云环境下数据库机密性保护技术研究综述

田洪亮 张 勇 李 超 邢春晓

(清华大学计算机科学与技术系 北京 100084)

(清华大学信息技术研究院 北京 100084)

(清华信息科学与技术国家实验室 北京 100084)

摘 要 随着云计算的迅猛发展,越来越多的企业和个人把数据外包到位于公有云的数据库系统上管理.然而,数据安全和隐私保护方面的顾虑已经成为阻碍用户更广泛采用云计算和云数据库的一大因素.在云数据外包业务和云数据安全需求的强力驱动下,云环境下数据库的机密性保护成为了重要的研究课题.该综述首先提出了云数据库机密性保护的五级安全模型,使得该文中涉及的众多跨领域、跨问题的安全技术得以在统一的框架中讨论;然后,针对该安全模型中的 3 级至 5 级的机密性威胁(云环境下的新型威胁),系统性地总结和分析了对其相应的三项机密性保护的关键技术——密文查询(3 级),可信硬件(4 级)和访问模式保护(5 级);进而在此基础上,介绍和比较了目前最先进的安全云数据库系统;最后,展望了云数据库机密性保护技术的研究趋势,指出了若干研究方向.

关键词 云计算;数据库系统;数据安全;密文查询;可信硬件;访问模式保护

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2017.02245

A Survey of Confidentiality Protection for Cloud Databases

TIAN Hong-Liang ZHANG Yong LI Chao XING Chun-Xiao

(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

(*Research Institute of Information Technology, Tsinghua University, Beijing* 100084)

(*Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing* 100084)

Abstract As cloud computing is getting popular, more and more individuals and organizations are considering outsourcing data to database management systems on public clouds. However, data security and privacy concerns hinder the further adoption of cloud computing and cloud databases. Driven by the needs of data outsourcing and data security on cloud, confidentiality protection for cloud databases has become an important research topic. In this survey, we start by introducing a five-level security model for confidentiality-protected cloud databases, which provides a unified framework for discussing various relevant techniques. This five-level security model identifies five types of security risks: interface risk (level 1), storage risk (level 2), memory data risk (level 3), program state risk (level 4), and access pattern risk (level 5). Among them, the risks from level 3 to level 5 are considered as new challenges in a cloud environment. Then, to address these security risks that are specific to cloud databases, we systematically review and analyze three key data security approaches: querying over encrypted data (level 3), trusted hardware (level 4) and access pattern protection (level 5). Querying over encrypted data can be seen as a specialized form of computation over encrypted data, which has

收稿日期:2016-08-13;在线出版日期:2017-05-25.本课题得到国家自然科学基金(91646202)、国家"八六三"高技术研究发展计划项目基金(SS2015AA020102)、千人计划和清华大学自主科研计划资助.**田洪亮**,男,1987 年生,博士研究生,主要研究方向为可信硬件技术、云计算.E-mail:tatetian@gmail.com.**张 勇**,男,1973 年生,博士,副研究员,主要研究方向为数据管理.**李 超**,女,1978 年生,博士,副研究员,主要研究兴趣包括存储系统和数据管理.**邢春晓**(通信作者),男,1967 年生,博士,研究员,博士生导师,主要研究领域为数据库、数字图书馆等.E-mail:xingcx@tsinghua.edu.cn.

been theoretically solved by fully homomorphic encryption. Unfortunately, all current constructions are too inefficient to be used in practice. However, there are certain kinds of queries can be efficiently processed on encrypted data. For example, range queries can be implemented by either leveraging order-preserving encryption or encrypted indexes. As another example, keyword querying over encrypted data can be efficiently implemented. Regarding the two types of queries, a number of state-of-the-art techniques are described in this paper. The main difficulties of querying over encrypted data are (1) handling arbitrary types of queries over encrypted data and (2) reducing the information leakage. To overcome these drawbacks, it is probably necessary to introduce into cloud environment a special kind of hardware, namely trusted hardware, where user data and computation within are isolated from the outside. Three trusted hardware technologies are described and compared, which are secure coproceesors, FPGAs and Intel SGX. FPGAs and Intel SGX are superior than secure coprocessors in terms of performance and cost, thus more likely to get widely adopted by cloud service providers. While trusted hardware protects the data and computation inside, it reveals its access to the storage outside. It has been shown by several studies that access pattern can leak sensitive information. To reduce information leakage from access pattern, there are two general techniques: private information retrieval and oblivious RAM. The target application scenario of these two techniques are different: the former assumes no encryption of data, while the latter requires encryption. A systematic review on the two techniques are presented in this paper. After reviewing the key techniques of protecting data confidentiality in cloud, we describe and compare the state-of-the-art secure cloud database systems, including those based on cryptography and those based on trusted hardware. Finally, we conclude with perspectives and directions for future research.

Keywords cloud computing; cloud databases; data security; querying over encrypted data; trusted hardware; access pattern protection

1 引 言

随着云计算的迅猛发展,越来越多的企业和个人把数据外包到公有云上管理.用户可以租用基础设施即服务(Infrastructure as a Service)提供的虚拟机,并在其上搭建自己的数据库系统.用户甚至可以免去动手安装配置的繁琐,直接采用云服务提供商的数据库即服务(Database as a Service),如谷歌的Cloud SQL[①]、微软的SQL Azure[②]和亚马逊的关系型数据库服务(Relational Database Service)[③]等.无论是上述哪种方式,都将数据迁移到了云端(即云数据外包),并利用了云环境下数据库系统做数据管理(即云数据库系统),最终令用户可以享受到云计算的廉价、便捷、弹性、可靠等优点.

云数据库,作为云计算的一种应用,具备云计算的诸多优点,同时也凸显了云计算在数据安全保障方面的不足;毫不夸张地说,数据安全和隐私保护方面的顾虑已经成为阻碍用户更广泛采用云计算和云数据库的一大因素.IDC调研[④]显示安全性是企业考虑是否采用云计算时的最大疑虑.这种担忧很大程度上来自对云服务提供商的不信任.首先,一旦数据外包到云端,用户实际上是放弃部分对数据的控制权,不得不转而依赖并信任云服务提供商,而这对掌握高价值、高敏感数据的客户(如金融、医疗和政府等)来说是难以接受的.其次,目前云服务提供商要求客户签订的服务协议,比如亚马逊[⑤],只提到可以采用"适当措施"以确保数据安全和隐私保护,对因超出"合理范围"因素而导致的损失概不负责;这

① Cloud SQL. Google Cloud Platform. https://cloud.google.com/sql/
② SQL Database. Microsoft Azure. http://azure.microsoft.com/en-us/services/sql-database/
③ Amazon Relational Database Service (RDS). http://aws.amazon.com/rds/
④ New IDC IT Cloud Services Survey: Top Benefits and Challenges. http://blogs.idc.com/ie/?p=730
⑤ AWS customer agreement. http://aws.amazon.com/cn/agreement/

种“尽力而为”程度的保证难以令客户放心.最后,谷歌员工窥探未成年人的 Gmail 通信记录①和瑞士银行泄露客户资料②等来自内部的数据泄露事件令客户不得不对云服务商内部人员的专业操守抱以谨慎怀疑的态度.综上可见,目前的公有云平台尤其是其中负责数据集中存储和管理的云数据库管理系统,对数据安全性的各项保障措施,并未赢得用户的足够信任,这制约着云计算以及云数据库系统未来被更广泛的采用.

在上述云数据外包业务和云数据安全需求的强力驱动下,学术界和企业界都开始研发云环境下数据库系统的数据安全保护技术.数据库系统的数据安全性至少包括三个方面[1]:机密性(Confidentiality)、完整性(Integrity)和可用性(Availability).对云环境下数据库系统在这三个维度中的任意一个的安全保护都已经有了大量的研究工作.限于篇幅,本综述聚焦于三者之首的机密性,即云环境下数据库系统的机密性保护.本文中的“安全云数据库系统”均指云环境下保障机密性的数据库系统.

1.1　云数据库的机密性保护

云数据库的机密性保护与传统数据库有很大不同.数据库管理系统是运行在数据库服务器上的.根据对数据库服务器的信任程度,一般可以将数据库服务器分为完全信任、部分信任或完全不信任[1].传统的环境下(如企业自有机房),我们往往假定服务器是可以完全信任或至少部分信任的,安全威胁主要来自服务器外部,安全措施主要是防止服务器被入侵.但在云环境下(尤其是公有云),服务器物理上位于云服务商的机房,逻辑上受云服务商的管理,云服务商拥有对云服务器的最终控制权.因此,对于涉及到敏感数据的用户来讲,最稳妥的做法应该是假定云服务器完全不可信.所谓“完全不可信”,即服务器上的系统软件(如操作系统和数据库管理系统等)可能被攻击者控制;各层面的访问控制机制可能被绕过;外存、内存和网络的数据都有可能被窃取.云数据库的机密性保护的主要目标是防范上述来自内部的安全威胁.

为保障云数据库中数据的机密性,一种直接的做法是数据加密,但加密后的数据如何安全并高效地做处理?目前主流商用数据库管理系统对数据加密的方案是,数据在外存中存储时加密、在内存中处理时解密;也就是说,外存中是密文,内存中是明文.在这个安全方案中,内存中的明文数据和明文密钥都存在被窃取的风险,因此安全性不理想.另一种保守但安全的做法是,云端对数据只加密存储而不处理,用户每次查询、检索、计算或分析都必须把相应的数据传回客户端,然后解密之后处理.这种做法放弃了对云计算资源的有效利用,不可避免地需要向客户端传回大量数据,因而,从性能上看这种方案是无法接受的.以上两种朴素的安全方法都没有做到在安全、功能和性能三者之间的兼顾与平衡(后文中会看到,安全、功能和性能的平衡是安全云数据库系统研究中反复出现的一个主题).

对云环境下数据库系统的安全保护,尤其是机密性保护,在过去十多年的时间里,获得了学术界和企业界的共同关注和广泛研究,取得了显著的研究进展,产生了大量的研究成果.现在,云数据库机密性保护已经发展成为一个涉及多个领域(如密码学、数据管理、体系结构等)、包含多个子方向(如密文查询、可信硬件和访问模式保护等)的庞大研究主题.由于兼具广度和深度,这一研究主题对新进入的学术研究者以及有兴趣的企业开发人员都是不小的挑战.一篇关于云数据库机密性保护的综述无疑可以提供有价值的参考,帮助读者更快地探索、理解和掌握有关技术.

本综述首先提出了云数据库机密性保护的五级安全模型(见第 2 节),并将它作为介绍后续所有相关工作的统一框架.然后,针对安全模型中的 3 级至 5 级的机密性威胁(云环境下的新挑战),本文对其相应的三项机密性保护的关键技术做了全面的总结和分析,包括密文查询(3 级,见第 3 节),可信硬件(4 级,见第 4 节)和访问模式保护(5 级,见第 5 节).进而在此基础上,本文详细地比较了目前最先进的安全云数据库系统(见第 6 节).最后,本文展望了云数据库机密性保护技术的发展趋势,指出了若干研究方向(见第 7 节).

1.2　相关综述

(1)云计算安全.云计算安全是所有与云计算有关的安全威胁和解决方案的总称.云计算安全所考虑的安全威胁包括云计算的所有层面:物理层(比如,雷电、火灾等),存储层(比如,数据加密、灾难备份等),网络层(比如,拒绝服务攻击、DNS 攻击等),系统层(比如,虚拟机漏洞、操作系统漏洞等),平台层(比如,程序语言实现的漏洞),数据库层(比如,

① Google Engineer Stalked Teens. Spied on Chats. http://gawker.com/5637234/creep-google-engineer-stalked-teens-spied-on-chats

② Germany Tackles Tax Evasion. http://online.wsj.com/news/articles/SB10001424052748704197104575051480386248538

SQL 注入攻击),应用层(比如,身份认证和权限管理).显然,云数据库安全是云安全的一个方面.更多关于云安全综述可见文献[2-5].

(2)云存储安全.云存储是通过云计算提供的数据存储服务.安全云存储的安全目标包括机密性、完整性和可用性等方面,其研究内容包括密文检索、密钥分发、密文除重(Deduplication)、完整性审计、密文访问控制等技术.本综述的目标——安全云数据库,在安全目标和安全技术方面与安全云存储有一定的重合与相通之处.不同之处在于,安全云数据库需要在保证相似的安全性的前提下支持更多也更复杂的功能需求,即数据库查询.在这种意义上,我们可以说云数据库安全比云存储安全更难.更多关于云存储安全综述可见文献[6-8].

(3)云数据库安全.就作者所知,目前尚无从系统整体的角度对安全云数据库做全面阐述的综述文章.与本文最相近的是文献[9],该文介绍了云计算中的数据安全与隐私保护.该文涉及了数据安全的机密性、完整性和可用性三个维度,而本文只专注于机密性保护,在机密性保护方面的讨论更加完整和深入.另外一个重要不同是,该文并不具体到某一种云服务,而本文专门针对云数据库.因而本文中的讨论不止于各种安全技术本身,更注重安全技术与数据库技术的结合.

2 五级安全模型

任何有关安全性的严肃讨论都离不开对"安全"的定义,本文也不能例外.虽然几乎每篇云数据库安全方面的相关论文都有定义"安全",甚至可以达到数学意义上的精确,但这些安全定义只适用于本文讨论的具体问题,该具体问题只是整个系统的局部,或者说是云数据库机密性保护的子问题之一.凭这些孤立的、局部的安全定义,用户仍然难以评估系统整体的安全性.

为了给出评估云数据库系统整体安全性的标准,也为了把跨领域、跨问题的不同安全技术放在统一框架下讨论,本节提出了适用于云环境的数据库机密性保护的五级安全模型(如图 1).下面具体描述这个安全模型的四个方面:参与者、系统、(五级)机密性风险和(五级)机密性保护.

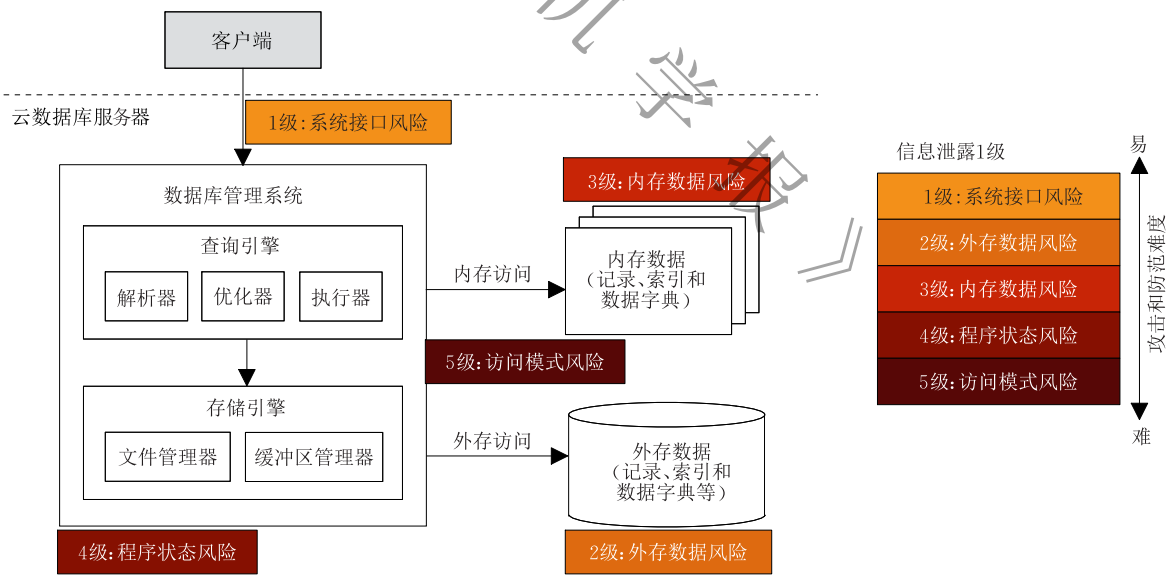


图 1 云环境下数据库系统的五级机密性风险(风险级别越高,对攻击者能力的要求越高;相应地,防范该风险的难度也越高)

2.1 参与者

在本安全模型中的参与者有三种角色:用户、管理员和攻击者.云数据库的用户也即数据拥有者,他们把数据外包到云端,希望获得高效、廉价和安全的数据管理服务,这里说的安全主要是指机密性.管理员,代表云服务提供商,负责安装、配置、管理和监控云数据库系统,其重要职责之一是保障系统的安全性,采取各种防范措施应对可能的攻击.攻击者,可能来自外部,也可能是内部的其他用户,甚至是管理员,因此攻击者可能具有特权.如前所述,本文主要关注的安全性是机密性,因此本文的安全模型假设攻击者是消极的(Passive),或者说是诚实但好奇的(Honest but Curious),其主要兴趣在于窥探和窃取用户数据;至于主动类型的恶意行为,如篡改破坏数

据、重放攻击和拒绝服务攻击等，超出了本文的讨论范畴.另外，旁路攻击(Side-Channel Attacks)和物理攻击(Physical Attacks)也不在本文的讨论范围.

2.2 系　统

整个系统是客户端-服务器架构的，用户通过数据库客户端发送查询和指令给位于云端数据库服务器上的数据库系统.后者在本模型中被描述为一个抽象的关系型数据库管理系统(Relational Database Management Systems)，包括查询引擎和存储引擎，分别负责管理内存中和外存中的数据.注意，在这个系统中，位于云端的数据库服务器是不可信的，其上的硬件(除了可信硬件，见 2.4 小节)与软件(除了可信硬件中运行的软件)都是不可信的；而客户端是完全由用户控制的，因而是可信的.

2.3 机密性风险

本安全模型试图对云数据库机密性的所有潜在风险做系统性地识别、归类和分级.这里将机密性风险分成从 1 级到 5 级(见图 1).前两级风险也适用于传统应用场景，而后三级风险是云环境下的新挑战.风险级别越高，对攻击者能力的要求越高，相应地，防范该风险的难度也越高.这五级机密性风险分别是：

1 级，访问接口风险.是指攻击者透过系统的访问接口(网络接口、编程接口、查询语言等)而泄露信息的风险，这包括因密码破解、软件漏洞或配置不当等原因导致攻击者得以通过系统接口来非法访问数据库，还包括正常用户远程访问系统的网络连接遭受攻击而泄露信息(比如中间人攻击)，也包括合法用户滥用权限、不恰当地查询数据.针对这些风险，一般采用网络加密、访问控制、程序补丁、以及监控审计等安全措施来防范.

2 级，外存数据风险.是指攻击者绕过系统接口而直接访问外存(如硬盘)中的敏感数据，比如，存储着数据库中数据的硬盘遗失或被盗，或者特权攻击者绕过数据库的访问控制直接从文件系统中或存储系统中读取数据.针对这类风险，主流的措施是对外存数据加密，这种加密可以是在应用层面、数据库层面(如微软 SQLServer 的透明硬盘加密①)、文件系统层或者硬件层面.

以上两种风险都属于传统数据库安全技术应对范畴，下面介绍在云环境下凸显出的新挑战.

3 级，内存数据风险.是指攻击者绕过访问限制(如数据库的访问控制，操作系统的进程间隔离等)直接读取数据库在内存中的敏感数据，比如在数据库缓冲区中的记录和索引.对内存数据加密固然能够保护机密性，但给数据处理带来了很大的困难；而一旦某些数据处理的操作得以实施又往往会造成一定程度的泄露信息.可见，完全防止内存数据的信息泄露是很困难的.内存数据风险是云数据库机密性保护的首要挑战.

4 级，程序状态风险.是指攻击者通过跟踪代码执行或记录程序状态来获得敏感信息.在数据库管理系统中，比较重要的程序状态包括，查询请求、中间结果以及查询计划等，这些都可能然泄露敏感信息.比如，数据库表中一个经过适当加密的列，静态来看该列的密文记录是不会泄露任何信息的；然而，一旦在这个列上的两个记录做比较操作，这个程序执行的中间结果就泄露了密文之间的大小关系.

这里需要强调的是，虽然“内存数据”和“程序状态”都主要存在于内存中，但它们有不同的侧重.“内存数据”主要指数据库中内容(如记录和索引)在内存中的表示，大部分存在于数据库缓冲区，占用内存较多；对于一个只查询而无更新的数据库来说，“内存数据”基本不变.而“程序状态”虽然也在内存中(还有寄存器中)，但它表示的是数据库运行时状态，主要存在于程序的堆和栈中，占用较小的内存；即使对于一个只读的数据库，只要它在处理查询过程中，程序状态就会不停地随着查询的执行而变化.由此可见，对于攻击者而言，程序状态短暂易逝、瞬息万变，因此这种信息泄露更难获取和利用.

5 级，访问模式风险.是指攻击者通过监控和分析数据库对内存和外存的访问模式(Access Patterns)从而推断出敏感信息.在这里，我们可以简单地认为内存和外存的访问接口要求包含三种信息：命令(比如读或写)，地址和数据.一系列的存储器访问所对应的数据(可能是加密的)、命令和地址等信息就构成了访问模式.访问模式的信息泄露看似无害，但一系列工作已经证明访问模式可以令攻击者推测出敏感信息.文献[10]展示了数据库的内存访问模式如何泄露正在执行 SQL 查询；文献[11]介绍了一种针对加密文档数据库的攻击，证明了仅凭借少量的先验知识和访问模式就可以高准确性地推测出敏感信息(比如加密文档中包含的关键词).相对于外内存数据和程序状态，访问模式的信息泄露的利用难度更大，因此该风险在本模型中的级别也最高.

① Transparent Data Encryption(TDE) of SQL Server. http://msdn.microsoft.com/en-us/library/bb934049.aspx

需要希望提醒读者的是,本模型的威胁分级是框架性的;实际上,每一级的安全风险(及其对应的保护技术,见下一小节)都可以再细分.比如,第3级的内存数据威胁可再分为:3.0级唯密文攻击(Ciphertext-Only Attacks),3.1级已知明文攻击(Known-Plaintext Attacks),3.2级选择明文攻击(Chosen-Plaintext Attacks),3.3级选择密文攻击(Chosen-Ciphertext Attacks);亦或者分为:3a级快照攻击(Snapshot Attacks)和3b级持久攻击(PersistentAttacks).如何细分取决于具体问题的应用场景和安全需求.本节提出的五级模型的意义在于为理解和讨论云环境下数据库的机密性保护提供一个整体的、宏观的视角.

2.4 机密性保护

与风险相对应,云数据库的机密性保护技术也可以分为五级.我们说某项机密性保护技术是 i 级的,如果它可以应对的最高级别的机密性风险是 i 级的(其中 $1\leq i\leq 5$).总的来讲,机密性保护技术的级别越高,其技术上的实现难度也越大;而从安全性上讲,不同级别的机密性保护技术之间的关系是互补的、而非蕴含的.根据这个定义,1级机密性保护技术(如访问控制)和2级机密性保护技术(如硬盘加密)属于传统数据库安全范畴,这里不多赘述.本文关注的3级到5级的机密性保护技术,这是云数据库安全研究的重点.具体来讲,本文将对三项云数据库机密性保护的关键技术做系统地总结和分析:

(1)密文查询技术(3级).该技术使得数据库可以直接在加密数据上做查询处理.由于数据始终保持加密,因此密文查询技术有助于云数据库防范1至3级机密性风险.详见第3节.

(2)可信硬件技术(4级).该技术是利用一种安全增强的硬件,其上的计算资源和存储资源受硬件保护,能保证计算和数据的机密性和完整性(虽然本文关注的是机密性,但可信硬件对完整性的保护也是一个重要优点),因而是可信的.可信硬件技术能够有效保护程序状态以及内存数据,因此可以防范3级和4级机密性风险.详见第4节.

(3)访问模式保护(5级).该技术使得程序在"不暴露意图"的前提下读写存储中的数据."不暴露意图"令攻击者难以从访问模式推测出敏感信息,因此可以防范5级机密性风险.详见第5节.

虽然密文查询、可信硬件和访问模式保护等并不能覆盖所有目前已知或未来可能的高级机密性保护技术(大于等于3级的),但这三种技术是目前主流的技术路线,它们(1)具有代表性,其解决问题的思路和技巧是典型的和可借鉴的;(2)具有多样性,有很多细分和变种,可根据具体需求来选择;(3)具有实用性,其性能达到或接近实际部署的程度,可作为基本构件来搭建安全云数据库系统.因此,本文称这三种技术为云数据库机密性保护的关键技术,并给予系统性的总结和分析.

类似地,云数据库系统可根据机密性保护的程度分为五级.我们说某个云数据库系统达到 i 级机密性,如果该数据库系统可以应对1级到 i 级的机密性风险(其中 $1\leq i\leq 5$).根据该定义,所有商用数据库产品,由于具备访问控制和网络加密等安全措施,都达到1级机密性;部分具备硬盘加密功能的商用数据库产品,达到2级机密性.以上1级、2级机密性的数据库系统属于传统范畴,3级及以上才能被认为是安全云数据库系统.安全云数据库系统的实例及其系统架构和设计原则将在第6节介绍.

3 密文查询(3级)

密文查询(Querying over Encrypted Data)是一种基于密码学方法的数据库查询技术,它利用特殊设计的数据加密方法或密码学协议,使得不经解密即对密文做某些处理成为可能.由于数据在外存和内存上始终保持加密,因而有效地减少2级和3级机密性风险.

数据库查询,特别是关系型数据库查询,是由一套有限种类且定义良好的操作子(Operator)组成,比如选择(Selection)、投影(Projection)、连接(Join)、并集(Union)、交集(Intersection)、聚集(Aggregation)等等.有些数据库操作,比如投影,只在乎记录有哪些字段,跟字段的具体值无关,因而投影操作可以直接在密文上执行(如果加密的颗粒度是字段,而不是记录).但更普遍的情况是,操作必须"理解"字段内容:选择操作需要确定某条记录是否符合查询条件,连接操作需要判断两条记录是否匹配,聚集操作可能需要做求和运算.

加密固然可以增强数据机密性,但也给数据库操作的执行带来困难.以概率性加密(Probabilistic Encryption)为例,它是泄露信息最少的一类加密方法,即使加密相同的明文每次也会得出不同的密文.一般来说,概率性加密后的数据是无法直接处理的.而与概率性加密相对的是,确定性加密(Deterministic Encryption),对给定的相同明文和密钥产生相同的

密文，这使得判等操作可以直接在密文上进行，但这也泄露了明文的概率分布信息. 从上面两个例子中可以看出，功能和安全是密文查询技术要解决的主要矛盾(当然，在性能上也必须是可接受的).

本节将介绍各种密文查询技术，这些技术在功能安全和性能方面都有不同的权衡取舍. 首先讨论的是将密文查询归结为全同态加密的可行性；接下来的两小节分别介绍范围密文查询和关键词密文查询；最后，讨论复合密文查询技术.

3.1　密文查询：作为密文计算的一种特例

密文查询可以认为是密文计算在加密数据上做计算的一个特例. 而密文计算的问题可以由全同态加密解决. 本小节介绍全同态加密的进展，并讨论它在安全云数据库上的应用前景.

直接在加密数据上做计算的问题，实际上远在云计算的应用场景出现之前就被提出来了. 1978 年，就在 RSA 密码系统诞生后不久，Rivest 等人就提出了全同态加密(Fully Homomorphic Encryption)的概念，试图解决在加密数据上做计算的问题[12]. 定义全同态加密之前，必须先定义同态加密：令 P 和 C 分别表示明文和密文的集合，K 表示密钥集合，我们说一个加密方法 $E:P\times K\rightarrow C$ 是同态的，如果对任何密钥 $k\in K$ 和任意 $p_1,p_2\in P$，加密方案 E 满足：

$$E_k(p_1\odot_P p_2)=E_k(p_1)\odot_C E_k(p_2),$$

其中 $\odot_P$ 和 $\odot_C$ 分别是定义在 P 和 C 上的两个操作. 由于 $(P,\odot_P)$ 和 $(C,\odot_C)$ 通常会构成一个群，而满足上式的两个群又被成为同态群，因此满足上面性质的加密方法被成为同态加密. 从上式可以看出，同态加密将明文上的运算转换成相应密文上的运算.

一个同态加密的用处取决于它能支持的操作符. 如果操作符是加法，加密方案 E 就叫加法同态加密(比如 Paillier 密码系统[13])；如果操作符是乘法，加密方案 E 就叫乘法同态加密(比如 ElGamal 密码系统[14]). 特别地，如果一个加密方案 E 同时支持加法和乘法，则 E 被成为全同态加密. 由于加法和乘法可以表达与、或和非这个三个逻辑运算，而后者又可以组成任意电路(即程序)，因此全同态加密在理论上支持在加密数据上做任意复杂的运算. 这样，密文查询问题也就可以由全同态加密解决.

虽然全同态加密的概念于 1978 年就提出了，但直到 2009 年才由 Gentry[15] 基于理想格(IdealLattices)理论设计出第一个构造. 然而，目前全同态加密实现的性能离实用还有很大的距离. Gentry 在后续工作[16]中优化并实现了他提出的基于理想格的全同态加密，但实验结果表明其公钥大小高达 2.3GB，公钥生成时间长达 2 小时. 其他研究者试图改进基于晶格的全同态加密[17]，或提出其他类型的构造，如基于整数的[18]和基于带误差学习(Learning With Errors)[19]. 但目前的全同态加密实现①仍需要 3 个小时以完成一次同态的 AES 加密②. 除了性能问题以外，如何在全同态加密的基础上支持全部的数据库操作仍需进一步研究. 文献[20]肯定了全同态加密可以支持复杂的选择、范围、连接和聚集操作，但也指出对于结果集大小不确定的查询，不能保证所有符合的记录都被返回.

全同态加密只泄露运算类型和结果集大小的信息，不会泄露任何关于数据本身的信息，因此可以达到理想的数据机密性. 但在全同态加密的性能达到实用程度之前，我们不得不尝试开销更低的其他性能的方法. 而性能的提高是以功能为代价的，因此接下来介绍的各种密文查询技术都只能支持特定种类的数据库查询.

3.2　范围密文查询

范围查询是一种最基础的、最常见的数据库查询. 比如，用 SQL 语言表示的一个范围查询 SELECT * FROM employees WHERE 20<=age and age<=30，它查找所有年龄在 20 到 30 岁之间的员工的记录.

本小节将介绍范围密文查询的两类解决方法：基于保序加密和基于加密索引. 保序加密方法保证密文之间的数值大小关系与明文一致，因此数据库系统的很多原有的查询算法、索引结构等无需任何修改即可适用于密文，这是保序加密很有吸引力的一个优点. 而加密索引则把密文的顺序关系以密码学的方法转换到数据结构之中，使得符合查询范围条件的记录可以通过索引快读地定位和读取. 由于顺序信息并不直接可见，加密索引相比保序加密而言泄露的信息更少.

3.2.1　基于保序加密的范围密文查询

保序加密(Order-Preserving Encryption)是一种保持数值大小关系的加密方法. 加密方法 E 是保序的，如果对任何密钥 k，以及明文 x 和 y 满足 $x<y$，则 $E_k(x)<E_k(y)$. 因为密文保持明文的大小关系，因此可以直接在密文上做比较操作，常见的

① HElib. https://github.com/shaih/HElib
② Performance of HElib. http://mpclounge.les.word-press.com/2013/04/hespeed.pdf

SQL 查询,如范围、MAX 和 MIN 等,也就可以方便的实现;数据库索引也完全无需改变,即可用于保序加密的密文.

保序加密方案的研究工作可以根据其达到的安全性分为 4 个阶段:(1)无严格安全性;(2)有严格安全性;(3)理想安全性;(4)超越理想安全性.对每一个阶段,本小节都将介绍最具代表性的工作.

(1)无严格安全性. Agrawal 等人[21]首次提出了保序加密的概念,并给出了一种不泄露明文概率分布的保序加密方案.该方案以明文已知的明文概率分布和目标的密文概率分布作为输入,保证密文①大小顺序与明文一致,②遵循目标概率分布.这个加密方案分为三个阶段:在建模阶段,利用分段线性样条曲线为明文和密文的概率分布建模;在匀化阶段,把明文集合的数值均匀分布;在转化阶段,根据目标概率分布,把匀化分布后的明文转化为密文.该文通过实验验证了该方案加密后的数据概率分布具有统计意义上的不可区分性.但该工作[21]以及其他不少后续工作[22-24]缺乏严格安全分析,因此这些方案有较大的安全隐患.

(2)严格安全性. Boldyreva 等人[25]首次对保序加密做严格的安全定义和分析.对保序加密而言,最理想的安全性莫过于除了顺序以外不泄露的任何其他信息.该文把这种理想安全性形式化为选择明文攻击下不可区分性(INDistinguishability under Ordered Chosen-Plaintext Attack,简称 IND-OCPA),但随后证明了这种安全性下密文长度必随明文长度而指数增长,因此理想安全性是不实际的.退而求其次,类比伪随机函数,该文给出了一个弱化的安全定义,即选择密文攻击下伪随机保序函数(Pseudorandom Order-Preserving Function under Chosen-Ciphertext Attack,简称 POPF-CCA),并给出了满足这个定义的保序加密方案.对于这种基于伪随机保序函数的保序加密方案,Boldyreva 等人在后续工作[26]中更深入地分析了其安全性,发现每条密文至少会泄露明文的一半比特.文献[27]定义了比 POPF-CCA 更安全的安全性——窗口单向性(WindowOne-Wayness),并构造了满足该安全定义的保序加密方案,保证在明文均匀分布的情况下,不泄露明文的低位比特.以上工作[25-27]的保序加密方案都未达到理想安全性.

(3)理想安全性. Popa 等人[28]首次构造出了达到理想安全性 IND-OCPA 的保序加密方案.该工作之所以能达到理性安全性是因为它与传统的加密方案不同:①交互式,该方案每次加密需要客户端和服务器之间多次通信;②可变的(Mutable),该方案已经产生的密文可在后续的新数据加密时做修改.该方案达到理想安全性的思路很朴素:如果一个明文加密后的值就是这个明文在所有已加密明文中的序号,那么该密文就只泄露顺序信息.为此,该方法的密文编码方式是以平衡二叉树的方式组织,每个节点都代表一个密文(及其对应的明文),左子树的密文对应的明文值总是小于右子树的(从而保持顺序),密文的二进制编码由根节点到它所在节点的路径决定(使得密文就是序号).当加密新明文时,服务器需要与客户端多次通信以确定新节点的插入位置;如果插入新节点后的树需要再平衡,那么部分子树就需要移动,这些子树中的节点对应的密文编码则需要改动.文中用理论和实验证明了密文重新加密的开销为 $O(\log(n))$ 次通信,其中 n 是唯一明文的数量.后续工作[29]构造出了另一种理想安全的构造,通信开销降低到了 $O(1)$,并大大降低了密文需要重新加密的概率,但客户端需要维护 $O(n)$ 大小的状态.

理想安全性,即仅泄露顺序信息,到底有多(不)安全?考虑一个极端情况,如果明文值域的所有明文都被(确定性地)保序加密,则明文和密文可直接一一对应,加密变得毫无意义.在实际数据集上的实证性安全研究[30,31]表明,攻击者可以通过离线的频率分析、关联分析等手段,部分揭露甚至完全解密(理想安全)保序加密的信息.由此可见,保序加密的理想安全性也并不理想.

(4)超越理想安全性.为了超越保序加密安全性的理论上限(即 IND-OCPA),目前有两种推广保序加密的方法.一种是隐藏频率保序加密(Frequency-HidingOrder-Preserving Encryption)[32],其基本思想是把重复明文所对应的密文的顺序随机化,比如明文序列 $X=\{1,2,2,3\}$ 经过隐藏频率保序加密之后的密文序列可能是 $Y_1=\{1,2,3,4\}$,也可能是 $Y_1=\{1,3,2,4\}$.这保证了频率信息的机密性,同时仍可支持大部分查询.另一种推广是显序加密(Order-Revealing Encryption)[33-35],它可视为是多输入功能加密(Multi-InputFunctional Encryption)的一种特例,这里的功能是一个可以比较两个密文所对应明文之间大小关系的函数.由于是通过一个函数、而非直接在密文上比较大小,显序加密放松了对密文结构的限制,因而可以保证密文更高的安全性.以上两种保序加密推广的安全性都优于传统保

序加密的理想安全性，可以有效防范基于频率的离线攻击[30].

3.2.2　基于加密索引的密文范围查询

加密索引把密文的顺序关系以密码学的方法转换到数据结构之中，使得符合查询过滤条件的记录可以通过索引快速地定位和读取.

文献[36,37]提出了桶划分(Bucketization)方法，可以认为是一种粗粒度的索引，它把数据库表的敏感字段的值域划分成若干个相同大小的区域，称为“桶”，并为每个桶分配一个随机的、独特的 ID. 数据库表的每条记录都加密，并为每个加密记录附加其各个敏感字段所在桶的 ID. 用户查询的时候数据库可以根据桶 ID 来筛选记录，这种筛选产生的中间结果难免包含假阳的记录，还需要传输到客户端解密处理得出最终结果. 桶划分方法在服务器端无需解密数据，但字段所属的区间 ID 有可能泄露明文的范围. 而且桶划分的划分粒度难以把握，区间过大，假阳结果多，影响性能；区间过小，区间 ID 泄露的范围信息过于精确，影响安全性.

针对文献[36,37]桶划分方案的安全较性低且通信开销较大的问题，文献[38]和文献[39]提出了树状结构的加密索引，并给出了严格的安全分析. 这两个工作都可以认为是层次式的桶划分方案.

具体来说，文献[38]中的索引结构是一颗平衡的二叉搜索树(树中的每个节点都相当于一个“桶”)，其叶子节点表示一条记录，非叶子节点包含其左子树和右子树记录的并集. 树中每个节点都附带一个布隆过滤器(Bloom Filter)[40]，它是一种空间紧凑的概率性数据结构，允许以 $O(1)$ 时间判断一个元素是否属于一个集合(但可能是假阳). 在布隆过滤器的帮助下，该索引可以快速判断某个子树下是否存在至少一条记录属于查询给定的区间. 该索引的安全性在于满足索引不可区分性(Index Indistinguishability)：(1) 其结构是不可区分的，即满足对任意两个数据集的索引，只要数据集的基数(Cardinality)一样，索引的结构就是一样的；(2) 其节点是不可区分的，树中任意两个节点所附带的(经过特殊设计的)布隆过滤器在概率分布上是没有差别的. 该索引的缺点是假阳，这是由布隆过滤器的本质属性决定的；作为一种弥补措施，假阳的概率可以通过调整布隆过滤器的参数来控制.

与文献[38]的思路完全不同，文献[39]把范围密文查询转换成了多关键词密文查询(见 3.3 小节“关键词密文查询”)：先利用区间覆盖技术(Range Covering)把值域在多个层次上划分为多个区间(有重叠)，并为每个区间分配唯一的关键词；然后把记录看作文档，记录所属的多个区间就相当于文档包含的多个关键词，这样对记录的范围查询可以等价转换为对文档的多个关键词查询. 关键词密文查询的实现通常都依赖于某种索引结构；这里可以简单地认为，这种索引的键是关键词，其值是包含关键词的文档集合(这里的“桶”). 该文给出了基于不同区间覆盖技术的多个构造，并分析和比较了其在安全和性能上的优劣. 该文所提出的方案的一个重要优点是关键词密文查询方法可以当做黑箱来使用，因此任何关键词密文查询的进展都可以自动改进该方案.

前述工作[36-39]的一个隐含假设是，用户范围查询之前加密字段都已经预先建立好了密文索引. 但在实际应用场景中，用户不得不考虑数据库索引所带来的存储和维护开销，预先为所有加密字段建立密文索引在实际应用中是很不方便的. 为了解决这个问题，文献[41]将数据库领域的自适应理念引入到了密文索引的研究上，提出了适用于列数据库的[42]、支持范围查询的自适应密文索引(Adaptive Indexing). 该文提出的密文索引在自适应方面是沿用了名为数据库分裂(Database Cracking)[43]的自适应索引技术：数据库存储的物理组织方式是按列的，且在最初存储时是无序的；之后在处理用户查询时，根据范围查询的区间端点逐渐分裂成片段，片段间有序，片段内无序；各个片段的端点及其对应记录的定位由一个自平衡二叉树(AVL Tree)维护，用于缩小区间查询的查找范围. 为了在密文上实现数据库分裂，该文提出了一种基于线性代数操作的可索引加密(Indexable Encryption)方法，它由两种互补的加密模式 A 和 B 组成，允许模式 A 和模式 B 的密文互相比较大小，但两个模式 A 的密文之间无法比较大小，同理模式 B 也不允许. 这种加密方法的特殊性质使得数据库中的记录(使用模式 A 加密)之间无法比较大小，但允许用户查询的区间端点(使用模式 B 加密)和数据库记录比较大小. 该工作的主要不足是基于可索引加密的自平衡二叉树在结构上会泄露信息，对此作者没有给予严格的分析和论证.

以上密文索引都是针对一维范围查询的，直接应用在多维范围查询上有性能和安全的局限. 针对这一问题，文献[44]提出了一种层次式密文索引，该索引在结构上与 R 树[45]相同. 在这个加密 R 树的

每个节点，密文查询算法需要判断该节点的最小边界框(加密的)是否与查询区域的超矩形(也是加密的)有交集.为了解决这个问题，作者利用了非对称点积保持加密(Asymmetric Scalar-Product Preserving Encryption)，这种加密方法首先提出并应用于加密的 K-近邻查询[46].这种加密方法的主要性质是，给定加密后的数据点 $P1$、$P2$ 和加密后的查询点 Q，可判断 $P1$ 和 $P2$ 哪个离 Q 更近.正是基于这个性质，该文密文索引中的最小边界框是否与查询区域相交的问题可以得到解决.在查询时，查询算法从 R 树的根节开始，用查询区域的超矩形与节点的最小边界框取交集；如果有交集，则查询递归执行，否则返回；最终返回所有存在交集的叶子节点的数据库记录.该方法在机密性上优于保序加密，但其索引结构在一定程度上会泄露顺序信息.对此问题，该文作者提出了用调整参数的方法来在安全性与效率之间保持平衡.

3.3 关键词密文查询

数值类型的密文数据需要支持范围查询，文本类型的密文数据则需要支持文本匹配查询，其中最基本的一种是查找包含某一个关键词的所有密文记录，也即关键词密文查询.在本小节中用“文档”指代记录，以强调记录的类型是文本；相应地，用“文档集”指代数据表.

本小节介绍四种具有代表性的对称可搜索加密(Symmetric Searchable Encryption)，可支持关键词密文查询(关于非对称可搜索加密的讨论可见综述[47])从索引角度看，这四个工作的区别是：第一个工作[48]不要求建立任何索引，查询处理需要扫描文档集中的每个文档；第二[49]和第三个工作[50]为每篇文档单独建立索引，因而查询的工作量与文档集的大小成正比，但不需要扫描每篇文档；第四个工作[51]为整个文档集建立索引，查询处理的工作量只与结果集的大小有关，与文档集的大小无关.

Song 等人在文献[48]中首次提出了关键词密文查询问题，并构造了一种简单实用的可搜索加密方案.该文中的方案将一篇明文文档中的每个词都分别转换成一个对应的双层加密结构：当用户查询的时候，会把检索词的陷门(Trapdoor)发给服务器，服务器可以尝试用这个陷门解开一篇密文中每个词的双层加密结构的外层加密，然后检查内层形式是否正确.如果形式正确，说明这个双层加密结构对应的词就(非常可能)是检索词；否则，说明对应的词不是.最后，服务器把所有包含检索词的文档返回给用户.这个方法实现简单，几乎无额外存储开销，但查询时必须遍历文档中的每一个词，且会泄露与检索词匹配的数量和位置.

为每篇文档建立索引可以快速判断文档是否匹配，且不泄露不必要的匹配信息；基于这个观察，文献[49,50]两个工作都基于索引实现可搜索加密.文献[49]首先提出了安全索引，并为安全索引定义了适应性选择关键词攻击下不可区分性(INDistinguishability against Adaptive Chosen Keyword Attack，后简称 IND1-CKA)的安全模型，然后构造出一个基于伪随机函数和布隆过滤器的、满足 IND1-CKA 的高效安全索引.该索引，简单地说，就是把一个文档中词是否出现的信息存储在一个特殊的布隆过滤器中；判断检索词是否在一个文档中出现也就是判断检索词的陷门是否满足文档对应的布隆过滤器.这里的布隆过滤器[40]的特殊之处在于，(1)必须用词的陷门来操作，陷门是以词和密钥为输入的伪随机函数产生，而只有用户拥有合法的密钥，能生成合法的陷门；(2)经过盲化(Blinding)处理，即向布隆过滤器插入一定数量的随机 1，使得任意两个长度相同的的文档，即使关键词数量不同，它们对应的布隆过滤器也不可区分.文献[50]提出了一种基于字典的安全索引，由伪随机比特序列作掩码，只有在用户查询的时候可以有选择性地恢复索引的一部分，剩下的部分仍然保持伪随机性.与文献[49]相同的是，这一安全索引也只需要 O(1)时间就可以判断一篇文档是否包含某一个检索词；不同的是，这个方法返回的结果集，不仅召回率是 100%，正确率也是 100%.

文献[51]为非适应性和适应性的攻击者分别构造了可搜索加密方案，并证明了它们的安全性.篇幅所限，这里只简要介绍他们提出的非适应安全可搜索加密方案.与文献[49,50]不同，这个工作建立的索引是针对整个文档集，而不是单个文档.因此，该方案为单位结果文档付出的服务器计算开销是 $O(1)$，优于前述的可搜索加密方法.文档集 $D=\{D_i\}$ 的可搜索加密索引由数组 A 和查询 T 组成.设 $W=\{w_i\}$ 是 D 中出现的所有关键词的集合；$L=\{L_i\}$ 是关键词结果集的集合，其中 L_i 是由包含关键词 w_i 的文档 ID 组成的链表，也即对任意链表 L_i 中的任意节点 $L_i[j]$ 都有 $w_i\in D_{L_i[j]}$.我们把 L 中的所有链表 L_i 打散，每个节点 $L_i[j]$ 用随机产生的密钥加密，然后随机写进数组 A 中.在节点 $L_i[j]$ 被加密前，扩充这个节点的信息，把 $L_i[j+1]$ 在 A 中的位

置和密钥也保存下来. 这样，只要知道 L_i 在 A 中的第一个节点的位置和密钥，就能解密整个 L_i. 而这个关键词 w_i 所对应的结果集链表 L_i 在 A 中的位置和密钥可以通过查询表 T 得到，这里 A 的数据结构是 FKS 字典[52]，后者保证查看任何关键词的最坏时间复杂度是 $O(1)$，存储复杂度是 $O(n)$，其中 n 是关键词的数量.

3.4　复合密文查询

目前没有任何一种加密方法能在性能开销可接受的条件下支持所有种类的数据库操作. 本节前面介绍了多种加密方法或协议，它们在功能、安全和性能方面上做出了不同的取舍权衡：概率性加密，泄露信息最少，但不允许任何操作；确定性加密，支持判等操作，但泄露了概率分布的信息；全同态加密，支持任何计算，但性能开销极大；半同态加密，支持加法(或乘法)，但不支持其他操作；保序加密，支持比较操作，但泄露顺序信息；加密索引，支持范围或关键词查询，不同的构造各有千秋，不支持算术运算；其他精巧的密码学构造，如可索引加密、非对称点积保持加密等等，也仅适用于特定的查询需求.

复合数据库查询，涉及多种数据库操作，这对本节前面介绍的各种密文查询方法带来了新挑战. 比如，SQL 查询 SELECT SUM(quantity) FROM orders WHERE quantity>100，在 quantity 列上不只需要范围查询，同时还要计算求和，这要求要么一种加密方法能同时允许比较和求和，要么需要某种方式“组合”使用两种加密方法. 再如，SQL 查询 SELECT p.name, o.quantity FROM products as p and orders as o WHERE p.id=o.id，涉及的连接操作要求在两个不同表的列之间做判等，这两个列的加密方法可能不同(出于功能考虑)，亦或者加密方法相同但密钥不同(出于安全性考虑)，这就要求不同的密文列之间能“兼容”，即允许密文操作的多个输入数据属于不止一种加密方法. 因此，为了克服单一加密方法在处理复合查询上的局限，不同加密方法必须能在一个数据库系统中有效地“组合”起来，甚至达到“兼容”的程度.

对于“组合”问题，文献[53]采用了洋葱加密(Onion Encryption)来组合不同功能的加密方法：对于一个数据字段，用多种加密方法以嵌套的方式逐层加密，越外层的加密方法的安全性越强，而功能越弱. 比如，RND(DET(JOIN(plaintext)))就是一种可能的洋葱加密，其中 RND 表示概率性加密，DET 表示确定性加密，JOIN 表示支持连接操作的加密；最外层的 RND 加密的安全性最强，而最内层的 JOIN 加密的功能更强. 数据最初加密时可用洋葱加密层次中尽可能外层的加密方法加密，以最大化安全性，之后查询处理中按需“剥开”外层加密，“露出”更强功能的内层. 为了满足多种功能，某一个列可能需要维护不止一套洋葱加密. 比如，对于上段中提到的第一个 SQL 查询例子，可以维护两套洋葱加密，最内层分别是保序加密和加法同态加密.

对于“兼容”问题，文献[53]提出了可调整连接(Adjustable Join)，文献[54]提出了数据可互操作性(Data Interoperability)；虽然两者的具体问题与方法都不尽相同，但本质上都采用了同样的技巧——构造加密方法使得服务器端不需要明文即可用新密钥对数据重新加密. 在文献[53]的可调整连接的方法，对于两个需要用相等关系连接的列，由于需要支持判断等，两个列都采用同一种确定性加密方法，但使用不同的密钥，以隐藏两个列的关联；当在这两个列上做连接操作时，对其中一个列使用另个列的密钥重新加密，这样重加密之后由于密钥相同，确定性加密保证两个列明文之间相等关系在密文上保持，因而可以直接在密文上进行连接操作. 在文献[54]中提出的数据可互操作性是指，一个操作的输出可以被另一个操作直接用作输入，即使这两个操作使用的加密方法不同. 该文给出了一种受安全多方计算(Secure Multiparty Computation)[55]启发的、具备数据可互操作性的密码学协议. 该方法中利用了一种基于 RSA 的加密方法，虽然其本身只是乘法同态加密的，但通过在必要时换密钥重新加密，可以在效果上支持加法、比较和连接等. 以上两种方法都使得一个操作可以接受不同方式加密的数据作为输入.

4　可信硬件(4 级)

可信硬件是一种安全增强的的硬件，能隔离其内部的计算和存储资源，使得攻击者无法从外部窥探和篡改可信硬件内部的状态，包括代码和数据. 因而，密文载入到可信硬件后中可解密处理，无需担心其机密性受到影响. 由于可信硬件中的内存数据和程序状态受到保护，因此可信硬件可有效地降低 3 级和 4 级机密性风险.

可信硬件在云环境下的优势来自其二重性. 从逻辑上讲，可信硬件服务于用户. 作为用户的延伸，它可以安全地存储用户的敏感数据，安全地运行用

户的应用程序.从物理上看,可信硬件处于不可信的服务器上.作为服务器的组件,它与服务器直接相连,通信延迟低、带宽高,而且可以被多个租户共享使用,摊销成本.这两方面的特点使得可信硬件成为解决云安全问题的一个有吸引力的选项.

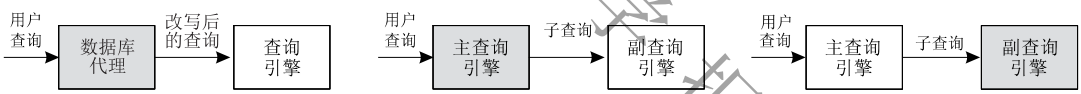
虽然可信硬件有其优点,但并不能取代密文查询技术;两者的关系不是互斥,而是互补.从安全角度看,可信硬件的安全保护比密文查询更全面:不只可以保护数据库中的数据,数据库执行状态的机密性和完整性也可以得到保障.但问题是可信硬件中的存储资源和计算能力都相对有限,如果将所有查询处理都放在可信硬件中,而不充分利用不可信硬件,那么系统整体性能表现难以达到最优.出于可信硬件与不可信硬件之间负载平衡的考虑,基于可信硬件的安全云数据库系统上也需要利用密文查询技术把部分工作负载放在不可信硬件上完成.

基于可信硬件的安全云数据库系统有多种可能的架构(见图 2);根据架构的不同,可信硬件扮演的角色也有差异:

架构 a[53,56]由可信的查询代理和不可信的查询引擎组成.可信的查询代理,位于可信硬件上,负责执行敏感操作,如接收和改写用户请求,管理用户密钥,对密文中间结果解密并处理,执行密码学协议等.而查询引擎是不可信的,运行在不可信数据库服务器上,它处理的所有敏感数据都必须是加密的以确保安全.这种架构的主要优点是,可信代理可以同时适配数据库客户端以及数据库管理系统,使得对客户端和数据库系统的修改最小化,有助于实际采用和部署.

架构 b[57]由可信的主查询引擎和不可信的副查询引擎组成.可信的主查询引擎,处于可信硬件中,负责接收接收用户查询、查询优化、查询调度和密钥管理等,其中的数据可以是明文形式处理.而不可信的副查询引擎,运行在不可信服务器上,其主要工作是分担主查询引擎的工作负载,完成主查询引擎分配的对密文的子查询或子任务.相比架构 a,架构 b 的主要优点是主查询引擎受到保护,其执行状态和控制逻辑不会被窥探和篡改.

架构 c[58]由不可信的主查询引擎和可信的副查询引擎组成.不可信的主查询引擎负责密文查询的主要处理流程,在必要时把密文子查询或子任务交给可信的副查询引擎处理,综合其结果并返回给用户.与架构 b 比较,架构 c 的优点是位于可信硬件中的功能相对简单的副查询引擎,有助于降低对可信硬件中的资源要求,简化可信硬件相关的开发.



(a) 架构a (b) 架构b (c) 架构c

图 2 基于可信硬件的云数据库系统的三种架构(其中灰色方框的组件处于可信硬件中)

目前,适用于大规模数据处理的可信硬件技术主要有三种:安全协处理器、现场可编程阵列和软件保护扩展.本小节下面将介绍这三种可信硬件技术的特点和安全性,并讨论它们在云环境下尤其是在安全云数据库系统中的应用前景.

4.1 安全协处理器

安全协处理器(Secure Coprocessors)是一个能抵御物理和逻辑攻击的通用计算环境,可以安全地加载和运行用户程序,安全地存储用户数据.它们通常以扩展卡的形式直接插在计算机主板的扩展槽上,作为(不可信)主机上 CPU 的协处理器,提供安全的计算环境,加速在主机上高耗时的密码学操作.大部分关于安全协处理器的应用性研究都是基于 IBM 的安全协处理器,比如 IBM 4765[59].IBM 的安全协处理器得到美国政府的认证,满足联邦信息处理标准(Federal Information Processing Standard)的最高安全级别的要求.除了安全机制,IBM 4765 的硬件逻辑专门实现了各种密码学操作,如加密解密、哈希函数、数字签名、随机数生成等.

安全协处理器的应用领域广泛,这里只介绍在数据库安全方面的相关研究.文献[60]从安全理论的角度对安全数据库的需求做了详细的分析,认为程序混淆(Program Obfuscation)对安全数据库是必要的,而鉴于完全基于密码学的程序模糊被认为是很困难的,安全协处理器可以起到程序模糊的作用.文献[61]从数据库即服务的角度也认可引入安全协处理器的必要性.该文认为桶划分方法(见 3.2.2 小节)有局限性,因此提出有必要在云数据库中整合安全协处理器的建议,克服桶划分方法的不足.文献[62]提出用安全协处理器实现主权连接(Sovereign Join),即在多个主权数据库(指管理权相互独立的)之间进行数据库的连接操作,既达到信

息整合的目的同时又保证不泄露每个主权数据库(除了结果集以外)的数据.

那么安全协处理器到底是否适合用于保护云数据库系统呢?文献[57]试图回答这一问题.传统观点认为安全协处理器只适合自动取款机等少数应用场景,这是因为安全协处理器(1)计算和存储资源十分有限,这主要受限于其防篡改包装的散热能力;(2)购置成本高昂,以 IBM 4764 为例,其零售价格高达 8000 美元(2011 年).但经过综合的成本分析之后,该文得出结论,在安全协处理器中单位数据库事务的成本比单纯在 CPU 中计算(在不可信服务器或者可信客户端)低一个数量级.这一方面是由于云数据中心的规模效应降低了安全协处理的摊销成本,另一方面要"归功"于直接在加密数据上做计算的极高计算开销,即使是非常简单的加法运算也至少相当于做大数模乘(Modular Multiplication)[13].

总的来说,我们认为安全协处理器是可信硬件用于云数据库系统的一种早期尝试.这方面的研究工作验证了可信硬件的技术路线是有潜力的,但我们认为安全协处理器在性能上有较大的瓶颈,除非物理攻击是主要安全顾虑之一,否则接下来将介绍的现场可编程阵列和软件保护扩展更有优势.

4.2　现场可编程阵列

现场可编程阵列(Field-Programmable Gate Arrays,后简称 FPGA)是一种在制造后可被客户配置的集成电路,其内部主要组件包括可编程的逻辑块,可配置的内部连线,以及可匹配外部电器要求的 I/O 块.因其灵活性和可定制性,FPGA 非常适合硬件的原型设计.并且,随着 FPGA 的性能、功耗和成本等方面的优化,FPGA 的应用范围还在不断扩展.其他关于 FPGA 发展历史、技术原理和未来趋势的分析见文献[63].

数据库界对 FPGA 的早期兴趣是把 FPGA 当作数据处理的加速器.文献[64]是这方面探索的最早工作之一,文章重点讨论了在 FPGA 上编程的难点和挑战.文献[65]利用 FPGA 的高并行性做数据集的频繁项集计算,取得了比 CPU 高数倍的吞吐量,并讨论了三种不同实现方案的利弊.文献[66]介绍了一个流查询到 FPGA 的编译器,给定一个流数据查询计划,编译器可以自动把它翻译成 FPGA 电路.文献[67]利用 FPGA 的高并行性做 XML 过滤,达到的吞吐量最多可以比软件实现高三个数量级.文献[68]介绍了如何在 FPGA 上实现高效的排序网络,并以此为例给出了在 FPGA 上做数据处理的指导方针.文献[69]设计了一个基于 FPGA 的数据库分析型操作的加速引擎,取得了 6 倍的性能提升.

不仅可作为加速器,更重要的是,FPGA 还可视为一种可信硬件.作为一个为特别目的定制的电路,FPGA 只提供有限的、定义良好的攻击面,比软件系统更难被攻破.与通用处理器不同的是,FPGA 中的内存空间是高度隔离的,"程序"空间通常通过配置逻辑单元和连接网络来定义,而"数据"空间则是由块内存的内容和触发器的状态来表示;这样就不会像软件系统因为缓冲区溢出的漏洞而被攻击者修改程序.而且,FPGA 厂商的产品(比如 Xilinx 7 系列)通常都提供 FPGA 比特流加密的功能,用户可以把 FPGA 比特流用 AES 加密,FPGA 在配置的时候用内部存储的密钥在线解密 FPGA 比特流.这个密钥是存储在板上的非易失性内存中,可以通过外部接又复写,但不能读取.解密、验证、装载 FPGA 比特流都是由 FPGA 上专门预设的启动逻辑完成的;与 FPGA 中的其他逻辑不同,这部分是无法更改的.FPGA 比特流加密的初衷是防止用户的知识产权被窃取,而在云环境下可以确保 FPGA 比特流的完整性.以 FPGA 作为可信硬件,文献[70]勾勒了基于 FPGA 的可信云计算,并以一个医疗应用具体说明了这种技术路线的可行性.文献[58]描述了 Cipherbase,第一个基于 FPGA 的安全云数据库系统,该系统如何有效结合软件与硬件来实现数据库的机密性保护会在 6.2 小节详细描述.

近几年,在云数据中心部署 FPGA 已经成为一种趋势,而这种趋势的最主要推动力并不是性能或安全,而是能耗.受限于能耗密度,扩展 CPU 核数来提高性能将难以为继,因此越来越多芯片上的计算单元将成为所谓的"暗硅"[71].鉴于这种趋势展望,利用率低但性能-功耗比高的专用计算单元(如 FPGA)变得更有吸引力.可重构计算[72]或富加速器架构(Accelerator-Rich Architecture)[73,74]的研究都看好 FPGA 未来被云数据中心大规模采用,虽然也指出仍有许多技术挑战需要面对和许多顾虑需要解决(比如低利用率、窄工作负载覆盖面,高设计费用和不友好的编程接又).学术界看好,工业界则已经开始行动.微软为 1600 多台 Bing 搜索服务器配备了 FPGA,大幅提高了性能-功耗比[75].百度也已经在处理深度学习任务的生产系统采用 FPGA 而不采用 GPU,其主要原因就是 FPGA 的能耗更低[76].

综上所述,我们认为 FPGA,因其在性能、安全和能耗等方面的优势,未来在云计算以及云数据库

系统等方面有良好的应用前景.

4.3 软件保护扩展

软件保护扩展(Software Guard Extensions,后简称 SGX)是英特尔 x86 指令集架构的一个扩展,该扩展使得应用程序可以创建受保护的运行时容器,称作"围圈(Enclave)".围圈是一个进程的地址空间中的一个块受保护的内存区域,该区域与外界隔离,即使是特权程序(如操作系统和虚拟机管理器)也无法窥探和篡改该区域内的代码和数据;也就是说,围圈的机密性和完整性是得到 CPU 保证的.关于 SGX 的全面介绍详见文献[77].

作为一种可信硬件技术,SGX 有诸多优点.首先,SGX 有比较好的兼容性:对于应用程序开发者来说,围圈对受保护的应用程序的 CPU 指令只有少数(虽然很重要的)的限制;在系统软件和编程框架的帮助下,这些限制对开发者会变得透明,因此易于开发 SGX 保护的应用(如数据库系统).其次,SGX 的一个设计目标就是虚拟化,即允许多个围圈同时运行,共用处理器上的资源.这为开发在云环境下基于 SGX 的多租户安全应用程序提供了便利.在这方面形成鲜明对比的是安全协处理器和 FPGA,这两种硬件难以虚拟化,不便于多个租户共享.再次,SGX 的性能开销较小:虽然 SGX 需要对内存加密解密,但依赖于 CPU 中为 SGX 专门设计的内存加密引擎,SGX 在 CPU 密集型基准测试上只平均产生 5%的额外性能开销[78].最后,考虑到英特尔在 CPU 的数据中心细分市场的领导地位,SGX 有望在数据中心的服务器上普及.因此,SGX 这种可信硬件技术无需云服务提供商付出额外的硬件购置成本,这点相比安全协处理器和 FPGA 是非常有吸引力的.

虽然 SGX 于 2016 年年初才正式推出,但学术界和工业界已经在诸多应用场景下尝试基于 SGX 的安全解决方案;这里着重介绍 SGX 在保护云环境下数据库系统或大数据系统的相关工作.文献[79]介绍了 VC3,该系统利用 SGX 保护用户提交的 Hadoop 分析任务的代码和数据的安全性.VC3 解决的一个主要挑战是,将 Hadoop 系统划分为可信与不可信的两部分,使得(1)只有用户代码加上少量公共辅助代码作为可信计算基(Trusted Computing Base)位于围圈中,(2)Hadoop 系统本身无需修改.同样是针对 Hadoop 生态系统,文献[80]介绍了 SecureKeeper,一个基于 SGX 的安全增强版的 ZooKeeper 14 ,后者负责 Hadoop 中各个服务之间的协调和同步.SecureKeeper 采用了一种新颖的多围圈架构,使得在 SGX 中可信计算基只有约 3000 行代码,而对 ZooKeeper 本身的代码几乎没有改动.除了 Hadoop 的组件以外,关系型数据库系统 SQL Server,以及键值型数据库 Memcached 和 Redis,也分别在文献[81]和文献[82]作为目标应用之一被移植到 SGX 环境下.这两篇文献都介绍了如何基于库操作系统(Library OS)把任意应用程序移植进 SGX 围圈中.以上四个工作代表了利用 SGX 保护应用程序的两种基本思路:(1)面向特定应用,如文献[79-80],(2)面向任意应用,如文献[81-82].前者的主要优点是,由于考虑到了应用的特殊性,性能和安全性更好;而后者的强项在于开发成本低,把对应用程序的代码修改降到最低.以上工作说明利用 SGX 保护云数据库系统是可行的.当然,SGX 可以作为实现安全云数据库的基础,但有些其他类型的攻击是 SGX 本身不能解决的,比如 Iago 攻击[83]和回滚攻击等[84].

综上,我们认为 SGX 是一项有前景的可信硬件技术,但 SGX 出现时间尚短,现有的 SGX 保护的数据库系统并不成熟,在性能和安全方面仍有局限.SGX 与数据库管理系统如何有效结合仍然是一个值得研究的问题.

5 访问模式保护(5 级)

访问模式保护技术使得程序可以在"不暴露意图"的前提下读写存储中的数据."不暴露意图"令攻击者难以从访问模式(Access Patterns)推测出敏感信息.这里说的访问模式,是指程序对存储器的一系列访问所泄露的信息,包括命令(读或写)、地址和数据(既可能是密文也可能是明文).即使利用密文查询或可信硬件对数据做加密,命令和地址仍然是公开的.数据库访问模式的信息泄露看似无害,但如 2.4 小节所说,现有工作已经证明攻击者可以从数据库的访问模式推测出敏感信息[10,11].

虽然存在针对特定数据结构或算法的访问模式保护技术的问题,但受限于篇幅,本节只介绍两种通用的访问模式保护技术:私密信息检索和输入无关内存.这两种密码学方法均能保证攻击者不能从程序的访问模式中获得任何有效信息,区别是前者的数据是不要求加密,而后者的数据必须是加密的.

5.1 私密信息检索

私密信息检索(Private Information Retrieval,

后简称 PIR)是在不暴露查询本身的情况下在远程数据库上检索并取回数据. 在 PIR 的场景中,数据库是公开的,数据不需要加密;要保护的并不是数据的机密性,而是用户查询(也就是访问模式)的隐私性. 适用于 PIR 的应用场景是很多的,比如 X.509 证书服务器上的证书是公开信息,但用户对证书的请求可能泄露它正访问网站的信息[85],再如基于地理位置的服务(Location-Based Service)的兴趣点(Point of Interest)信息是公开的,但用户位置的信息却涉及隐私[86]. PIR 也可以运用在云数据库系统的场景下,如果云数据库里包含一些公开数据,但用户希望保护其对公开数据查询的隐私性.

PIR 是 Chor 和 Goldreich 等人于 1995 年首次提出的[87]. 问题的一个常见提法是,给定一个数据库 x,它由一个长度为 n 的二进制串 $x_1x_2\cdots x_n$ 组成,并在 k 台不互相通信的服务器上都存有副本;用户有一个索引 i,他希望得到数据库中 x_i 的值. 为此,用户给所有 k 个服务器都发送查询并收到其响应,其中第 j 个服务器收到的查询是 q_j,其返回的响应是 $Ans_j(q_j)$. 最后,用户根据 $Ans_1(q_1)$ 到 $Ans_k(q_k)$ 求出 x. 以上过程的通信复杂性是 $\sum_{j=1}^{k}|q_j|+|Ans_j(q_j)|$. 设计 PIR 方案就是构造 q_j 和 $Ans_j(q_j)$,使(1)用户能最终获得 x_i,(2)数据库不能据此推测出关于 i 的信息,(3)通信开销尽可能小. 一个平凡的 PIR 方案是,令数据库把所有数据都发给用户,这显然是正确的. 但这样个平凡 PIR 的通信开销 $O(n)$. 那是否存在开销更小的方案呢?

(1)信息论意义安全 PIR. Chor 等人在文献[87]中证明任何从信息论意义上(假定攻击者具有无限量的计算资源)安全的单服务器 PIR 方案($k=1$)都有 $\Omega(n)$ 通信复杂度;也就是说,前面提到的平凡 PIR 已达到最优的通信复杂性. 更进一步,多服务器 PIR 方案($k>1$)是可以做到优于线性的通信复杂度的,Chor 等人提出了 $O(n^{1/k})$ 通信复杂度的方案. 后续研究对多服务器 PIR 的改进,见综述[88].

(2)计算意义安全 PIR. Chor 和 Gilboa 在文献[89]中提出把对 PIR 的安全性由信息论安全降低到计算安全,即对攻击者计算能力的假定从"无限量"降低到"多项式",以期实现通信复杂度更优的 PIR;这种计算安全意义下的 PIR 称为计算性私密信息检索(Computational PIR,后简称 cPIR). 他们基于伪随机数生成和单向函数存在的假设下构造出通信复杂度 $O(n^{\varepsilon})$ 的双服务器 cPIR 方案,其中 ε 可以任意小. 令人意外的是,安全性从信息论意义放松到计算意义还使得单数据库的 PIR 成为可能. 之后,Kushilevitz 等人在文献[90]中基于二次剩余(Quadratic Residuosit)假设提出了第一个单服务器的 cPIR 方案,通信复杂度也是 $O(n^{\varepsilon})$,其中 ε 可以任意小. 这之后,又陆续有其他工作提出单服务器 cPIR 方案,进一步优化通信复杂度. 关于单服务器 cPIR 的其他分析见文献[91].

虽然 PIR 问题已经取得了很大进展,但我们不得不承认 PIR 离大范围实用还有很大距离. 早期 PIR 工作的优化目标都是通信复杂性,对服务器端的计算复杂性只要求是多项式的即可. 2007 年,文献[92]对单服务器 cPIR 实用性的研究表明,cPIR 仍然比传输整个数据库到客户端慢几个数量级,并指出主要性能瓶颈在服务器的计算开销,呼吁后续工作把重点放在优化服务器端的计算复杂性. 2012 年,文献[93]中回顾了 cPIR 实用性问题,他们分析了文献[94](2008 年)中提出的单服务器 cPIR 的性能,得出了与之前相反的结论:cPIR 比传输整个数据库快 1～3 个数量级. 但文献[95]指出,根据文献[93]的结果,对一个 28 GB 大小的数据库,以家用带宽访问一个数据块也要耗时 25 分钟. 这说明,对大型数据库,PIR 还远未达到实用程度.

5.2　输入无关内存

输入无关内存(Oblivious RAM,后简称 ORAM)[96]是由 Goldreich 于 1987 年提出的概念,其最初的目标是隐藏程序对内存访问的真实意图以使得程序更难以被逆向工程. 由于各种存储器的访问接又的相似性,ORAM 也同样适用于外存和文件系统等. 显然,ORAM 在安全云数据库系统中是有用武之地的,因为后者依赖于内存、外存、文件系统等来存取数据.

ORAM 的安全性,简而言之,就是保证对存储的访问模式与程序的输入无关,这样存储的访问模式就不会泄露程序输入的信息(比如用户的查询和数据). 下面我们以保护对外存的访问模式为目标,给出 ORAM 的安全性定义. 假设在一台不可信服务器上,有一个用户程序运行在可信硬件中,该用户程序需要访问不可信硬件上的硬盘,后者接受以块为单位的读写命令 $read(bid,data)$ 和 $write(bid,data)$,分别表示读和写一个数据块,其中 bid 表示块序号. 令

$$\vec{y}=((rw_1,bid_1,data_1),\cdots,(rw_m,bid_m,data_m))$$

表示一个长为 m 的操作序列,其中 rw_i 表示操作是

读或写，bid_i表示块序号，$data_i$表示读出或写入的数据.这里的操作序列 $\vec{y}$ 就是可信用户程序对不可信存储的访问模式，攻击者可以监视到.为了隐藏操作序列 $\vec{y}$ 的真实意图，一个 ORAM 构造 A 通过把用户的原始操作序列 $\vec{y}$ 转换成一个新的操作序列 $A(\vec{y})$.我们说一个 ORAM 构造 A 是安全的，如果对于任意两个同样长度的操作序列 $\vec{y}$ 和 $\vec{z}$，它们被 A 转换后的操作序列 $A(\vec{y})$ 和 $A(\vec{z})$ 是计算不可区分的(Computationally Indistinguishable).

直觉上看，ORAM 的“任何长度相同的操作序列不可区分”的安全定义要求不可信服务器：(1)不能区分操作是读还是写，因此读操作也必须附加写操作，写操作也必须有读操作；(2)不能理解数据块的内容，因此数据块都需要加密；(3)不能发现同样的块被再次访问，因此数据块被访问一次之后必须被重新加密(每次加密后的密文不同)；(4)不能区分数据块之间“冷热”，因此所有数据块必须“看起来”被均匀地访问；(5)不能推测出访问顺序(如顺序或随机)，因此逻辑上连续的块必须在物理上打乱位置存储.

许多研究工作致力于设计高效的 ORAM 构造，它们几乎都把数据分层次组织，并可以分为两大类：哈希层次法和树状层次法.

(1)哈希层次法. Goldreich 和 Ostrovsky 在 1996 年发表了一篇 ORAM 奠基性论文[97]，对后续研究影响深远.文中，他们提出了一种经典的哈希层次法的 ORAM 构造，有 $\log(n)$ 层，每层都是一个哈希表，第一层的大小是 $O(1)$，下面每一层的大小都是上一层的 2 倍.数据块从第一层开始插入，一旦一层满了，就整层打乱重排之后归并到下一层.为了隐藏真实的访问意图，要访问 ORAM 中的一个数据块，必须每个层都访问一个数据块，且数据块一旦被访问就要被重新安置.块重新安置的一个简单办法是把这条块从原位置移除，重新加密后插入到第一层.这个 ORAM 构造的平均操作时间开销是 $O(\log^3 n)$，存储要求是 $O(n\log n)$.后续有许多工作[98-101]都沿着上述基于哈希的层次式 ORAM 的框架进行改进，其中开销最低的是 Kushilevitz 等人[101]提出的 ORAM 构造，它的时间摊销开销是 $O(\log^2(n)/\log\log n)$.哈希层次法的一大弊端是需要周期性的整层打乱重排，这导致虽然访问每条记录的摊销开销是低于线性的，但最坏开销仍然是 $\Theta(n)$的；对实际应用来讲，这意味着系统会周期性地出现高延迟，这是令用户难以接受的.虽然后续有工作[102-104]改进了哈希层次法的最坏开销，但要么增加其他开销，要么技巧较为复杂.

(2)树状层次法. Shi 等人在文献[105]提出了树状结构的 ORAM，不仅免去了周期性整层打乱重排的需要，而且概念上更简明.在这里，ORAM 不再是多层的哈希表，而是一颗二叉树，每个节点都是一个最多可以包含 $\log n$ 个数据块的桶，一共有 $\log n$ 层；桶的具体实现是可选的，比如一个简单的数组，或是一个简单的 ORAM(如文献[97]中的“平方根”ORAM 构造).要读写一个数据块，首先通过查找位置表(Position Map)找到它所在的路径(从根节点到叶子节点)，然后遍历路径上的所有节点桶找到数据块，最后把块重新加密后移到根节点桶，并随机赋给它一个新的路径.为了保证各节点的桶(有很大概率)不溢出，每一轮操作，都从每层随机选择两个节点的桶，并各弹出其中的一个数据块，然后将移除的数据块移到它的路径中的下一层节点中.这个树状层次 ORAM 的最坏开销是 $O(\log^3(n))$，需要 $O(n\log n)$的存储开销.受这个工作的启发，后续又有诸多树状的 ORAM 被提出[106-110]，其中特别值得一提的是 Path ORAM[109]，其实现非常简单，开销仅为 $O(\log n)$.

ORAM 正在走向实用.文献[111]以优化实际性能为目标(而非渐进复杂度)，大幅降低了 ORAM 的实际开销，实验结果表明他们改进的 ORAM 在 1TB 数据集上的平摊开销是 20 倍 ～35 倍(相对于一次普通方式的访问).文献[112]解决传统 ORAM 不能支持并发操作的问题，他们改进后的 ORAM 可以在 1TB 数据集上每秒处理数个请求.文献[95]搭建基于 ORAM 的大规模分布式存储系统 Shroud，首次在数据中心级规模评估了 ORAM 的实用性.文献[113]设计和搭建了名为 ObliviStore 的高性能、分布式 ORAM 存储系统，并在 11 台机器上达到了 31.5 MB/s 的吞吐量.ObliviStore 的后续工作[114]又针对高峰负载优化了响应时间.由此可见，随着 ORAM 的性能提升，未来拥有大规模数据的实际系统，包括云数据库系统，都有可能采用 ORAM 以保护访问模式.

6 安全云数据库系统

虽然前面介绍的密文查询、可信硬件和访问模式保护等三种技术是云数据库机密性保护的关键，但仅凭这些技术无法构成完整的云数据库系统.本

节以四个最先进的安全云数据库系统为例，介绍安全云数据库系统的系统架构和设计原则.需要指出的是，这些安全云数据库系统的机密性保护均 3 级或 4 级，目前尚未见到达到 5 级机密性的、接近实用的安全云数据库系统的报道.

6.1　基于密文查询的安全云数据库系统

本小节介绍并比较基于密文查询的安全云数据库系统(如图 3)：面向事务型应用的 CryptDB[53]，以及面向分析型应用的 Monomi[56]. CryptDB 与 Monomi 中的敏感数据在硬盘和内存中始终保持加密，因此这两款数据库达到 3 级机密性保障.这两个安全云数据库系统均由麻省理工学院人工智能实验室研发，秉承一致的设计思路，具体体现在下面 3 个方面：

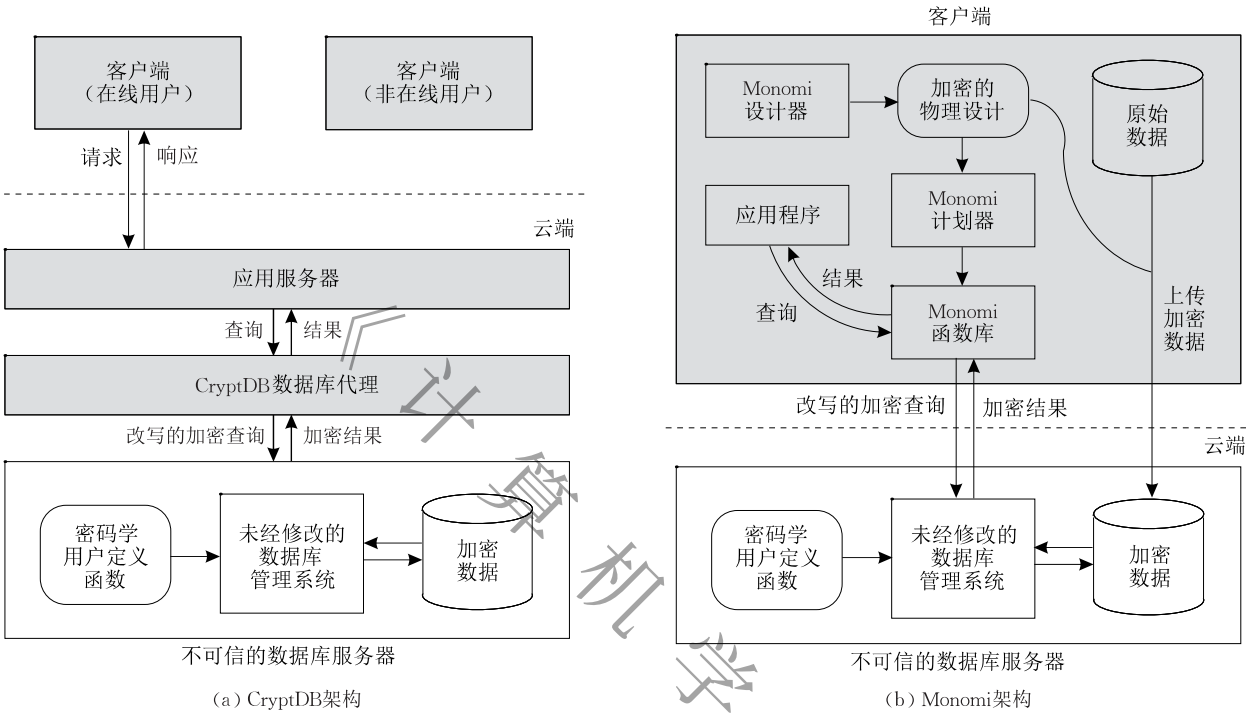


图 3　基于密文查询的安全云数据库系统(CryptDB 针对事务型负载，典型场景是以数据库为后台的网站应用，而 Monomi 则针对分析型负载.图中方框表示组件，圆角方框表示重要数据结构，灰色表示是可信的)

(1) 利用 SQL 操作符种类有限的便利.大部分的 SQL 查询只涉及到少数种类的 SQL 操作符组成，如判等、比较、匹配、聚集(如求和)和连接等.这些常见的 SQL 操作都有特定的加密方法支持(见第 3 节).各种密码学操作(如密文比较、密文关键词匹配和密文加法等)都被实现为数据库的用户定义函数(User Defined Functions)，而数据库接收到的 SQL 查询也用密文和用户定义函数恰当地改写，这样数据库系统无需做任何代码修改即可以安全地处理加密数据和加密查询.

(2) 引入安全云数据库适配器.不仅数据库系统无需修改，应用程序也(几乎)不用做任何改动，这要归功于所谓的“安全云数据库适配器”——CryptDB 中的数据库代理和 Monomi 中的函数库.安全云数据库适配器负责改写用户 SQL 查询，转发查询给数据库，解密处理中间结果(一些密文上的操作无法在云数据库上完成)，和返回最终结果给用户.以上保证数据安全的额外措施因适配器的引入而对用户透明.安全数据库适配器会接触到各种密钥，因此必须运行在可信服务器或可信硬件上.

(3) 根据负载特点优化加密方案.CryptDB 引入了洋葱加密(见 3.4 小节)使得数据库可以根据查询负载调整加密方法，以达到安全和功能的权衡.Monomi 提出了多种优化技术，比如行预计算、空间高效加密、组同态加密等等.但这些技术并不通用，其优化效果取决于负载.因此，Monomi 引入根据负载优化数据库物理布局的设计器以及在运行时选择高效查询计划的计划器.

6.2　基于可信硬件的安全云数据库系统

本小节介绍并比较基于可信硬件的最先进安全云数据库系统(如图 4)：基于安全协处理器的 TrustedDB[62]，以及基于 FPGA 的 Cipherbase[58]. TrustedDB 和 Cipherbase 不仅保证敏感数据在外存中加密和(不可信)内存中加密，而且在可信硬件

中的程序状态也不会泄露(Cipherbase 在不可信硬件中的程序状态也在算法层面有一定的保护措施,详见原文[58]),因此 TrustedDB 和 Cipherbase 达到 4 级机密性保障.作为基于可信硬件的安全云数据库系统,TrustedDB 和 Cipherbase 都遵循下面 3 个相同的设计原则,只是实现方式有区别:

(1)确保可信硬件上代码的正当性,并实现对外的可验证性.IBM 的安全协处理器将程序分为 4 个级别(段 0～3),分别运行 Miniboot 0、Miniboot 1、操作系统和应用程序(在 TrustedDB 中主要是嵌入式数据库);每一层程序的载入和验证由更低一层负载完成,后者保存有相应的公钥,可以验证程序附带的数字证书;最底层的 Miniboot 0 是硬件出厂时永久写入,不可修改,它的公钥-私钥对是由 IBM 工厂提供.Miniboot 还提供外向验证(Outbound Authentication)的功能,可以向外界证明设备的软硬件均未被篡改.FPGA 有专门的、不可修改的启动逻辑,在启动时从与 FPGA 连接的非易失性内存中读取加密的 FPGA 比特流,用于配置 FPGA 可编程部分的逻辑;加密 FPGA 比特流的密钥预先存储在 FPGA 内部的非易失性内存中,这个密钥可以验证 FPGA 比特流的合法性.而 FPGA 比特流中存有 Cipherbase 的私钥,Cipherbase 以此验证 FPGA 配置的正当性.

(2)只在可信硬件内处理明文数据.可信硬件可以安全地处理明文数据,因为它的软硬件保证其中的数据不会被外界窥探.虽然两个工作都利用可信硬件的这个特点,但方式是不一样的:在 TrustedDB 中可信硬件是主导地位,而 Cipherbase 中可信硬件是从属地位.TrustedDB 采用的是一种双数据库管理系统的松耦合架构,安全协处理器中的嵌入式数据库负责解析用户请求,产生查询计划,划分公开和私有子查询,向主机上的数据库管理系统发送子查询,最后合并结果;在这个架构中,可信硬件中的嵌入式数据库"指挥"主机上的数据库.与之相反的是,Cipherbase 只有一个主机上的数据库管理系统,FPGA 是作为这个数据库的表达式求值引擎.比如当数据库需要评估表达式 $E(x)<E(y)$ 是否成立时,会把它发送给 FPGA,后者在其内部首先解密 $E(x)$ 和 $E(y)$,然后计算出 $x<y$ 的值,最后把结果发回给主机上的数据库;可以说,Cipherbase 利用 FPGA 模拟了全同态加密.

(3)充分利用主机的计算和存储资源.为提高主机的利用率、避免可信硬件成为性能瓶颈,TrustedDB 和 Cipherbase 都采取了多种措施.TrustedDB 在生成执行计划时,尽量把大部分工作都交给主机,少部分工作留给安全协处理器.安全协处理器中的存储有限,不可能容纳整个数据库.因此,TrustedDB 扩展了嵌入式数据库(采用 SQLite)的缓冲区池,使得它可以按需读取/写回和解密/加密主机上的数据页.由于架构不同,Cipherbase 并不能采用 TrursedDB

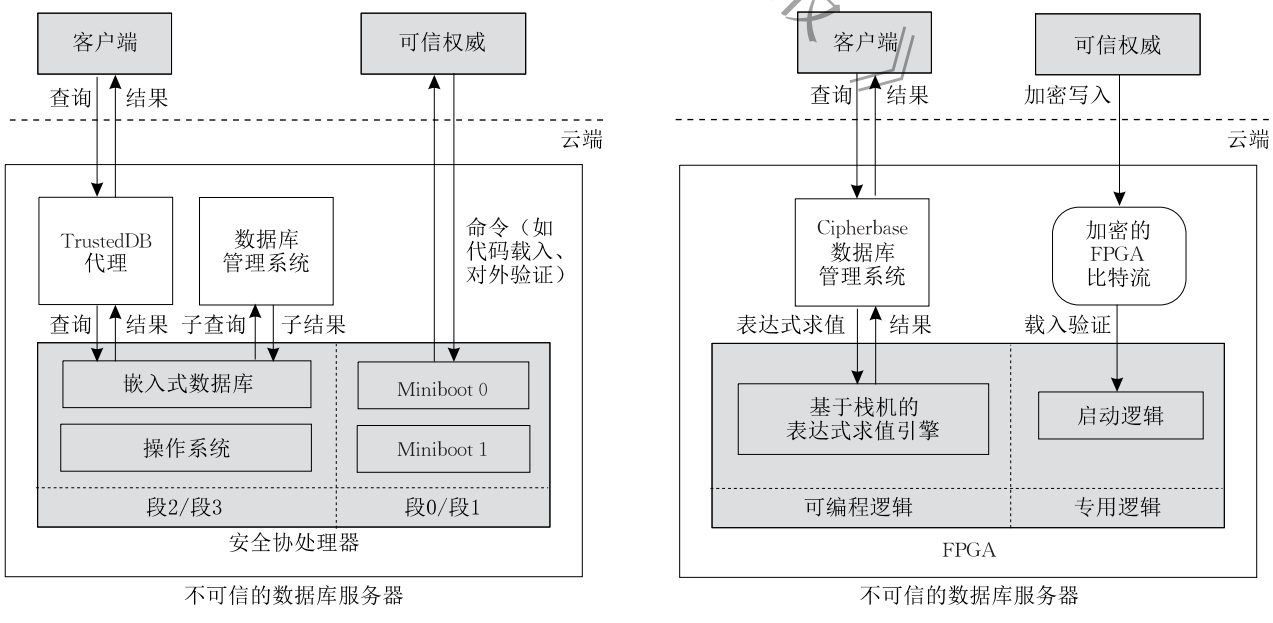


图 4 基于可信硬件的安全云数据库系统(TrustedDB 是基于安全协处理器的松耦合架构,安全协处理器和主机各运行一个独立完整的数据库管理系统.Cipherbase 是基于 FPGA 的紧耦合架构,主机上运行数据库管理系统,FPGA 则是它的基于栈机(Stack Machine)的表达式求值引擎.图中灰色部分表示可信的组件)

的优化方法，而是采用密文查询技术，比如在达到安全要求的前提下，对主键采用确定性加密，方便键值查询；再如，对一些字段采用保序加密，方便区间查询.

7　未来研究趋势

最近几年，随着针对云数据库系统的安全技术尤其是机密性保护技术取得显著进展，工业界已经开始沿着学术界探索出的技术路线和系统架构对安全云数据库系统进行产品化和商业化的尝试，如微软公司的始终加密(Always Encrypted)技术(适用于在 SQL Server 2016 以及 Azure SQL 数据库服务)和谷歌公司的加密大查询(Encrypted Big Query)客户端(仍处于实验阶段). 从目前的发展趋势看，我们预测，未来五年内将会出现获得市场广泛认可、并取得商业成功的安全云数据库产品.

虽然前景光明，但仍有许多技术问题有待解决.

(1) 通用型强大密码学工具的特殊化. 目前的各种密文查询方案，为了性能上达到实用的程度，往往依赖于相对简单的密码学原语，比如单向哈希函数、对称加密函数以及非对称加密函数等，而非更强大的密码学工具，比如全同态加密(见 3.1 小节)、程序混淆(Program Obfuscation)[115]、功能加密(Functional Encryption)[116]和多方安全计算(Multi-Party Secure Computation)[55]等. 这限制了密文查询方案在功能上或安全上所能达到的上限. 而通用型的强大密码学工具之所以在密文查询方案中很少被采用，要么是因为已经被证明通用型的构造是根本不可能[115,117]，要么是因为最先进的实现仍然离实用还有很大的距离[118,119]. 鉴于以上情况，我们认为未来针对安全云数据库的应用型密码学研究的一个有益的研究思路是，根据安全云数据库应用场景的特点对这些功能强大的通用型密码学工具做一些功能上的牺牲或安全上的放松，即特殊化，以期达到实用的程度. 在这个方向上，已经有一些研究工作取得了不错的结果：比如，文献[54]设计了一种只支持算术运算的多方安全计算协议，实现了在密文上支持 TPC-H 数据库基准测试的所有查询；再如，文献[33]给出了一种秘钥多输入功能加密方案，可以计算任何多项式规模的分支程序(Branching Programs)，其中一个重要特例就是密文比较，可以达到比任何保序加密更好的安全性.

(2) 数据中心可信硬件的普及化. 2015 年 7 月，英特尔宣布以 176 亿美元的价格收购全球最大的 FPGA(见 4.2 小节)生产商 Altra，并预测到 2020 年将有三分之一的云数据中心部署 FPGA. 2015 年 8 月，英特尔发布了代号为 Skylake 的新一代微体系架构的处理器，并从这一代处理器开始支持 SGX(见 4.3 小节). 考虑到英特尔在 CPU 数据中心细分市场的绝对统治地位，不难想见 FPGA 和 SGX 等可信硬件技术将在数据中心快速普及. 因此，我们认为此前在学术界并不被十分关注的可信硬件技术将成为研究热点. 这方面的研究问题包括但不限于：创造新的编程工具和框架，降低 FPGA 开发的难度和工作量[120-121]；研究 FPGA 在云环境下的虚拟化、调度以及部署技术，提高 FPGA 的设备利用率[122-124]；开发密码学和数据库操作在 FPGA 中的高效实现[125-126]；研究 SGX 环境下的形式化验证(Formal Verification)方法[127]；开发经过形式化验证的数据库系统，以确保 SGX 中的安全关键代码的正确性[128].

(3) 云数据库安全技术的实证化. 过去十余年围绕云数据库机密性的保护已经积累了大量研究工作，其中一些甚至已经被工业界采纳和部署. 这些实用的安全技术或系统在设计时往往出于性能或功能方面的考虑牺牲了一定的安全性，但妥协后的安全性究竟会泄露了多少信息？有何种可能的攻击方法？实际部署时有哪些注意事项？为了回答这些问题，我们认为未来需要更多研究从攻击者的角度评估相关安全技术在实际数据集上和实际应用系统上的安全表现. 这个研究趋势从最近一两年发表的实证性研究便可见一斑：文献[30]在实际医疗数据集上恢复了保序加密的明文；文献[129]展示了范围查询的访问模式和通信大小如何泄露明文信息；文献[130,131]描述了针对支持密文关键词查询的实用系统的攻击，成功恢复了大量密文关键词.

(4) 安全技术与数据库技术的融合化. 经典的安全云数据库系统 CryptDB 是基于一种可信代理的系统架构(见 6.1 小节)，该架构也得到了后续许多安全系统[54,132]的采用，其主要原因是这种分离式的架构使得现有数据库系统无须任何修改，因而兼容性好、易于部署. 然而，由于安全机制对数据库系统完全透明，这种架构实际上未能充分利用在数据库层面改善性能和安全的机会. 因此，我们认为，安全技术与数据库系统的深度整合，即融合化，是未来

的发展方向.在这个方向上,有许多问题有待深入研究.比如,如何重构数据库系统的架构以有效整合可信硬件? Cipherbase 的架构融合了 FPGA(见 6.2 小节),但目前仍然不清楚数据库架构如何适应 SGX 的"两世界"编程模型[77],以及数据库架构如何同时有效整合 FPGA 和 SGX 以利用二者各自的优势.再比如,如何以更模块化的方式为数据库系统增加密文查询功能? 目前最用户友好的、扩展数据库系统的方式是用户自定义函数,然而用户自定义函数不支持与第三方通信,这导致基于交互式密码学协议的密文查询方案(比如文献[28])不能以用户自定义函数的方式实现,必须要依赖于某种数据库之外的中间层,这增加了数据传输的开销,同时也造成了各个研究"重新造轮子"的浪费.除了以上架构层面的问题,查询执行算法、代价计算模型、物理存储优化和数据插入更新等方面也都存在与安全方案更好整合的空间,比如文献[41]创新性地通过数据库自适应索引实现范围密文查询(详见 3.3 小节).

8 结束语

在云数据外包业务和云数据安全需求的强力驱动下,云环境下数据库的机密性保护成为了重要的研究课题.本综述首先提出了云数据库机密性保护的五级安全模型,使得本文中涉及的众多跨领域、跨问题的安全技术得以在统一的框架中讨论;然后,针对该安全模型中的 3 级至 5 级的机密性威胁(云环境下的新型威胁),系统性地总结、分析了与其相应的三项关键安全技术——密文查询(3 级)、可信硬件(4 级)和访问模式保护(5 级);进而在这三项关键安全技术的基础上,介绍、比较了目前最先进的安全云数据库系统;最后,展望了云数据库机密性保护技术的研究趋势,指出了若干研究方向.作者希望本综述能为今后的研究者和开发者提供有价值的参考.

参 考 文 献

[1] Shmueli E, Vaisenberg R, Elovici Y, et al. Database encryption: An overview of contemporary challenges and design considerations. ACM SIGMOD Record, 2010, 38(3): 29-34

[2] Chow R, Golle P, Jakobsson M, et al. Controlling data in the cloud: Outsourcing computation without outsourcing control//Proceedings of the 2009 ACM Workshop on Cloud Computing Security(CCSW'09). Chicago, USA, 2009: 85-90

[3] Feng Deng-Guo, Zhang Min, Zhang Yan, Xu Zhen. Study on cloud computing Security. Journal of Software, 2011, 22(1): 71-83(in Chinese)
(冯登国, 张敏, 张妍, 徐震. 云计算安全研究. 软件学报, 2011, 22(1): 71-83)

[4] Lin Chuang, Su Wen-Bo, Meng Kun, et al. Cloud computing security: Architecture, mechanism and modeling. Chinese Journal of Computers, 2013, 36(9): 1765-1784(in Chinese)
(林闯, 苏文博, 孟坤等. 云计算安全: 架构, 机制与模型评价. 计算机学报, 2013, 36(9): 1765-1784)

[5] Ding Yan, Wang Huai-Min, Shi Pei-Chang, et al. Trusted cloud service. Chinese Journal of Computers, 2015, 38(1): 133-149(in Chinese)
(丁滟, 王怀民, 史佩昌等. 可信云服务. 计算机学报, 2015, 38(1): 133-149)

[6] Fu Ying-Xun, Luo Sheng-Mei, Shu Ji-Wu. Survey of secure cloud storage system and key technologies. Journal of Computer Research and Development, 2013, 50(1): 136-145 (in Chinese)
(傅颖勋, 罗圣美, 舒继武. 安全云存储系统与关键技术综述. 计算机研究与发展, 2013, 50(1): 136-145)

[7] Feng Chao-Sheng, Qin Zhi-Guang, Yuan Ding. Techniques of secure storage for cloud data. Chinese Journal of Computers, 2015, 38(1): 150-163(in Chinese)
(冯朝胜, 秦志光, 袁丁. 云数据安全存储技术. 计算机学报, 2015, 38(1): 150-163)

[8] Tan Shuang, Jia Yan, Han Wei-Hong. Research and development of provable data integrity in cloud storage. Chinese Journal of Computers, 2015, 38(1): 164-177(in Chinese)
(谭霜, 贾焰, 韩伟红. 云存储中的数据完整性证明研究及进展. 计算机学报, 2015, 38(1): 164-177)

[9] Xiao Ren-Yi. Survey of privacy preserving data queries in cloud computing. Journal on Communications, 2014, 35(12): 168-177(in Chinese)
(肖人毅. 云计算中数据隐私保护研究进展. 通信学报, 2014, 35(12): 168-177)

[10] Maas M, Love E, Stefanov E, et al. Phantom: Practical oblivious computation in a secure processor//Proceedings of the 20th ACM Conference on Computer & Communications Security(CCS'13). Berlin, Germany, 2013: 311-324

[11] Islam M S, Kuzu M, Kantarcioglu M. Access pattern disclosure on searchable encryption: Ramication, attack and mitigation//Proceedings of the Network and Distributed System Security Symposium (NDSS'12). San Diego, USA, 2012: 15

[12] Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms. Foundations of Secure Computation, 1978, 4(11): 169-180

[13] Paillier P. Public-key cryptosystems based on composite degree residuosity classes//Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (Advances in Cryptology—EUROCRYPT'99). Prague, Czech Republic, 1999: 223-238

[14] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms//Proceedings of the Workshop on the Theory and Application of Cryptographic (Advances in Cryptology—EUROCRYPT' 85). Techniques Linz, Austria, 1985: 10-18

[15] Gentry C. Fully homomorphic encryption using ideal lattices//Proceedings of the 41st ACM Symposium on Theory of Computing(STOC 2009). Bethesda, USA, 2009, 9: 169-178

[16] Gentry C, Halevi S. Implementing Gentry's fully-homomorphic encryption scheme//Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic (Advances in Cryptology — EUROCRYPT 2011). Tallinn, Estonia, 2011: 129-148

[17] Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping//Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS'12). Cambridge, UK, 2012: 309-325

[18] van Dijk M, Gentry C, Halevi S, et al. Fully homomorphic encryption over the integers//Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Advances in Cryptology—EUROCRYPT 2010). Monaco and Nice, French Riviera, 2010: 24-43

[19] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. SIAM Journal on Computing, 2014, 43(2): 831-871

[20] Wang S, Divyakant A, Amr E A. Is homomorphic encryption the holy grail for database queries on encrypted data? Department of Computer Science, University of California, Santa Barbara, California, USA: Report 2012-01, 2012

[21] Agrawal R, Kiernan J, Srikant R, et al. Order preserving encryption for numeric data//Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. Paris, France, 2004: 563-574

[22] Ozsoyoglu G, Singer D A, Chung S S Anti-tamper databases: Querying encrypted databases//Proceedings of the 17th Annual IFIP WG 11.3 Working Conference on Database and Applications Security (DBSec 2003). Colorado, USA, 2003: 133-146

[23] Kadhem H, Amagasa T, Kitagawa H. MV-OPES: Multi-valued-order preserving encryption scheme: A novel scheme for encrypting integer value to many different values. IEICE Transactions on Information and Systems, 2010, 93(9): 2520-2533

[24] Liu D, Wang S. Programmable order-preserving secure index for encrypted database query//Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing (CLOUD'12). Hawaii, USA, 2012: 502-509

[25] Boldyreva A, Chenette N, Lee Y, et al. Order-preserving symmetric encryption//Proceedings of 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Advances in Cryptology — EUROCRYPT 2009). Cologne, Germany, 2009: 224-241

[26] Boldyreva A, Chenette N, O'Neill A. Order-preserving encryption revisited: Improved security analysis and alternative solutions//Proceedings of the 31st Annual Cryptology Conference (Advances in Cryptology—CRYPTO 2011). Santa Barbara, USA, 2011: 578-595

[27] Malkin T, Teranishi I, Yung M. Order-preserving encryption secure beyond one-wayness. Cryptology ePrint Archive, Report: 2013/409, 2013

[28] Popa R A, Li F H, Zeldovich N. An ideal-security protocol for order-preserving encoding//Proceedings of the 34th IEEE Symposium on Security and Privacy (SP'13). San Francisco, USA, 2013: 463-477

[29] Kerschbaum F, Schropfer A. Optimal average-complexity ideal-security order-preserving encryption//Proceedings of the 21st ACM Conference on Computer and Communications Security. Scottsdale, USA, 2014: 275-286

[30] Naveed M, Kamara S, Wright C V. Inference attacks on property-preserving encrypted databases//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS 2015). Denver, USA, 2015: 644-655

[31] Durak F B, DuBuisson T M, Cash D. What else is revealed by order-revealing encryption?//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS 2016). Hofburg Palace, Austria, 2016: 1155-1166

[32] Kerschbaum F. Frequency-hiding order-preserving encryption//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS 2015). Denver, USA, 2015: 656-667

[33] Boneh D, Lewi K, Raykova M, et al. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation//Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Sofia, Bulgaria, 2015: 563-594

[34] Chenette N, Lewi K, Weis S A, Wu D J. Practical order-revealing encryption with limited leakage//Proceedings of the 23rd International Conference on Fast Software Encryption (IACR-FSE). Bochum, Germany, 2016: 474-493

[35] Lewi K, Wu D J. Order-revealing encryption: New constructions, applications, and lower bounds//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS 2016). Vienna, Austria, 2016: 1167-1178

[36] Hacigumus H, Iyer B, Li C, et al. Executing SQL over encrypted data in the database-service-provider model//Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data (SIGMOD'02). Madison, Wisconsin, 2002: 216-227

[37] Hore B, Mehrotra S, Tsudik G. A privacy-preserving index for range queries//Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04). Toronto, Canada, 2004: 720-731

[38] Li R, Liu A X, Wang A L, Bruhadeshwar B. Fast range query processing with strong privacy protection for cloud computing//Proceedings of the 40th International Conference on Very Large Data Bases (VLDB 2014). Hangzhou, China, 2014: 1953-1964

[39] Demertzis I, Papadopoulos S, Papapetrou O, et al. Practical private range search revisited//Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data (SIGMOD'16). San Francisco, USA, 2016: 185-198

[40] Bloom B H. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 1970, 13(7): 422-426

[41] Karras P, Nikitin A, Saad M, et al. Adaptive indexing over encrypted numeric data//Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data (SIGMOD'16). San Francisco, USA, 2016: 171-183

[42] Stonebraker M, Abadi D J, Batkin A, et al. C-store: A column-oriented DBMS//Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05). Trondheim, Norway, 2005: 553-564

[43] Idreos S. Database Cracking: Towards Auto-Tuning Database Kernels[Ph. D. dissertation]. Centrum Wiskunde & Informatica, Netherland, 2010

[44] Wang P, Ravishankar C V. Secure and efficient range queries on outsourced databases using Rp-trees//Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE'13). Brisbane, Australia, 2013: 314-325

[45] Guttman A. R-trees: A dynamic index structure for spatial searching//Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data (SIGMOD 1984). Boston, USA, 1984: 47-57

[46] Wong W K, Cheung D W, Kao B, Mamoulis N. Secure kNN computation on encrypted databases//Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD'09). Rhode Island, USA, 2009: 139-152

[47] Bosch C, Hartel P, Jonker W, Peter A. A survey of provably secure searchable encryption. ACM Computing Surveys, 2015, 47(2): 18-51

[48] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data//Proceedings of the 21st IEEE Symposium on Security and Privacy (S&P'00). Oakland, USA, 2000: 44-55

[49] Goh E J. Secure indexes. Cryptology ePrint Archive, Report: 2003/216, 2003

[50] Chang Y C, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data//Proceedings of the 3rd Conference of Applied Cryptography and Network Security (ACNS). New York, USA, 2005: 442-455

[51] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: Improved definitions and efficient constructions//Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06). Alexandria, USA, 2006: 79-88

[52] Fredman M L, Komlós J, Szemerédi E. Storing a sparse table with $O(1)$ worst case access time. Journal of the ACM, 1984, 31(3): 538-544

[53] Popa R A, Redfield C, Zeldovich N, Balakrishnan H. CryptDB: protecting confidentiality with encrypted query processing//Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP'11). Cascais, Portugal, 2011: 85-100

[54] Wong W K, Kao B, Cheung D W, et al. Secure query processing with data interoperability in a cloud database environment//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD'14). Snowbird, USA, 2014: 1395-1406

[55] Yao A C. Protocols for secure computations//Proceedings of the 23rd Symposium on Foundations of Computer Science. Singer Island, USA, 1982: 160-164

[56] Tu S, Kaashoek M F, Madden S, et al. Processing analytical queries over encrypted data//Proceedings of the 39th International Conference on Very Large Data Bases. Riva del Garda, Trento, 2013, 6(5): 289-300

[57] Bajaj S, Sion R. TrustedDB: A trusted hardware-based database with privacy and data confidentiality. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(3): 752-765

[58] Arasu A, Blanas S, Eguro K, et al. Orthogonal security with cipherbase//Proceedings of the 6th Biennial Conference on Innovative Data Systems Research (CIDR'13). Asilomar, USA, 2013

[59] Arnold T W, Buscaglia C, Chan F, et al. IBM 4765 cryptographic coprocessor. IBM Journal of Research and Development, 2012, 56(1.2): 10:1-10:10

[60] Kantarcioglu M, Clifton C. Security issues in querying encrypted data//Proceedings of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security. Storrs, USA, 2005: 325-337

[61] Mykletun E, Tsudik G. Incorporating a secure coprocessor in the database-as-a-service model//Proceedings of the International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA'05). Washington, USA, 2005: 38-44

[62] Agrawal R, Asonov D, Kantarcioglu M, et al. Sovereign joins//Proceedings of the 22nd International Conference on Data Engineering (ICDE'06). Atlanta, USA, 2006: 26-26

[63] Kuon I, Tessier R, Rose J. FPGA architecture: Survey and challenges. Foundations and Trends in Electronic Design Automation, 2008, 2(2): 135-253

[64] Mueller R, Teubner J, Alonso G. Data processing on FPGAs//Proceedings of the 35th International Conference on Very Large Data Bases (VLDB'09). Lyon, France, 2009: 910-921

[65] Teubner J, Mueller R, Alonso G. FPGA acceleration for the frequent item problem//Proceedings of the 26th IEEE International Conference on Data Engineering (ICDE'10). Long Beach, USA, 2010: 669-680

[66] Mueller R, Teubner J, Alonso G. Glacier: A query-to-hardware compiler//Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD'10). Indianapolis, Indiana, 2010: 1159-1162

[67] Moussalli R, Salloum M, Najjar W, et al. Massively parallel XML twig ltering using dynamic programming on FPGAs//Proceedings of the 27th IEEE International Conference on Data Engineering (ICDE'11). Hannover, Germany, 2011: 948-959

[68] Mueller R, Teubner J, Alonso G. Sorting networks on FPGAs. The International Journal on Very Large Data Bases, 2012, 21(1): 1-23

[69] Sukhwani B, Min H, Thoennes M, et al. Database analytics acceleration using FPGAs//Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques (PACT'12). Minneapolis, USA, 2012: 411-420

[70] Eguro K, Venkatesan R. FPGAs for trusted cloud computing //Proceedings of the 22nd International Conference on Field Programmable Logic and Applications (FPL'12). Oslo, Norway, 2012: 63-70

[71] Esmaeilzadeh H, Blem E, St Amant R, et al. Dark silicon and the end of multicore scaling//Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA'11). San Jose, USA, 2011: 365-376

[72] Madhavapeddy A, Singh S. Reconfigurable data processing for clouds//Proceedings of the 19th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM'11). Salt Lake City, Utah, 2011: 141-145

[73] Cong J J. Accelerator-rich architectures: From single-chip to datacenters//Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'14). La jolla, USA, 2014: 139-140

[74] Cong J, Ghodrat M A, Gill M, et al. Accelerator-rich architectures: Opportunities and progresses//Proceedings of the 51st Annual Design Automation Conference on Design Automation Conference (DAC'14). San Francisco, USA, 2014: 1-6

[75] Putnam A, Caulfeld A M, Chung E S, et al. A reconfigurable fabric for accelerating large-scale datacenter services//Proceedings of the ACM/IEEE 41st International Symposium on Computer Architecture (ISCA'14). Minnesota, USA, 2014: 13-24

[76] Edwards C. Growing pains for deep learning. Communications of the ACM, 2015, 58(7): 14-19

[77] Costan V, Devadas S. Intel SGX explained. Cryptology ePrint Archive, Report: 2016/086, 2016

[78] Gueron S. A memory encryption engine suitable for general purpose processors. Cryptology ePrint Archive, Report: 2016/204, 2016

[79] Schuster F, Costa M, Fournet C, et al. VC3: Trustworthy data analytics in the cloud using SGX//Proceedings of the 2015 IEEE Symposium on Security and Privacy. San Jose, USA, 2015: 38-54

[80] Brenner S, Wulf C, Goltzsche D, et al. SecureKeeper: Confidential ZooKeeper using Intel SGX//Proceedings of the 16th Annual Middleware Conference (Middleware). Trento, Italy, 2016: Article 14, 13 pages

[81] Baumann A, Peinado M, Hunt G. Shielding applications from an untrusted cloud with haven. ACM Transactions on Computer Systems, 2015, 33(3): 3-10

[82] Arnautov S, Trach B, Gregor F, et al. SCONE: Secure Linux containers with Intel SGX//Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). Savannah, USA, 2016: 689-703

[83] Checkoway S, Shacham H. Iago attacks: Why the system call API is a bad untrusted RPC interface//Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2013). Houston, USA, 2013: 253-264

[84] Strackx R, Piessens F. Ariadne: A minimal approach to state continuity//Proceedings of the 25th USENIX Security Symposium. Vancouver, Canada, 2016: 875-892

[85] Iliev A, Smith S W. Protecting client privacy with trusted computing at the server. IEEE Security and Privacy, 2005, 3(2): 20-28

[86] Ghinita G, Kalnis P, Khoshgozaran A, et al. Private queries in location based services: anonymizers are not necessary//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Cata. Vancouver, Canada, 2008: 121-132

[87] Chor B, Goldreich O, Kushilevitz E, Sudan M. Private information retrieval//Proceedings of the 36th Annual Symposium on the Foundations of Computer Science (FOCS 1995). Milwaukee, USA, 1995: 41-50

[88] Gasarch W. A survey on private information retrieval. Bulletin of the EATCS, 2004, 82(72-107): 1

[89] Chor B, Gilboa N. Computationally private information retrieval//Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC 1997). El Paso, USA, 1997: 304-313

[90] Kushilevitz E, Ostrovsky R. Replication is not needed: Single database, computationally-private information retrieval//Proceedings of the 54th IEEE Annual Symposium on Foundations of Computer Science (FOCS 1997). Miami Beach, USA, 1997: 364-364

[91] Ostrovsky R, Skeith III W E. A survey of single-database private information retrieval: Techniques and applications//Proceedings of the 10th International Conference on Practice and Theory in Public-Key Cryptography (Public Key Cryptography—PKC 2007). Beijing, China, 2007: 393-411

[92] Sion R, Carbunar B. On the computational practicality of private information retrieval//Proceedings of the 14th Network and Distributed Systems Security Symposium (NDSS'07). San Diego, USA, 2007

[93] Olumofin I, Goldberg I. Revisiting the computational practicality of private information retrieval//Proceedings of the 16th International Conference on Financial Cryptography

and Data Security (FC 2012). Kralendijk, Bonaire, 2012: 158-172

[94] Melchor C A, Gaborit P. A fast private information retrieval protocol//Proceedings of the 2008 IEEE International Symposium on Information Theory (ISIT'08). Toronto, Canada, 2008: 1848-1852

[95] Lorch J R, Parno B, Mickens J W, et al. Shroud: Ensuring private access to large-scale data in the data center//Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST'13). San Jose, USA, 2013: 199-213

[96] Goldreich O. Towards a theory of software protection and simulation by oblivious RAMs//Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987). New York, USA, 1987: 182-194

[97] Goldreich O, Ostrovsky R. Software protection and simulation on oblivious RAMs. Journal of the ACM, 1996, 43(3): 431-473

[98] Williams P, Sion R, Carbunar B. Building castles out of mud: Practical access pattern privacy and correctness on untrusted storage//Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS'08). Alexandria, USA, 2008: 139-148

[99] Pinkas B, Reinman T. Oblivious RAM revisited//Proceedings of the 30th Annual Cryptology Conference (Advances in Cryptology—CRYPTO 2010). Santa Barbara, USA, 2010: 502-519

[100] Goodrich M T, Mitzenmacher M, Ohrimenko O, et al. Oblivious RAM simulation with efficient worst-case access overhead//Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop (CCSW'11). Chicago, USA, 2011: 95-100

[101] Kushilevitz E, Lu S, Ostrovsky R. On the (in) security of hash-based oblivious RAM and a new balancing scheme//Proceedings of the 23rd annual ACM-SIAM Symposium on Discrete Algorithms (SODA'12). Tokyo, Japan, 2012: 143-156

[102] Ostrovsky R, Shoup V. Private information storage//Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC 1997). El Paso, USA, 1997: 294-303

[103] Boneh D, Mazieres D, Popa R A. Remote oblivious storage: Making oblivious rampractical. Massachusetts Institute of Technology, USA: Report 1721.1/62006, 2011

[104] Stefanov E, Shi E, Song D. Towards practical oblivious RAM. arXiv preprint, Report: 1106.3652, 2011

[105] Shi E, Chan T H H, Stefanov E, et al. Oblivious RAM with $O((\log N))$ worst-casecost//Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security (Advances in Cryptology—ASIACRYPT 2011). Seoul, South Korea, 2011: 197-214

[106] Gentry C, Goldman K A, Halevi S, et al. Optimizing oram and using it efficientlyfor secure computation//Proceedings of the 13th International Symposiumon Privacy Enhancing Technologies (PETS 2013). Bloomington, USA, 2013: 1-18

[107] Chung K M, Liu Z, Pass R. Statistically-secure ORAM with $\tilde{O}(\log^2 n)$ overhead//Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Hanoi, Vietnam, 2014: 62-81

[108] Chung K-M, Pass R. A simple ORAM. Cryptology ePrint Archive, Report: 2013/243, 2013

[109] Stefanov E, van Dijk M, Shi E, et al. Path oram: An extremely simple oblivious ramprotocol//Proceedings of the 20th ACM Conference on Computer & Communications Security (CCS'13). Berlin, Germany, 2013: 299-310

[110] Pinkas B, Reinman T. A simple recursive tree oblivious RAM. Cryptology ePrint Archive, Report: 2014/418, 2014

[111] Stefanov E, Shi E, Song D. Towards practical oblivious RAM//Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS'12). San Diego, USA, 2012

[112] Williams P, Sion R, Tomescu A. Privatefs: A parallel oblivious file system//Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS'12). Raleigh, USA, 2012: 977-988

[113] Stefanov E, Shi E. Oblivistore: High performance oblivious cloud storage//Proceedings of the 34th IEEE Symposium on Security and Privacy (SP'13). San Francisco, USA, 2013: 253-267

[114] Dautrich J, Stefanov E, Shi E. Burst ORAM: Minimizing ORAM response times for bursty access patterns//Proceedings of the 23rd USENIX Security Symposium. San Diego, USA, 2014: 749-764

[115] Barak B, Goldreich O, Impagliazzo R, et al. On the (Im) possibility of obfuscating programs//Proceedings of the 21st Annual International Cryptology Conference (Advances in Cryptology—CRYPTO 2001). Santa Barbara, USA, 2001: 1-18

[116] Shen E, Shi E, Waters B. Predicate privacy in encryption systems//Proceedings of the 6th Theory of Cryptography Conference (TCC 2009). San Francisco, USA, 2009: 457-473

[117] Agrawal S, Gorbunov S, Vaikuntanathan V, et al. Functional encryption: New perspectives and lower bounds//Proceedings of the 33rd Annual International Cryptology Conference (CRYPTO 2013). Santa Barbara, USA, 2013: 18-21

[118] Bellare M, Hoang V T, Keelveedhi S, Rogaway P. Efficient garbling from a fixed-key blockcipher//Proceedings of the 34th IEEE Symposium on Security and Privacy (IEEE S&P). San Francisco, USA, 2013: 478-492

[119] Lepoint T, Naehrig M. A comparison of the homomorphic encryption schemes FV, and YASHE//Proceedings of the 7th International Conference on the Theory and Application of Cryptographic Techniques (AFRICA CRYPT 2014). Marrakesh, Morocco, 2014: 318-335

[120] Martinez V F, Styles H. Unlocking FPGAs using high level synthesis compiler technologies//Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'15). Monterey, USA, 2015: 27-27

[121] Shanker P. Spatial debug & debug without Re-programming in FPGAs: On-chip debugging in FPGAs//Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'16). Monterey, USA, 2016: 3-3

[122] Byma S, Steffan J G, Bannazadeh H, et al. FPGAs in the cloud: Booting virtualized hardware accelerators with openstack//Proceedings of the 22nd IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM '14). Boston, USA, 2014: 109-116

[123] Asiatici M, George N, Vipin K, et al. Designing a virtual runtime for FPGA accelerators in the cloud//Proceedings of the 2016 26th International Conference on Field Programmable Logic and Applications (FPL'16). Lausanne, Switzerland, 2016: 1-2

[124] Iordache A, Pierre G, Sanders P, et al. High performance in the cloud with FPGA groups//Proceedings of the 9th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'16). Shanghai, China, 2016: 1-10

[125] Casper J, Olukotun K. Hardware acceleration of database operations//Proceedings of the 2014 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'14). Monterey, USA, 2016: 151-160

[126] Halstead R J, Absalyamov I, Najjar W A, Tsotras V J. FPGA-based multithreading for in-memory hash joins//Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR'15). Asilomar, California, USA, 2015

[127] Sinha R, Rajamani S, Seshia S, et al. Moat: Verifying confidentiality of enclave programs//Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. Denver, USA, 2015: 1169-1184

[128] Malecha G, Morrisett G, Shinnar A, et al. Toward a verified relational database management system. ASCM SIGPLAN Notices, 2010, 45(1): 237-248

[129] Kellaris G, Kollios G, Nissim K, O'Neill A. Generic attacks on secure outsourced databases//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 1329-1340

[130] Pouliot D, Wright C V. The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 1341-1352

[131] Grubbs P, McPherson R, Naveed M, et al. Breaking web applications built on top of encrypted data//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 1353-1364

[132] Popa R A, Stark E, Valdez S, et al. Building web applications on top of encrypted data using Mylar//Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14). Seatle, USA, 2014: 157-172

TIAN Hong-Liang, born in 1987, Ph. D. candidate. His main research interests include trusted hardware, and cloud computing.

ZHANG Yong, born in 1973, Ph. D., associate professor. His main research interests focus on data management.

LI Chao, born in 1978, Ph. D., associate professor. Her main research interests include storage system and data management.

XING Chun-Xiao, born in 1967. Ph. D., professor. His main research interests include digital library and database technology.

Background

As cloud computing is getting popular, more and more individuals and organizations are considering outsourcing data to database management systems on public clouds. However, data security and privacy concerns hinder the further adoption of cloud computing and cloud databases. Driven by the needs of data outsourcing and data security on cloud, confidentiality protection for cloud databases has become an important research topic. In this survey, we start by introducing a five-level security model for confidentiality-protected cloud databases, which provides a unified framework for discussing various relevant techniques. This five-level security model identifies five types of security risks: interface risk (level 1), storage risk (level 2), memory data risk (level 3), program state risk (level 4), and access pattern risk (level 5). Among them, the risks from level 3 to level 5 are considered as new challenges in a cloud environment. Then, to address

these security risks that are new in cloud environments, we systematically review and analyze three key data security approaches: querying over encrypted data (level 3), trusted hardware (level 4) and access pattern protection (level 5). Querying over encrypted data can be seen as a specialized form of computation over encrypted data, which has been theoretically solved by fully homomorphic encryption (FHE). Unfortunately, all current constructions of FHE are too inefficient to be used in practice. However, common types ofdatabase queries, like range queries and keyword queries, can be efficiently processed on encrypted data. Range querying can be processedby either leveraging order-preserving encryption or encrypted indexes. And keyword querying over encrypted data can be supported by searchable symmetric encryption. For the two types of queries, a number of state-of-the-art techniques are described in this paper. The main difficulties of querying over encrypted data are (1) handling arbitrary types of queries over encrypted data and (2) reducing the information leakage. To overcome these drawbacks, it is probably necessary to introduce into cloud environment a special kind of hardware called trusted hardware, where user data and computation within are isolated from the outside. Three trusted hardware technologies are described, including secure coproceesors, FPGAs and Intel SGX. FPGAs and Intel SGX are superior than secure coprocessors in terms of performance and cost, thus more likely to get widely adopted by cloud service providers. While trusted hardware protects the data and computation inside, it reveals its access to the storage outside. It has been shown by several studies that access pattern can leak sensitive information. To reduce information leakage from access pattern, there are two general techniques: private information retrieval and oblivious RAM. The target application scenario of these two techniques are different: the former assumes no encryption of data, while the latter requires encryption. A systematic review on the two techniques is presented in this paper. After reviewing the key techniques of protecting data confidentiality in cloud, we describe and compare the state-of-the-art secure cloud database systems, including those based on cryptography and those based on trusted hardware. Finally, we conclude with perspectives and directions for future research. This work was supported by the National Natural Science Foundation of China (91646202), the National High Technology Research and Development Program (863 Program) of China (Grant No. SS2015AA020102), the 1000-Talent Program, Tsinghua University Initiative Scientific Research Program.