安全数据重复数据删除研究报告



**班级：硕士五班**

**学号：2017282110221**

**姓名：周谦**

目录

[摘要 1](#_Toc500695637)

[1. 简介 2](#_Toc500695638)

[2. 相关工作 3](#_Toc500695639)

[3. 威胁模型 5](#_Toc500695640)

[3.1假设 6](#_Toc500695641)

[3.2 组成部分 8](#_Toc500695642)

[4. 系统设计 11](#_Toc500695643)

[安全重复数据删除概述 12](#_Toc500695644)

[5. 结论 14](#_Toc500695645)

摘要**：**随着世界各地开始用数字存储来存储档案，使得能够提供对安全数据以一个具有成本效益的方式进行存储的系统的需求日益增长。通过在文件内部和文件之间识别只存储一次普通的数据块，重复数据删除可以通过增加给定数量的存储空间来节省成本。不幸的是，重复数据删除是利用相同的内容，而加密则是试图使所有内容看起来是随机的;相同的内容用两种不同的密钥加密会得到非常不同的内容。因此，将重复数据删除的空间效率与加密技术的保密方面结合起来是有问题的。

我们已经开发了一种解决方案，可以在单服务器存储和分布式存储系统中同时提供数据安全和空间效率。加密密钥由块数据以一致的方式生成;因此，相同的块将总是对相同的密文进行加密。此外，密钥不能从加密的块数据中推断出来。因为每个用户需要访问和解密组成文件的块信息使用了只有用户才知道的密钥进行加密，所以即使是系统也无法知道块是由哪一个用户使用的。

# 简介

企业和消费者越来越意识到安全、档案数据存储的价值。在商业领域，数据保存通常是由法律规定的，而且数据挖掘也已经被证明是利于商业战略的制定。对于个人来说，档案储存被用来保存情感和历史，如照片、电影和个人文件。此外，虽然很少有人会说商业数据需要安全，但隐私对个人来说同样重要;医疗记录和法律文件等数据必须保存很长时间，但不能公开。

矛盾的是，越来越多的档案数据正推动着对成本效益存储的需求; 廉价的存储允许保存所有可能最终证明有用的数据。为了达到这个目的，被称为单实例存储的重复数据删除被用作一种最大化给定存储容量的方法。重复数据删除可以识别文件内部和之间的常见字节序列，并且只存储每个块的单个实例，而不考虑它出现的次数。通过这样做，重复数据删除可以极大地减少存储大数据集所需的空间。

数据安全是现代存储系统中越来越重要的另一个领域，令人遗憾的是，重复数据删除和加密在很大程度上是截然相反的。为了减少存储空间，重数据删除复利用了数据相似性。相反，密码学的目标则是使密文与理论上的随机数据难以区分。因此一个安全的重复数据删除系统的目的是在不损害通过单实例存储技术实现的空间效率的情况下，保证数据的安全性以防来自内部和外部的损坏。

为此，我们提出了两种安全的重复数据删除的方法：身份验证和匿名。虽然这两个模型是相似的，但它们各自有着稍微不同的安全属性。两者都可以应用到单服务器存储和分布式存储。在前者，单服务器存储中，客户机与存储数据和元数据的单一文件服务器交互。在分布式存储中，元数据存储在独立的元数据服务器上，数据存储在一系列基于对象的存储设备上。

我们的安全去重复策略的两种模型都依赖于一些基本的安全技术。首先，我们利用聚合加密来启用加密，同时仍然允许在公共块上进行重复数据删除。聚合加密使用块的散列函数作为加密密钥:任何客户机对给定的块进行加密都会使用相同的密钥，因此相同的明文值将对相同的密文值进行加密，而不管谁对它们进行加密。虽然这种技术确实泄漏了特定密文的信息，而且明文已经存在，但不知道明文的敌手无法从加密块中推断出密钥。其次，所有数据块和加密都发生在客户端;纯文本数据从来没有传输过，这也就加强了系统对抗来自内部和外部的威胁。最后，将块与给定文件之间的映射用惟一键进行加密，这限制了一个密钥对于单个文件的影响。此外，密钥存储在系统中，这样用户只需要维护一个私有密钥，而不用考虑他们访问的文件的数量。

# 相关工作

当前使用单个实例存储的系统依赖于以下三种主要的重复数据删除策略之一：整个文件、固定大小的块和可变大小的块。首先，整个文件，通常使用一个文件的哈希值作为它的标识符。因此，如果两个或多个文件散列到相同的值，则假定它们具有相同的内容，并且只存储一次(这之中不包括冗余副本)。这种内容可寻址存储的形式被用于EMC中心系统。Farsite和Windows单实例存储也在每个文件的基础上执行重复数据删除操作，尽管它们都使用传统的标识符，并使用单独的数据结构来处理重复数据。第二种类型的重复数据删除，对每块进行重复数据删除，以超大的档案存储系统为例。在超大存储中，文件在重复数据删除之前被分割成固定大小的块，所以共享相同内容(但不是全部)的文件仍然可以进行存储节省。第三种，也是最灵活的形式，将文件分解成可变长度的“块”，在滑动窗口上使用散列值；通过使用诸如Rabin指纹之类的技术，可以非常有效地完成分块工作。不同长度的块就可以用于LBFS、Shark和Deep Store。

许多分布式文件系统，如OceanStore，SNAD，Plutus和电子保险库通过使用密钥加密来解决文件保密问题。在这些系统中使用密码技术的范围从所有输入数据已经被加密的假设到定义系统的中央架构元素。但是，这些系统中没有一个试图通过重复数据删除来实现可能的存储效率。Panasas并行文件系统，Ceph 和Lustre 等高性能分布式文件系统通常比“标准”分布式文件系统的安全性要低，交换性能更高，安全性更低。虽然已经尽可能的为Ceph增加更多的安全性，但这种努力只涉及认证，而不涉及加密。

与提供安全，重复数据删除存储效率的系统相反，一些系统则使用了会导致存储开销的安全模型。例如，PASIS和POTSHARDS通过秘密共享实现安全存储。虽然非常适合长期安全性，但这种技术会产生非常高的存储开销。同样，隐写文件系统，如steganographic File System和Mnemosyne ，通过使用随机数据块提供了并不那么可信的存储内容。在秘密共享和隐写中，存储开销可能是明文数据大小的许多倍。

除了数据保密之外，一些系统也满足了匿名的要求。特别是在内容分发领域，有一种系统可以隐藏数据主机、发布者和读者的身份。例如，Publius使用加密的数据和秘密共享密钥来提供一个有审查性的web发布平台，这种平台让作者有着完全的匿名性。数据加密还为存储主机提供了一定程度的可否认性; 由于没有明确的拥有者，节点的操作员可以声称他们不知道存储在他们节点上的明文数据。

在所有这些文件系统中，只有Farsite将重复数据删除与安全性结合在一起。在其最初的设计中，其目标是利用桌面级计算机网络中未使用的磁盘空间，并将其呈现为中央文件服务器。在最初的实现中，安全性是通过文件加密提供的，每个用户都使用了对称和非对称密钥的组合。这项工作的延伸是通过重复的文件合并来实现更好的空间效率。为此，作者开发了聚合加密，将数据的散列用作加密密钥。这允许用户将相同的明文加密到相同的密文。但是，与我们的工作不同的是，Farsite只在整个文件级别上合并。我们的系统在子文件级别上合并数据，从而节省空间，只用相似的文件，而不是相同的文件。另外，在Farsite设计中，客户端生成密钥及其标识符。我们发现，如果重复数据删除的键/值存储未经过验证，则其内容可能容易受到有针对性的冲突攻击。最后，我们介绍一种允许匿名用户场景中安全的重复数据删除存储的模型。

# 威胁模型

为了妥善设计和评估一个安全的存储系统，威胁模型必须明确规定。 在本节中，我们将要确定模型中存在的威胁、我们的假设以及必须在安全重复数据删除系统中考虑的攻击。

作为建立安全模型的一部分，建立一致的符号是很重要的。我们使用了两个主要的加密函数:加密和哈希。加密函数取两个参数，表示为e(K,P)= C，其中K为加密密钥，P为明文，C为密文。一个加密函数有一个对应的解密函数，它使用一个密钥和密文来恢复原始的明文，表示为d(K,C)= P。

哈希的表达方式类似于加密。没有使用加密密钥的简单哈希被表示为单参数函数，hash(P)= h，使用哈希函数和加密密钥Ki(称为HMAC(key- hash Message Authentication Code))生成纯文本P的散列，提供完整性和消息身份验证。我们用下面的符号来表示这个函数：HMACi(P)。

## 3.1假设

我们最基本的假设之一就是加密数据实际上是随机的。这意味着，在重复数据删除时，随机数据的存储收益非常低。我们通过检查一个系统的存储利用率来支撑这一说法，在这个系统中，每个用户使用各自不同的密钥对数据进行加密，并且系统将加密的数据删除。为了简便起见，我们假设加密的数据被划分成固定大小的块，但是对于可变长度的块也可以进行类似的讨论。在任何时间点上，系统存储长度l的k逻辑块，每2块可以取= m值的任意一个。我们递归地求出块中的物理存储利用率，，公式如下：

如果系统中存在单个块，则逻辑和物理块的数量是相等的，因此我们的基本情况是s1 = 1。假定在加密块上的均匀分布，第二块将以概率1 / m匹配。通常，第k块以概率sk-1 / m匹配。如果k> m，则非常接近于m，这导致大量的重复数据删除。不幸的是,当k≫m时,性能会受到影响，文件索引的大小可能会变得很难。在实践中,我们发现m≫k和重复数据删除的效用是非常小的,因为sk≈k。由此，我们得出结论，传统加密数据的重复数据删除在很大程度上是无效的。由于我们假设加密数据是随机的，我们可以使用上面的公式对随机数据的重复数据删除进行建模。对这个随机数据进行块签名操作，如果它是一个好的散列函数，它也应该是均匀分布的。对于非平凡大小的块（超过几个字节），匹配的可能性非常小，因此除极少数情况外，每个块都是唯一的，必须全部存储起来。

我们的第二个假设是，对于相对较短的归档方案，加密提供了足够的安全级别：如果数据的生命周期在几年的时间内，那么由攻击者基于现代密码系统产生的密文攻击将无法得到加密密钥或者导出相应的明文值。我们认识到，在几十年的长期情景下，这种假设可能不成立。而将这项工作扩展到安全的长期领域，则有可能成为未来的一个方向。

接下来，我们假设一个可以充分模拟用户的对手可以访问该用户的数据。换句话说，如果恶意用户获得了足够的用户用户名和密码的信息，比如参与系统的协议，那么该用户将获得该协议的标准结果。这个场景在几乎所有安全系统中都适用。

由于我们的解决方案在密钥信息的生成中使用散列函数，因此我们假定它们是密码安全的。更具体地说，我们假设它们都是能够抵抗弱和强烈的碰撞。前者指出，找到两个散列到相同输出值的输入值是棘手的问题。 后者指出，给定一个散列值，找到一个哈希到相同输出值的值是棘手的。

档案存储通常用作一次写入，可读取存储; 因此它强调吞吐量而不是低延迟的性能。大多数现有的大规模重复数据删除系统被用作档案商店。这些系统通常不经常出现，因为它们非常适合读取繁重的工作负载，如软件分发 这种使用模式与强调低延迟访问和频繁写入的分层存储解决方案的顶级存储层完全不同，而且与仅以高吞吐量写入为唯一目标的备份解决方案不同。 我们假设这种对吞吐量的强调使系统能够适应合理的延迟消耗。

我们正在考虑的数据生命周期假定为几年，而不是几十年。 虽然这比在一线存储中经常遇到的文件生命周期要长，但它并不像安全的存档系统所支持的无限生命期那样长。

## 组成部分

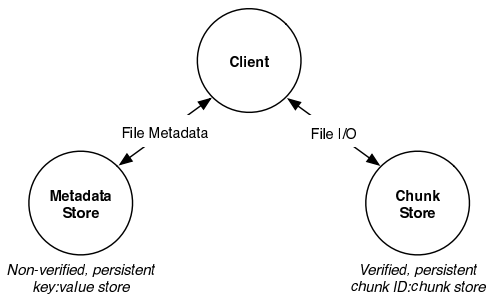
如图1所示，在协议级别，存储模型中有三个主要角色：客户端，元数据存储和块存储。这种布局映射到单服务器和分布式存储架构。在单服务器体系结构中，元数据存储区和块存储区位于同一个系统上，而分布式存储系统可能选择断开元数据存储区和块服务器的连接，从而处理各个集群中的各个任务。

图 1：存储模型中的三个主要参与者及其交互。在分布式存储系统中，元数据服务器和块服务器的职责由不同的系统集群来处理。在单服务器模型中，元数据和块服务器位于同一系统上。

用户通过客户端与系统交互，这是输入和输出的起点。如图1所示，它是存储模型中其他组件的中心联系点。与存储模型中的其他组件不同，客户端没有任何持久性存储要求，尽管系统假定用户可以安全地访问其密钥。

元数据存储负责维护用户需要的信息，以便从块中重建文件（如地图和加密密钥）。我们使用一个简单的，未经验证的key：value体系结构对这个永久性存储进行建模。在这样的系统中，当用户向元数据服务器提交键值对时，服务器不需要验证密钥是否正确对应于该值。例如，如果密钥是值的散列，则服务器不需要验证值的散列与用户提交的密钥是否相同。

第三个部分（块存储）的作用是持久地存储数据块，并根据其ID满足块的请求。块存储也被建模为key：value存储，然而，与元数据存储不同，块存储必须能够验证密钥在值方面的正确性。这是由于在块存储中可能发生下文所描述的有针对性的碰撞攻击。

在重复数据删除块存储中，可以使用有针对性的冲突攻击将假值与给定密钥相关联。随机冲突和目标冲突之间的关键区别在于，用户可以利用某些数据的可预测内容——在图2中，恶意用户利用形式字母中的相似性 - 生成有效的块标识符。如果攻击者可以首先将这些标识符提交给垃圾块，并且如果块存储不能验证标识符的正确性，那么具有相同标识符的后续提交将被重复数据删除到垃圾块。

除了存储模型的三个部分之外，我们的系统还可以识别两个敌人，通过与系统的关系来确定：外部和内部。外部攻击者存在于系统之外。这个对手没有任何内部的权限，比如用户账号，只能拦截邮件或者试图破坏用户的账号。

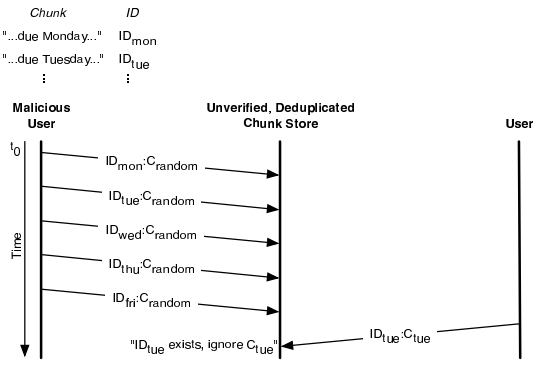
相比之下，内部攻击者或恶意的内部人员至少具有有限的内部访问权限。 这些攻击者进一步根据访问级别进行定义，从简单的用户帐户访问到特权级别的访

图 2：恶意用户利用可预测数据（在本例中为带有截止日期的表单字母）生成有效的块ID，并将这些ID与非法块关联的目标冲突攻击。 如果用户是第一个提交ID，则后续块将被重复数据删除为垃圾值。

问（可能由恶意管理员持有）。内部攻击者的存在意味着元数据存储和大块存储都不被认为是可信的。

我们提供的两种安全模式的目标是向用户提供来自外部和内部攻击者的数据保护级别，而不管攻击者的访问级别如何（或者至少减少在被攻击后丢失的数据量）。每个模型都提供了一组额外的安全特性。

# 系统设计

在本节中，我们将介绍我们的两个主要安全重复数据删除模型：已验证和已匿名。 两者虽然类似，但每个模型都提供了一个轻微的功能集。 我们从描述安全特性开始，然后介绍我们的安全重复数据删除技术的基本设计。 最后，我们介绍两个模型的细节。具体来说，我们描述每个元数据和块存储的内容，以及它们各自的摄取和提取过程。

与加密关联最密切的安全属性是保密性，即只有授权用户才能读取明文数据。 通常，授权是通过密钥分发来处理的; 如果用户能够合法地获取正确的密钥，则她可以访问和解密数据。 我们提供的两种模型都提供了针对外部和内部敌人的数据保密性。

匿名允许用户隐藏身份。这个功能有两个方面。首先是对用户提交请求的匿名性，即，读写请求不能归属于特定的用户。第二个方面是关于存储内容的匿名性，这表明系统不能确定哪个数据是由特定账户拥有或可以访问的。

撤销是能够删除用户对给定文件的访问权限。为了以细粒度完成这项工作，就像每个用户的撤销一样，系统必须包含认证; 在不知道用户身份的情况下，每个用户的撤销在系统中显然是难以实现的。撤销计划可以通过在撤销时发生的行为来描述。在主动撤销中，访问立即被删除。这通常很开销很大，并且可能涉及相当数量的密码计算。在惰性撤销中，访问只在数据更改时被删除。因此，用户无法看到撤销后发生的任何变化，但可能继续访问他们以前有权查看的内容。

## 安全重复数据删除概述

在匿名和认证模型中，客户端通过将文件转换为一组块来开始获取信息。 这通常是使用基于内容的分块过程完成的，该分块过程根据文件的内容生成块。 这种方法的优点是它可以跨文件匹配共享内容，即使该内容不存在于给定的固定偏移量的倍数处。 该算法基于阈值A和在文件上移动的宽度w的滑动窗口来选择块。 在文件的每个k位置处，计算窗口内容的指纹Fk，k + w-1。 如果Fk，k + w-1> A，则选择k作为块边界。 结果会是一组可变大小的块，其中块之间的边界基于数据的内容。

文件分块和加密都发生在客户端上。 在客户端而不是服务器上执行这些任务有很多好处。首先，它减少了服务器上的处理量。其次，通过对客户端上的块进行加密，数据永远不会以明文方式发送，从而降低许多被动的外部攻击的影响。 第三，具备特权的恶意的内部人员不能访问数据的明文，因为服务器不需要保存加密密钥。

客户端使用Farsite系统中引入的收敛加密对块进行加密。使用这种方法，客户端使用加密密钥决定从明文内容得到加密; Farsite和我们的系统都使用明文的密码散列作为密钥。由于相同的明文会导致使用相同的密钥，不管谁加密，给定的明文总是导致相同的密文。

与其他方法相比，这一策略提供了许多优点。 正如我们在第3节所示，如果每个用户使用自己的密钥进行加密，那么通过重复数据删除所节省的存储空间将大大减少，因为使用两个不同密钥加密的同一个块会导致不同的密文（具有很高的概率）。其次，尝试在多个用户帐户之间共享随机密钥引发了密钥共享问题。第三，不知道数据明文值的用户不能生成密钥，因此不能从密文中获取明文。这点尤其重要，因为与服务器对数据进行加密的方法相比，即使根级管理员没有密钥也不能访问块的明文值。

在原来的描述中指出这种方法的主要安全缺点是它泄漏了一些信息。特别地，收敛加密揭示了两个密文字符串是否解密为相同的明文值。但是，在使用重复数据删除的系统中，这种行为是必须的，因为它允许系统删除重复的明文数据块，但只能观察密文；信息泄露是通过重复数据删除实现空间效率所需妥协的一部分。

每个密文块必须被分配一个标识符。在我们的系统中，系统中的每个块都使用加密块的散列值进行标识，这种技术有时也被称为基于内容的命名。

使用加密块的散列的替代方案是使用明文块散列的散列，即，加密密钥的散列是块标识符。这种方法提供了许多有吸引力的特性。首先，性能得到改善。在这两种方法中，用户执行两个散列：密钥生成散列和标识符生成散列。 假设密钥长度小于块长度，执行两个块散列将比块散列和密钥散列花费更多。 其次，如果标识符可以从密钥中导出，那么文件到块映射只需要保存密钥，而不是密钥和标识符。然而，使用密钥的散列作为标识符存在很大的缺点：块存储不能验证块的基于内容的标识符是正确的。如3.2节所解释的，未经验证的块签名允许来自有针对性的冲突攻击的使用。

加密块本身存储在块存储中。在可能有多个块存储的分布式存储模型中，块列表还可以包括定位正确存储设备所需的信息。或者，基于块的标识符可以使用确定性放置算法定位正确的存储设备。

# 结论

我们为安全重复数据删除存储开发了两种模式：认证和匿名。这两个设计表明，安全性可以与重复数据删除相结合，从而提供多种安全特性。在我们提出的模型中，通过使用融合加密来提供安全性。这种技术首先在Farsite系统的背景下提出，提供了一种确定性的生成加密密钥的方式，使得两个不同的用户可以将数据加密到相同的密文。在经过身份验证和匿名的模型中，都会为每个文件创建一个映射，以描述如何从块重建文件。该文件本身是使用唯一密钥加密的。在已认证的模型中，通过使用非对称密钥对来管理该密钥的共享。在匿名模式中，存储是不可变的，文件共享是通过离线共享地图密钥并为每个授权用户创建地图引用来进行的。

在我们的评估中，我们分析了每个模型在安全性方面的安全性。我们发现这个系统对外部攻击者来说基本是安全的。此外，我们的模型没有明确防范的安全威胁可以通过添加标准的安全通信技术来解决，例如传输后期的安全性。恶意内部人员的安全攻击在很大程度上可以从设计避免服务器端加密中得到缓解。由于内部人员从来不会接触到纯文本或加密的密钥，所以以不可检测的方式更改元数据值的能力大大降低。在大块存储中安全性更为明显，安全块的内容处理本质上使恶意更改的检测非常明显。最后，我们检查了主要威胁导致的信息泄露，发现最严重的安全漏洞是由于客户密钥丢失造成的。然而，在这种密钥丢失的情况下的损害被限制在用户的文件中。而且，在大多数安全系统中，违反客户密钥是一个严重的威胁。