云存储中外包数据确定性删除研究综述

任正伟[1,2],李雪婷[1],王丽娜[3],童 言[4],徐士伟[4],丁 炜[5]
(1. 武汉科技大学计算机科学与技术学院,湖北武汉 430065; 2. 智能信息处理与实时工业系统湖北省重点实验室,湖北武汉 430065; 3. 武汉大学国家网络空间安全学院,湖北武汉 430072; 4. 华中农业大学,湖北武汉 430070; 5. 中国地震局地震研究所地震大地测量重点实验室,湖北武汉 430071)

摘 要: 云存储的外包特性使得数据的所有权与管理权/持有权分离,导致数据安全成为用户关注的焦点之一. 作为云存储中数据安全的一个组成部分和数据生命周期的最后一个阶段,外包数据确定性删除研究的是如何证实留存于云服务提供商、数据使用者和网络中的数据是失效、不可恢复的,从而防止数据滥用和隐私泄漏等安全隐患. 其主要研究思路是利用密码学相关理论和技术将数据删除问题转换为密钥的安全管控和删除问题,即在假定加密算法是安全的情况下,安全管控和删除密钥将使得外包的密文数据不能被解密和访问,从而实现数据在计算上的删除. 本文首先介绍了外包数据确定性删除的研究背景及其主要研究思路,并阐述了作者对于该问题的思考,包括该问题的模型及其蕴含的关键科学问题;之后,分类梳理了国内外研究现状,分析了每类方法的特点和发展趋势;接着,通过几个有着众多用户的应用案例展示了外包数据确定性删除所预想的部分功能;最后,探讨了该领域未来的研究方向.

关键词: 云存储; 数据安全; 确定性删除; 数据加密; 密钥管理; 密钥使用条件
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112(2022)10-2542-19
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220204

A Survey on Assured Deletion of Outsourced Data in Cloud Storage

REN Zheng-wei[1,2], LI Xue-ting[1], WANG Li-na[3], TONG Yan[4], XU Shi-wei[4], DING Wei[5]
(1. *School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, Hubei* 430065, *China*; 2. *Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan, Hubei* 430065, *China*; 3. *School of Cyber Science and Engineering, Wuhan University, Wuhan, Hubei* 430072, *China*; 4. *Huazhong Agricultural University, Wuhan, Hubei* 430070, *China*; 5. *Key Laboratory of Earthquake Geodesy, Institute of Seismology, China Earthquake Administration, Wuhan, Hubei* 430071, *China*)

Abstract: The outsourcing feature of cloud storage separates the ownership and management/possession of data, making data security being one of the research topics. In cloud storage, as a part of data security and the last stage of the data life cycle, assured deletion of the outsourced data is aimed to make sure that the data retained by the cloud service provider, data user, and network is invalid and unrecoverable, which can achieve the goal of preventing the potential security risks such as data abuse and privacy leakage. The main research methodology of data assured deletion is to convert the data deletion problem to controlling and deleting the encryption key securely using cryptography theories and technologies. More precisely, when the encryption algorithm is secure, the outsourced encrypted data cannot be decrypted and accessed any more if the encryption key is deleted securely. As a result, it can be considered that the data has been deleted computationally. This paper summaries and reviews the assured deletion problem of outsourced data in cloud storage. Firstly, we introduce the research background and the main research methodology of assured deletion of the outsourced data. Secondly, we expound our thinking on this problem, including the general model of this problem and the critical scientific problems it contains. Thirdly, we systematically survey the state-of-art of existing work in a classified manner and analyze the characteristics and research trends of each classification. Then, we demonstrate some of the expected functions of outsourced data as-

收稿日期:2022-02-25;修回日期:2022-06-15;责任编辑:朱梅玉
基金项目:国家自然科学基金(No.61902285);湖北省自然科学基金(No.2019CFB099,No.2021CFB314);武汉引力与固体潮国家野外科学观测研究站开放研究基金资助课题(No.WHYWZ202109);中央高校基本科研业务费专项基金(No.2662022XXYJ003);应用数学湖北省重点实验室(湖北大学)开放基金(No.HBAM202101)

sured deletion via several application cases with many users. Finally, the future research roadmaps of this field are discussed.

Key words: cloud storage; data security; assured deletion; data encryption; key management; condition of using the key

1 引言

云计算和云存储的概念已经普及化,在产业界也得到了广泛的部署和应用,但是用户对其安全性尤其是对数据安全的担忧也始终是其发展过程中的阻碍之一. 这种担忧的根源在于云计算、云存储的外包特性会使得数据的所有权与管理权/持有权分离,即数据被存储于云端,由云服务提供商(Cloud Service Provider, CSP)持有和管理,用户则不再物理持有数据. 但是,CSP往往不是完全可信的,既可能因为存储了众多数据而成为攻击的重点,也可能会因为内部管理不规范或员工违规操作而导致数据泄漏,这也就形成了云计算、云存储的便利性和用户数据安全之间的矛盾. 为此,用户在使用云计算和云存储服务时,必然要求能够从技术的角度确信其数据的安全性,并且能够参与到数据的安全管控之中,而不是无从选择地依赖和信任CSP的服务承诺. 因此,数据安全仍是云计算和云存储中需要深入研究的领域[1~7].

学术界对云计算和云存储模式下的外包数据安全已有较多的研究,包括外包数据的机密性[3]及可搜索加密技术[4]、完整性[5]、可用性[6]和计算可验证性[7]等. 但是,作为云存储中数据安全的一个组成部分和数据生命周期的最后一个阶段,外包数据确定性删除(Assured Deletion)[1,8~12]的研究相比上述研究领域,整体上还处于起步阶段[10,11],其关键理论和技术与实际应用还有很大的差距,应用场景和功能需求也需要扩展和完善.

数据确定性删除,也可被称为数据自销毁(self-destructing)[13~18]或数据可信删除[10],尽管表述不同,但其内涵是一致的,是随着云存储模式的普及而被提出来的. 其研究的是当数据生命周期到达或不满足使用条件应被删除时,如何证实脱离了数据属主(Data Owner, DO)物理控制的数据在CSP处、数据使用者(Data User, DU)端及网络中是失效、不可恢复的,从而防止数据滥用和隐私泄漏等安全隐患.

本文首先介绍了云存储中外包数据确定性删除问题的研究背景和主要研究思路,然后阐述了对该问题的思考,包括该问题的模型及其蕴含的关键科学问题;之后,以分类的方式,对国内外相关工作及每类工作的发展趋势进行了系统的梳理和分析;接着,又以几个有着众多用户的应用/系统作为案例,展示了外包数据确定性删除所预想的部分功能;最后,探讨了该领域后续可能的研究方向.

2 问题陈述

2.1 问题背景

在云存储模式下,为了保护外包数据的机密性,用户一般会将数据加密后以密文形式存储于云端. 但即使如此,数据也仍然面临着被泄漏和滥用的安全隐患. 首先,外包数据会被存储在CSP处;其次,在传输和使用过程中,外包数据也可能被网络运营商和数据使用者DU留存. 这也就意味着数据面临的安全威胁是多方面的,攻击者可以从CSP,DU或网络传输设备及路由设备处获得密文数据. 此时,一旦加密数据的密钥也泄漏了(如DU被攻击或未对密钥采取安全保护措施等都可能导致密钥泄漏),数据将没有安全可言. 因此,在确保外包数据机密性和可用性的同时,还应当研究其不可用问题,即如何安全删除数据,使其失效、不可恢复,确保数据在"不应用"时"不可用".

用户有权删除其个人数据/信息,这是很多法律法规所明确规定的,如欧盟于1995年通过的《数据保护指令》中就规定了"删除权". 2012年,欧盟在修订后的《数据保护指令》中又首次提出了"被遗忘权"(right to be forgotten),被遗忘权被认为是删除权的强化升级版. 2018年5月正式生效适用的《通用数据保护条例》(General Data Protection Regulation, GDPR)以欧盟法律的形式正式确立了被遗忘权,规定了被遗忘权的行使要件及限制条件. 在GDPR中,被遗忘权可以被概括为:数据主体有要求数据控制者删除关于其个人数据的权利,数据控制者有责任在特定情况下及时删除用户的个人数据[19~21]. 在我国,2012年的《全国人民代表大会常务委员会关于加强网络信息保护的决定》[22]、2016年的《中华人民共和国网络安全法》[23]、2021年的《中华人民共和国个人信息保护法》[24]也均有关于个人信息删除的规定. 尤其是《中华人民共和国个人信息保护法》第四十七条的规定尤为详细,其表述为"有下列情形之一的,个人信息处理者应当主动删除个人信息;个人信息处理者未删除的,个人有权请求删除:(一)处理目的已实现、无法实现或者为实现处理目的不再必要;(二)个人信息处理者停止提供产品或者服务,或者保存期限已届满;(三)个人撤回同意;(四)个人信息处理者违反法律、行政法规或者违反约定处理个人信息;(五)法律、行政法规规定的其他情形". 法律、行政法规

规定的保存期限未届满，或者删除个人信息从技术上难以实现的，个人信息处理者应当停止除存储和采取必要的安全保护措施之外的处理.

上述法律、法规虽然规定了用户有权根据其需要删除其个人数据/信息，但在云存储的外包模式下，如何从技术上使得用户的这些权利得以保障却仍是需要研究的课题. 这是因为此时数据属主DO已经不再物理持有数据，DO也就无法直接从物理上删除数据，只能依赖持有数据的CSP，DU或网络运营商来执行数据删除操作. 这将面临着以下挑战：首先，CSP和DU一般都不是完全可信的，它们可能不会如实地删除数据，却向DO返回数据已被删除的信息；其次，为了提高服务的可靠性和可用性，CSP一般会对数据进行备份并异地存储，因此，CSP在删除数据时，也可能有意或无意删除了原始数据却保留了备份；最后，数据是通过不可信的网络传输的，在跨地域跨组织的大规模网络中，数据及其副本[25]的删除更难以保证.

云存储中外包数据的确定性删除即是在此背景下被提出的，其主要研究思路是利用密码学相关理论和技术将数据删除问题转换为密钥的安全管控和删除问题[8,11,12]，即DO在将数据外包存储于云端之前，先对其加密，在假定加密算法是安全的情况下，DO安全管控密钥即可实现对数据的控制. 密钥被安全删除后，留存于CSP处、DU端和网络中的密文数据将不能被解密和访问，这时数据是不可用的，其效果等同于数据被删除了. 这种删除虽不是物理上的，但是在计算上是可以被DO所确信的，因而被称为确定性删除.

2.2 系统模型

根据云存储中外包数据确定性删除的主要研究思路，本文在通用云存储架构下提出了如图1所示的外包数据确定性删除模型. 该模型中有3个参与方：数据属主DO、云服务提供商CSP、数据使用者DU. CSP拥有丰富的存储和计算资源，向外提供存储和计算服务；DO将数据外包存储于CSP处；DU从CSP处获取并使用数据.

从DO的角度而言，CSP和DU都是不完全可信的，传输数据的网络也是不可信的，它们都可能留存数据. 为此，DO需要从技术上确信留存于CSP，DU和网络中的数据在特定条件下是无效的、不可用的.

为此，DO需先对数据加密，再将密文数据外包存储在云端. 满足数据访问和使用策略的DU可以从CSP处获得密文数据，并从DO处获得对应的、被附加了使用条件的密钥. 当密钥使用条件满足时，DU可在本地解密和使用数据；当密钥使用条件不满足时，DU端将安全删除其获得的密钥. 若DU在密钥使用条件不满足时仍需访问数据，可再次向DO申请密钥. 由于密钥只在DU端被使用，因此，一旦DU端的密钥被安全删除了，CSP，DU和网络中的密文数据均不能被解密和访问，也就实现了外包数据的确定性删除.

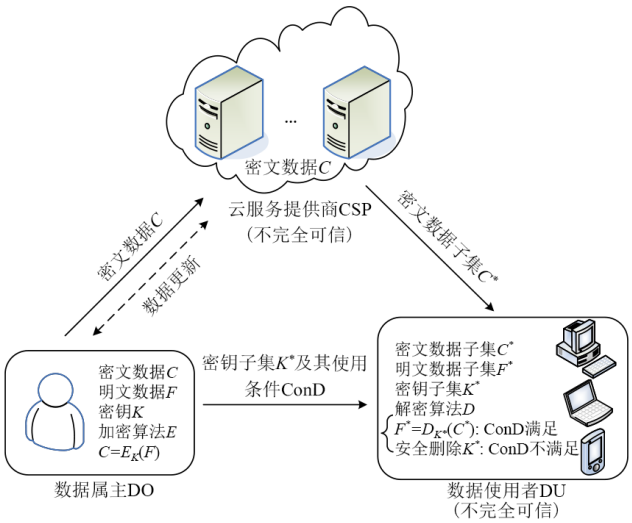


图1 云存储中外包数据确定性删除模型

2.3 关键科学问题

在图1所示的云存储中外包数据确定性删除模型中，本文认为蕴含着以下两个关键科学问题.

（1）云存储模式下因所有权与管理权（持有权）分离而引起的数据控制权问题. 在图1所示的模型中，数据被外包存储在云端，由云服务提供商CSP持有和管理，且在传输和使用过程中，还可能被网络运营商和数据使用者DU留存，但数据属主DO才是数据的所有者，仍应拥有数据的部分控制权，包括决定数据的使用粒度和更新方式、在什么条件下才能被使用等. 因此，在DO没有物理持有数据且无法完全信任CSP，DU和网络运营商的情况下，如何实现数据的安全管控，确信数据在指定条件不满足时是失效、不可用的，是第一个需要解决的科学问题.

（2）云存储模式下同时拥有持有权和使用权而引起的数据控制权问题. 在图1所示的模型中，DU分别从CSP和DO处获得密文数据和密钥，这也就意味着DU可以获得数据的完全控制权. 数据被加密后，在加密算法是安全的情况下，没有密钥将无法使用数据，数据的控制权也就难以获得，因而，密钥是关键. 因此，在DU既要使用数据（假定DU无主观上的恶意行为，如主动泄漏明文数据）但又无法被完全信任（如可能没有安全保管密钥、遭受攻击等）的情况下，如何实现密钥在DU端的安全存储和受限使用，且在使用条件不满足时将被安全删除，使得DU不能完全控制数据，是第二个需要解决的科学问题.

3 国内外研究现状与分析

根据删除对象和删除方式的不同，学术界目前对

于数据删除的研究大体上可分为两类:安全删除(secure deletion)和确定性删除(assured deletion). 安全删除是指通过物理方式直接删除数据本身(密钥也可看作是数据),并确保删除效果的安全性,即数据在物理上不存在或不能被恢复,其具体实现过程依赖底层的物理存储介质和/或文件系统. 确定性删除并不关心底层的物理存储介质和文件系统,而是在假定加密算法是安全的情况下,通过删除密钥使得密文数据不能被解密和访问,从而实现数据在计算上的删除.

需要指出的是,虽然确定性删除中的密钥删除最终还是要通过安全删除方法来实现,但它们的关注点、技术路线都不同[21],因此有必要对它们进行区分,以明确各自的功能需求和适用场景. 本文的主题和重点是外包数据的确定性删除方法,对于数据安全删除方法,由于其技术和实现的特定性,本文只是简略介绍了部分工作,以作为在实际物理存储介质和/或文件系统中删除密钥时的参考.

3.1　数据安全删除国内外研究现状

文献[26]将数据安全删除定义为当敌手具备访问系统的某些方式后仍然不能从系统中恢复被删除的数据,并详细分析了系统的不同层次如控制器层、设备驱动层、文件系统层、用户应用层等对数据删除的需求和处理方式,讨论了如何通过层与层之间的接口来实现具体物理介质上的数据删除,还根据攻击者的能力对攻击模型进行了分类,从而指导用户根据场景和需求来选择合适的删除方案. 之后,学者们针对闪存、固态硬盘、Android系统等具体的存储介质和文件系统提出了对应的数据安全删除方法.

文献[27]讨论了如何实现Android系统下的数据删除. 文献[28]和文献[29]对移动设备闪存中数据的远程擦除和安全删除问题进行了讨论. 文献[29]还在Android系统中实现了对应的原型系统. 文献[30]在文献[31]的基础上,将块擦除和页擦除结合起来以实现固态硬盘中的数据安全删除,但是在块内部进行页擦除可能导致合法数据不可访问等错误. 文献[32]提出的ErasuCrypto方案则是将块擦除方法和数据加密方法联合在一起,存储介质被划分为多个区域,处于同一组的数据将用相同的组密钥加密,在执行数据安全删除时,通过贪心算法找到无效数据和有效数据比值最高的组或块,然后擦除块或者删除组密钥. 该方案将组密钥也存储在介质中,可能导致密钥泄漏,并且由于需要解密、转移和重新加密有效数据,方案的开销也较大. 文献[33]提出了一种密钥派生加密算法来实现密钥的生成和管理、数据的加解密,该方案将块擦除和密钥删除方法结合起来支持数据的安全删除,并且通过隐式的枚举分析法来减小删除过程中的页转移和块擦除开销. 文献[34]提出了一种可屏蔽底层存储介质和文件系统的安全删除框架,可同时支持硬盘和NAND闪存上文件级的删除操作. 文献[35]基于文献[26]中的分层结构,对基于硬盘和NAND闪存的删除方法进行了层次归类和对比分析. 文献[33]、文献[36]、文献[37]分别在不同存储介质上对数据安全删除的性能优化问题进行了研究.

由于安全删除的关注点和具体实现与物理存储介质和/或文件系统紧密相关,相同的方法或技术在不同的存储介质或文件系统中的删除结果和性能开销可能会大不相同,因而安全删除方法和技术往往要基于特定的存储介质或文件系统来讨论,一般不具有通用性.

3.2　数据确定性删除国内外研究现状

数据确定性删除一般不考虑底层物理存储介质和文件系统,其方法和技术更有通用性,但一般也需要分类讨论,以便更好地展现不同方案之间的异同. 文献[11]和文献[21]对先前的研究工作进行了总结,但其分类讨论缺乏统一的视角. 在对国内外已有工作的调研、梳理和分析的基础上,本文从密钥删除过程中密钥控制方式的角度,将数据确定性删除方法分为三类:基于时间的方法、基于策略的方法和基于本地环境的方法.

对于每类方法,本文着重阐述其基本思路、共性的技术路线和还需要进一步研究解决的问题,中间辅以部分典型确定性删除方案的具体介绍,其他方案的具体细节则不详细展开.

3.2.1　基于时间的数据确定性删除方法

在基于时间的方法[8,13~18,38~41]中,密钥的控制方式被限定为使用时间,由一个第三方环境(相对于数据属主DO和云服务提供商CSP而言)在预定时间点到达后删除密钥. 这个第三方环境既可以是集中式的,也可以是分布式的. 在集中式环境中,由一个可信服务器作为第三方来管理和删除密钥,该类方案不适合大规模、用户动态变化的场景,而且需要假定第三方服务器可信,这在实际应用中有时难以保证. 因此,如何通过分布式环境来实现密钥的删除成为学者们的关注点.

Geambasu等[13]首次利用分布式哈希表(Distributed Hash Table,DHT)网络的动态变化特性,设计和实现了Vanish系统,以实现邮件及其副本的自我销毁. Vanish系统首先用对称加密算法加密数据(如邮件及其副本),然后用Shamir秘密共享算法[42]将密钥分割为密钥片段,再将这些密钥片段随机地分发到一个动态变化的DHT网络(如Vuze)中. 经过一段时间后(默认情况下为8个小时),大部分存储了密钥片段的节点会清空其存储的密钥片段或退出网络. 这时,由于不能再从该DHT网络中提取到足够的密钥片段,邮件接收方将不能重新计算原始密钥和解密邮件,邮件服务器和

网络中的邮件及其副本也就被认为是自我销毁了. 在文献[14]中,Geambasu 等进一步指出可以构建一个通用的数据自毁框架,通过抽象接口对包括 DHT 网络、Apache 服务器、WWW 网页等在内的不同动态环境进行封装,从而支持不同的密钥存储系统.

在 Vanish 系统的启发下,学者们开展了一系列的改进和完善工作[8,15~18,38~41],极大地推动了基于时间的数据确定性删除方法的研究. 这些改进和完善之处主要是在将密钥分发到 DHT 网络之前,对其进行不同的处理,包括将单一密钥扩展为多密钥、支持密钥的安全加密及动态更新等,以适用于不同的应用场景,并且防止 DHT 网络中存在的跳跃攻击和嗅探攻击. 基于 DHT 网络的确定性删除方法的系统模型如图 2 所示.

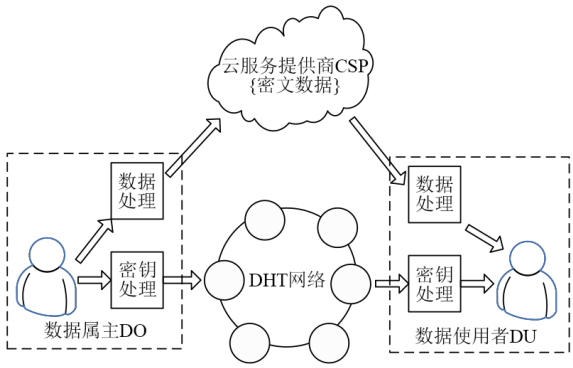


图 2 基于 DHT 网络的数据确定性删除模型

文献[8]结合密钥派生树和 DHT 网络,实现了一种适用于云存储系统的数据确定性删除方法. 其基本思想是先用哈希函数构造一棵二叉密钥派生树以实现大规模密钥的生成和管理,然后在数据块级加密数据以支持数据的细粒度访问,最后将经秘密共享方式处理后的密钥分发到 DHT 网络中,使得密钥在授权时间到达后不能被恢复,进而导致数据不能被解密. 具体而言,该方案包含了 4 个参与方,即数据属主 DO、数据使用者 DU、云服务提供商 CSP、分布式哈希表 DHT 网络以及 7 个算法,即加密密钥生成算法 DataKeyGen,加密算法 Encryption,最小树密钥集生成算法 MixTrKeySet,树密钥分发算法 TrKeyDis,树密钥提取算法 TrKeyExtract,加密密钥恢复算法 DataKeyRecover,解密算法 Decryption. 其数据访问包括了 Initial, KeyDistribution 和 DataAccess 3 个阶段,如图 3 所示.

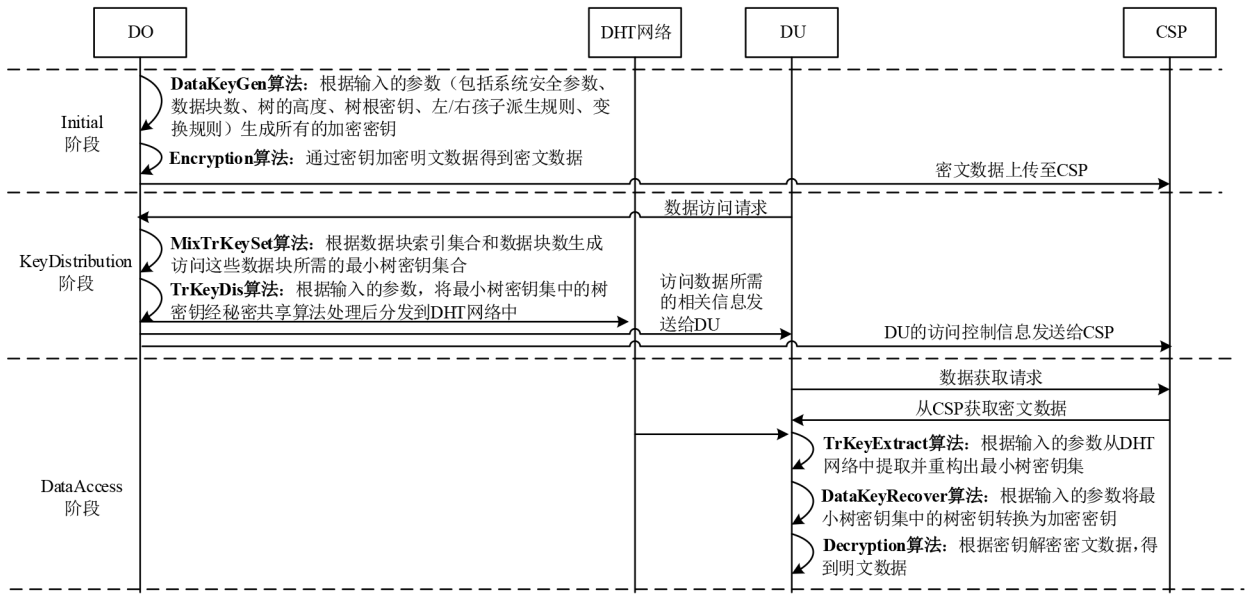


图 3 文献[8]中的数据确定性删除过程

在 Initial 阶段,DO 通过 DataKeyGen 算法生成加密密钥,并用 Encryption 算法加密数据块,将密文数据上传至 CSP. 在 KeyDistribution 阶段,DO 根据 DU 的数据访问请求,通过 MixTrKeySet 算法生成访问数据所需的最小树密钥集,通过 TrKeyDis 算法将最小的树密钥集分发到 DHT 网络中,并将访问数据所需的相关信息(如伪随机函数及其种子集合、门限信息等)发送给 DU,将 DU 的访问控制信息发送给 CSP. 在 DataAccess 阶段,DU 从 CSP 处获得密文数据块后,通过 TrKeyExtract 算法从 DHT 网络中提取并重构出最小树密钥集,再通过 DataKeyRecover 算法恢复出加密密钥,最后通过 Decryption 算法得到明文数据,从而实现数据的访问. 在此过程中,树密钥、叶子节点和加密密钥的计算都是由后台处理程序执行的,对 DU 是透明的,DU 只知道有时效限制的随机种子. 当授权时间到达后,DHT 网络的动态特性会使得网络节点中存储的密钥分片被清除. 这

时,通过随机种子定位到的网络节点上将不再存有密钥分片信息,其结果就是原始的树密钥及加密密钥将不能被重构出来.没有密钥就不能解密和访问数据,也就是数据被确定性删除了.

针对文献[8]中没有讨论数据/密钥动态更新的问题,文献[38]在其基础上进一步讨论了密钥的更新问题,以支持数据的修改、插入、删除操作,并进一步指出可以通过收敛加密来同时实现数据副本删除和数据去重.文献[15~18,39~41]也在Vanish系统的基础上,对数据确定性删除方法进行了不同应用场景下的功能需求扩展.其具体方案可被概括为对密文数据进行抽取,使得存储于云端的密文数据不再是完整的,再将抽取的密文数据与采用不同算法(包括基于属性的加密、基于身份的加密、带时间属性的策略属性加密等算法)加密后的密钥组合后再分发到DHT网络中,从而增强方案的抗攻击能力.但在上述工作中,密钥是单一的,数据是静态的,因而不能支持云存储模式下大规模数据的细粒度访问和动态操作.

在早期的基于DHT网络的方案中,数据的生命周期值是固定的.文献[43]认为数据的存活时间应当是可变的,当数据仍然受到足够关注时应当保持其可用性,反之则应当删除数据.基于该观点,文献[43]利用域名系统(Domain Name System,DNS)的缓存机制实现了一个数据删除协议和原型系统.文献[44]提出了一种云环境下的数据多副本安全共享与关联删除方案,通过引入授时中心使得DO可以自定义数据的访问期限和存储期限,并且在存储期限过期后实现副本的关联删除.

表1给出了基于时间的数据确定性删除代表性方案的对比.从表1中可以看到,基于时间的方案都对密钥进行了安全处理,但是这些方案大多只支持单一密钥,并未考虑密钥的动态更新问题.而且,这些方案的密钥使用环境都是普通执行环境(Rich Execution Environment,REE),也即通用的软硬件环境,该环境中没有一个可为敏感数据和代码提供保护且不受包括操作系统等在内的外部不可信软硬件干扰的安全区域.与REE相对应的则是可信执行环境(Trusted Execution Environment,TEE)[4,45~47].TEE利用可信平台模块(Trusted Platform Module,TPM)[45],Intel SGX[46],ARM TrustZone[47]等可信硬件技术在本地构建了一个额外的可为敏感数据和代码提供保护且不受包括操作系统等在内的外部不可信软硬件干扰的安全区域.

在大多数基于时间的方案中,密钥也不能由DU保存在本地.一旦DU能够在本地保存密钥,DO所设定的时间限制将不会起作用.这就导致DU在每次访问数据前,都需要从网络中获取密钥,使得方案的通信开销与数据访问次数成线性增长关系.此外,现有方案中的数据使用方——用户端(包括DU和DO)的安全问题没有被充分考虑,用户尤其是DU端被假定是可信的,即用户不会主动泄漏密钥,使用完密钥后会删除密钥,即使密钥是在REE中被使用的,也认为其在使用过程中是受保护的.但是,如何确保密钥在用户端的安全使用和删除却未被详细讨论.

总体而言,现有的基于时间的方案可以解决本文提出的第一个科学问题,但对本文提出的第二个科学问题却缺少应对措施.即在其他实体只有密文数据、没有密钥的情况下,基于时间的方案可以安全管控和删除数据,但在DU已经获得密文数据和密钥后,现有方案难以保证时间限制能够继续有效,使得DU不能完全控制数据.

表1　基于时间的数据确定性删除代表性方案的对比

	密钥处理方式	时间控制方式	密钥类型	密钥动态更新	DU保存密钥	密钥使用环境
文献[8]	秘密共享	DHT网络	多密钥	否	否	REE
文献[13]	秘密共享	DHT网络	单一密钥	否	否	REE
文献[15]	基于属性的加密方法	DHT网络	单一密钥	否	否	REE
文献[16]	基于身份的加密方法	DHT网络	单一密钥	否	否	REE
文献[17]	基于身份加密的定时发布加密方法	DHT网络	单一密钥	否	否	REE
文献[38]	秘密共享	DHT网络	多密钥	是	否	REE
文献[43]	门限方式	DNS服务器	单一密钥	否	否	REE
文献[44]	基于属性的加密方法	授时中心	单一密钥	否	是	REE

3.2.2　基于策略的数据确定性删除方法

在基于策略的方法[48~51]中,密钥与用户自定义的数据访问策略相关联,对策略进行操作即可控制密钥的使用与删除.如在FADE方案[48]中,每个文件都有对应的访问策略(访问策略可以是单独的策略,也可以是多个策略的布尔组合),文件由数据密钥加密,数据密钥进一步由控制密钥加密,控制密钥则与访问策略相关联.当文件的访问策略被撤销时,与之相关联的控制密钥将被删除,使得数据密钥不可用,进而导致数据不能被解密,从而实现了数据的确定性删除.具体而言,该方案包含了3个参与方:数据属主DO、密钥管理者(key manager)和存储云(storage cloud).其文件/策略的操作包括文件上传、文件下载、策略删除和策略更新,如图4所示.

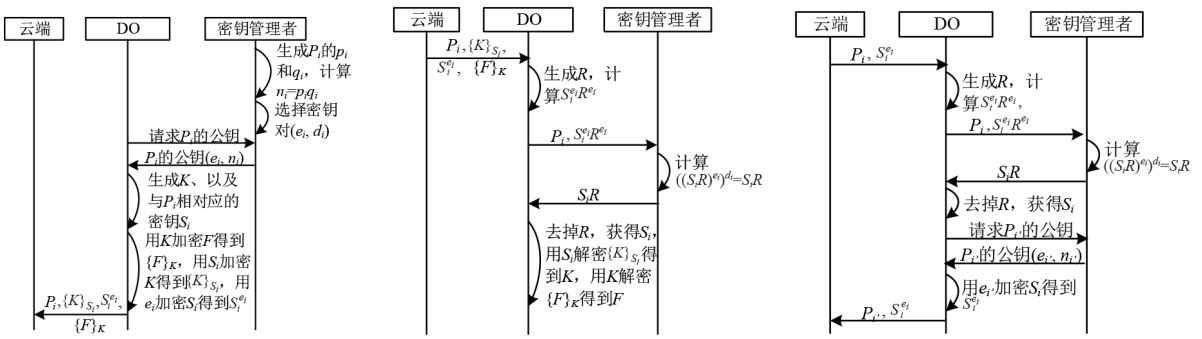
(1)文件上传. 对于每一个访问策略P_i,密钥管理者先为其生成相应的RSA大素数p_i和q_i,并计算$n_i=p_iq_i$;然后随机选取其RSA控制密钥对(e_i,d_i). DO需要加密与P_i相关联的文件F时,先向密钥管理者请求P_i的公钥(n_i,e_i);然后随机生成一个数据密钥K,以及与P_i相对应的密钥S_i;再用K加密F得到$\{F\}_K$,用S_i加密K得到$\{K\}_{S_i}$,用e_i加密S_i得到$S_i^{e_i}$;最后将P_i,$\{K\}_{S_i}$,$S_i^{e_i}$和$\{F\}_K$一起上传至云端.

(2)文件下载. 当DO需要下载文件时,首先从云端获取P_i,$\{K\}_{S_i}$,$S_i^{e_i}$和$\{F\}_K$,然后生成一个随机数R,计算$S_i^{e_i}R^{e_i}$,将$S_i^{e_i}R^{e_i}$发送给云端. 云端计算$((S_iR)^{e_i})^{d_i}=S_iR$,将结果$S_iR$返回给DO. DO去掉$R$后得到$S_i$,用$S_i$解密$\{K\}_{S_i}$得到$K$,用$K$解密$\{F\}_K$得到$F$.

(3)策略删除. 当DO要求密钥管理者撤销策略P_i时,密钥管理者将删除与P_i对应的私钥d_i以及p_i和q_i,从而使得S_i无法从$S_i^{e_i}$中被恢复出来;没有S_i也就不能从$\{K\}_{S_i}$和$\{F\}_K$中恢复出K和F了,也就实现了与P_i相关联的文件F的确定性删除.

(4)策略更新. 当需要将策略P_i更新为$P_{i'}$时,密钥管理者首先为$P_{i'}$生成新的RSA公私钥对. DO就可以用与$P_{i'}$相对应的公钥$(n_{i'},e_{i'})$对S_i重新加密,并将$P_{i'}$和$S_i^{e_{i'}}$重新上传至云端,这样就不需要从云端下载和操作密文数据了.

针对FADE方案中访问策略和控制密钥都是由密钥管理者集中管理的问题,文献[49]通过秘密共享方案对其进行了改进,将单个密钥管理者扩展为多个密钥管理者. 文献[50]通过形式化分析指出文献[48]和文献[49]的方案都存在着中间人攻击等安全问题,并提出了改进方法.



(a) 文件上传 (b) 文件下载 (c) 策略更新

图4 文献[48]中的文件/策略操作

文献[51]给出了基于策略的安全删除方法的通用形式化模型和安全定义,将数据访问策略中的属性和被保护的数据建模为有向无环策略图或策略图的组合,然后结合图论理论和秘密共享方案阐述了密钥的删除方法. 该方法可以制定和表达更为灵活的控制策略,可以满足更多的应用需求. 作者们还在Linux文件系统下实现了原型系统. 图5给出了一个包含6个属性(源节点)和6个保护类(内部节点)的策略图[51],用以建模在文件所有者、项目分类、有效期限、审计等不同属性下的保护需求. 其中,内部节点是门限节点,需要实现二进制的"AND"和"OR"操作. 对于"OR"操作,只要有一个属性满足即表示数据将不可访问;对于"AND"操作,则需要所有属性都满足时才使得数据不可访问. 如保护类p_3的支配策略是"Alice OR Exp_2015",那么只要对Alice拥有的保护类执行了删除操作或者删除了有效期限为Exp_2015的文件,p_3中的数据将变得不可访问. 在具体构造上,数据的删除最终是通过更新主密钥或者擦除部分主密钥实现的,主密钥则是保存在可擦除内存中的. 但是,该方案没有指明谁来执行该操作,如果执行者没有如实更新或擦除主密钥,那么该方案将不能实现预期目标.

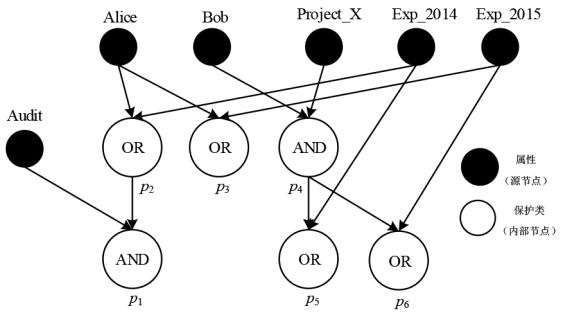


图5 文献[51]中的策略图

在基于策略的方法中,密钥的使用与删除与数据访问策略相关. 因此,数据访问策略由谁来管理和执行至关重要. 基于属性的加密算法(Attribute-Based Encryption,ABE)可以将属性与密文、访问策略与密钥相关联,使得满足特定属性的用户都能获得密钥并访问数据,从而实现云计算环境下多用户对数据的共享和

使用,并且在不依赖第三方的情况下,访问策略也能被正确执行. 因此,一些工作[15,18,40,45,52~59]在设计方案时引入了ABE方法来管理访问权限和分发密钥,并通过属性的撤销或更新来实现密文数据的不可访问,从而避免第三方密钥管理者集中管理策略时存在的单点失效和不可信问题. 此外,也有学者将基于身份的加密算法引入到方案[60]中以支持轻量级的密钥管理、密钥协商和撤销.

文献[52]将确定性删除的应用场景由云存储模式扩展到了雾计算、物联网场景下,以满足更多的工业应用需求. 在该方法中,智能终端(如传感器、嵌入式设备等)产生数据,并且用对称加密算法加密数据,然后用基于密文策略的属性基加密算法(Ciphertext-Policy ABE,CP-ABE)来加密数据密钥,每个用户的属性集里还额外包含了一个用于删除的虚假属性. 加密后的数据和密钥被上传给雾设备,雾设备在本地保存密钥的密文,将密文数据上传给云服务器. 删除数据时,雾设备首先与智能终端通过密钥交换协议生成一个删除密钥,然后再利用该删除密钥来改变与虚假属性相关的密钥密文,从而使密钥密文的访问控制结构发生改变. 其结果就是用户将不能再继续访问数据. 此外,智能终端也可根据删除密钥来验证删除的结果,即雾设备是否真的更新了密钥密文,使得数据确实不可访问. 但是,该方案引入虚假属性会带来额外的开销,每次也只能删除一个文件,而且需要一个额外的可信第三方.

文献[53]则是利用基于密钥策略的属性基加密算法(Key-Policy ABE,KP-ABE)来保护数据,要删除数据时,由DO指定要修改的属性,云服务器更新属性所对应的密文,并生成一个基于MHT(Merkle Hash Tree)的删除证据返回给DO. 其方案也需要一个可信第三方来生成删除(重加密)密钥. 文献[54]在改进的基于策略的可穿透加密的基础上,提出了一种细粒度自我控制的外包数据删除方案,该方案利用了可穿透策略与访问策略的逻辑联系,设计了从可穿透策略到访问策略的转换方法,再利用ABE方法中的密钥协商技术实现密钥的更新操作,通过密钥的更新使得密文数据不能再被解密,从而实现了数据的删除. 文献[55]提出了一种基于CP-ABE和线性秘密共享的数据删除方案,该方案利用策略图来描述用户、策略、属性和文件之间的关系,并且认为当一个文件对于所有用户都是未授权时,该文件对于CP-ABE方案而言是被删除了的. 该方案通过选择密钥属性和更新相关的密文以确保所有用户不被授权,并由第三方通过MHT来更新密文,从而实现数据的可验证删除.

基于ABE的方法可以实现灵活的策略表达,也可以通过删除密钥、撤销属性、更新密文或密钥等多种方式实现数据的确定性删除,是当前研究的热点. 但是,由于需要存储加密后的密钥,基于ABE的方法会引起较大的存储开销,其加密/解密的计算开销也高于只使用对称加密算法的方案. 而且,其中属性的撤销与更新[52~54,58,61]、策略的表达粒度和组合能力等都是需要深入研究的问题. 此外,在大部分基于ABE的工作中,密钥同样不能由DU保存在本地,否则ABE算法中对于DU属性的限制条件将失效,DU就能不受限制地使用密钥和数据. 如在文献[52]和文献[55]中,密钥的密文分别保存在雾设备和云服务器中,这就导致DU在每次访问数据时都需要通过网络从雾设备或云服务器处获取加密后的密钥,使得方案的计算和通信开销也是与数据访问次数成线性关系.

表2给出了基于策略的数据确定性删除代表性方案的对比. 从表2中可以看到,在基于策略的方法中,密钥一般也是单一的、静态的,而且是在普通环境REE中使用的,即数据加密密钥在用户端的安全使用和删除问题同样没有被详细讨论. 此外,不同于基于时间的方法中明确将密钥的使用期限作为密钥的删除条件,当前基于策略的工作对于密钥删除条件(即密钥在什么条件下应当被删除)的讨论较为模糊,通常情况下认为当用户不满足指定策略时就不能获得数据解密密钥,也就不能访问数据,但没有显式地定义什么条件下用户的属性需要与访问控制策略不匹配.

表2　基于策略的数据确定性删除代表性方案的对比

	策略表达方式	密钥/数据存储及删除方式	删除条件	第三方	密钥类型	密钥使用环境
文献[45]	CP-ABE	(DU端)可信应用删除密钥	使用次数	不需要	静态、单一	TEE
文献[48]	策略及其布尔组合	(密钥管理者)撤销文件的访问策略	未明确	需要	静态、单一	REE
文献[51]	有向无环策略图或策略图的组合	删除源节点触发门限节点的布尔值为真	可自定义	不需要	静态、一个属性对应一个密钥	REE
文献[52]	CP-ABE	(雾设备)更新密钥的密文	未明确	需要	静态、单一	REE
文献[53]	KP-ABE	(云服务器)更新数据的密文	未明确	需要	静态、单一	REE
文献[54]	可穿透策略+KP-ABE	(DO)更新密钥	未明确	不需要	静态、单一	REE
文献[55]	CP-ABE	(云服务器)更新密钥的密文	未明确	需要	静态、单一	REE

总体而言,现有的基于策略的方案可以解决本文提出的第一个科学问题,但是很少明确讨论本文提出的第二个科学问题,即在其他实体只有密文数据、没有密钥的情况下,现有方案可以安全管控和删除数据. 但是,在DU已经获得密文数据和密钥后,如何在DU端限制密钥的使用,并且保证限定条件能够继续有效,使得DU不能完全控制数据,现有工作大多没有明确讨论.

3.2.3 基于本地环境的数据确定性删除方法

基于本地环境的方法[45,62~73]侧重于讨论如何在本地环境中实现密钥的删除. 这里的本地环境既可以是DU端的环境,也可以是DO端的环境. 此时,密钥直接在DO和DU之间传递,或者DO就是数据的使用者. 本地环境既可以是普通执行环境REE,也可以是可信执行环境TEE.

REE下的确定性删除是指在通用软硬件环境下研究密钥的安全删除问题,这方面的代表性工作包括文献[62~65]. 文献[62]基于图论对密钥管理过程进行了建模,指出不同方案在该模型下具有内在统一性,然后通过密钥暴露图对攻击者的能力进行建模,当表示密钥的节点在图中不可达时即意味着密钥被安全删除了,该工作还讨论了基于B-Tree的密钥更新问题,但密钥的使用条件只被限定为时间. 文献[64]设计了递归加密的红黑密钥树以支持多密钥的更新和删除,但只讨论了红黑密钥树上的插入和删除操作,没有涉及节点的修改操作. 文献[65]基于密钥模函数和模树讨论了多密钥的更新和删除,使得在更新过程中无关节点不受影响. 文献[66]提出了一种基于比特流变换的方案,将文件分为固定大小的块,对每个块进行比特流变换,原始数据加密后上传给云,比特流数据和密钥则由DO保存在本地,DO在本地删除比特流数据和密钥即可实现数据的确定性删除.

TEE下的确定性删除首先利用可信硬件技术在本地构建TEE,然后再在TEE中研究密钥的安全使用与删除,其框架一般如图6所示. 密钥管理模块和加解密引擎位于TEE中,分别负责密钥的加解密(密封和解密封)和数据的加解密,在访问完数据后且密钥删除条件不满足时,继续将数据和密钥以密文形式存储在不可信的文件系统中. 当密钥不满足使用条件时,密钥管理模块将安全地删除密钥,确保密文数据不能被解密.

基于TEE的确定性删除代表性工作包括文献[45,67~73]. 文献[45]利用可信平台模块TPM的安全存储功能和单调计数器功能实现了一种在DU端限制密钥使用次数的方法. 此外,该方法还利用CP-ABE算法将密钥加密后存储在云端,使得多用户可以共享访问数据. 具体而言,其系统架构如图7所示,包含了4个参与方:数据属主DO、数据使用者DU、云服务提供商

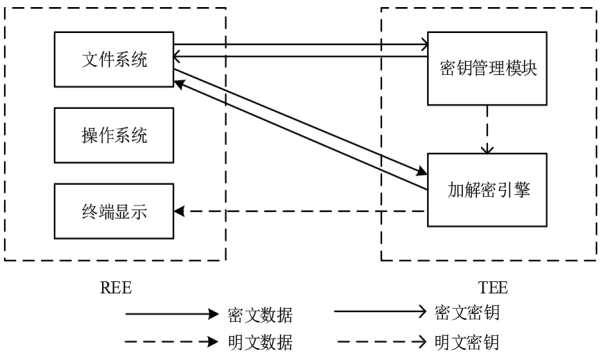


图6 基于TEE的数据确定性删除框架

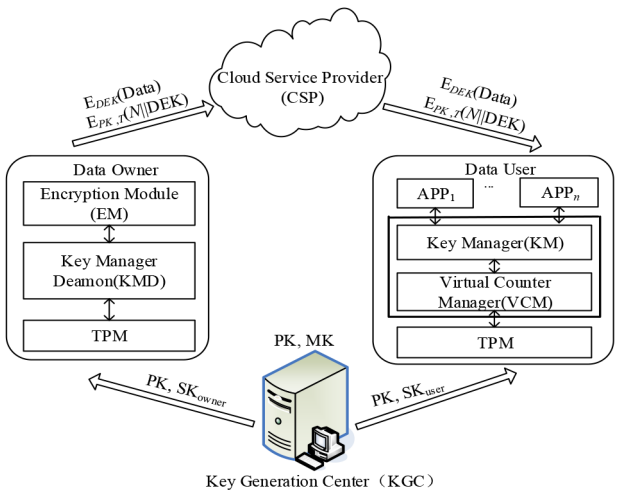


图7 文献[45]的系统架构图

CSP、密钥生成中心KGC. 其数据确定性删除的过程如下:DO先用对称密码算法对数据进行加密,并为数据加密密钥(Data Encryption Key,DEK)指定最大使用次数N. 然后,DO用CP-ABE算法对N和DEK加密,并将加密后的数据和密钥一起上传给CSP. 满足特定属性的DU从CSP处获取到密文数据和密钥后,DU端的可信密钥管理模块(Key Manager,KM)用DU的私钥恢复出DEK,并与虚拟计数器管理器模块(Virtual Counter Manager,VCM)交互为该DEK生成一个初始值为0的可信虚拟单调计数器V_Counter,然后执行密钥存储算法(见算法1)在本地存储密钥. 当DU端的应用程序要访问数据时,KM会执行密钥加载算法(见算法2)加载密钥. 在此过程中,若KM发现该DEK已使用的次数超过了其最大使用次数,即VC>N时,就销毁该DEK,使得数据不可访问,也就实现了密钥的使用次数受限和数据的确定性删除. 若密钥加载成功,KM就用DEK解密数据,将明文数据返回给应用程序,并且再次执行密钥存储算法,将密钥安全存储在本地. 此外,文献[67]利用TPM和虚拟化技术,设计并实现了Dissolver系统,使得当指定时间点到达时,密钥管理模块将强制销毁只出现在受保护空间中的用户数据及密钥. 文献[68~71]

算法1　文献[45]中的密钥存储算法

1. KM调用VCM的increment命令使V_Counter的值VC加1,并获取增加后的VC值.
2. KM计算N,VC和DEK的哈希摘要值digest,即digest=hash(N||DEK||VC).
3. KM将N||DEK||digest加载到TPM内部,TPM用绑定密钥User_BindKey加密N||DEK||digest,即encBlock=$E_{User\_BindKey\_pub}$(N||DEK||digest),并将encBlock返回给KM.
4. KM将digest,encBlock保存到本地文件系统中.

算法2　文献[45]中的密钥加载算法

1. KM从本地文件系统中读取密钥文件列表,获取密钥的摘要值和加密后的密钥,即digest和encBlock.
2. KM将encBlock加载到TPM内部,TPM用绑定密钥User_BindKey的私钥解密encBlock得到N,DEK和digest,将N,DEK和digest返回给KM.
3. KM向VCM请求与DEK相绑定的V_counter的当前值VC.若VC>N,则将该DEK销毁,返回密钥加载失败信息;否则,执行4.
4. KM计算N,VC和DEK的哈希摘要值digest′=hash(N||DEK||VC),判断digest′与digest是否相同.若相同,则返回密钥加载成功信息;否则,返回密钥加载失败信息.

在不同场景下讨论了基于TPM的密钥安全删除方法.其中,文献[68]利用了TPM和CPU的可信执行模式提供的安全存储功能,通过特殊的删除口令删除密钥,并可对删除结果进行验证.文献[69]将数据加解密和密钥删除放置在TPM内部进行,在不访问TPM内部代码的情况下,能够验证加解密和删除操作的正确性.文献[70]和文献[71]认为硬件安全模块或者TPM都可以用来作为安全删除加密密钥的可信存储空间,任何在该存储空间里被删除的数据都被认为是不可恢复的,即使攻击者能够访问存储介质也不能重构出密钥.文献[70]和文献[71]还设计和实现了基于TPM的多密钥保护和删除方法.文献[72]和文献[73]在云存储模式下分别利用SGX及其单调计数器功能和TrustZone及其安全文件,设计并实现了可以限定密钥使用次数的方法.

云存储中的数据具有数据量大、多用户共享、细粒度访问和操作、动态更新等需求,其对应的密钥也应是大规模的且可动态更新的.但在上述工作中,文献[45,67,69,72,73]讨论的都是静态单一密钥;文献[62,64]支持多密钥的生成及其动态更新,但在更新密钥时,相关路径上的密钥会被解密后再重新加密,该过程会引起较多的计算和通信开销;文献[62,64,65]的多密钥生成策略不够良好,不能很好地满足云存储模式下多用户共享使用数据的要求.此外,由于侧重于密钥的生成、更新和可信的删除环境的构建(非物理存储介质和文件系统层面的删除环境),此类方法对于密钥的其他方面有时会缺乏考虑,如文献[64,65,69,71]均没有讨论密钥的控制使用条件,即密钥在什么条件下才能被使用、才应当被删除.表3给出了基于本地环境的数据确定性删除代表性方案的对比.

表3　基于本地环境的数据确定性删除代表性方案的对比

	密钥使用环境	密钥删除条件	密钥类型	密钥生成策略/组织方式	密钥动态更新
文献[45]	TEE(TPM)	使用次数	单一密钥	无	否
文献[62]	REE	使用时间	多密钥	B-Tree	是
文献[64]	REE	未明确	多密钥	递归加密的红黑密钥树	是
文献[65]	REE	未明确	多密钥	基于模函数和模树的派生树	是
文献[67]	TEE(TPM)	使用时间	单一密钥	无	否
文献[68]	TEE(TPM+CPU)	未明确	单一密钥	无	否
文献[69]	TEE(TPM)	未明确	单一密钥	无	否
文献[71]	TEE(TPM)	未明确	多密钥	级联加密树	是
文献[72]	TEE (SGX)	使用次数	单一密钥	无	否
文献[73]	TEE (TrustZone)	使用次数	单一密钥	无	否

总体而言,基于本地环境的确定性删除方法可以解决本文提出的第一个科学问题,而且也可以解决本文提出的第二个科学问题.即在DU已经获得密文数据和密钥后,可以将密钥的管理和删除放置于TEE中,由TEE为密钥的存储及其处理提供高安全的隔离环境,确保DU无法接触到密钥和获得数据的控制权.这就要求此类方法需要进一步研究SGX,TrustZone等新型可信硬件上的TEE构建方法和删除证据生成方法,支持多用户、多样化的密钥使用条件,并确保使用条件的有效性、证据的不可伪造、否认和可验证性[69].而且,在设计具体方案时,还需要考虑如何通过良好的生成策略来支持多密钥的高效生成和有效组织,以及高效的密钥动态更新.

3.2.4　数据确定性删除结果的可验证性

在以往的国内外工作中,密钥处理方式,策略表达方式,密钥删除方式和条件,密钥类型、生成策略、动态更新、使用环境等是研究的关注点.但是,近年来一些研究者认为删除结果的验证问题也应当被重视,即要

求能够对删除结果进行显式地验证,以对执行删除操作的主体进行约束和追责.因此,本节将可验证性[12,52,53,55,69,74~83]作为数据确定性删除的一种需求和目标进行单独描述.

文献[69]指出现有确定性删除方案均为黑盒方案,即用户只能相信返回的结果却难以验证结果的真实性,从而提出了"信任但验证"的假设,使得用户能够验证加密和删除操作的正确性,并且删除密钥时还需生成一个签名作为承诺.该签名实际上就是密钥删除的公开证据,如果后面发现密钥被恢复了,则可基于该证据对执行删除操作的主体进行追责.文献[53,55,74,78,83]都采用了MHT来对删除结果进行验证.其中,文献[74]设计了基于秩的MHT来验证删除结果,但文献[75]指出该方案存在着安全漏洞,进而提出了相应的解决方法.文献[78]针对多副本关联的场景,利用预删除序列和MHT来实现数据完整性和删除的可验证,但是其方案需要一个可信第三方来管理数据密钥.为此,文献[79]采用向量承诺设计了一种支持数据迁移和删除的方案,使得在无需可信第三方的情况下,数据属主可以对数据迁移和删除结果进行验证.文献[80]基于可计数的布隆过滤器提出了一种可支持云间数据安全迁移和删除的方案,在无需可信第三方的情况下,可在数据被删除后,生成可被公开验证的证据,使得数据属主可以对删除结果进行验证.文献[81]则是通过挑战-应答协议的响应时间来对删除结果的正确性进行验证.文献[82]提出了将删除证据放置到区块链上的方案,任何验证者都可以通过对删除证据的验证来核实删除结果.

在基于时间的数据确定性删除方法中,虽然密钥删除的结果是被证实了的,但这种验证是基于试验和统计分析的,即在真实的DHT网络中进行密钥分发和提取试验,统计并分析网络中的密钥片段随时间的变化关系,进而确信确实不能从网络中恢复出完整的密钥.这种验证难以定位责任主体并证实其行为,即密钥片段的不可用是由于DHT网络整体上的动态变化而导致的,而不是网络节点主动执行了密钥片段删除操作.

学者们近来对于外包数据确定性删除结果可验证性的研究则是希望能够证实删除主体与删除对象之间具有唯一且可追溯的关系,其实质是要求执行密钥删除操作的责任主体对其密钥删除行为做出可验证的公开承诺,从而对其进行约束和追责.这就要求承诺的主体和删除对象都必须是唯一的、不可伪造的、不可否认的,且两者的关联关系容易被追溯和验证.

3.2.5 数据确定性删除方法小结

前文根据密钥删除过程中密钥控制方式的不同,从三个方面梳理和分析了外包数据确定性删除国内外当前的研究现状,并单独讨论了删除结果的可验证性.但是,上述分类并不是完全相互独立的,它们之间也存在着交叉和重叠.如文献[62~65]虽被放在第三类方法中描述,但其中的多密钥管理和更新却是每类方法都应研究和解决的问题;文献[15~18]被归在第一类方法中,但其中的密钥访问控制却是第二类方法的关注点.去掉前面两类方法中用户尤其是数据使用者DU是可信的假设,就需要具体考虑密钥在本地的安全使用和删除问题.此外,文献[52,53,55]的密钥删除过程属于第二类方法,文献[68,69]的密钥删除过程属于第三类方法,但它们也都分析和讨论了删除结果的可验证性问题.

表4对这三类方法进行了综合分析.从前文的分析和表4可知,外包数据确定性删除当前的研究工作主要存在以下不足:(1)多数方案考虑的是静态单一密钥场景,即用一个密钥加密全部数据,数据也是静态的,不涉及插入、删除、修改等动态操作,不能很好地满足云存储模式下大规模数据的细粒度访问及动态操作需求;(2)多数方案未能提供良好的密钥生成和分发策略,对云存储模式下多用户共享数据特性的支持度不高,也即密钥的使用是一次性的,删除密钥将使得其他用户也不能访问数据,或者DO需要为同一份数据维护多个不同的密钥;(3)多数方案在设定密钥使用条件时,只考虑了时间因素或者没有明确定义密钥删除条件,功能较为单一,适用场景有限,而且密钥删除是由第三方环境实现的,时间粒度依赖于第三方环境自身的特性,且难以确认密钥删除的责任主体.

表4 数据确定性删除方法的综合分析

	密钥类型	密钥生成策略	密钥/数据动态更新	密钥分发方式	密钥使用条件	密钥删除方式	结果可验证性
第一类方法	多数为单一密钥	较少考虑	较少考虑	分割或加密后分发到第三方环境	时间	第三方环境的动态变化导致密钥不可用	基于试验验证
第二类方法	多数为单一密钥	较少考虑	较少考虑	加密后分发,可多用户共享	DU需满足指定的条件	属性或策略的撤销或更新导致密钥不可用	有方案采用密码学方法
第三类方法	有单一密钥,也有多密钥	多密钥方案有考虑	多密钥方案有考虑	可在DO和DU间点到点分发	较少考虑	密钥更新或在TEE中删除密钥	有方案采用密码学方法

4　应用案例

为数据的访问和使用加上控制条件(如使用期限、使用次数等),在实际生产生活中有着广泛的需求和应用,如很多应用都支持通过邮箱或短信验证码找回或重置用户口令,但是后台服务器一般都会为返回给用户的随机链接或验证码设置一个有效时间段. 在有效时间段内,点击链接或输入验证码能够找回或重置口令;超过了有效时间,链接或验证码就会失效. 此外,还有很多应用通过限制口令输入的次数来保护用户的资产和信息,如在银行的取款机上若连续3次输入错误的口令,银行卡就会被吞掉;很多应用在连续输入几次错误的口令后会锁定用户账户等.

这些实际应用中的保护措施与云存储中外包数据确定性删除想要实现的目标和效果是类似的. 但在上述场景中,数据访问控制条件设置和执行的主体都是可信的后台服务器,其技术实现也相对较为灵活简单. 在云存储等外包模式下,数据属主DO是数据的所有者,云服务提供商CSP是数据的管理者,数据使用者DU是数据的使用方. 在这种架构下,数据访问控制条件的设置主体应当是DO,数据访问控制条件的执行主体应为CSP和DU. 但是,CSP和DU对于DO而言,都不是完全可信的,它们不一定会如实地执行DO所设置的数据访问控制条件,这就使得外包数据确定性删除的技术实现变得较为复杂.

作为保护数据机密性和隐私性的一个支撑技术,外包数据确定性删除尚未作为一个单独的应用或产品出现,但已经有一些应用/系统实现了确定性删除所预想的部分功能,即俗称的"阅后即焚"功能. 本节将选取几个代表性的应用/系统作为应用案例来直观地展示这些功能,以帮助读者更好地理解外包数据确定性删除的含义和以期达到的效果.

4.1　App应用

4.1.1　Snapchat

Snapchat是一款"阅后即焚"照片分享应用,利用该应用程序,用户可以拍照、录制视频、添加文字和图画,并将它们发送给自己在该应用上的好友. 这些照片及视频被称为"快照"("Snaps"),所有快照的有效期为1~10 s. 在用户将快照发送给好友后,这些快照会在用户设定的有效期到达后被销毁. 若接收方在此期间试图对快照进行截图,发送方将会得到通知.

对Snapchat数据包进行分析后,会发现其数据包都是安全套接字层(Secure Sockets Layer,SSL)包,无法被解密. 也就是说,Snapchat使用的是https协议,以保证通信过程中数据的安全. 而且,Snapchat还对https代码进行了处理,以避免其证书被劫持.

Snapchat接收快照的过程大致如下:(1)Snapchat获取到有未读取消息后,在对话框里显示"点击加载",提示用户有新消息可读;(2)用户点击"点击加载"后,将生成一个统一资源定位符(Uniform Resource Locator,URL),从该URL下载对应的文件到本地内存中;(3)在用户本地内存中用预先设置的密钥对文件进行AES解密,使得内存中的文件以明文形式存在;(4)在用户本地生成一个随机密钥,重新对内存中的明文进行AES加密,并将密文和该随机密钥都保存为文件,此时,"点击加载"按钮将变成"按住查看",在用户看完照片之前,该URL都是有效的;(5)用户点击"按住查看"后,Snapchat使用保存的随机密钥对文件进行解密,并显示出来,然后删除本地文件,并向服务端发送该快照的编号,通知服务器该文件已读,服务器就会使先前的URL失效,同时通知发送者,此消息已读.

4.1.2　钉钉

钉钉是一款成熟的商业办公聊天软件,其提供的密聊功能,也能够实现消息的"阅后即焚".

用户在聊天输入框中输入字符串"***"后,就会进入"密聊"模式. 此时,双方的头像都会被打上马赛克,昵称也会被隐藏;发送给对方的消息,会在对方打开聊天界面后,进入倒计时(当前默认的时间为30 s),计时结束后,消息即被删除. 在整个密聊过程中,消息是不能被拷贝或转发的,聊天界面也不能被截屏和录屏.

4.1.3　支付宝

支付宝的聊天功能中提供的"悄悄话"选项,也能实现"阅后即焚"功能.

点击支付宝的聊天输入框右边的"+"号,在展开的菜单里面就有"悄悄话"选项. 开启"悄悄话"后,发送给对方的消息会带上一把锁,发送方看到的锁是打开的状态,而接收方看到的消息是锁住的,需要点开才能查看消息内容. 接收方点开锁住的消息后,系统就会进入倒计时(当前默认的时间为10 s),计时结束后,系统将删除消息. 但是,与钉钉不同的是,在"悄悄话"过程中,聊天界面能被截屏和录屏,这降低了其安全性.

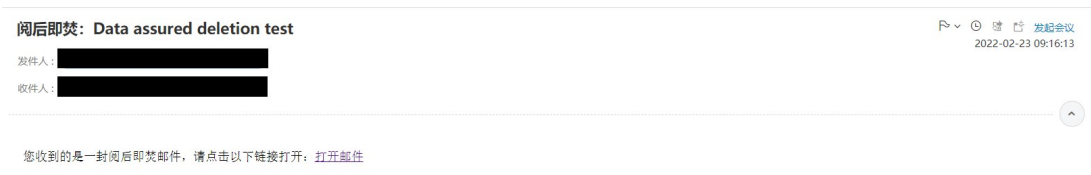
4.2　邮箱应用

笔者单位内部自行研制和部署的邮件系统也提供了"阅后即焚"功能. 而且,与上述App应用中只支持限时看消息不同,该邮件系统同时支持限定邮件的阅读次数和阅读期限,并且可由用户自行设定接收方对邮件的阅读次数和阅读期限. 当前的限定阅读次数为1~99次,邮件有效期则为所设日期当天的24时.

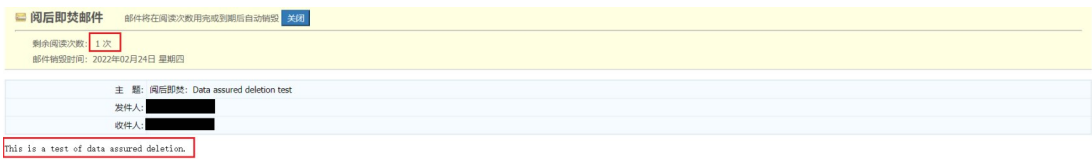
在发送邮件时,用户可启用"阅后即焚"功能,并且可以指定对方的阅读次数和阅读期限,系统会获取当前系统时间,并将邮件的有效期设置为所设日期当天的24时,如图8(a)所示. 在指定的阅读次数用完或邮

(*a*) 开启"阅后即焚"功能

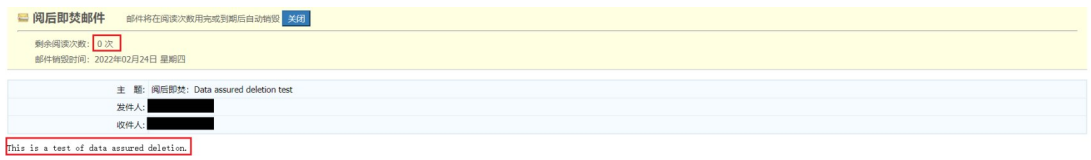
(*b*) 发送方查看邮件的状态



(*c*) 接收方收到"阅后即焚"邮件



(*d*) 接收方第1次阅读邮件



(*e*) 接收方第2次阅读邮件

(*f*) 发送方显示邮件已被销毁

(*g*) 接收方无法继续阅读邮件

图8 某邮件系统的"阅后即焚"功能

件到期后,系统将会自动销毁邮件. 发送方还可以随时查看接收方对邮件的操作情况,如邮件被对方阅读了几次、是否被销毁了,如图8(*b*)所示.

接收方收到邮件后,系统会提示该邮件是一封阅后即焚邮件,并生成一个新的URL链接,而不是直接显示邮件内容,如图8(*c*)所示. 接收方需要点击该URL链接,并输入邮箱的登录口令,才能看到邮件内容. 此时,剩余的阅读次数会自动减1,如图8(*d*)和图8(*e*)所示. 当剩余阅读次数为0时,系统将会销毁邮件内容,并将信息返回给发送方,如图8(*f*)所示. 此时,虽然接收方可以继续通过先前的URL打开邮件,但已经看不到邮件的内容了,如图8(*g*)所示.

图8只展示了该邮件系统如何限定接收方对邮件的阅读次数. 该系统在限定邮件的有效期时,其过程与上述过程是类似的,即在有效期内接收方可以任意阅读邮件内容(前提是限定的阅读次数没有用完,可以通过设置较大的阅读次数来实现),在超过限定时间后,邮件内容也将被服务器销毁.

4.3　应用案例小结

上述的应用案例虽然都提供了“阅后即焚”功能,但是其应用场景和模式还都较为简单,如Snapchat、钉钉、支付宝都只能限定消息的有效期,而且钉钉、支付宝中的消息有效期还不是由用户根据需要自行设定的,而是由系统统一设定的. 这就会导致所有消息的有效期都是一样的,就可能存在着有的长消息还没被用户看完或者理解清楚,就被删除了,给用户造成不便.

笔者所在单位的邮件系统虽然同时考虑了邮件的阅读次数和有效期,并且也支持用户自定义阅读次数和有效期,但是其对邮件阅读条件的控制粒度也较粗. 在用户开启“阅后即焚”功能后,邮件的阅读次数和有效期会被同时开启,而不是由用户根据需要进行选择. 此时,邮件的阅读次数和有效期是“或”的关系,即任一条件成立时,邮件都将被自动销毁. 这就可能导致当发送方的限定时间设置不当时,接收方还没来得及查看邮件,邮件就已经被销毁了. 而且,该系统目前也只对内部用户提供了“阅后即焚”功能,即只有都是该域名的邮箱用户才能正常查看“阅后即焚”邮件. 该系统也可以给其他域名的邮箱用户发送“阅后即焚”邮件,其他域名的邮箱用户虽然也能收到该邮件,但因为不是域内用户,将无法打开图8(*c*)所示的URL链接,也就无法查看邮件内容.

此外,上述邮件系统不能同时开启“阅后即焚”和“邮件加密”功能,也就是说,“阅后即焚”邮件都是未经加密就发给接收方的,这与外包数据确定性删除的场景还有些差异. 在数据外包情况下,数据是以密文形式在网络上传输的,在服务器端及数据使用者端也是以密文形式存储的. 在Snapchat中,文件虽然是加密后传到用户本地的,但在用户本地内存中是以明文形式存在的,没有进一步的保护措施,如果攻击者能够在这一步中获取到内存的内容,就可以保存快照了,使得其“阅后即焚”失效.

通过对上述几个有着众多用户的实际应用/系统的分析可知,当前的“阅后即焚”应用虽然实现了外包数据确定性删除所预想的部分功能,但仍有很多工作需要继续开展. 在上述的应用案例中,消息/邮件的接收方被认为是不可信的,但是服务器似乎被认为是可信的. 因此,数据访问控制条件的设置和执行主体很多时候都还是服务器,即由存储着数据的服务器来计时/计数,进而删除消息/邮件. 这时就无法解决数据使用者DU端也可能存有数据时的数据删除问题.

因此,后续的工作应是将数据访问控制条件的设置权限交回给数据属主DO,由DO根据需要自行设定数据访问的控制条件,然后在不完全可信的服务器和DU端构建可信的执行环境TEE,使得控制条件能够在TEE中得以如实如期地执行,并在条件不满足时,在TEE中安全地删除密钥,使得留存于服务器和DU端的数据不可用. 当然,作为实际的应用/系统,还有很多其他的因素也需要加以考虑,如防止对应用/系统的截屏和录屏等,这在外包数据确定性删除的学术研究中很少被考虑.

5　研究展望

在图1所示的模型中,本文将外包密文数据的访问流程分为3个阶段:密钥生成、密钥分发、密钥使用与删除. 因此,本文认为外包数据确定性删除未来的研究也可以围绕着这3个阶段来展开,即在密钥生成阶段研究动态场景下的大规模密钥生成与更新方法,在密钥分发阶段研究密钥及其使用条件的安全分发方法,在密钥使用与删除阶段研究密钥的受限使用与可验证的删除方法.

5.1　动态场景下的多密钥生成与更新

加密数据时离不开密钥的生成和管理. 因此,在密钥生成方法中,首先需要研究如何为众多数据块生成互不相同、但又存在联系的密钥,并对这些数量众多的密钥进行有效、高效的组织和管理,以支持大规模数据的细粒度访问;其次,要研究如何确保密钥生成是可重复的、结果是确定的,以支持多用户共享使用数据,避免密钥使用的“一次性”,即在数据生命周期内,一个DU端将密钥删除了,导致其他DU不能访问数据,或者DO要为同一数据生成和维护多个密钥;最后,要研究如何在密钥生成方式中支持密钥的原地动态更新,以支持数据的动态操作,原地动态更新要求密钥更新不会影响到无关密钥,也不会改变密钥的整体结构,从而避免对无关数据块的解密和重新加密而引起额外的通

信和计算开销.

5.2 密钥及其使用条件的安全分发

DU端是密钥最有可能被泄漏的地方. 因此,DO在将密钥分发给DU之前,首先应确信DU端具有所需的安全环境. 由于DU可通过不同的终端设备(如PC、移动终端等)来访问数据,为此,需要研究不同终端设备的可信执行环境TEE构建和远程证明方法. 其次,在分发的过程中,应只分发访问数据所需的最小密钥集合[84](若多个互不相同的密钥均由同一个密钥生成,则该密钥即为这多个密钥的最小密钥集合),以节省通信开销,且避免无关密钥也被分发. 为此,还需要针对多密钥研究最小密钥集合的高效判定和生成方法. 最后,DU在获得密钥后不能无限制地使用密钥. 因此,在分发密钥的同时,还应当指定密钥的使用条件(如密钥的使用期限、使用次数、使用环境等),并根据终端设备的硬件特性来设计密钥及其使用条件的安全迁移方法.

5.3 密钥的受限使用与可验证的删除

在DU端对密钥进行安全管控,是确保外包数据安全的关键. 因此,首先要研究如何基于可信执行环境TEE实现密钥的状态追踪,以根据当前的使用状态和预定的使用条件来判断密钥是否可用,确保密钥始终按照条件被使用. 这就要求密钥的状态必须是当前的,不能被回滚、重放,使用条件在该过程中不能被篡改、绕过. 其次,当密钥使用完后且还不应被删除时,密钥及其使用条件应被安全存储在不可信的普通执行环境REE中. 为此,需要研究TEE与REE之间的状态切换及数据传输方法,以确保静态存储的密钥不会被泄漏、使用条件不会被篡改. 最后,要研究密钥的可验证删除方法,即当使用条件不满足时,在TEE中安全地删除密钥,并生成唯一的、不可伪造和否认的、可验证的证据,在发现密钥泄漏后,DO可基于该证据对责任主体进行追责.

6 结束语

作为云存储中数据安全的一个组成部分和数据生命周期的最后一个阶段,外包数据确定性删除研究的是外包数据在“不应用”时的“不可用”问题,即当数据生命周期到达或不满足使用条件应被删除时,如何证实脱离了数据属主物理控制的数据在云服务提供商处、数据使用者端及网络中是失效、不可恢复的,从而防止数据滥用以及隐私泄露等安全隐患.

本文对云存储中外包数据确定性删除问题进行了分析和综述. 本文首先阐述了该问题的研究背景及主要研究思路,即利用密码学相关理论和技术将外包数据删除问题转换为密钥的安全管控和删除问题;根据该研究思路,提出了云存储中外包数据确定性删除的模型,以及其中所蕴含的关键科学问题;然后,对国内外现有工作进行了分类梳理,分析了每类方法的特点和发展趋势,并且综合对比归纳了该领域当前还需要解决的科学及技术问题;接着,以几个有着众多用户的应用/系统作为案例,展示了外包数据确定性删除所预想的部分功能;最后,依据该问题的模型和外包密文数据的访问流程,从密钥的生成、分发、使用与删除三个方面探讨了该领域未来的研究方向.

参考文献

[1] 丁滟, 王怀民, 史佩昌, 等. 可信云服务[J]. 计算机学报, 2015, 38(1): 133-149.
DING Y, WANG H M, SHI P C, et al. Trusted cloud service[J]. Chinese Journal of Computers, 2015, 38(1): 133-149. (in Chinese)

[2] 冯朝胜, 秦志光, 袁丁. 云数据安全存储技术[J]. 计算机学报, 2015, 38(1): 150-163.
FENG C S, QIN Z G, YUAN D. Techniques of secure storage for cloud data[J]. Chinese Journal of Computers, 2015, 38(1): 150-163. (in Chinese)

[3] 田洪亮, 张勇, 李超, 等. 云环境下数据库机密性保护技术研究综述[J]. 计算机学报, 2017, 40(10): 2245-2270.
TIAN H L, ZHANG Y, LI C, et al. A survey of confidentiality protection for cloud databases[J]. Chinese Journal of Computers, 2017, 40(10): 2245-2270. (in Chinese)

[4] SUN W H, ZHANG R D, LOU W J, et al. REARGUARD: Secure keyword search using trusted hardware[C]//Proceedings of the 38th IEEE Conference on Computer Communications(INFOCOM' 18), Honolulu: IEEE, 2018: 801-809.

[5] 谭霜, 贾焰, 韩伟红. 云存储中的数据完整性证明研究及进展[J]. 计算机学报, 2015, 38(1): 164-177.
TAN S, JIA Y, HAN W H. Research and development of provable data integrity in cloud storage[J]. Chinese Journal of Computers, 2015, 38(1): 164-177. (in Chinese)

[6] REN Z W, WANG L N, WANG Q, et al. Dynamic proofs of retrievability for coded cloud storage systems[J]. IEEE Transactions on Services Computing, 2018, 11(4): 685-698.

[7] KOSBA A, PAPAMANTHOU C, SHI E. xJsnark: A framework for efficient verifiable computation[C]//Proceedings of the 39th IEEE Symposium on Security and Privacy(S&P' 18). San Francisco: IEEE, 2018: 944-961.

[8] 王丽娜, 任正伟, 余荣威, 等. 一种适于云存储的数据确定性删除方法[J]. 电子学报, 2012, 40(2): 266-272.
WANG L N, REN Z W, YU R W, et al. A data assured deletion approach adapted for cloud storage[J]. Acta Electronica Sinica, 2012, 40(2): 266-272. (in Chinese)

[9] RAMOKAPANE K M, RASHID A, SUCH J M. Assured deletion in the cloud: Requirements, challenges and future directions[C]//Proceedings of 2016 ACM on Cloud Com-

puting Security Workshop(CCSW'16). Vienna: ACM, 2016: 97-108.

[10] 魏凯敏, 翁健, 任奎. 大数据安全保护技术综述[J]. 网络与信息安全学报, 2016, 2(4): 00046-1-00046-11.
WEI K M, WENG J, REN K. Data security and protection techniques in big data: A survey[J]. Chinese Journal of Network and Information Security, 2016, 2(4): 1-11. (in Chinese)

[11] 熊金波, 李凤华, 王彦超, 等. 基于密码学的云数据确定性删除研究进展[J]. 通信学报, 2016, 37(8): 167-184.
XIONG J B, LI F H, WANG Y C, et al. Research progress on cloud data assured deletion based on cryptography [J]. Journal on Communications, 2016, 37(8): 167-184. (in Chinese)

[12] ZHENG D, XUE L, YU C, et al. Toward assured data deletion in cloud storage[J]. IEEE Network, 2020, 34(3): 101-107

[13] GEAMBASU R, KOHNO T, LEVY A A, et al. Vanish: Increasing data privacy with self-destructing data[C]//Proceedings of the 18th USENIX Security Symposium(USENIX Security'09). Montreal: USENIX Association, 2009: 299-316.

[14] GEAMBASU R, KOHNO T, KRISHNAMURTHY A, et al. New Directions for Self-Destructing Data Systems[R]. Technical Report, University of Washington, 2011, UW-CSE-11-08-01.

[15] 熊金波, 姚志强, 马建峰, 等. 基于属性加密的组合文档安全自毁方案[J]. 电子学报, 2014, 42(2): 366-376.
XIONG J B, YAO Z Q, MA J F, et al. A secure self-destruction scheme for composite documents with attribute based encryption[J]. Acta Electronica Sinica, 2014, 42(2): 366-376. (in Chinese)

[16] 熊金波, 姚志强, 马建峰, 等. 面向网络内容隐私的基于身份加密的安全自毁方案[J]. 计算机学报, 2014, 37(1): 139-150.
XIONG J B, YAO Z Q, MA J F, et al. A secure self-destruction scheme with IBE for the Internet content privacy [J]. Chinese Journal of Computers, 2014, 37(1): 139-150. (in Chinese)

[17] 姚志强, 熊金波, 马建峰, 等. 云计算中一种安全的电子文档自毁方案[J]. 计算机研究与发展, 2014, 51(7): 1417-1423.
YAO Z Q, XIONG J B, MA J F, et al. A secure electronic document self-destructing scheme in cloud computing[J]. Journal of Computer Research and Development, 2014, 51(7): 1417-1423. (in Chinese)

[18] XIONG J B, LIU X M, YAO Z Q, et al. A secure data self-destructing scheme in cloud computing[J]. IEEE Transactions on Cloud Computing, 2014, 2(4): 448-458.

[19] FERNANDES M, RODRIGUES SILVA A, GONÇALVES A. Specification of personal data protection requirements-analysis of legal requirements from the GDPR regulation[C]//Proceedings of the 20th International Conference on Enterprise Information Systems(ICEIS'18). Madeira: SCITEPRESS-Science and Technology Publications, 2018: 398-405.

[20] LINDQVIST J. New challenges to personal data processing agreements: Is the GDPR fit to deal with contract, accountability and liability in a world of the Internet of Things?[J]. International Journal of Law and Information Technology, 2017, 26(1): 45-63.

[21] HUA M Y, ZHAO Y Y, JIANG T. Secure data deletion in cloud storage: A survey[J]. International Journal of Embedded Systems, 2020, 12(2): 253-265.

[22] 中华人民共和国国家互联网信息办公室. 全国人民代表大会常务委员会关于加强网络信息保护的决定[EB/OL]. (2021-12-29)[2022-02-21]. http://www.cac.gov.cn/2012-12/29/c_133353262.htm.

[23] 中华人民共和国国家互联网信息办公室. 中华人民共和国网络安全法[EB/OL]. (2016-11-07)[2022-02-21]. http://www.cac.gov.cn/2016-11/07/c_1119867116_2.htm.

[24] 中华人民共和国国家互联网信息办公室. 中华人民共和国个人信息保护法[EB/OL]. (2021-08-20)[2022-02-21]. http://www.cac.gov.cn/2021-08/20/c_1631050028355286.htm.

[25] 刘田甜, 李超, 胡庆成, 等. 云环境下多副本管理综述[J]. 计算机研究与发展, 2011, 48(S3): 254-260.
LIU T T, LI C, HU Q C, et al. Multiple-replicas management in the cloud environment[J]. Journal of Computer Research and Development, 2011, 48(S3): 254-260. (in Chinese)

[26] REARDON J, BASIN D, CAPKUN S. SoK: secure data deletion[C]//Proceedings of the 34th IEEE Symposium on Security and Privacy(S&P'13). Berkeley: IEEE, 2013: 301-315.

[27] SHU J L, ZHANG Y Y, LI J R, et al. Why data deletion fails? A study on deletion flaws and data remanence in Android systems[J]. ACM Transactions on Embedded Computing Systems, 2017, 16(2): 61(1-22).

[28] LEOM M D, CHOO K K R, HUNT R. Remote wiping and secure deletion on mobile devices: A review[J]. Journal of Forensic Sciences, 2016, 61(6): 1473-1492.

[29] YANG L, WEI T, ZHANG F W, et al. SADUS: Secure data deletion in user space for mobile devices[J]. Computers & Security, 2018, 77: 612-626.

[30] JIA S J, XIA L N, CHEN B, et al. NFPS: Adding unde-

tectable secure deletion to flash translation layer[C]//Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. Xi'an: ACM, 2016: 305-315.

[31] WEI M, GRUPP L M, SPADA F E, et al. Reliably erasing data from flash-based solid state drives[C]//Proceedings of the 9th USENIX Conference on File and Stroage Technologies(FAST'11). San Jose: ACM, 2011: 105-117.

[32] LIU C, AGHAEI KHOUZANI H, YANG C M. ErasuCrypto: A light-weight secure data deletion scheme for solid state drives[J]. Proceedings on Privacy Enhancing Technologies, 2017, 2017(1): 132-148.

[33] XIONG J B, CHEN L, BHUIYAN M Z A, et al. A secure data deletion scheme for IoT devices through key derivation encryption and data analysis[J]. Future Generation Computer Systems, 2020, 111: 741-753.

[34] DIESBURG S, MEYERS C, STANOVICH M, et al. TrueErase: Leveraging an auxiliary data path for per-file secure deletion[J]. ACM Transactions on Storage, 2016, 12(4): 18(1-37).

[35] ZHANG Q L, JIA S J, CHANG B, et al. Ensuring data confidentiality via plausibly deniable encryption and secure deletion-a survey[J]. Cybersecurity, 2018, 1(1): 1-20.

[36] CHEN S H, YANG M C, CHANG Y H, et al. Enabling file-oriented fast secure deletion on shingled magnetic recording drives[C]//Proceedings of the 56th Annual Design Automation Conference(DAC'19).Las Vegas: ACM, 2019: 103(1-6).

[37] LI B Z, DU D H C. TASecure: Temperature-aware secure deletion scheme for solid state drives[C]//Proceedings of the 2019 on Great Lakes Symposium on VLSI(GLSVLSI'19). Tysons Corner: ACM, 2019: 275-278.

[38] LI C L, CHEN Y, ZHOU Y Z. A data assured deletion scheme in cloud storage[J]. China Communications, 2014, 11(4): 98-110.

[39] XIONG J B, LI F H, MA J F, et al. A full lifecycle privacy protection scheme for sensitive data in cloud computing[J]. Peer-to-Peer Networking and Applications, 2015, 8 (6): 1025-1037.

[40] 张坤, 杨超, 马建峰, 等. 基于密文采样分片的云端数据确定性删除方法[J]. 通信学报, 2015, 36(11): 108 - 117.
ZHANG K, YANG C, MA J F, et al. Novel cloud data assured deletion approach based on ciphertext sample slice [J]. Journal on Communications, 2015, 36(11): 108-117. (in Chinese)

[41] 王敏燊, 熊金波, 林倩, 等. 基于密钥分发和密文抽样的云数据确定性删除方案[J]. 计算机应用, 2018, 38(1): 194-200.
WANG M S, XIONG J B, LIN Q, et al. Cloud data assured deletion scheme based on key distribution and ciphertext sampling[J]. Journal of Computer Applications, 2018, 38(1): 194-200. (in Chinese)

[42] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.

[43] ZARRAS A, KOHLS K, DURMUTH M, et al. Neuralyzer: Flexible expiration times for the revocation of online data[C]//Proceedings of the 6th ACM Conference on Data and Application Security and Privacy(CODASPY'16). New Orleans: ACM, 2016: 14-25.

[44] 熊金波, 沈薇薇, 黄阳群, 等. 云环境下的数据多副本安全共享与关联删除方案[J]. 通信学报, 2015, 36(S1): 136-140.
XIONG J B, SHEN W W, HUANG Y Q, et al. Security sharing and associated deleting scheme for multi-replica in cloud[J]. Journal on Communications, 2015, 36(S1): 136-140. (in Chinese)

[45] 王丽娜, 任正伟, 董永峰, 等. 云存储中基于可信平台模块的密钥使用次数管理方法[J]. 计算机研究与发展, 2013, 50(8): 1628-1636.
WANG L N, REN Z W, DONG Y F, et al. A management approach to key-used times based on trusted platform module in cloud storage[J]. Journal of Computer Research and Development, 2013, 50(8): 1628-1636. (in Chinese)

[46] CHEN G X, ZHANG Y Q, LAI T H. OPERA: Open remote attestation for Intel's secure enclaves[C]//Proceedings of the 26th ACM SIGSAC Conference on Computer and Communications Security(CCS'19). London: ACM, 2019: 2317-2331.

[47] ZHAO S J, ZHANG Q Y, QIN Y, et al. SecTEE: a software-based approach to secure enclave architecture using TEE[C]//Proceedings of the 26th ACM SIGSAC Conference on Computer and Communications Security(CCS'19). London: ACM, 2019: 1723-1740.

[48] TANG Y, LEE P P C, LUI J C S, et al. FADE: secure overlay cloud storage with file assured deletion[C]//Proceedings of the 6th EAI International Conference on Security and Privacy in Communication Networks(SecureComm'10). Singapore: Springer, 2010: 380-397.

[49] TANG Y, LEE P P C, LUI J C S, et al. Secure overlay cloud storage with access control and assured deletion[J]. IEEE Transactions on Dependable and Secure Computing, 2012, 9(6): 903-916.

[50] ALI M, MALIK S U R, KHAN S U. DaSCE: Data security for cloud environment with semi-trusted third party[J]. IEEE Transactions on Cloud Computing, 2017, 5(4):

642-655.

[51] CACHIN C, HARLAMBIEV K, HSIAO H C, et al. Policy-based secure deletion[C]//Proceedings of the 20th ACM Conference on Computer and Communications Security(CCS' 13). Berlin: ACM, 2013: 259-270.

[52] YU Y, XUE L, LI Y N, et al. Assured data deletion with fine-grained access control for fog-based industrial applications[J]. IEEE Transactions on Industrial Informatics, 2018, 14(10): 4538-4547.

[53] XUE L, YU Y, LI Y N, et al. Efficient attribute-based encryption with attribute revocation for assured data deletion [J]. Information Sciences, 2019, 479: 640-650.

[54] HAO J L, LIU J, WU W, et al. Secure and fine-grained self-controlled outsourced data deletion in cloud-based IoT[J]. IEEE Internet of Things Journal, 2020, 7(2): 1140-1153.

[55] CHENG Y T, YANG L, YU S, et al. Achieving efficient and verifiable assured deletion for outsourced data based on access right revocation[C]//Proceedings of the 18th International Conference on Cryptology and Network Security(CANS' 19). Fuzhou: Springer, 2019: 392-411.

[56] SHAN F F, LI H, LI F H, et al. An attribute-based assured deletion scheme in cloud computing[J]. International Journal of Information Technology and Web Engineering, 2019, 14(2): 74-91.

[57] TIAN Y C, SHAO T, LI Z. An efficient scheme of cloud data assured deletion[J]. Mobile Networks and Applications, 2021, 26: 1597-1608.

[58] TIAN J F, WANG Z D. Fine-grained assured data deletion scheme based on attribute association[J]. Computers & Security, 2020, 96: 101936(1-9).

[59] MA J, WANG M S, XIONG J B, et al. CP-ABE-based secure and verifiable data deletion in cloud[J]. Security and Communication Networks, 2021, 8855341(1-14).

[60] BENTAJER A, HEDABOU M, ABOUELMEHDI K, et al. An IBE-based design for assured deletion in cloud storage[J]. Cryptologia, 2019, 43(3): 254-265.

[61] KAUSHIK S, GANDHI C. Capability based outsourced data access control with assured file deletion and efficient revocation with trust factor in cloud computing[J]. International Journal of Cloud Applications and Computing, 2020, 10(1): 64-84.

[62] REARDON J, RITZDORF H, BASIN D, et al. Secure data deletion from persistent media[C]//Proceedings of the 20th ACM Conference on Computer and Communications Security(CCS' 13). Berlin: ACM, 2013: 271-284.

[63] ROCHE D S, AVIV A, CHOI S G. A practical oblivious map data structure with secure deletion and history independence[C]//Proceedings of the 37th IEEE Symposium on Security and Privacy(S&P' 16). San Jose: IEEE, 2016: 178-197.

[64] MO Z, XIAO Q J, ZHOU Y A, et al. On deletion of outsourced data in cloud computing[C]//Proceedings of the 7th IEEE International Conference on Cloud Computing (CLOUD' 14). Anchorage: IEEE, 2014: 344-351.

[65] MO Z, QIAO Y, CHEN S G. Two-party fine-grained assured deletion of outsourced data in cloud systems[C]// Proceedings of the 34th IEEE International Conference on Distributed Computing Systems(ICDCS' 14). Madrid: IEEE, 2014: 308-317.

[66] YAO W B, CHEN Y J, WANG D B. Cloud multimedia files assured deletion based on bit stream transformation with chaos sequence[C]//Proceedings of the 17th International Conference on Algorithms and Architectures for Parallel Processing(ICA3PP' 17). Helsinki: Springer, 2017: 441-451.

[67] 张逢喆, 陈进, 陈海波, 等. 云计算中的数据隐私性保护与自我销毁[J]. 计算机研究与发展, 2011, 48(7): 1155-1167.
ZHANG F Z, CHEN J, CHEN H B, et al. Lifetime privacy and self-destruction of data in the cloud[J]. Journal of Computer Research and Development, 2011, 48(7): 1155-1167. (in Chinese)

[68] ZHAO L Y, MANNAN M. Gracewipe: secure and verifiable deletion under coercion[C]//Proceedings of the 22th Annual Network and Distributed System Security(NDSS' 15). San Diego: Internet Society, 2015: 1-16.

[69] HAO F, CLARKE D, ZORZO A F. Deleting secret data with public verifiability[J]. IEEE Transactions on Dependable and Secure Computing, 2016, 13(6): 617-629.

[70] WAIZENEGGER T. Deletion of Content in Large Cloud Storage Systems[D]. Stuttgart: Institut für Parallele und Verteilte Systeme der Universität Stuttgart, 2017.

[71] WAIZENEGGER T, WAGNER F, MEGA C. SDOS: using trusted platform modules for secure cryptographic deletion in the swift object store[C]//Proceedings of the 20th International Conference on Extending Database Technology(EDBT' 17). Venice: DBIS, 2017: 550-553.

[72] REN Z W, CHEN X S, TANG J S, et al. Limited times of data access based on SGX in cloud storage[C]//Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics(SMC' 21). Melbourne: IEEE, 2021: 3146-3151.

[73] REN Z W, LI X, XU S W, et al. Restricting the number of times that data can be accessedincloud storage using TrustZone[C]//Proceedings of the 22th IEEE/ACM Internation-

al Symposium on Cluster, Cloud and Internet Computing (CCGrid'22). Messina: IEEE/ACM, 2022: 289-296.
[74] XUE L, NI J B, LI Y N, et al. Provable data transfer from provable data possession and deletion in cloud storage[J]. Computer Standards & Interfaces, 2017, 54: 46-54.
[75] LIU Y D, WANG X A, CAO Y F, et al. Improved provable data transfer from provable data possession and deletion in cloud storage[C]//Proceedings of International Conference on Intelligent Networking and Collaborative Systems(INCoS'18). Bratislava: Springer, 2018: 445-452.
[76] YANG C S, TAO X L. New publicly verifiable cloud data deletion scheme with efficient tracking[C]//Proceedings of International Conference on Security with Intelligent Computing and Big-data Services(SICBS'18). Bratislava: Springer, 2018: 359-372.
[77] HALL B, GOVINDARASU M. An assured deletion technique for cloud-based IoT[C]//Proceedings of the 27th International Conference on Computer Communication and Networks(ICCCN'18). Hangzhou: IEEE, 2018: 1-9.
[78] DU L, ZHANG Z W, TAN S C, et al. An associated deletion scheme for multi-copy in cloud storage[C]//Proceedings of the 18th International Conference on Algorithms and Architectures for Parallel Processing(ICA3PP'18). Guangzhou: Springer, 2018: 511-526.
[79] YANG C S, WANG J F, TAO X L, et al. Publicly verifiable data transfer and deletion scheme for cloud storage [C]//Proceedings of the 20th International Conference on Information and Communications Security(ICICS'18). Lille: Springer, 2018: 445-458.
[80] YANG C S, TAO X L, ZHAO F, et al. Secure data transfer and deletion from counting bloom filter in cloud computing[J]. Chinese Journal of Electronics, 2020, 29(2): 273-280.
[81] LUO Y C, XU M, FU S J, et al. Enabling assured deletion in the cloud storage by overwriting[C]//Proceedings of the 4th ACM International Workshop on Security in Cloud Computing(SCC'16). Xi'an: ACM, 2016: 17-23.
[82] YANG C S, CHEN X F, XIANG Y. Blockchain-based publicly verifiable data deletion scheme for cloud storage [J]. Journal of Network and Computer Applications, 2018, 103: 185-193.
[83] YANG C S, LIU Y L, TAO X L. Assure deletion supporting dynamic insertion for outsourced data in cloud computing[J]. International Journal of Distributed Sensor Networks, 2020, 16(9): 155014772095829(1-14).
[84] REN Z W, LI X J, WANG L N, et al. Minimal key set of binary key-derivation tree in cloud storage[J/OL]. Soft Computing,2021,https://doi.org/10.1007/s00500-021-06065-w.

作者简介

任正伟 男,1986年4月出生于湖北省武汉市.2014年于武汉大学获工学博士学位.武汉科技大学计算机科学与技术学院教师、硕士生导师.主要研究方向为数据安全、应用密码学.
E-mail: zhengwei_ren@whu.edu.cn

李雪婷 女,1997年12月出生于湖北省枣阳市.武汉科技大学计算机科学与技术学院硕士研究生.主要研究方向为可信硬件.
E-mail: 1569698204@qq.com

王丽娜 女,1964年10月出生于辽宁省营口市.武汉大学国家网络安全学院教授、博士生导师.主要研究方向为信息安全.
E-mail: lnwang@whu.edu.cn

童 言 男,1982年10月出生于湖北省黄冈市.华中农业大学理学院副教授.主要研究方向为密码学和区块链.
E-mail: tongy@mail.hzau.edu.cn

徐士伟 男,1985年6月出生于江西省宜春市.华中农业大学信息学院副教授.主要研究方向为信息安全和区块链.
E-mail: xushiwei@mail.hzau.edu.cn

丁 炜 男,1979年5月出生于湖北省武汉市.中国地震局地震研究所地震大地测量重点实验室高级工程师.主要研究方向为软件体系结构.
E-mail: tingwhere@whu.edu.cn