分　数:	
评卷人:	

華中科技大學

研究生（数据中心技术）课程论文（报告）

题 目：云存储的安全机制

学　　号　M202073589

姓　　名　蔡寒均

专　　业　电子信息

课程指导教师　施展、童薇

院（系、所）　计算机科学与技术学院

2020 年 12 月 20 日

云存储的安全机制

蔡寒均[1)]

(华中科技大学计算机科学与技术学院 武汉 430074)

摘 要 随着云存储的迅猛发展，越来越多的用户选择使用云存储存放自己的资料。云存储的最大特点在于存储即服务，用户可以通过公有 API 将自己的数据上传到云端保存。但由于用户丧失了对数据的绝对控制权，一些数据安全的隐患也由此产生。为了消除安全隐患，并在保证安全性的同时尽可能地提高系统的服务质量，近年来国内外机构作了大量研究，从而开启了云存储中的一个研究方向——安全云存储系统，首先介绍了云存储系统的安全需求，然后阐述了安全云存储系统的研究现状，并总结了现有安全云存储系统中的一些关键技术的现状与不足之处，其中包括可搜索加密机制、基于属性的加密机制、可追溯的加密机制、基于 AONT 加密的云存储安全机制等；最后指出了在算法上优化的思路和未来的研究方向。

关键词 云存储；安全云存储系统；可搜索加密；AONT；属性加密

Security mechanism of cloud storage

CAI Han-Jun[1)]

[1)] (School of Computer Science and Technology, School of Computer Science and Technology, Wuhan 430074)

Abstract With the rapid development of cloud storage, more and more people prefer to store their owner data in remote cloud storage to avoid troublesome data management in local storage systems. The most famous feature of cloud storage is the concept that storage as a service, users can store their own data into clouds by public APIs. However, because of losing absolute control of data, users storing their own data in cloud storage will suffer a series of security problems, such as data peeping, data tampering, and so on. In order to solute those security problems and improve the quality of secure cloud system based on enhance its security, researchers have investigated lots about cloud security problem in recent years, which established a research branch of the cloud storage- -secure cloud storage system. This paper introduces the security demand of secure cloud storage system； expounds the current status of cloud storage system； summarizes the key technologies of currently secure cloud storage systems, such as encryption key's distribution and management, attribute-based encryption, searchable encryption, ciphertext-based data deduplication, provable data possession and proof of retrievability mechanism, data assured delete, etc. At the end of paper we discuss the future research directions of secure cloud storage system.

Key words cloud storage; secure cloud storage system; searchable encryption; AONT; attribute-based encryption

1 引言

随着计算机技术和互联网应用的迅速发展，数据正以几何级数的方式增长，人们对存储空间的需求也越来越大。在这一趋势下，近年来云存储的提出与发展以及存储即服务的理念为人们提供了大量廉价的存储空间，同时也向传统的数据存储方式发起了挑战。

研发界对实现分散式云存储（DCS）服务的关注度越来越高，其特点是可以使用多个节点以分散方式存储资源，从而见证了这种格局的变化。在这种服务中，各个资源被分割成碎片，分配给不同的节点（通过复制来提供可用性保证）。访问资源需要检索其所有碎片。分布式控制系统的主要特点是由独立节点（提供多个权限存储网络）组成的协作和动态结构，这些节点可以加入服务并提供存储空间，通常是以某种回报作为交换。基于区块链的技术促进了这一演变，提供了有效的低摩擦电子支付系统，支持使用服务的报酬。提供了有效的低摩擦电子支付系统，支持使用服务的报酬。在 Storj、SAFE Network Vault、IPFS 和 Sia 等平台上，用户可以出租其未使用的存储和带宽，为网络的其他用户提供服务，这些用户使用网络加密货币支付该服务。

数据的安全性包含 CIA（confidentiality，integrity and availability）3 个方面，即机密性、完整性和可用性。机密性是指任何人或团体在非授权的情况下不得查看到数据明文；完整性是指数据在存储过程和传输过程中未被篡改，或者能够检测出此数据已被篡改；可用性是指用户可以通过云存储接口随时使用自己的数据。为了解决数据的安全问题，国内外对此作了大量的研究，分别从机密性、完整性和可用性 3 方面提出了一些新的系统架构来保证数据的安全性。

目前保证数据机密性的主流方法是将数据进行加密。数据被加密后，用户只需要保护好自己的密钥就可以保证数据在存储过程和传输过程中的机密性。但由于云端保存的所有数据都是加密的，云存储在无法偷窥用户数据内容的同时，也无法以传统的方式提供一些常见的功能，例如数据搜索、数据删冗等。在安全云存储系统中如何提供这些功能也非常值得研究。

本文从云存储系统中的安全问题与需求出发，详细介绍了目前已有的安全云存储系统的现状与关键技术，并指出未来安全云存储系统的研究方向。

2 背景

本节我们将阐述云存储安全机制的一些相关背景。

2.1 云存储系统的安全需求

数据安全是云存储系统中最重要的安全需求之一。云存储系统中数据的安全性可分为存储安全性和传输安全性两部分，每部分又包含机密性、完整性和可用性 3 个方面。

l）数据的机密性云存储系统中的数据机密性是指无论存储还是传输过程中，只有数据拥有者和授权用户能够访问数据明文，其他任何用户或云存储服务提供商都无法得到数据明文，从理论上杜绝一切泄漏数据的可能性。

2）数据的完整性云存储系统中数据的完整性包含数据存储时和使用时的完整性两部分。数据存储时的完整性是指云存储服务提供商是按照用户的要求将数据完整地保存在云端，不能有丝毫的遗失或损坏。数据使用时的完整性是指当用户使用某个数据时，此数据没有被任何人伪造或篡改。

3）数据的可用性云存储的不可控制性滋生了云存储系统的可用性研究。与以往不同的是云存储中所有硬件均非用户所能控制。因此，如何在存储介质不可控的情况下提高数据的可用性是云存储系统的安全需求之一。

2.2 数据加密

无服务器平台通常用容器隔离 lambda。因此，优化容器初始化是 lambda 冷启动问题的关键部分。在 Linux 中，容器化不是单一的内聚抽象。相反，像 Docker 这样的通用工具通常用于使用各种 Linux 机制来构建容器，以分配存储、逻辑隔离资源和隔离性能。Linux 提供的灵活性也为设计各种特殊用途的容器系统创造了机会。在这一节中，我们希望通过分析相关的 Linux 抽象的性能特征来为 SOCK 和其他专用容器系统的设计提供信息。

容器存储

1.2.1 可搜索加密

可搜索加密可分为 4 个子过程：

Step 1： 加密过程。用户使用密钥在本地对明

文文件进行加密，并将其上传至服务器。

Step 2：陷门生成过程。具备检索能力的用户，使用密钥生成待查询关键词的陷门，要求陷门不能泄露关键词的任何信息。

Step 3：检索过程。服务器以关键词陷门为输入，执行检索算法，返回所有包含该陷门对应关键词的密文文件，要求服务器除了能知道密文文件是否包含某个特定关键词外，无法获得更多信息。

Step 4：解密过程。用户使用密钥解密服务器返回的密文文件，获得查询结果。

图 1 可搜索加密过程

从如今的应用角度，可搜索加密问题模型分为 4 类。

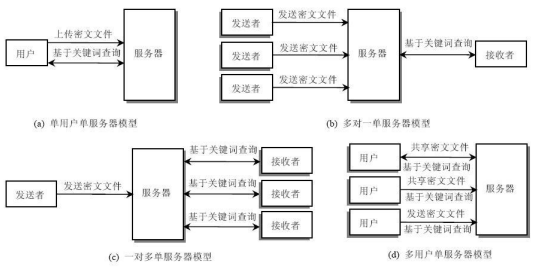

图 2 可搜索加密应用模型分类

1） 单用户（单服务器）模型。

用户加密个人文件并将其存储于不可信赖外部服务器，要求：① 只有该用户具备基于关键词检索的能力；② 服务器无法获取明文文件和待检索关键词的信息。所提到的应用问题以及单用户模式的云存储服务都是单用户模型的实例。

2） 多对一（单服务器）模型。

多个发送者加密文件后，将其上传至不可信赖外部服务器，以期达到与单个接收者传送数据的目的。要求：① 只有接收者具备基于关键词检索的能力；② 服务器无法获取明文文件信息。需要指出的是，不同于单用户模型，多对一模型要求发送者和接收者不能是同一用户。文献中的应用问题和具备简单共享机制的云存储服务都是多对一模型的实例。

3） 一对多（单服务器）模型。

与多对一（单服务器）模型类似，但为单个发送者将加密文件上传至不可信赖外部服务器，借此与多个接收者共享数据。该模型遵循着一种广播共享的模式，文献中的研究问题是一对多模型的实例。

4) 多对多（单服务器）模型。

在多对一模型的基础上，任意用户都可成为接收者，其通过访问控制和认证策略以后，具备基于关键词的密文检索方式提取共享文件的能力。要求：① 只有合法用户（例如能够满足发送者预先指定的属性或身份要求）具备基于关键词检索的能力；② 服务器无法获取明文文件信息。该模型既是多对一模型的扩展，同时也是云计算中复杂共享机制的抽象，具备广阔的应用前景。

1.2.3 属性加密

属性加密（ABE），又称模糊的基于身份的加密（Fuzzyidentity-basedencryption），最先由 Waters 提出，也被看作是最具前景的支持细粒度访问的加密原语。

与以前的公钥加密方案，如 RSA 和身份基加密最大的不同点就是，ABE 实现了一对多的加解密。不需要像身份加密一样，每次解密都必须知道接收者的身份信息，在 ABE 中它把身份标识被看做是一系列的属性。当用户拥有的属性超过加密者所描述的预设门槛时，用户是可以解密的。但这种基于预设门槛的方案不具有通用性。因为在语义上无法表述一个普遍通用的情景。

基于属性加密主要分为两大类：密文策略的属性加密（CP-ABE）和密钥策略的属性加密（KP-ABE）。在 CP-ABE 中密文和加密者定义的访问策略相关联，密钥则是和属性相关联；在 KP-ABE 中密文则是和属性相关，而密钥与访问策略相关联。属性加密算法一般包含四个部分：

Setup 阶段：也称系统初始化阶段，输入系统安全参数，产生相应的公共参数（PK）和系统主密钥（MK）；

KeyGen 阶段：也称密钥生成阶段，解密用户向系统提交自己的属性，获得属性相关联的用户密钥（SK）；

Enc 阶段：也称加密阶段，数据拥有者对数据进行加密得到密文（CT）并发送给用户或者发送到公共云上；

Dec 阶段：也称解密阶段，解密用户获得密文，用自己的密钥 SK 进行解密。

1.2.3 AONT 加密

最初 Rivest 提出 AONT 的目的是在加密过程中作为一种预处理手段，结合一般加密模式，来使得该密码受到的非法攻击（如穷尽密钥搜索）更加困难。AONT 所具有的性质保证了它可以用于任何其他的加密体制。因此，为了更好的发挥 AONT 的独特性质，我们把 AONT 和分组密码的一些工作模式结合起来进行研究。在首先简单介绍了分组密码的几个工作模式，并仔细分析了 Rivest 给出的实例“包变换”所具有的一些特点后，为了避免“包变换”所带来的安全及运行速度方面的不足，我们将 AONT 放在普通加密模式之后并结合单向陷门函数构造了一个新的加密模式，该模式在加密时能比“包变换”更安全快速地运行，而且在明文信息较短时更为安全可靠。

AONT 最初的提出可以用作构造一种新的工作模式来代替以前 DES 和 AES 在实际运用过程中的几种工作模式，而为了进一步加强加密体制的安全性和可实际操作性，我们将 AONT 变换与通常的工作模式结合起来使用，构造出了新的工作模式．在介绍新的工作模式之前，我们将 DES 和 AES 所采用的几种工作模式先做一简单介绍。

工作模式是一种算法，它刻画了如何利用分组密码提供信息安全服务。1980 年 NBS（即现在的 NIST）公布了四种 DES 的工作模式，它们分别是 ECB， CBC，CFB，OFB。在 AES 即将诞生之际，2000 年 3 月 NIST 为 AES 公开征集保密模式，后来于 2001 年从 15 种候选工作模式中选定了 5 种工作模式，分别是 ECB，CBC，CFB， OFB 和 CTR，前四种用法类似于 3DES，CTR 是一种新的工作模式，在特定的环境下它有一些值得注意的优点。

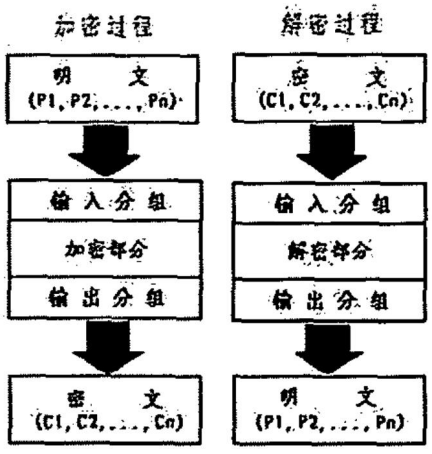

图 3 ECB 模式示意图

2.3 安全云存储系统设计的一般原则

安全云存储系统是云存储系统的一个子集，它指的是包含了安全特性的云存储系统。安全云存储系统的设计者常常会提出一些安全方面的假设，然后根据这些假设建立系统的威胁模型与信任体系，最终设计并实现系统或原型系统。一般来说，安全云存储系统设计时需要考虑如下几个方面：

1）安全假设。在安全领域中，最好的假设是除自己以外的所有实体都不可信。但是在云存储系统中，数据被存放在云端，拥有者对数据丧失了绝对控制权，使得这一假设只存在理论上的可行性。因此，云存储安全系统的设计者需要针对不同的应用场景提出相应的安全假设，并以此为前提来保证系统的安全性。

2）威胁模型和信任体系。设计者基于安全假设相关实体进行分析，由此得出相关实体是否可信，然后将这些实体模型化或体系化，由此得出相应的威胁模型和信任体系。

3）保证系统安全的关键技术。设计者往往会根据自己系统的应用场景与特征，采取一些相关技术来保证系统的安全性，这些技术也称为安全云存储系统的关键技术。盾体，在保证系统安全性的同时必然会在一定程度上降低系统效率。在安全云存储系统中，设计者需要对系统的安全与效率进行均衡，使得系统能够在适应所需的安全需求的同时，为用户提供可接受的性能。

3 研究现状

在本节中，本文将介绍在云存储安全机制中的可搜索加密，可追溯的属性加密以及基于 AONT 算法的保护资源机制的研究现状。分别是三篇论文的主要内容。

3.1 安全云存储系统的研究现状

从存储系统的技术支撑与发展来看，文件系统是构建云存储系统的重要部分。CFS 用是最早的加密文件系统之一，它是一个用户态的虚拟加密文件系统，可以挂在在其他文件系统之上，为使用者提供文件/文件名加密保护的功能。此后，NCryptfs，ECFS，Cepheus，TCFS 等都是在 CFS 的基础上研究开发的。NCryptfs 是一个内核态的加密文件系统，它将 CFS 的思想从用户态提升到内核态，同时为用户提供了方便的共享机制。ECFS 在加密数

据的基础上提出了校验数据散列值（Hash value）的方式，提供了数据的完整性保护功能。Cepheus 提出了三方架构的模式，提出一个可信的第三方服务器进行用户密钥的管理，引入了锁盒子机制进行用户分组管理，同时提出了懒惰权限撤销的思想。TCFS 提出了多级密钥的加密方式，使用一个主密钥加密原来的文件密钥。

随着网络存储系统的发展，加密文件系统的理念也逐渐网络化、系统化，最终演变成安全网络存储系统。一般的安全网络存储系统至少包括客户端与服务器两部分，客户端由系统的使用者进行操作，为用户数据提供数据加解密、完整性校验以及访问权限控制等功能；服务器作为数据及元数据的存储介质，对数据没有任何的访问或使用权限。

安全网络存储系统中比较典型的有 Plutus，SAND 以及 Corslet 等。Plutus 是 Cepheus 思想在网络存储系统中的扩展。在 Plutus 系统中，客户端负责所有的密钥分发与管理，在共享过程中为用户数据与元数据提供端到端的机密性和完整性保护。SAND 提出了一种密钥对象的数据结构，为网络存储系统提供了端到端的安全解决方案。Corslet 是一个栈式文件系统，通过引入可信第三方服务器，消除了用户对底层存储系统的依赖，在不可信的网络环境下为用户提供端到端的数据私密性、完整性的保护以及区分读写的访问权限控制功能。

云存储的廉价、易扩展等特性使得它一出现就成为人们研究的热点，由于用户将数据存放在云存储中便意味着丧失了对数据的绝对控制权，云存储系统对安全性有着十分迫切的需求。在这种需求的驱使下，Microsoft 于 2009 年提出了 Cryptographic，Cloud Storage。 Cryptographic Cloud Storage 系统以加密的方式为数据提供机密性保护、以审计的方式为数据提供持有性保护，同时为系统提供了细粒度的访问控制功能，并在系统的原型设计中使用了可搜索的加密机制（searchable encryption）、基于属性的加密机制（attribute-based encryption）、数据持有性证明 (probable of data possession) 等技术，在提高系统整体性能的同时增强了用户的体验效果。

同业界一样，学术界也很重视云存储系统的安全问题。2010 年，Tang 等人在 FADE 可系统中提出了一种解决云存储系统中数据可信删除（assured delete）的方法；Mahajan 等人在 Depot 系统中提出了一种最小化云存储中可信任（可用性方面）实体的方式，只要有一个正确（可访问）的客户端或服务器上有用户需要的数据，用户就可以通过网络获取到正确的数据；Shraer 等人在 Venus 系统中提出了基于一个核心集（core Set）的信任体系，通过三方架构的方式为用户提供安全功能。2011 年，Bessani 等人在 DEPSKY 中提出了云中云的思想，在一定程度上减轻了数据机密性问题和运营商锁（vendor data lock-in）的问题。

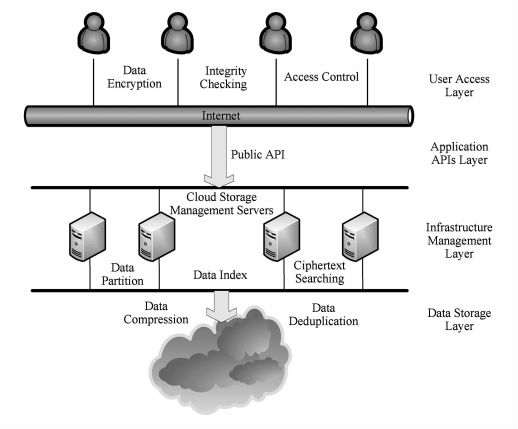

图 4 安全云存储系统通用架构

3.2 多用户可搜索加密算法

可搜索加密的思想是服务器可以在不解密的情况下对加密数据执行搜索。一般而言，可搜索加密可分为带关键字搜索的公钥加密（PEKS）和可搜索对称加密（SSE）。

大多数可搜索的对称加密方案都是在单用户环境中，客户端既是数据所有者又是数据用户，服务器负责存储加密数据、执行数据搜索并返回相应的结果。这种 SSE 适合单个用户在云存储中存储个人敏感数据的需要。然而，云存储的好处之一是数据共享的便利性，单一用户的 SSE 方案无法满足这种需求。

在这项工作中，我们提出了一个多用户可验证可搜索对称加密（MVSSE）方案（图 5），该方案允许对多个用户共享的加密数据进行有效搜索。与现有的使用单关键字索引的 SSE 方案相比，多关键字搜索效率较低，即要求搜索每个关键字一次，然后在所有返回集合中进行逻辑运算（合取、析取等），我们的 MVSSE 使用了一个双关键字索引，可以通过各种逻辑操作显著加速多关键字搜索。此外，我们的方案还实现了 SSE 的其他特性，如对不诚实服务器的可验证性和支持动态数据操作。我们的工作有四个方面的贡献：

1） 制定了 MVSSE 的安全定义，即隐私性和可靠性。

2） 提出了一种利用双关键字索引减少搜索时间的有效方案。它也是第一个支持可验证性和云存储动态操作（即添加、删除和修改文件）的多用户 SSE。

3） 我们定义了 MVSSE 的理想功能 FMVSSE，它也可以被将来的研究用来证明任何新提出的方案的 UC 安全性。

4） 我们证明了 UC 安全与安全 MVSSE 的安全要求（即隐私性和可靠性）之间的等价性。然后我们证明了我们的 MVSSE 方案对非自适应对手是 UC 安全的，即该方案安全地实现了我们理想的功能 FMVSSE，这意味着隐私性和可靠性。

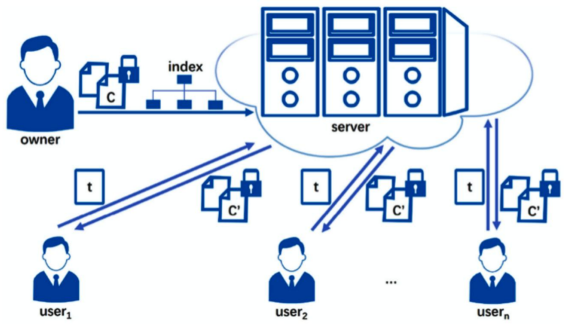

图 5 安全云多用户 SSE

如图 5 所示，我们的系统包括三个不同的方面：一个是在云服务器上存储文档的数据所有者，一个是允许搜索数据所有者存储文档的数据用户组，一个是为数据所有者提供存储服务和为数据用户提供搜索服务的云服务器。

数据所有者负责为数据用户生成系统参数和私钥。然后对文档进行加密，生成相关索引和相应的可验证信息。最后将加密后的文档和索引存储在云服务器上，并将可验证的信息发送给系统用户。它还可以在云服务器上添加、修改和删除文档。

数据用户允许数据用户搜索存储在云服务器上的文档。当数据用户想要搜索包含一些关键字的文档时，它会生成一个陷门并与云服务器通信以检索目标文档。最后，数据用户还应该能够对搜索结果执行有效性检查，以决定是否接受搜索结果。

云服务器负责存储数据所有者上传的文档和索引，并用文档和验证信息响应有效数据用户的搜索查询，以便用户对搜索结果进行有效性检查。

本文假设云服务器是一个恶意对手。这意味着云服务器对文档以及相应的索引和关键字很好奇，并试图收集比允许的泄漏更多的信息。此外，云服务器还可以删除或伪造文档，只返回部分搜索结果。值得注意的是，尽管一些恶意行为，如伪造搜索结果或返回较少的结果可能并非云服务器故意为之，但它们仍可能由于恶意软件或软件漏洞而发生。假设数据所有者是诚实的。换句话说，它诚实地生成参数，将文档和相关索引一起外包给云服务器。同时，它还为授权数据用户提供可信的密钥和可验证的信息。数据用户也应该是诚实而好奇的。在每次搜索过程中，他们都会诚实地生成一个陷门并与云服务器通信。

3.3 可追溯的属性加密算法

基于属性的加密（ABE）是一种非常有前途的方法。保护外包数据，为云存储服务提供细粒度的访问控制。在 CP-ABE 上下文中，密文和密钥分别用访问策略和描述属性集进行标记。在云存储服务 CP-ABE 系统中，数据拥有者可以针对潜在授权云用户应该拥有的属性指定访问策略，授权的云用户将获得与其属性集相对应的访问凭证，并可以访问外包的数据。如果云用户拥有的属性集满足数据所有者指定的访问策略，则云用户被授权。直观地说，CP-ABE 不仅实现了对云中外包数据的细粒度访问控制，而且为云中外包数据的保护提供了可靠的方法。

然而，在实践中，也存在着一个亟待解决的重要问题。在 CP-ABE 中，拥有一组属性的用户可以通过密文的访问结构来解密密文。与多个解密属性关联的用户共享了不同的权限。因此，很难追踪故意泄露密钥的恶意用户。因此，恶意用户可能会故意将自己的解密密钥泄露给他人牟利。

在许多例子中，密钥泄漏破坏了加密系统的预期用途。例如，Staddon 等人提出了一种用于文档编辑的 ABE 系统，其中部分文档通过加密被“屏蔽”。那些属性满足编校文档策略的用户就可以解密这些被屏蔽的区域，并完全读取文档的内容。例如，法律文件可能会在报告中抹去未成年人的名字。只有执法人员、法官或高级政治家（根据政策）才能解密这些信息。在这种情况下，如果秘密钥匙泄露给新闻界，文件中未成年人的隐私权就会受到侵犯，也许当地法律也会受到侵犯。Bethencourt 等人对 ABE 的另一个建议用法是联邦调查局加密文件。这些备忘录是由联邦调查局特工加密的，因此只有拥有特定证件的人才能解密和访问。由于在 ABE 系统中没有授权密钥的追索权，低收入的特工

可能会被诱惑向私家侦探或新闻界出售他的秘密钥匙。

对于云存储服务，恶意数据消费者可能会将自己的访问凭证泄露给他人，以获取经济利益，几乎没有被抓的风险。我们以 Alice（属性为“fAlice；Department of Research；Engineerg”）和 Bob（属性为“fBob；Department of Research；Engineerg”）为例。作为一种表达式的一对多加密机制，CP-ABE 使数据能够在具有基于角色的描述性属性的访问策略下加密和外包到云中，例如“研究和工程师部门”或“高级工程师”。同时，Alice 和 Bob 都能够委托与属性集“fDepartment of Research；Engineerg”对应的解密密钥（即访问凭证）。如果一个人在因特网上出售这种属性集的密钥（或访问凭证），就很难确定谁是原始卖家。

功能加密通常被用作商业应用的原型。在功能加密的一些特殊情况下 (如基于身份的加密, IBE)，解密特权是密钥所有者的专有权，跟踪公开的解密密钥的所有者是很简单的。实际上，在所有的一对一加密机制（如 IBE）中，跟踪任务可以简单地通过将用户的身份嵌入其密钥来实现。然而，在 CP-ABE（一对多加密机制）的情况下，加密的数据通常与一些潜在用户共享，而不知道谁将接收这些数据。简单地将身份嵌入到用户的密钥中是行不通的，因为这将与 CP-ABE 的细粒度访问控制属性相冲突。因此，我们需要新的思想和技术。由于 CP-ABE 的特性，具有相同属性的用户共享相同的解密特权，这使得他们有机会出于商业目的泄露秘密密钥，而不会被发现。这实际上是恶意的密钥委托问题。这种恶意的密钥委托问题严重限制了 CP-ABE 的实用性。因此，有必要考虑 CP-ABE 系统中泄露解密权限的恶意用户的可追踪性。泄露解密特权大致有两种。第一种是泄漏解密密钥，第二种是泄漏解密黑匣子。相应地，大致有两种可追溯性。第一种是白盒跟踪，给定一个格式良好的解密密钥，跟踪算法可以识别出泄露该密钥的恶意用户。第二种是黑盒跟踪，给定一个解密黑盒，跟踪算法可以识别出使用其密钥构建设备的恶意用户。如所述。

黑盒跟踪场景、解密密钥甚至给定解密黑匣子的解密算法都可以隐藏起来。在这种情况下，跟踪算法仍然可以跟踪密钥被用于构造解密黑盒的恶意用户。直观地说，黑盒可追溯性比白盒可追溯性强。而在实践中，白盒可追溯性比黑盒可追溯性更高效、更容易实现。本文旨在实现一个高效、完全安全的云存储服务 CP-ABE 系统，该系统具有白盒可跟踪性，能够有效地抵抗恶意密钥委托，并能有效地跟踪泄露访问凭证的恶意云用户。

首先提出了两种用于叛徒追踪的非交互式承诺：具有完全绑定密钥的非交互式可跟踪承诺和具有完全隐藏密钥的非交互式可跟踪承诺。我们进一步提出了一个完全安全的白盒可追踪的云存储服务 CP-ABE 系统（T-CPABE-pbCom 系统）。我们的系统使数据所有者能够为他们要外包的数据定义访问策略，任何授权的云用户都将获得与其拥有的属性集相对应的访问凭证。对于非法泄漏访问凭证到外包数据，我们的系统可以跟踪泄漏访问凭证的所有者。

我们给出了承诺计划的定义。承诺方案是一个两阶段的交互协议, 双方称为发送方 S 和接收方 R。发送方 S 可以将自己提交给一个随机值 R，从而满足以下两个相互冲突的要求。

1. 隐藏：在第一个阶段（称为承诺阶段）结束时，R 没有获得任何关于 S 值的知识。即使接收者试图作弊，也必须满足这一要求。

2. 绑定：给定承诺阶段的交互记录，最多存在一个 R 可以稍后 (即在第二阶段, 也称为揭示阶段) 接受为承诺的合法“打开”的值。即使发送者试图作弊，也必须满足这一要求。

另外，我们应该要求协议是可行的，在这个意义上，如果 S 和 R 都诚实地遵循它，那么在揭示阶段结束时，R 得到 S 提交的值。

我们将用户的身份隐式地插入具有完全绑定密钥的可提取承诺中，并使用完全绑定密钥（即可提取密钥）来跟踪叛徒。如文所述，本方案的解密在身份空间的大小上需要多项式时间，因此，上述方案只能用于对短消息进行加密。显然，可以使用承诺提交较长的标识，例如会话密钥，使用任何操作模式将短块上的密码转换为任意长块上的密码。我们注意到，可以通过预先计算 gq 的（多项式大小）幂表来加速解密，使得解密可以在恒定的时间内发生。

3.4 基于AONT算法的保护资源机制

如果说安全问题和（或实际）失去控制的感觉是集中式云的一个问题和减缓因素，那么对于分散式云存储来说，情况更是如此，在这种情况下，网络的动态性和独立性可能暗示着所有者对其资源在何处以及如何管理的控制权进一步降低。实际

上，在集中式云系统中，CSP 通常被认为是诚实的，但很好奇，然后被信任执行授权用户请求的所有操作（例如，在所有者请求时删除一个文件）。不鼓励顾客服务提供商恶意行事，因为这显然会影响其声誉。相反，分散系统的节点在不影响声誉的情况下可以提供经济利益（如出售已删除文件的内容）时，可能会出现恶意行为。客户机端加密通常假定在 DCS 中提供了第一个关键的保护层，但它会使资源面临威胁，特别是在长期内。例如，在加密密钥暴露的情况下，或者在恶意节点未根据所有者的请求删除其碎片以尝试整体重建资源的情况下，资源仍然易受攻击。

因此，在 DCS 场景中，加密密钥的保护是不够的，因为它仍然暴露在上述威胁之下。一般的安全原则是依赖于一个以上的防御层。在本文中，我们提出了一个额外的正交保护层，可以减轻这些风险。

然而，从积极的方面来看，我们注意到，分散控制系统的性质也提高了服务的可靠性，因为一系列独立方的参与减少了单一故障可能限制对存储资源的访问的风险。除此之外，DCS 系统的独立结构——如果加上有效的资源保护和对网络中节点的谨慎分配——使它们有希望切实加强对依赖分散网络存储数据的所有者的安全保障。

在本文中，我们提出了一个解决方案，使资源所有者能够安全地将其资源存储在 DCS 服务中，与其他用户共享，同时仍然能够安全地删除它们。我们的贡献是三倍的。首先，利用全有或无转换（AONT）提供的保护保证，我们设计了一种仔细控制网络中节点的资源切片和分配的方法，以确保可用性 (即检索所有切片以重建资源) 和安全性 (即防止恶意方）联合收集组成资源的所有切片）。所提出的解决方案还使资源所有者能够在需要时安全地删除其资源，即使在 DCS 中的某些节点行为不当时也是如此。其次，我们研究了在分散网络上进行资源切片和分配的不同策略，并从可用性和安全性保证方面分析了它们的特点。第三，我们提供了一个问题模型，使所有者能够控制切片的粒度和分配的多样化，以确保目标可用性和安全性保证。我们在一个现有的 DCS 系统上进行了一些实验，证明了该模型的有效性。我们的解决方案提供了一种有效的方法来保护分散云存储中的数据，并确保了可用性和保护性，以响应新兴分布式控制系统场景的当前开放问题，包括安全删除。事实上，虽然考虑了明显相似的要求，但通用的秘密共享解决方案（例如 Shamir）不适用于整个资源内容（而不仅仅是加密密钥）需要保护的场景，因为它们的存储和网络成本（例如：Shamir 方法中的每个共享的大小与必须保护的整个数据的大小相同）。

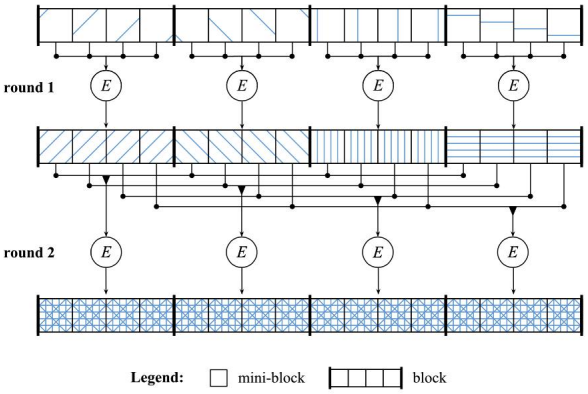

图 6 MIX&SLICE 的例子

AONT 加密为外部存储器转换资源的加密模式。此模式需要使用加密密钥。密钥驱动的加密代表一级保护，AONT 加密模式的使用进一步加强了安全性。AONT 加密模式将明文资源（任何形式的原始内容）转换为密文，其属性是转换的整个结果需要获得原始明文。AONT 实际上保证加密资源的比特之间完全相互依赖（混合），这样加密资源的一部分不可用会阻止原始明文的任何部分的重建。一方可以访问加密资源的一部分（但不能访问整个加密资源）：i）如果知道加密密钥，它将无法重建资源的任何部分（即，它将无法从它拥有的 AONT 加密部分中获取任何信息；唯一的选择是尝试一个暴力对缺失部分的可能配置进行强制攻击，但其可能较大的大小使此攻击不可行）；ii）如果不知道加密密钥，则无法对猜测该密钥进行暴力攻击，因为任何密钥（即使是正确的密钥）如果不应用于整个资源都将无效。AONT 保护方案可以使用常见的加密函数来构建，比如对称加密和哈希函数。保证完全混合的 AONT 方案的一个例子是 Mix&Slice，它也被用于原型的实现中。直观地说，Mix&Slice 是通过应用不同的加密轮来工作的，每一轮加密都是在上一轮加密产生的位的精心设计的组合上进行的。使用 Mix&Slice，i 轮加密工作在块上，每个块包括 b 个小块，保证由双小块组成的资源完全混合。演示了一个混合两轮加密的示例。第一轮混合相邻的小块，而第二轮混合代表第一轮中不同计算的小块，提供整个资源内容的混合（如图中的模式

编码所示）。Mix&Slice 保证加密资源中的每个位都依赖于其明文表示中每个位的值。在我们的上下文中 ANOT 保证了对组成资源的各个片（和碎片）的保护，因此也保护了资源本身（包括它的整个部分和任何部分）。事实上，AONT 根据信息理论，对所需资源的每一部分进行重构。然后通过缺少信息内容来提供保护。

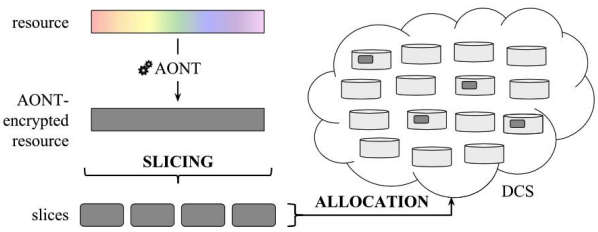

图 7 参考方案

图 7 展示了我们的参考方案。本文的重点是在 DCS 系统中，合理地设计资源切片，并将生成的切片分配到不同的节点。注意，在本文中，我们使用术语“切片”来指资源的切割，而术语“切片”则指这样一个过程的结果。因此，片是资源的一块，表示一个分配单元，而 shard 表示分配给节点的资源的一部分（shard 可以包括多个片）。我们的方法侧重于切片和分配，对于要使用的特定 AONT 技术是不可知的，只要确保有针对性的强保护保证，并且对于所采用的特定 DCS。

在我们的方法中，将资源分为多个切片以分布在不同的节点上，这是由需要保证的可用性和保护属性指导的。可用性（尽管节点出现故障或暂时无法访问）是通过复制提供的，安全性是通过防止恶意联盟提供的。恶意节点（及其联盟）希望通过不返回它们存储的资源片段来使资源不可用，或者通过不删除它们存储的资源片段并将这些片段返回给付费的（未授权的）用户来提供对资源的访问，即使在资源被删除之后也是如此。在处理切片之前，我们描述了资源分布的复制和联盟抵抗特性。

我们假设(转换后的)资源在客户端经过 AONT 加密（如前一节所述）。为了简单起见，我们将省略这种关于转换的明确注释，我们将简单地使用术语 resource 来表示 AONT 加密的资源。此外，我们假设一个资源由不同的片组成，以便在 DCS 中分布。

我们将一个资源建模为一组 $S=\{S_1, \ldots, S_s\}$片，分配给 DCS 的节点，用 N 表示。

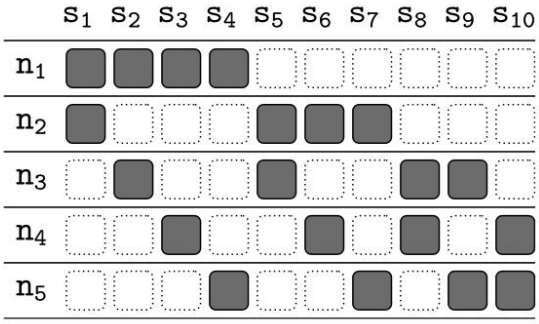

图 8 分配函数示例

分配函数指定如何将切片分配到 DCS 中的节点。在同域中考虑节点集（与单个节点不同）可以适应复制。排除空的节点集确保无损分布（即，每个片被分配到至少一个节点）。图 8 说明了一个分配函数的例子，考虑到将一个资源分成十个部分，分配给 DCS 中的五个节点（图中不报告分配中未使用的节点）。图中每个节点有一行，每个切片有一列。将一个切片分配给一个节点是用一个灰色框表示的，它位于表示节点的行和表示切片的列之间。带有虚线框的空框表示切片未分配给节点。

我们确定了一个分配的两个主要属性，描述了由复制提供的可用性和由分配的多样化所提供的对节点可能的恶意联盟的保护。

我们根据系统中维护的副本数量来描述复制提供的可用性。虽然原则上为每个切片维护的副本数量可能不同，但我们假设所有切片使用相同数量的副本。这源于这样一个事实，即我们假设节点不与单个可靠性配置文件相关联。因为重建资源需要所有的片，所以对任何片使用较少的副本会降低资源的可用性，这将由这样一个下限决定。将复制函数的定义形式化。

4 评估

在本节中，本文将三篇文章的算法在效果评估上的对比和结果。

4.1 多用户可搜索加密算法

在本节中，我们给出了我们在 C++中实现的实验结果。我们的实验是在一台使用英特尔酷睿 i7-4770cpu（8 核 3.4ghz）和 16gb ram、运行 64 位 windows7 的 PC 机上进行的。系统性能通过 NIPS 全文（包含 1500 个文件）和安然电子邮件（包含 39861 个）进行评估。

图 9 表示出了设置阶段的时间成本。由于该阶段只包含生成系统参数和多用户密钥，因此时间成本与系统用户（最大）数量成线性关系。当系统用户少于 100 人时，时间成本小于 1 秒。当有 10000 个用户时，它的成本不到 78 秒，这仍然是合理的，因为设置是一次性的过程。

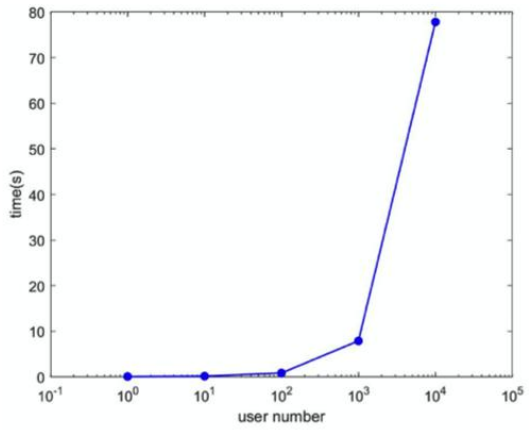

图 9 设置

图 10 表示出了存储阶段的时间成本，即，构建索引。对于 NIPS 数据集，我们分别测量数据集中 20%、40%、60%、80%和 100%的文档建立索引的时间。虽然存储阶段比其他阶段花费更多的时间，但它是用户在将文档上载到服务器之前一次性完成的过程。在确定文档数量的情况下，存储阶段的时间成本与关键字数量呈二次增长关系。当文档数量为 1500 个关键字的数量是 50 个，大约需要花费 68.0954 秒。

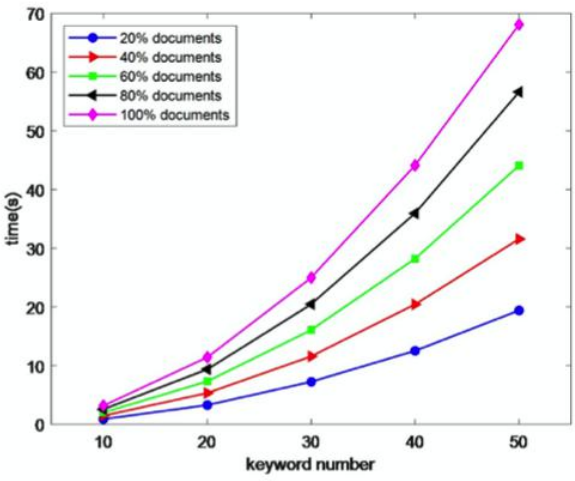

图 10 存储

在 TokenGen 阶段，用户进行的计算大约花费 1 毫秒，而服务器进行的计算大约需要 6 毫秒。它们都独立于文档和关键字的数量。

我们将搜索阶段分为三个步骤来衡量它的时间消耗：用户端的搜索，服务器端的搜索和用户端的验证。

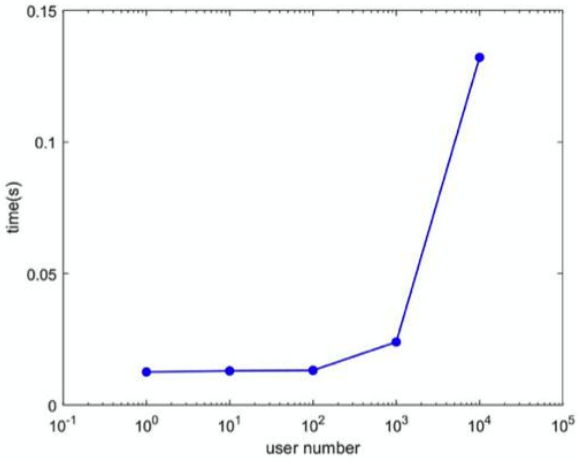

图 11 搜索（用户）

图 11 表示出了用户侧的搜索的时间成本。这只取决于用户的数量。当系统用户少于 100 个时，花费不到 13 毫秒。当有 10000 个用户时，它的成本不到 0.14 秒，仍然非常高效。

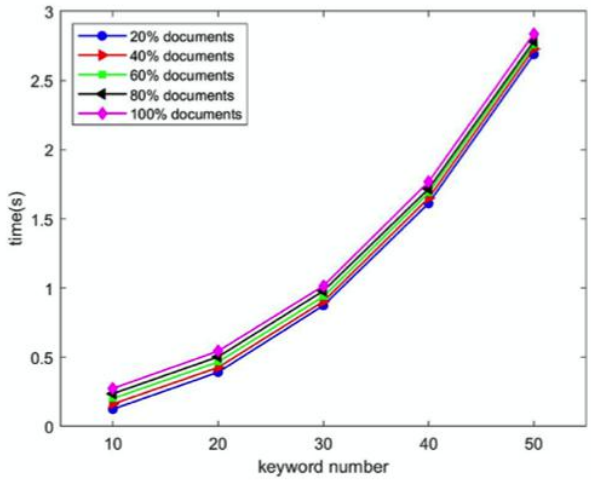

图 12 搜索（服务器）

图 12 表示出服务器侧的搜索阶段的时间开销。由于服务器需要计算两个累加器的见证值，因此时间消耗相对于关键字数量呈二次增长。NIPS 数据集（1500 个文件）和 50 个关键字只需不到 3 秒钟的时间。

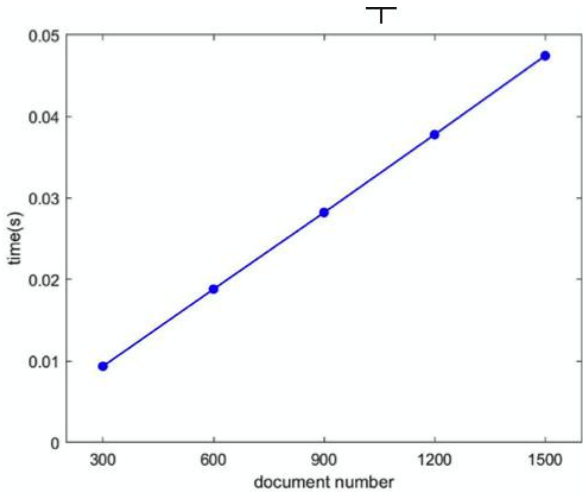

图 13 验证

图 13 表示出在用户侧的搜索阶段中的验证的时间成本。

	# of rounds	trapdoor computation	index computation	search	multi-user	hide search pattern	adaptive adversary	verifiability	verification cost	update
[1]	1	$\mathcal{O}(1)$	–	$\mathcal{O}(nL)$	✗	✓	–	✗	–	✗
[2]	1	$\mathcal{O}(r)$	$\mathcal{O}(n\|\Delta\|)$	$\mathcal{O}(n)$	✗	✓	✓	✗	–	✗
scheme1[3]	1	$\mathcal{O}(1)$	$\mathcal{O}(n\|\Delta\|)$	$\mathcal{O}(n)$	✗	✓	✗	✗	–	✓
scheme2[3]	2	$\mathcal{O}(1)$	$\mathcal{O}(m)+\mathcal{O}(n\|\Delta\|)$	$\mathcal{O}(n)$	✗	✓	✗	✗	–	✓
SSE-1[4]	1	$\mathcal{O}(1)$	$\mathcal{O}(m)+\mathcal{O}(n\|\Delta\|)$	$\mathcal{O}(\mathcal{R}(w))$	✗	✓	✗	✗	–	✗
SSE-2[4]	1	$\mathcal{O}(n)$	$\mathcal{O}(n\|\Delta\|)$	$\mathcal{O}(\mathcal{R}(w))$	✗	✓	✓	✗	–	✗
[5]	1	$\mathcal{O}(n)$	$\mathcal{O}(mn)$	$\mathcal{O}(\mathcal{R}(w))$	✗	✓	✗	✓	$\mathcal{O}(n)$	✗
[6]	1	$\mathcal{O}(1)$	$\mathcal{O}(m)$	$\mathcal{O}(1)$	✗	✓	✗	✓	$\mathcal{O}(\mathcal{R}(w))+\mathcal{O}(n)$	✓
[18]	1	$\mathcal{O}(\log(n\|\Delta\|))$	$\mathcal{O}(n\|\Delta\|)$	$\mathcal{O}(\mathcal{R}(w)\log^2(n\|\Delta\|))$	✗	✗	–	✓	$\mathcal{O}(n)$	✓
[19]	1	$\mathcal{O}(l)$	$\mathcal{O}(\|e\|^l)$	$\mathcal{O}(1)$	✗	✗	–	✓	$\mathcal{O}(l)$	✗
our scheme	2	$\mathcal{O}(1)$	$\mathcal{O}(m^2)$	$\mathcal{O}(1)$	✓	✓	✗	✓	$\mathcal{O}(\mathcal{R}(w))+\mathcal{O}(n)$	✓

表 1 与其他相关工作的比较

总的来说，我们的方案在实现多用户搜索的同时，具有与现有的 SSE 方案相当的性能。

4.2 可追溯的属性加密算法

实验是在一台笔记本电脑上进行的，使用英特尔酷睿 i5-5200U，频率为 2.20ghz，内存为 4gb，windows7 操作系统为 Service Pack 1。我们使用基于曲线的密码系统 A1 来实现。编程语言是 Java，JDK32-1.6.0 和 JPBC-2.0.0。

在 CP-ABE 系统中，密文策略的复杂性会影响加密时间和解密时间。为了说明这一点，我们以 (S_1 和 S_2…和 S_l) 的形式生成密文策略来模拟最坏的情况，其中 S_i 是一个属性。我们的目的是通过比较每个阶段所用的总时间与中的原始 CP-ABE，来评估我们的 TCPABE-pbCom 系统的效率。

	T	TP	MAS	FS	SM
[26]	√	none	×	×	×
[25]	√	none	×	×	√
[28]	√	linear	√	√	√
[27]	√	sub-linear	√	√	√
[31]	√	constant	√	×	√
[33]	√	constant	√	×	√
this work	√	none	√	√	√

表 2 与其他相关工作的比较

不考虑可追溯性问题。如图 14 所示，我们检查执行单个阶段的时间成本（包括设置时间、KeyGen 时间、加密时间和解密时间）。我们的系统在考虑可追溯性问题后花费了更多的时间，这一点也不奇怪。在没有引入 PBA 的情况下，我们的原始 CP-ABE 实现了显著的可追溯性 CP-ABE 和我们的 T-CPABE-pbCom 系统进行了对比。我们注意到，我们介绍的添加可追溯性问题的方法是通用的（这也适用于其他 CP-ABE 系统），很容易应用我们的技术来获得更有效的系统。

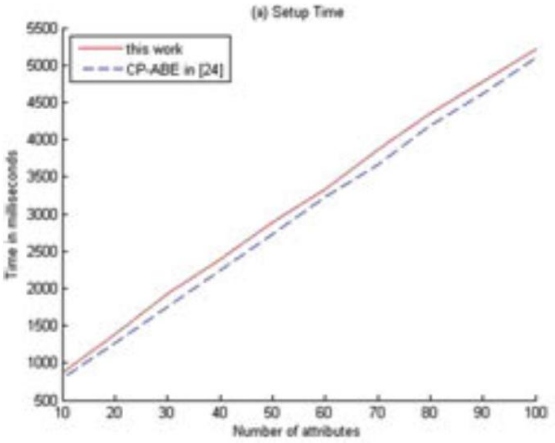

图 14 实验结果（设置）

我们可以使用其他的方法中使用的技术将 T-CPABE-pbCom 系统扩展到 prime order 设置系统。具体地说，我们可以利用冈本和高岛提出的对偶向量空间框架，在素序群中建立一个假设，用来模拟复合序群中一般子群决策假设的效果。正如所述，我们可以获得一个工具箱，用于将我们的复合序结构转换为可以从 DLIN 证明安全的素数序构造。使用其中开发的翻译技术（基于 DPV）可以将复合序结构（即我们系统的底层结构）转换为素级模拟。请注意，基于中开发的 DPV 的翻译技术通常是适用于依赖于取消性质的复合序构造，并且从子群决策假设的变体证明是安全的。值得注意的是，我们系统的底层 CP-ABE 结构和我们使用的可提取承诺基本上和关键地依赖于取消属性，并且从子群决策假设的变体中证明是安全的，这些假设是由中的翻译技术（基于 DPV）支持的构造。此外，我们还可以采用文中介绍的技术，将复合序群模拟成素序群，以提高系统的效率。

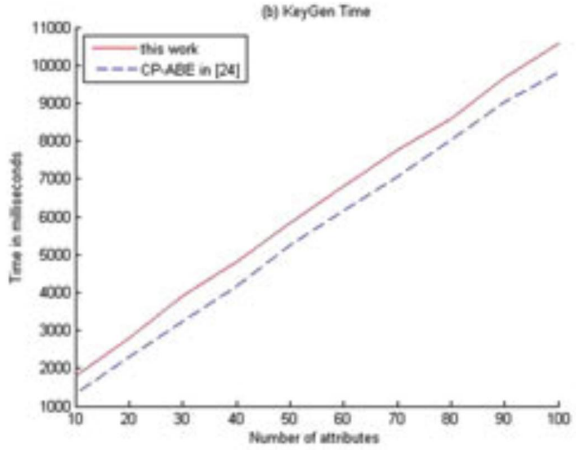

图 15 实验结果（密钥）

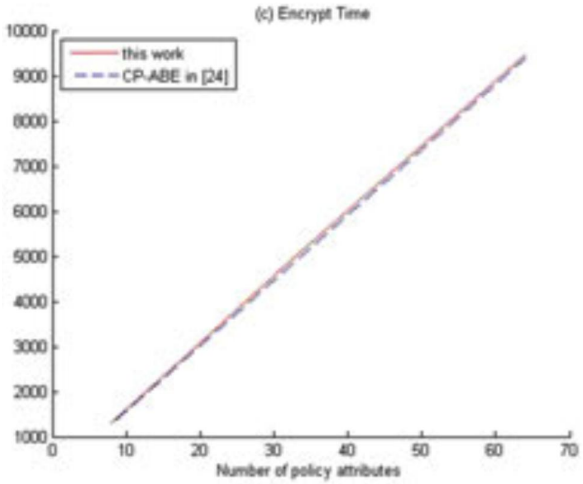

图 16 实验结果（解密）

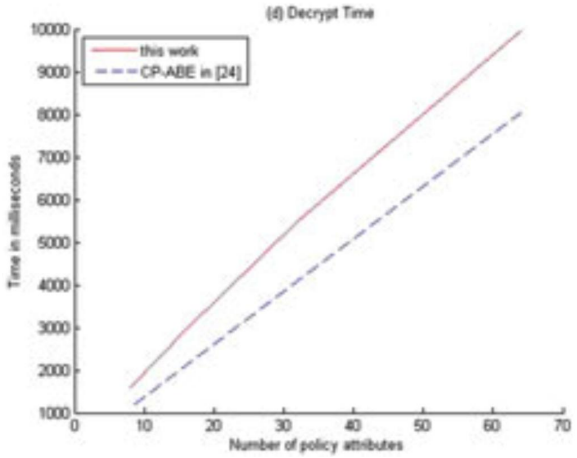

图 17 实验结果（加密）

我们可以将 T-CPABE-pbCom 系统扩展到一个可撤销的云存储服务系统。具体来说，我们可以利用文献中的密文委托和分段密钥生成技术来实现撤销的功能。我们注意到，由于我们提出的系统在跟踪算法中使用了承诺，所以系统不需要维护一个包含所有用户标识的身份表 T。这带来了这样一个优点：当一些恶意用户被撤销时，系统不需要维护（即更新）。

4.3 基于AONT算法的保护资源机制

为了验证我们的建议的好处，我们将其应用到现有的 DCS 网络中。在现有的 DCS 网络 (如 Storj、Sia、IPFS 和 Maidsafe Safenetwork）中，我们选择了 Storj，因为据我们所知，它是目前最先进和最受支持的 DCS。与这些 DCS 相关的加密货币的市场估值（Storj 用于 Storj，Siacoon 用于 Sia，Filecoin 用于 IPFS，MaidSafeCoon 用于 Maidsafe）证明了这些解决方案的重要性：在提交之日，这些方案的全球市值超过 4 亿美元。Storj 网络目前有超过 100000 个节点提供容量，可用数据超过 100PB，计划在 2019 年实现 10 倍的增长。

Storj 是一种协议，它协调分散的网络以在对等方之间创建和执行存储契约。每个对等方可以与其他对等方协商合同，从其他对等方上传和下载数据，并定期验证其数据的可用性和完整性。Storj 利用分布式哈希表（DHT）将有兴趣签订存储合同的各方联系起来。在讨论中，我们保留了模型中的术语，并将资源外包给分散网络的各方（Storj 的承租人），以及提供存储空间以换取以数字货币作为存储节点的报酬的各方（Storj 的农民）。网桥节点支持系统中的正确操作，并负责验证资源的完整性和可用性。在下面，我们将描述描述我们的实现的技术选择、实验结果以及关于细粒度检索所产生影响的一些考虑因素。

在 Storj 中实施 Min_slices 和 Min_nodes 分配策略需要更改开源实现的客户端库。特别是，Storj 目前提供了三个主要的客户机，一个用 C 语言编写，必须从源代码构建，另一个用 JavaScript 编写，设计为由节点。JS 运行时，以及一个用 Python 编写并与任何 Python 环境兼容的。我们在 Python 实现中集成了我们的技术，这也是为了便于与 Mix&Slice 的实现集成, Mix&Slice 除了作为 AONT 加密之外，还支持其他保护要求（例如基于加密的访问控制和策略撤销）。Storj 的设计使客户端独立于网桥和存储节点。我们在 Python 客户机上的工作允许我们访问整个网络的服务。

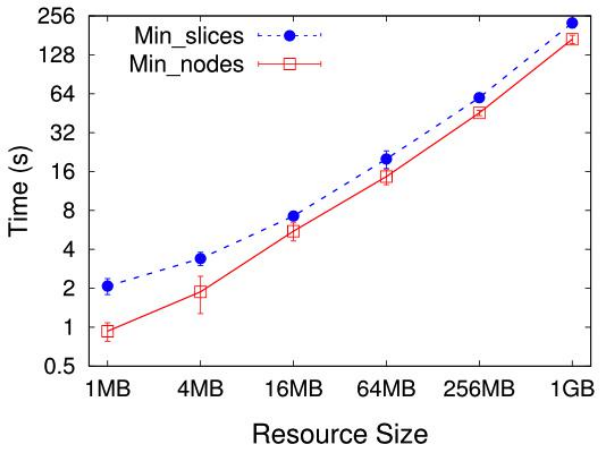

图 18 最小片分配策略中的完成时间（a）和总吞吐量（b）

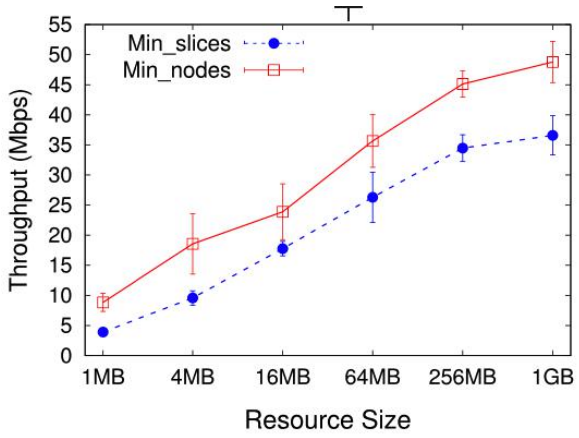

图 18 最小节点分配策略中的完成时间 (a) 和总吞吐量 (b)

最小节点分配策略，我们在客户机中引入一个模块，该模块激活多个并行线程 (在考虑的配置中，我们使用 10 个并发线程）来打开对存储节点的访问请求。在 Storj 中，来自所有者的访问请求同时涉及存储节点和网桥节点。事实上，每次所有者需要检索碎片时，她都会向网桥发出请求，网桥会返回一个令牌以及存储所需碎片的节点的 IP 地址 (请注意，记录了该访问请求，并且所有者为该节点创建了加密货币支付）。然后，客户端将令牌用作指向节点的 HTTP 请求的参数。我们的实验考虑了系统在管理所有者和节点之间对话方面的性能。特别是，我们比较了两种分配策略在不同资源大小下的访问时间。

我们在客户机中实现了 Min_slices 和 Min_nodes 分配策略, 并为节点分配了 slice (在 Storj 术语中，分配给节点的所有片形成了一个 shard）。shard 创建和资源重建的性能比 Storj 网络中存储节点的吞吐量高出一个数量级（Mix&Slice 以几百 MB/s 的速度运行，而我们在 Storj 中观察到的最大吞吐量大约低两个数量级）。在我们的实验中，我们着重评估完成访问请求所需的时间，因为解密请求的时间没有显著影响。我们注意到，AONT 的使用迫使每个访问请求只有在客户端上有完整的资源可用时才能进行客户端解密。如果这是特定应用领域的一个问题（例如，资源非常大），可以通过分割大量资源并将我们的方法应用于产生的（较小的）块来提供缓解。然后可以独立下载和解密每个块。这将减少对资源的访问时间，但也可能延迟传输的完成，因为管理更多访问请求的开销会降低有效带宽。实验证实了这一观察结果，因为它们表明，管理大型资源具有性能优势。

我们注意到，与 Min_slices 策略相比，Min_nodes 策略表现出中等的优势。这一好处来自于与多个节点交互所节省的开销。节点性能的自然变化也是影响吞吐量的一个因素，有些节点的性能比其他节点快。只要性能有限的节点数（慢节点）小于 r−1，并且至少有 k+1 个性能良好的节点（快速节点）服务于 shard，则 Min_nodes 策略就可以很好地工作。对于 Min_slices，将 k+1 碎片中的一个分配给一组节点就足够了，其中并行接触的 t 节点碰巧慢于遭受访问中的显著延迟。

5 总结

安全云存储系统是云存储领域中的一个重要研究方向。本文介绍了云存储系统的安全需求、安全云存储系统的现状和一般架构，详细阐述了在现有安全云存储系统中所使用到的一些关键技术，并指出了未来安全云存储系统的研究方向。

云存储的服务性质让用户失去了对数据的绝对控制权，从而产生了云存储环境中特有的安全隐患。为此，安全云存储系统的设计者根据不同的应用场景，提出安全假设并建立相应的威胁模型与信任体系，采用合适的关键技术，设计并实现了各式各样的安全云存储系统。然而，任何安全机制都是有代价的，安全云存储系统无论是在效率上还是方便性上都要低于一般的云存储系统。

从总体上看，未来安全云存储系统的研究方向是在保证用户数据和访问权限信息安全的前提下，尽可能地提高系统效率，并提供一般云存储系统所具备的功能。目前安全云存储系统在密文的搜索、重复数据删除、数据持有性证明等功能的支持上仍有待加强，需要大家共同研究与探索。由于安全云存储系统有着无可比拟的安全优势，相信在不久的将来，它会逐步取代一般云存储系统，在实际应用中接受用户的检验。

参 考 文 献

[1] X. Liu, G. Yang, Y. Mu and R. H. Deng, "Multi-User Verifiable Searchable Symmetric Encryption for Cloud Storage," in IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 6, pp. 1322-1332, 1 Nov.-Dec. 2020, doi: 10.1109/TDSC.2018.2876831.

[2] J. Ning, Z. Cao, X. Dong and L. Wei, "White-Box Traceable CP-ABE for Cloud Storage Service: How to Catch People Leaking Their Access Credentials Effectively," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 5, pp. 883-897, 1 Sept.-Oct. 2018, doi: 10.1109/TDSC.2016.2608343.

[3] E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa and P. Samarati, "Securing Resources in Decentralized Cloud Storage," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 286-298, 2020, doi: 10.1109/TIFS.2019.2916673.