故障的服务付出了巨大的努力，但云服务中断偶尔会发生（例如Dropbox在2013年中断了10个小时）是不可避免的。这样的中断可能会对存储系统造成严重的破坏，最终使得受损的数据无法恢复。由于重复数据删除的性质，数据重复数据删除进一步增加了对数据可靠性的担忧，而无论上载次数如何，只保留一份数据副本。

除了存储系统的故障之外，数据丢失可能是由内部攻击者故意造成的，特别是由恶意的CSP造成的。换句话说，为了在不增加任何成本的情况下保持足够的磁盘空间，它们可以非法地擦除存储器中未被客户端最近访问的一部分数据，而不通知客户端删除。其他内部攻击者，如黑客，其目的是为云存储服务引发DoS，也可能会损坏云存储中的数据。

## 外部攻击者造成的威胁

外部攻击者能够利用重复数据删除技术来发起对云存储服务的攻击。在本节中，我们将通过重复数据删除技术介绍外部攻击者对云存储造成的各种安全威胁。

3.2.1信息泄漏通过旁边渠道

如最近的研究[15 16 17]，客户端的跨用户重复数据删除协议可以被利用作为一个旁道，恶意用户（或客户端）等外部攻击者可以从其他用户的敏感数据学习信息。侧通道是通过客户端重复数据删除的两个固有属性建立的：（1）通过网络的数据传输对于攻击者是可见的，并且（2）用于上传数据（例如，数据的散列）的指纹可以用于确定云服务器上的副本的存在。通过利用旁道，攻击者可以合理安装以下攻击：

识别具体数据的存在：假设攻击者想要确认存储器中存在感兴趣的数据，比如文件F。攻击者可以通过上传F并监视其网络活动来轻松确认F的存在。如果F的全部内容通过网络上传，那么攻击者就会知道F在云存储中不存在。否则，可以假定F已经被上传到存储器。

学习数据内容：急于学习有趣文件F的攻击者可以利用旁道进行在线猜测攻击。使用一个包含F的多个候选字典，攻击者重复运行一个CSP的重复数据消除协议，用于每个猜测的F’直到他或她观察到重复数据删除事件。

3.2.2未经授权的数据访问

除了通过侧通道泄漏信息之外，客户端重复数据删除还容易受到另一种攻击，即允许未经授权的外部攻击者访问其他用户在云存储中的敏感数据[18]。在将数据上传到云存储之前，客户端重复数据删除系统要求用户将数据的小指纹发送到云服务器，这通常是数据的散列值。使用此指纹作为搜索关键字，CSP将尝试使用相同的指纹查找存储的数据。如果找到匹配的数据，则跳过数据的上传，并且CSP将数据的所有权授予上传者（用户）。一个新的安全威胁源于这样一个事实，即CSP通过使用通常不被视为秘密的散列值来验证外包数据的所有权，特别是在数据可预测的情况下。因此，即使没有实际的数据，恶意用户也可以用猜测的散列值来证明CSP的所有权。外部攻击者可能利用此漏洞来获取他们不应被授权访问的有趣数据的所有权。通过提供散列，攻击者可以说服CSP他或她实际拥有数据，从而能够从服务器上下载。

Mulazzani等人[16]在Dropbox上发现了上述漏洞，Dropbox是最大的商业CSP之一。为了在Dropbox中演示这种攻击，他们修改了一个Dropbox客户端程序的加密模块，以便代替正常的散列计算，为Dropbox服务器提供一个有趣文件的特定散列值，客户端无权访问该文件。通过运行这个修改后的客户端程序，他们发现Dropbox没有检测到哈希值的操纵，甚至可以随意访问客户端存储的任何文件。

3.2.3滥用云存储服务

通过上述客户端重复数据删除中的安全漏洞，外部攻击者甚至可以将云存储服务用于其他恶意目的。以下情况说明了他们如何滥用云存储服务：

建立一个秘密通道：假设两个攻击者A和B试图使用（客户端）重复数据删除所产生的辅助通道保密地相互通信。他们同意使用两个不同的随机文件F0和F1将一位数据传送给另一方。为了发送“1”，攻击者A会将F1上传到云存储。经过一段时间后，B上传F1并通过网络观察重复数据删除事件。如果发生了F 1的重复数据删除，则B最终得知“1”是从A发送的。同样，A也能够使用另一个文件F0向B发送“0”。

用作内容交付网络：利用客户端重复数据消除的云存储服务可能被滥用为内容交付网络（CDN）。假设攻击者A有一个大的文件F，可能包含侵犯版权的内容（例如盗版电影），他或她希望将F传递给另一个攻击者B。而不是使用合法的CDN来传递F，A会下载F到CSP。一旦F的上传完成，A就将相应的F的散列值通过任意频道发送给B。然后，B仅通过向CSP提供F的散列就可以很容易地获得对F的所有权并下载F的全部内容。如Halevi等人[18]所述，A和B的行为与云存储的商业模式冲突服务，主要是为了支持大量的上传（如数据备份）而不是下载。

3.2.4数据中毒

在一般重复数据删除过程中，初始上传者（即首次上传数据的客户端）应该将数据和相应的标签发送给CSP。通过使用标签作为搜索关键字，CSP能够检查这个上传数据的重复。但是，如果数据是以加密形式外包的，则CSP无法验证加密数据和相应标签之间的一致性。这个问题使得重复数据删除系统容易受到数据中毒的攻击。例如，假设外部攻击者将原始数据文件F更改为F'，然后上传F'和与F相对应的标签。然后，由后续合法数据所有者外包的相同数据文件F将被CSP重复删除。之后，每当他们想要从CSP下载F时，他们将得到恶意修改的数据文件F'而不是原来的F。

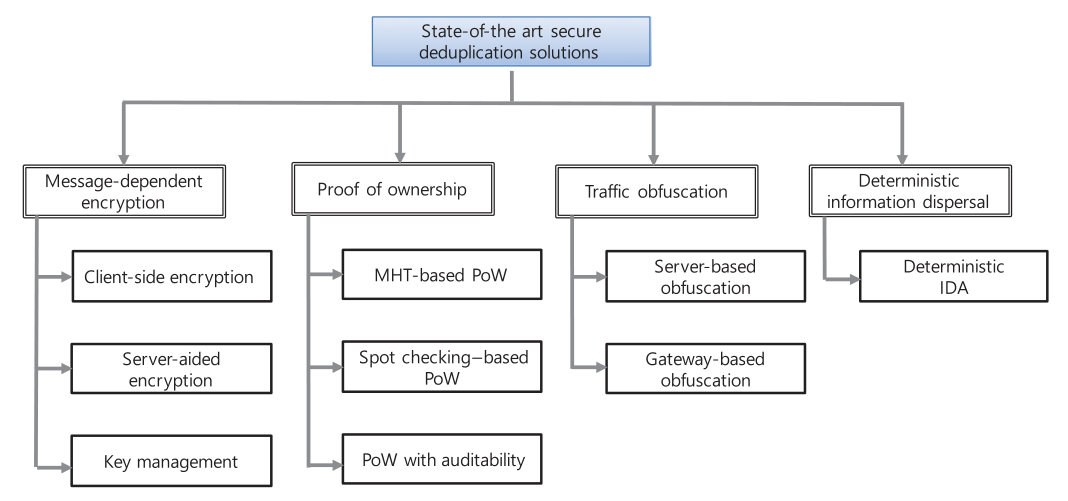
# 安全重复计划的分类和评估标准

安全的重复数据删除方案的目标是减轻上述安全威胁，从而提供一种解决方案，允许云存储安全地利用重复数据删除技术。如第2节所述，重复数据删除系统的多样性以及来自内部/外部攻击者对云存储的许多可能的攻击导致了文献中各种安全的重复数据删除解决方案。基于实现安全目标的关键思想和设计决策，我们将最先进的安全重复数据删除方案分为四种不同的方法：消息相关加密，PoW，流量混淆和确定性信息分散。每种方法都覆盖了重复数据删除系统的不同领域，因此它们所处的安全威胁也因此而异。第5节将介绍安全重复数据删除方法的细节。在本节中，我们将简要介绍一下安全重复数据删除方案的分类和评估标准。

## 分类

图4.1显示了最先进的安全重复数据删除方案的分类。根据每种方法所采用的主要思想构建分类，以解决重复数据删除系统的安全问题：

图4.1 图最先进的安全重复数据删除解决方案的分类



消息依赖加密：这种方法主要关注保护外包数据的机密性，防止不可信和恶意的CSP。为了实现这个安全目标，加密技术被用于构建一个方案。然而，这种方法并没有遵循传统加密算法的随机密钥生成，而是从依赖于外包数据的信息源确定性地计算加密密钥，从而允许CSP找到由多个用户加密的重复密文。根据与消息相关密钥生成的主要设计决策，这种方法可以分为三个子类：客户端加密，服务器辅助加密和密钥管理。大部分依赖于消息的加密方案所涵盖的系统主要局限于基于单一云架构的重复数据删除系统。

所有权证明：这种方法专注于解决客户端重复数据删除系统中的各种安全问题。 PoW基本上是在外包用户和CSP之间执行的挑战-响应协议。 PoW允许CSP验证想要外包数据的用户是否拥有该数据，而无需上传数据的全部内容。配备PoW的重复数据删除系统应该是安全的，以防止内部/外部攻击，包括数据泄露，数据丢失，未经授权的访问和滥用服务。就协议中使用的数据结构及其提供的功能而言，基于PoW的解决方案可以分为三个子类：基于Merkle哈希树（MHT）的PoW，基于点检查的PoW和具有可审计性的PoW。基于PoW的方案通常考虑在客户端重复数据删除系统中提供安全性，以支持跨多个用户进行重复检查（即用户间重复的检查边界）。

流量模糊处理：这种方法解决了客户端重复数据删除中的安全问题，特别是威胁，包括滥用云存储和其他利用旁路的攻击。来自外部攻击者的安全威胁来源于客户端重复数据删除的性质，允许他们在执行重复数据删除协议期间观察网络流量模式。基于流量混淆的解决方案试图通过混淆这些模式来削弱这种能力，从而通过观察网络流量来了解任何信息是不可行的。根据运行混淆算法的位置，该方案可以分为基于服务器的混淆和基于网关的混淆。流量混淆方法通常包括具有用户重复检查边界的客户端重复数据删除系统。

确定性信息分散：这种方法解决了云存储中机密性的保存和外包数据的可用性问题，尤其是多重体系结构上的服务器端重复数据删除系统。在多个CSP上的云存储服务中，可以通过将数据分散到多个云存储服务器来提高外包数据的可用性。但是，数据分散使加密数据的重复数据删除成为一个具有挑战性的问题。确定性信息分散通过应用秘密共享技术而不是加密算法来解决这个问题。与依赖于消息的加密类似，该方法用随数据确定性计算的值替换随机元素，从而允许每个CSP查找分散数据的重复副本。

## 评价标准

安全的重复数据删除方案根据两个因素进行评估：（a）每个方案提供的安全属性；（b）CSP和客户端系统开销。关于第3节中介绍的安全威胁，需要安全的重复数据删除方案来满足以下安全属性：

（1）保密性：外包给云存储的原始数据内容不应泄漏给除拥有数据的用户以外的任何人。

（2）可用性：即使在系统发生故障的情况下，拥有数据的用户也应该始终可以使用外包的数据。

（3）侧通道阻力：在客户端重复数据删除系统中，通过观察网络上的重复数据删除流量，攻击者不可能获知有关外包数据的任何信息。

（4）所有权的真实性：在客户端重复数据删除系统中，只有拥有原始数据的有效客户端（或用户）才能被验证为云存储中外包数据的真正拥有者。

（5）数据完整性：客户或CSP应该能够验证外包数据没有被攻击者修改或删除。

应用安全重复数据删除可能会导致CSP不可避免的成本。因此，每个方案也在性能和效率方面在以下方面进行评估：（1）计算开销，（2）存储开销，（3）通信高架。

# 开放的安全问题和挑战

虽然广泛的研究已经解决了云重复数据删除系统中的安全性，隐私性和可靠性方面的许多问题，但仍然存在着有趣且关键的开放式研究挑战。为了激发研究兴趣和努力来增强安全的重复数据删除技术，本节将介绍几个未解决的问题。

## 数据所有权管理

在所有权管理和撤销的情况下，假设多个用户拥有一个外包给云存储的密文。随着时间的推移，一些用户可能会要求CSP删除或修改其数据，然后CSP从相应数据的所有权列表中删除用户的所有权信息。然后应该阻止撤销的用户在删除或修改请求后访问存储在云存储中的数据。另一方面，当用户上传云存储中已经存在的数据时，应当阻止用户访问在他或她获得所有权之前所存储的数据，并将其上传1。在实际的云系统中，这些动态的所有权变更可能会频繁发生，因此应谨慎管理，以避免云服务的安全性降级。然而，由于大多数现有的安全重复数据删除方案中的加密密钥是确定性地导出的，并且在初始密钥之后很少更新，所以先前的重复数据删除方案在具有动态所有权更改的环境中仍然无法实现安全的访问控制推导。因此，只要被吊销的用户保留加密密钥，他们就可以随时访问云存储中的相应数据，而不管其所有权是否有效。因此，数据所有权管理和撤销是安全重复数据删除的一个具有挑战性的开放性问题。

## 实现语义上的安全重复数据消除

语义安全的加密方案使关心隐私的客户能够在客户端加密数据。然而，语义加密的纯应用可能使得重复数据删除不可能，因为完全独立的密文很可能由相同的文件产生。因此，协调重复数据删除和语义安全的加密是一个具有挑战性的研究课题。 CE [19]是一个有效的解决方案，但是由于加密是确定性的（即在语义上不是安全的），所以它容易对数据散列进行离线暴力攻击，尤其是在数据可预测的情况下。实现语义上安全的重复数据删除的一个有效的解决方案是允许初始上传者使用随机生成的加密密钥对其数据进行加密，并将其仅分发给安全地共享相同数据的用户[5 20 44 23 21]。不幸的是，一些方案需要一个或多个额外的外部KS的帮助，这是一个在实际云系统中很难满足的强假设[44 23]，甚至容易受到在线暴力攻击。其他方案在没有附加KS的帮助下实现语义安全性，但是使随后的上传者能够学习相同数据的以前的上传者的身份[5 22]，这违反了用户的隐私，或要求每个以前的上传者在线[24]，这会显着降低效率和重复数据删除效率。因此，实现语义上安全的重复数据删除而不降低重复数据删除效率或借助其他KS是安全重复数据删除的关键问题。

## 分散重复数据删除系统中的PoW

重复数据删除后的外包数据很可能会因数据丢失或损坏而受到影响，因为重复数据删除技术只保留一个数据副本，而不是存储多个副本。为了解决外包数据的可靠性，分散的方法是最近的研究重点。分散式重复数据删除方案应用于多云存储架构，将多个云存储服务整合为一个存储服务。许多研究人员[52 53]已经研究了将加密数据外包给多重存储的重复数据删除技术。他们的工作主要集中在设计确定性的IDA，将数据转换成具有一定冗余度的多个确定性股份，并将这些股份分散到多个CSP中。这些方法在一定程度上保护外包数据免受数据损坏，并保持对包括CSP在内的内部攻击者的安全。但是，它们不能直接应用于多语音体系结构中的客户端重复数据删除系统。客户端重复数据消除需要PoW解决方案，以防止外部攻击者通过利用重复数据删除来引发信息泄露攻击或未经授权的数据访问。 PoW的安全性一般取决于外部数据的最小熵，这是外部攻击者所认为的。一旦min-entropy k的数据被分散到n份中，每个份额都有min-entropy ki，使得ki≤k和。对于攻击者而言，一个个打破每个份额的“分而治之”战略，要比一次打破整个原始数据容易得多。因此，在多语音结构中无法保证外包数据所需的PoW安全级别。我们需要进一步解决这个问题，通过使用分散的客户端重复数据删除系统来提高效率，同时保证安全性和可用性。

## 重复数据删除对安全构成新的威胁

我们介绍应用重复数据删除技术时可能暴露于云存储系统的新型安全威胁。首先，让我们考虑一个可以灵活地启用或禁用重复数据删除的场景。在这种灵活的重复数据删除中，云存储系统可能容易受到攻击者的信息泄露攻击。例如，假设给定存储器中的一组文件F，则在时间t1和t3处启用重复数据删除，并且在时间t2处禁用，其中t1 <t2 <t3。通过利用第3.2.1节中描述的重复数据删除的副通道，试图推断某个文件f是否在F中的攻击者可以获得关于f被包括在F（t1或t3）中的时间的进一步的敏感信息，除了文件简单存在的信息之外。

对于第二个安全威胁，让我们考虑一个支持并发读写块操作的云存储系统。如Paulo和Pereira [6]所述，将重复数据删除技术应用于读取和写入操作频繁的存储设备会导致底层存储系统性能严重下降。然后，通过利用重复数据删除的性能开销，攻击者将能够进行DoS攻击，并容易在这样的云存储系统上产生巨大的访问延迟。这些新类型的攻击在文献中还没有被考虑，因此我们认为这些问题也是重要的安全重复数据删除问题，因为这是未来有希望的工作。

# （即时）安全性重用方案的实用性

重复数据删除所面临的安全威胁是实现云系统（如云存储上的信息泄漏和数据损坏）的最重要的障碍。正如前面的部分所回顾的，各种先进的安全重复数据删除技术有希望增强针对每种不同威胁的云存储的安全性。但是，尽管安全问题取得了成功的对策，但在将这些安全解决方案广泛应用于实际应用之前还有一个问题需要克服。本节讨论安全重复数据删除方案的几个实用性问题。

为了掌握目前安全重复数据删除技术在实际应用中的使用情况，我们对Amazon [50]，Dropbox [25]，Google Drive [45]，OneDrive [46]，Bitcasa [35] Wuala [47]。除存储服务外，我们还研究了以下常见存储和重复数据删除产品：OpenDedup [26]，Q-Cloud Protect [27]，CloudBerry [28]，DupLESS [29]，CDStore [30]，Duplicati [31]和StorReduce [32]。具体而言，我们通过调查由服务提供商和产品供应商提供的公开可用白皮书和其他技术文档，调查了这些服务和产品的安全特性的技术细节。

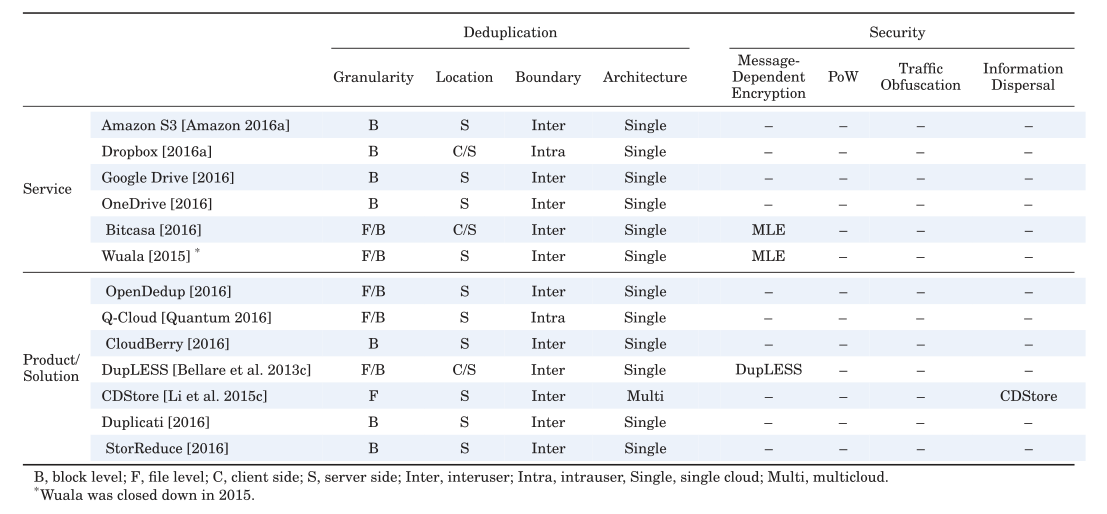
表6.1 目前安全重复数据删除方案的使用

表6.1列出了我们在各种云存储服务和产品中使用安全重复数据删除技术的调查和分析结果。每个服务和产品的安全特性都是根据我们的技术分类和第四部分提出的评估标准进行评估的。重复数据删除特性是根据以下标准进行评估的：粒度（块级别与文件级别），位置（服务器端与客户端），边界（interuser与intrauser）和架构（单个云与多个云）。结果表明，尽管重复数据删除系统利用率很高，但相关的安全特性很少部署在云存储服务和产品中。虽然某些服务涉嫌将安全技术应用于重复数据删除[33 34 26 32 27]，它们不符合我们的安全要求（在第4.2节中定义）。例如，Amazon S3提供服务​​器端加密，以实现重复数据删除和加密两个目标[33]。它只是允许存储服务器在服务器自己选择的密钥下使用传统的加密算法（例如AES-256）加密外包的数据。因此，这种方法是完全不安全的，因为服务器（即内部攻击者）仍然可以访问外包数据的内容。根据我们的调查，只有两个商业存储服务Bitcasa [35]和Wuala [47] - 使用依赖于消息的加密方案来启用对加密数据的重复数据删除。而且，对于重复数据删除产品，除了DupLESS以外，大部分产品[29]和CDStore [30]被发现不适用于任何安全的重复数据删除方案。

尽管对安全重复数据删除至关重要，但在实际应用中如此低的安全技术利用率指出，目前安全实施的必要性和有效性之间仍存在巨大的障碍。在这方面，我们观察到这个障碍有两个原因：（1）现实世界所需要的理论安全性和实际效率之间的巨大差距，以及（2）缺乏足够数量的良好实施的用于安全的软件实现重复数据删除。我们在下面的章节中提供关于这些问题的更多细节。

## 理论与实践之间的巨大差距

安全的重复数据删除方案通常伴随着在底层存储系统中失去一些效率的成本。因此，对于现实世界中的许多场景，在确定安全级别时，应该认真考虑实际效率和安全性之间的折衷。第5节介绍的各种技术在不同的威胁和云体系结构模型下实现了不同的安全目标。然而，一些技术需要巨大的计算或网络资源，最终可能使CSP和产品供应商不愿在其系统中采用安全的重复数据删除方案。

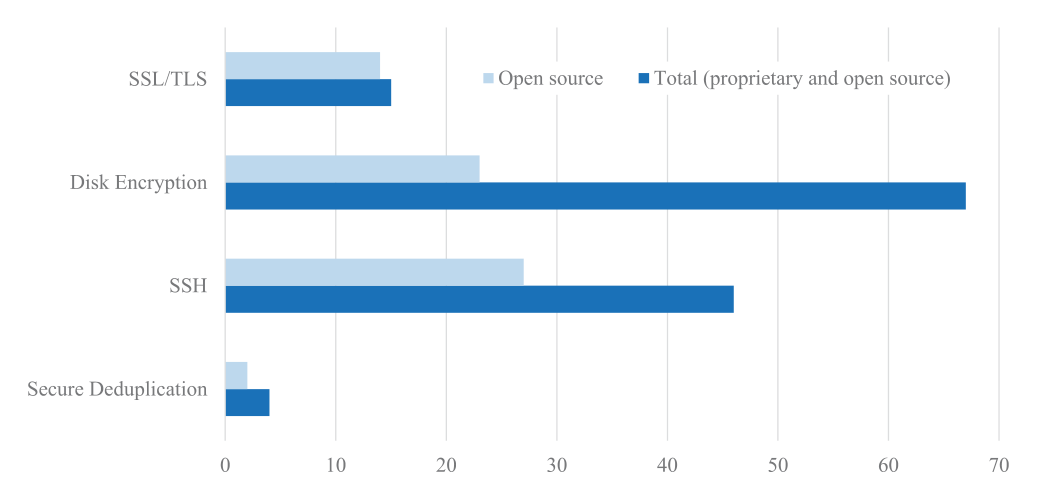
例如依赖于消息的加密是云存储的基本算法，可通过加密数据实现重复数据消除。Liu等人[37]的计划是有前途的依赖于消息的加密方案之一，因为它提供了基础云存储体系结构的灵活性，同时即使对可预测的消息分布也提供了强大的数据保密性。 Liu等人的方案的算法通过使用同态加密来实现[38 39]，这是一种最先进的密码算法，允许对密文进行算术运算。然而，尽管它的新颖性，但同态加密的实现水平还远远不能实际使用。最近的基准[40]为了在Intel i7-4600U CPU上最快地实现Paillier同态加密原语[38]，每加密记录0.5毫秒，这比使用AES-NI的英特尔i7-980X上的AES慢19万倍[41]。同态加密的计算效率低下是Liu等人在实际云存储系统中应用的最大障碍之一。

另一个例子可以在PoW方案和流量混淆方案中找到。为了防止未经授权的数据访问和客户端重复数据删除的信息泄露，这些解决方案不可避免地在云服务提供商一方产生了一定数量的网络流量。尽管这些方法保持了安全性的理论上的可靠性，但消耗相当多的带宽资源对于服务提供商来说绝不会是便宜的。大多数提供商（如亚马逊网络服务（AWS）和CloudFlare）都从多个互联网服务提供商（ISP）购买带宽，以确保与其云存储的全球连接。例如，CloudFlare以每秒每兆比特每秒的最大带宽为CloudFlare向第一层ISP支付约35美元[42]。因此，将PoW和流量混淆应用到云存储服务的成本实际上是不可忽视的，这可能会限制它们被服务提供商采用。

## 缺乏足够数量的软件实现

云存储系统构建在各种相关技术上，包括虚拟化，存储，网络和安全，其中大部分由大量复杂的软件实现组成。由于软件工程方面的进步，云服务提供商和产品供应商的开发人员不必重新创建存储服务和产品。相反，他们可能会将每种技术的软件包或库集成到系统中，方法是购买或从开源项目获取源代码。因此，使用包括商业专有软件包和开放资源在内的现有软件实施方式对于在实践中有效实现特定技术是必要的。

图6.1 每个安全解决方案的现有软件实施数量（2016年）



就安全技术而言，使用正确性和安全性证明的软件，而不是从头开始重新实现，而对复杂的密码学和安全性没有深刻的了解是至关重要的。没有它们的实现可能会导致底层软件系统的严重漏洞。例如据报道，Android应用程序的大量安全事件是由SSL / TLS等安全解决方案实施不当造成的[43]。

在这种情况下，我们指出限制安全重复数据删除方案的普遍使用的另一个障碍是由于该技术缺乏良好实施的软件。图6.1显示了表6.1中安全重复数据删除方案的软件实现数量，以“安全重复数据删除”表示。为了比较，我们还对Secure Shell（SSH）[36]，磁盘加密[48] ，以及图6.1中的SSL / TLS [49]。根据分析，我们发现目前总共只有四种安全重复数据删除技术（使用专有软件包或开源）实现。其中，两个开源实现-DupLESS [29]和CDStore [30]已被公开用于学术目的，而不是实际使用。与SSH，磁盘加密和SSL / TLS相比，实际使用率较高且软件实现充足，因此可用于安全重复数据删除的软件要少得多。为了将安全重复数据删除技术成功应用到实际应用中，为原始安全重复数据删除技术开发正确性和安全性验证软件并将其提供给云服务开发人员至关重要。

# 结论

安全的重复数据消除是一种有效的技术，可以最大限度地减少云存储空间，同时保护云数据的安全性和隐私性。在本文中，我们根据重复数据删除系统环境和考虑到各种威胁的安全性目标，引入了最先进的重复数据删除技术的分类。具体而言，我们调查了现有的安全重复数据删除方案，并将其分类为密码解决方案和协议解决方案，包括依赖于消息的加密，PoW，流量混淆以及确定性信息分散方法。然后，我们讨论了云存储安全重复数据删除的未来研究方向。就我们所知，这是对安全重复数据删除技术的首次调查和讨论。虽然重复数据删除系统及其优点现在已经被大多数当前的云服务提供商广泛接受，但是在云数据的安全性和隐私性方面存在许多实际问题。由于仍然有一些有趣而关键的开放式研究挑战，我们预测在未来几年，安全重复数据删除的研究将会继续增长。

参考文献

[1] Cisco. 2015. CISCO Global Cloud Index (2012-2017). Available at http://www.cisco.com/c/en/us/solutions/service-provider/global-cloud-index-gci/index.html.

[2] Dutch T. Meyer and William J. Bolosky. 2012. A study of practical deduplication. ACM Transactions on Storage 7, 4, 1–20.

[3] Nagapramod Mandagere, Pin Zhou, Mark A. Smith, and Sandeep Uttamchandani. 2008. Demystifying data deduplication. In Proceedings of the ACM/IFIP/USENIX Middleware 2008 Conference Companion(Companion’08). 12–17.

[4] Hussam Abu-Libdeh, Lonnie Princehouse, and Hakim Weatherspoon. 2010. RACS: A case for cloud storage diversity. In Proceedings of the 1st ACM Symposium on Cloud Computing. 229–240.

[5] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. 2013a. Message-locked encryption and secure deduplication. In Advances in Cryptology—EUROCRYPT 2013. Lecture Notes in Computer Science, Vol. 7881. Springer, 296–312.

[6] Jo˜ao Paulo and José Pereira. 2014. A survey and classification of storage deduplication systems. ACM Computing Surveys 47, 1, 1–30. DOI:http://dx.doi.org/10.1145/2611778

[7] Andrei Z. Broder. 2000. Identifying and filtering near-duplicate documents. In Combinatorial Pattern Matching. Lecture Notes in Computer Science, Vol. 1848. Springer, 1–10. DOI:http://dx.doi.org/10.1007/3-540-45123-4\_1

[8] Michael Rabin. 1981. Fingerprinting by Random Polynomials. Center for Research in Computing Technology, Aiken Computation Laboratory, Harvard University, Cambridge, MA.

[9] Mingqiang Li, Chuan Qin, and Patrick P. C. Lee. 2015b. CDStore: Toward reliable, secure, and cost-efficient cloud storage via convergent dispersal. In Proceedings of the 2015 USENIX Annual Technical Conference(ATC’15). 1–20.

[10] Danny Harnik, Oded Margalit, Dalit Naor, Dmitry Sotnikov, and Gil Vernik. 2012. Estimation of deduplication ratios in large data sets. In Proceedings of the 2012 IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST’12). 1–11.

[11] Deepak R. Bobbarjung, Suresh Jagannathan, and Cezary Dubnicki. 2006. Improving duplicate elimination in storage systems. ACM Transactions on Storage 2, 4, 424–448.

[12] Michal Kaczmarczyk, Marcin Barczynski, Wojciech Kilian, and Cezary Dubnicki. 2012. Reducing impact of data fragmentation caused by in-line deduplication. In Proceedings of the 5th Annual International Systems and Storage Conference. 15

[13] Keren Jin and Ethan L. Miller. 2009. The effectiveness of deduplication on virtual machine disk images. In Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference. 7.

[14] Chih Wei Ling and Anwitaman Datta. 2014. InterCloud RAIDer: A do-it-yourself multi-cloud private data backup system. In Distributed Computing and Networking. Lecture Notes in Computer Science, Vol. 8314. Springer, 453–468. DOI:http://dx.doi.org/10.1007/978-3-642-45249-9\_30

[15] Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg. 2010. Side channels in cloud services: Dedu plication in cloud storage. IEEE Security and Privacy Magazine 8, 6, 40–47.

[16] Martin Mulazzani, Sebastian Schrittwieser, Manuel Leithner, Markus Huber, and Edgar R. Weippl. 2011.Dark clouds on the horizon: Using cloud storage as attack vector and online slack space. In Proceedings of the 2011 USENIX Security Symposium.

[17] Tobias Pulls. 2012. (More) side channels in cloud storage. Privacy and Identity Management for Life 375, 102–115

[18] Shai Halevi, Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg. 2011. Proofs of ownership in remote storage systems. In Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS’11). 491–500.

[19] John R. Douceur, Atul Adya, William J. Bolosky, Dan Simon, and Marvin Theimer. 2002. Reclaiming space from duplicate files in a serverless distributed file system. In Proceedings of the 2002 22nd International Conference on Distributed Computing Systems. IEEE, Los Alamitos, CA, 617–624.

[20] Jia Xu, Ee-Chien Chang, and Jianying Zhou. 2013. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In Proceedings of the 8th ACM Symposium on Information, Computer, and Communications Security (ASIACCS’13). 195.

[21] J. Liu, N. Asokan, and B. Pinkas. 2015. Secure deduplication of encrypted data without additional independent servers. In Proceedings of the 22th ACM Conference on Computer and Communications Security(CCS’15). 874–885.

[22] Jia Xu, Ee-Chien Chang, and Jianying Zhou. 2013. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In Proceedings of the 8th ACM Symposium on Information, Computer, and Communications Security (ASIACCS’13). 195.

[23] Yitao Duan. 2014. Distributed key generation for encrypted deduplication. In Proceedings of the 6th Edition of the ACM Cloud Computing Security Workshop (CCSW’14). 57–68.

[24] J. Liu, N. Asokan, and B. Pinkas. 2015. Secure deduplication of encrypted data without additional independent servers. In Proceedings of the 22th ACM Conference on Computer and Communications Security(CCS’15). 874–885.

[25] Dropbox. 2016a. Home Page. Retrieved December 7, 2016, from http://www.dropbox.com

[26] OpenDedup. 2016. Home Page. Retrieved December 7, 2016, from http://www.opendedup.org

[27] Quantum. 2016. Home Page. Retrieved December 7, 2016, from http://www.quantum.com/products/ cloudservices/q-cloud/index.aspx.

[28] CloudBerry. 2016. Home Page. Retrieved December 7, 2016, from http://www.cloudberrylab.com

[29] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. 2013c. Source code of DupLESS. Retrieved December 7, 2016, from http://cseweb.ucsd.edu/∼skeelvee/dupless/.

[30] Mingqiang Li, Chuan Qin, and Patrick P. C. Lee. 2015c. Source Code of CDStore. Retrieved December 7, 2016, from https://github.com/chintran27/CDStore

[31] Duplicati. 2016. Home Page. Retrieved December 7, 2016, from http://www.duplicati.com

[32] StorReduce. 2016. Home Page. Retrieved December 7, 2016, from http://www.storreduce.com

[33] Amazon.2016b.Protecting Data Using Server-Side Encryption. RetrievedDecember 7,2016,from http://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html.

[34] Dropbox. 2016b. Dropbox Business Security: A Dropbox Whitepaper. Retrieved December 7, 2016, from https://www.dropbox.com/static/business/resources/Security\_Whitepaper.pdf.

[35] Bitcasa. 2016. Home Page. Retrieved December 7, 2016, from https://www.bitcasa.com

[36] Fingolfin. 2016. SSH Implementation Comparison. Retrieved December 7, 2016, from http://ssh- comparison.quendi.de/

[37] J. Liu, N. Asokan, and B. Pinkas. 2015. Secure deduplication of encrypted data without additional independent servers. In Proceedings of the 22th ACM Conference on Computer and Communications Security(CCS’15). 874–885

[38] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In Advances in Cryptology—EUROCRYPT 1999. Lecture Notes in Computer Science, Vol. 1592. Springer, 223–238. DOI:http://dx.doi.org/10.1007/3-540-48910-X\_16

[39] Taher Elgamal. 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31, 4, 469–472. DOI:http://dx.doi.org/10.1109/TIT.1985.1057074

[40] Christine Jost, Ha Lam, Alexander Maximov, and Ben J. M. Smeets. 2015. Encryption Performance Improvements of the Paillier Cryptosystem. Retrieved December 7, 2016, from https://eprint.iacr.org/2015/864

[41] Intel. 2010. Breakthrough AES Performance with Intel AES New Instructions. White Paper. Retrieved December 7, 2016, from http://www.intel.it/content/dam/www/public/us/en/documents/white-papers/aes-breakthrough-performance-paper.pdf.

[42] CloudFlare. 2014. The Relative Cost of Bandwidth Around the World. Retrieved December 7, 2016, from https://blog.cloudflare.com/the-relative-cost-of-bandwidth-around-the-world/

[43] Sascha Fahl, Marian Harbach, Thomas Muders, Matthew Smith, Lars Baumgärtner, and Bernd Freisleben.2012. Why Eve and Mallory love Android: An analysis of Android SSL (In)Security. In Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS’12). 50–61.

[44] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. 2013b. DupLESS: Server-aided encryption for deduplicated storage. In Proceedings of USENIX Security Symposium 2013. 179–194.

[45] Google Drive. 2016. Home Page. Retrieved December 7, 2016, from https://www.google.com/drive/

[46] OneDrive. 2016. Home Page. Retrieved December 7, 2016, from https://onedrive.live.com

[47] Wuala. 2015. Home Page. Retrieved December 7, 2016, from https://en.wikipedia.org/wiki/Wuala

[48] Wikipedia. 2016a. Comparison of Disk Encryption Software. Retrieved December 7, 2016, from https://en. wikipedia.org/wiki/Comparison\_of\_disk\_encryption\_software.

[49] Wikipedia. 2016b. Comparison of TLS Implementations. Retrieved December 7, 2016, from https://en. wikipedia.org/wiki/Comparison\_of\_TLS\_implementations.

[50] Amazon. 2016a. Amazon S3. Retrieved December 7, 2016, from https://aws.amazon.com/s3.

[51] Min Fu, Dan Feng, Yu Hua, Xubin He, Zuoning Chen, Wen Xia, Yucheng Zhang, and Yujuan Tan. 2015. Design tradeoffs for data deduplication performance in backup workloads. In Proceedings of the 13th USENIX Conference on File and Storage Technologies (FAST’15). 331–344.

[52] J. Li, X. Chen, X. Huang, S. Tang, Y. Xiang, M. Hassan, and A. Alelaiwi. 2015. Secure distributed deduplication systems with improved reliability. IEEE Transactions on Computers 64, 12, 3569–3579. DOI:http://dx.doi.org/0.1109/TC.2015.2401017

[53] J. Li, X. Chen, M. Li, P. P. C. Lee, and W. Lou. 2014a. Secure deduplication with efficient and reli-able convergent key management. IEEE Transactions on Parallel and Distributed Systems 25, 6, 1615–1625.