大数据计算环境下的隐私保护技术研究进展

钱文君[1),2),3)] 沈晴霓[1),2),3)] 吴鹏飞[1),2),3)] 董春涛[1),2),3)] 吴中海[1),2),3)]

[1)](北京大学软件与微电子学院 北京 102600)

[2)](北京大学软件工程国家工程研究中心 北京 100871)

[3)](高可信软件技术教育部重点实验室(北京大学) 北京 100871)

摘 要 批处理、流式计算和机器学习等分布式的大数据计算环境在云上的广泛部署与应用,为云用户带来了极大的便利,但随之带来的隐私数据泄露事件愈演愈烈.如何在这种云上部署的大数据计算环境下保护数据隐私成为一个研究热点,本文对近些年国内外在该领域的最新隐私保护研究成果及进展进行了全面综述.针对上述大数据计算环境下的参与角色及应用场景,结合不同角色的敌手模型,从计算过程涉及的数据输入、计算和输出等三个环节出发,依据计算数据为明文、密文或可信硬件保护条件下可能存在的隐私泄露风险,总结了对应的5类主要研究方向,包括:基于数据分离的隐私保护、基于数据干扰的隐私保护、基于安全多方计算的隐私保护、基于硬件增强的隐私保护和基于访问模式隐藏的隐私保护等,从隐私性、可用性和性能等方面对比分析了现有研究工作的优缺点;最后,展望了大数据计算环境下隐私保护技术的未来研究方向.

关键词 大数据隐私保护;数据分离;数据干扰;安全多方计算;硬件增强;访问模式隐藏

中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2022.00669

Research Progress on Privacy-Preserving Techniques in Big Data Computing Environment

QIAN Wen-Jun[1),2),3)] SHEN Qing-Ni[1),2),3)] WU Peng-Fei[1),2),3)] DONG Chun-Tao[1),2),3)]
WU Zhong-Hai[1),2),3)]

[1)](*School of Software and Microelectronics, Peking University, Beijing* 102600)

[2)](*National Engineering Research Center for Software Engineering, Peking University, Beijing* 100871)

[3)](*Key Laboratory of High Confidence Software Technologies of Ministry of Education (Peking University), Beijing* 100871)

Abstract The widespread deployment and applications of distributed big data computing environment, such as batch processing, stream computing, and machine learning in cloud, have brought great convenience to users for efficiently processing massive amounts of data, but the privacy issues caused by data breaches are becoming more and more serious. How to protect private data in such a big data computing environment deployed on the cloud has become a research hotspot. This paper provides a comprehensive overview of the latest research achievements and progress of big data privacy protection in this field, mainly including domestic and foreign research work in recent years. Firstly, the participating roles and application scenarios in the above-mentioned big data computing environment are introduced. Combining the adversary models of different roles, we start from the three links of data input, computation, and output

收稿日期:2020-11-20;在线发布日期:2021-06-28.本课题得到国家自然科学基金(61672026, 61232005)资助.**钱文君**,博士研究生,主要研究领域为大数据安全与隐私、差分隐私.E-mail: wenjunqian@pku.edu.cn. **沈晴霓**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为操作系统与虚拟化安全、云计算和大数据安全与隐私、可信计算.E-mail: qingnishen@ss.pku.edu.cn. **吴鹏飞**,博士研究生,主要研究领域为分布式系统安全、隐私保护、大数据安全. **董春涛**,博士研究生,主要研究领域为分布式系统安全、大数据安全、可信计算. **吴中海**,博士,教授,博士生导师,中国计算机学会(CCF)杰出会员,主要研究领域为大数据系统与分析技术、大数据与云安全、高可信嵌入式系统.

involved in the distributed computing process, and divide the existing privacy issues into three categories: privacy leakage of native individual data occurs during the data input stage, private data is stolen by attackers inside the cloud during the data computation process, and sensitive information is maliciously inferred by untrusted data consumers (i. e. users who use cloud computing platform and pay for cloud service form cloud providers) during the data output stage. Secondly, we summarize the corresponding five main research directions based on the possible privacy leakage risks under conditions such as plaintext, ciphertext, or trusted hardware protection, including privacy protection based on data separation, privacy protection based on data interference, privacy protection based on secure multi-party computation, privacy protection based on hardware enhancement, and privacy protection based on access pattern hiding. For each type of privacy protection scheme, privacy challenges, adversary model, privacy issues, mainstream privacy-preserving techniques, and existing limitations are sorted out and analyzed. Furthermore, the advantages and disadvantages of existing privacy-preserving techniques are compared in terms of privacy, utility, and performance. Specifically, these techniques have different characteristics and limitations, and are suitable for different application scenarios. In order to protect individual privacy in the data input stage, previous work adopt many effective techniques such as data separation, data anonymization, and local differential privacy. Besides, in order to ensure the confidentiality and privacy of sensitive data involved in the computing process, the existing mainstreaming privacy-preserving solutions are based on secure multi-party computation, hardware enhancement, and access pattern hiding, including main techniques such as garbled circuit, secret sharing, homomorphic encryption, Intel software guard extensions, oblivious random access machine and oblivious shuffle. In addition, it should be noted that privacy leakage may occur during the data output stage. Attackers outside the cloud can use their known background knowledge to analyze the output of big data computing, and then obtain sensitive information that can be traced back to a specific individual, and consequently steal privacy of the original input data. In order to defend against such attackers, it is effective to adopt data anonymization or differential privacy technique. Finally, the future research trends of privacy-preserving techniques in big data computing environment are prospected at the end of this paper.

Keywords big data privacy protection; data separation; data interference; secure multi-party computation; hardware enhancement; access pattern hiding

1 引 言

随着云计算与大数据技术的发展,亚马逊、微软、华为与阿里等主流云服务提供商(Cloud Service Provider, CSP)支持云端部署分布式存储和计算框架,主要包括批量计算框架(如MapReduce[1])、流式计算框架(如Spark Streaming[2]、Storm①、Flink[3])和机器学习框架(如TensorFlow[4])等,为用户提供持续可靠、可扩展且高吞吐量的大数据存储和计算服务.但是,在这种外包的大数据计算环境下,由于数据所有权和使用权的分离,在计算过程涉及的数据输入、计算和输出等阶段都有可能发生隐私数据泄露的风险.因此,如何在大数据计算环境下保护敏感数据的隐私性(privacy),同时保证数据的可用性(utility)和计算的高效性(efficiency)成为大数据隐私保护领域的研究热点之一.

近年来,隐私数据(private data)泄露事件频频发生,造成的影响也越来越严重.从泄露的数据类型来看②,泄露最多的隐私数据是个人基本信息,其次是用户账号密码信息,再者是个体敏感信息.并且个体敏感信息泄露呈现明显增长的趋势,主要包括人脸图像、指纹和虹膜等生物识别敏感信息,交易收入敏

① Storm, distributed realtime computation system. http://storm.apache.org/

② 天枢实验室. http://blog.nsfocus.net/inventory-of-data-breaches-at-home-and-abroad-in-2019/

感信息和医疗病历敏感信息等. 国内外隐私泄露事件举例①：2016年5月，美国职业社交网站LinkedIn宣布近1.67亿用户的电子邮箱地址和密码发生泄露，并被黑客组织公开销售；2017年9月，美国知名信用机构Equifax遭黑客攻击，导致近1.43亿用户的信用卡和驾照号码等个人信息被泄露；2018年3月，美国社交媒体Facebook承认其近5000万用户的个人信息被一款性格测试软件非法收集；2019年2月，中国深网视界科技有限公司SenseNets被曝出超过250万人的人脸数据发生泄露；2020年5月，某脱口秀艺人控诉中信银行为“配合大客户的需要”，在未经本人允许的情况下违法泄露了个人账户交易. 为了避免隐私泄露带来的负面影响和经济损失，一系列隐私保护条例和法规被相继颁布②. 例如，国内已经实施的《中华人民共和国网络安全法》和最近通过的《中华人民共和国个人信息保护法》，明确规定了个人信息收集、处理和利用的基本规范和主要法律责任；国际上，欧盟已经实施的《通用数据保护条例》(General Data Protection Regulation，GDPR)，加强了欧洲居民的个人数据保护；美国加利福尼亚州已经颁布且正式生效的《加州消费者隐私法案》(California Consumer Privacy Act，CCPA)，旨在加强消费者的数据安全与隐私保护.

但是，仅仅从立法层面约束隐私泄露事件的发生是不够的. 面对多样化的业务场景和问题挑战，从技术层面引入一些隐私保护技术是非常必要的. 近年来，云上数据隐私问题已经受到了学术界和工业界的广泛关注和重视. 分析大数据计算环境下数据处理流程，主要存在三类隐私泄露问题：数据输入阶段的原始数据(raw data)隐私泄露，计算过程中的隐私数据被攻击者窃取，以及不可信的数据消费者在结果输出阶段试图推断出数据隐私.

首先，在数据输入阶段，如果对数据所有者的敏感信息不采取标记和去隐私处理，那么有关个体的隐私数据将可能被不可信的云服务提供商或者其他攻击者恶意窃取，造成个体隐私的直接泄露. 为了在数据输入阶段保护个体隐私，目前行之有效的手段是采取数据分离或者数据干扰等方法. 其一，数据分离方法主要考虑到隐私数据的位置，一般将数据所有者的非敏感数据上传到公有云，敏感数据被分离到本地的私有云，这保证了敏感数据在可信的私有云环境进行存储和计算. 但是，在实践中发现，数据分离方法会增加私有云与公有云之间的通信开销，甚至通信数据存在被恶意敌手截获的风险. 例如，主流的MapReduce计算框架是基于单个云而设计的，并不适用于混合云环境；另外，联邦学习下本地和第三方参数服务器之间传输的参数也属于模型隐私. 其二，本地化差分隐私(Local Differential Privacy，LDP)技术是目前数据干扰方法中保护输入阶段个体隐私的一种重要手段. 该技术不要求数据所有者必须信任云服务提供商，通过对敏感信息进行本地化随机响应，达到干扰真实数据的效果. 不可避免地，在云端对失真数据进行分布式计算会严重地影响结果的准确性. 因此，如何在保证原始数据隐私性的同时，有效地提高数据的可用性已经成为学术界关切的热点问题.

其次，在数据计算阶段，如果存储在云端的数据直接以明文的形式参与计算，那么不可信的云服务提供商或者计算参与方可以伪装成半诚实敌手直接窥探到部分甚至整体数据，进一步推测出个体隐私信息，造成计算过程中的隐私泄露. 为了保证数据的机密性和计算隐私性，目前行之有效的手段是对传输数据进行加密，即加密传输，并结合安全多方计算(Secure Multi-Party Computation，SMC)、硬件增强或者访问模式隐藏等主流方法实现隐私计算. 其中，设计安全多方计算协议需要依赖混淆电路(Garbled Circuit，GC)、秘密共享或者同态加密(Homomorphic Encryption，HE)等密码学技术. 在互不信任的多个参与方之间协作计算时，SMC保证任何一方都无法窃取其他各方的数据隐私. 特别地，同态加密使得在密文上执行计算成为可能，即密文计算. 但是，当应用到复杂的计算任务时，其执行效率较低且计算开销较高. 为了解决密文计算带来的性能瓶颈，学术界一方面对更加实用且高效的安全多方计算协议开展研究，另一方面依赖可信硬件保护提出“加密传输-明文计算”的优化策略[5-6]. Intel SGX(Software Guard Extensions)属于硬件增强方法中的代表性技术，它为明文计算提供了安全的可信执行环境(Trusted Execution Environment，TEE). 相比SMC，Intel SGX技术既能保护数据的机密性和隐私性，也保证计算代码的安全执行. 尽管如此，攻击者仍然能够通过观察内存层的访问模式[7]和网络层的访问模式[8]，进一步地推测出数据隐私. 不经意随机访问机(Oblivious Random Access Machine，ORAM)和不经意混洗(oblivious shuffle)是目前主流的两种访问模式隐藏技术，它们能够实现不经意

① FREEBUF 数据安全. https://www.freebuf.com/articles/database/231332.html

② NSFOCUS_Research. https://www.secrss.com/articles/13857

计算，防止攻击者观察计算过程中的访问模式.

再者，在计算结果输出阶段，如果数据不经过去隐私化处理而直接发布，那么攻击者可以结合背景知识分析输出结果，窃取其中可以追溯到特定个体的敏感信息，造成输出阶段的隐私泄露. 为了解决输出隐私问题，目前主要采用数据干扰方法，如数据匿名(data anonymization)和中心化差分隐私(Centralized Differential Privacy，CDP)技术. 但是干扰数据会影响数据的可用性，因此需要考虑隐私性和可用性权衡问题.

近年来，国内外学者研究了隐私保护技术在数据挖掘隐私、大数据安全与隐私以及机器学习隐私等领域的应用，形成了一些综述性的文章：文献[9]重点分类阐述了数据失真、数据加密和限制发布等隐私保护技术在数据库领域的应用；文献[10]重点梳理了基于直方图、划分和回归分析的差分隐私技术在数据发布和分析中的应用；文献[11]重点对比分析了隐私保护数据挖掘(Privacy-Preserving Data Mining，PPDM)中的数据匿名和数据扰动技术；文献[12]聚焦大数据安全与隐私领域，梳理及总结了隐私保护、信任和访问控制等角度的关键技术，包括数据匿名、数据水印、数据溯源和风险自适应的访问控制等技术；文献[13]从MapReduce计算的数据安全性和隐私性出发，调研并分析了所面临的安全和隐私挑战、敌手能力以及现有的安全和隐私协议等内容；文献[14-15]聚焦机器学习隐私，重点分析和总结了差分隐私、同态加密和安全多方计算等技术在该领域的研究成果. 除此之外，已有的大多数综述文章侧重于梳理某项或者某类隐私保护技术的基础理论与应用[16-17]，形成专项技术综述，却缺少对大数据计算全过程面临的隐私问题及不同隐私保护技术的总结分析. 特别是目前大数据隐私问题严重地影响了计算框架的推广与应用，因此有必要梳理有关大数据计算隐私的研究进展.

本文区别于已有综述文章，重点梳理了与大数据计算环境相关的隐私保护研究工作. 如图1所示为本文结构图，展示了不同隐私保护技术之间的联系与分类依据.

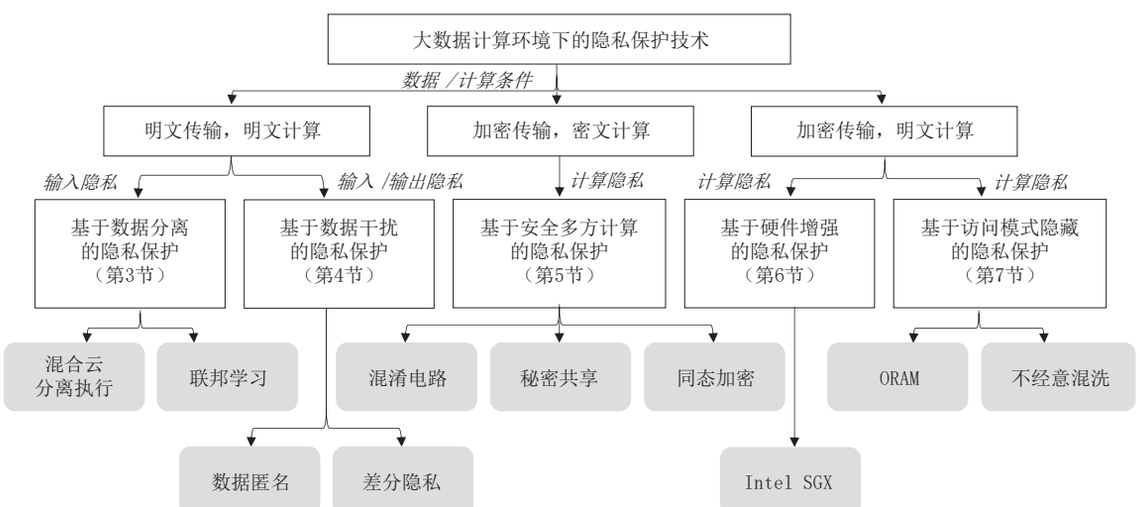


图1 文章结构

本文侧重于从计算过程涉及的数据输入、计算和输出等三个环节出发，依据计算数据为明文、密文或可信硬件保护等条件下可能存在的隐私泄露风险和技术挑战，将分离执行、联邦学习、差分隐私、安全多方计算、Intel SGX等主流的隐私保护技术划分为5大类，包括基于数据分离的隐私保护、基于数据干扰的隐私保护、基于安全多方计算的隐私保护、基于硬件增强的隐私保护和基于访问模式隐藏的隐私保护等. 每一类隐私保护技术侧重于解决不同阶段所面临的隐私问题，并依赖不同的解决思路. 例如，差分隐私通过拉普拉斯、指数和随机响应等扰动机制干扰数据，保证数据输入阶段或输出阶段隐私；安全多方计算依赖混淆电路、秘密共享和同态加密等密码学手段，设计协议保证计算过程中数据的机密性；联邦学习通过在本地联合训练模型保护训练数据的隐私，但是已有研究表明联邦学习存在着参数泄露的风险，需要进一步地依赖差分隐私或者安全多方计算等技术保护训练阶段的模型隐私；Intel SGX通过提供可信的执行环境保证数据以明文形式计算的安全性. 此外，本文从隐私性、可用性和性能等方面对比分析了现有研究

工作的优缺点；最后对未来研究方向进行探讨及展望，为今后进一步研究提供参考.

2　大数据计算环境现状及隐私问题

本节对大数据计算环境现状及存在的隐私问题进行概述，首先介绍了大数据计算环境下的参与角色和部署框架，然后分析了敌手模型、存在的隐私问题与挑战、以及主要研究方向.

2.1　大数据计算环境现状

随着云计算、人工智能以及物联网技术的发展，不同于传统的数据处理，大数据时代的海量数据依赖于云平台提供的可扩展资源实施安全可靠的高效计算. 如图2所示为典型的基于云平台的大数据计算环境，参与角色可以细分为：数据所有者(data owner)、数据持有者(data holder)、云服务提供商、数据消费者(data consumer). 一般情况下，数据所有者和数据持有者被当作同一方，统称为数据提供方(data provider). 但是，在许多实际应用场景中，数据所有者、数据持有者和数据消费者往往不是同一方，例如患者、基因组数据提供商、基于云的健康咨询服务商与制药公司等示例关系. 出于研究、监控或共享的目的，数据持有者往往需要向云服务提供商提供数据访问权限，同时也要考虑为数据所有者的数据提供足够的隐私保护. 参与角色之间的关系分析如下：

(1)数据所有者：生成和创建原始数据的实体，对原始数据具有所有权，负责控制其数据的生成、收集和共享. 原始数据可能是网络数据、社交数据、日志数据或者其他的非结构化数据等，其中可能包含着个人基本信息、社交账号密码或者人脸图像等敏感信息.

(2)数据持有者：经过认证和授权持有数据的实体. 一般情况下，能够生成或创建文件的数据持有者也被视为数据所有者，可以认为是同一个实体. 数据持有者往往不愿意共享数据，但是出于商业合作、科学研究或知识共享的目的，它使用云服务提供商提供的技术和服务存储或处理数据，例如银行、医院、学术机构或者政府机构等.

(3)云服务提供商：作为提供云服务的第三方平台，它以按需付费的形式为数据持有者或者数据消费者提供网络、存储或计算等资源，支持分布式存储、分布式计算和机器学习等服务，例如进行科学仿真分析、数据库查询操作、机器学习或者图像处理等应用.

(4)数据消费者：也称为云用户，经过数据所有者授权对数据具有使用权. 它既可以直接访问部署在云平台的服务，也可以向云平台提交自编程的应用程序请求计算服务，例如企业员工、医生或终端用户等. 有些数据消费者也可能是提供数据的一方，即数据消费者和数据所有者是同一个实体. 如个性化服务场景，用户允许服务提供商收集个人信息，以获取更精准的推荐服务.

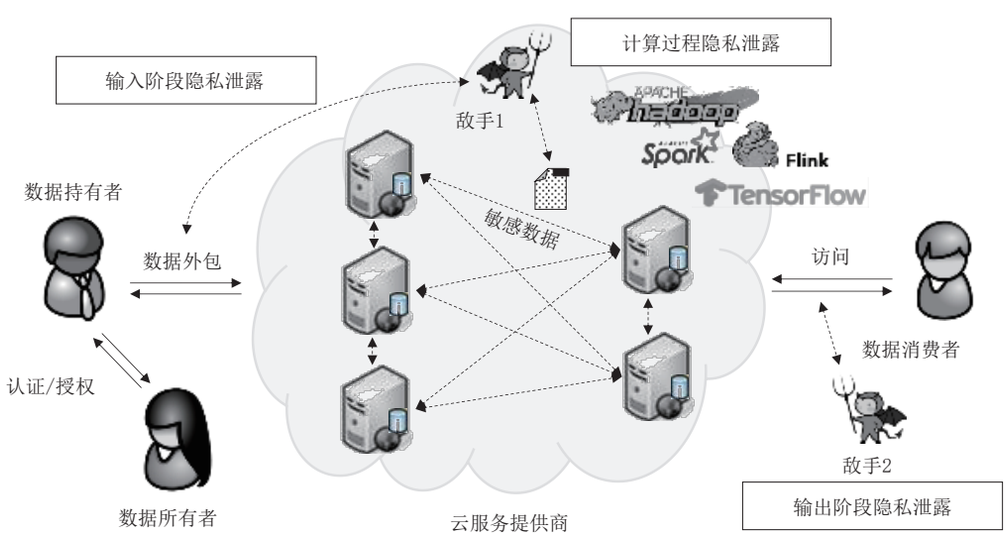


图2　基于云平台的大数据计算环境

在大数据计算环境下，云服务提供商主要为大数据应用提供分布式存储和计算服务，部署主流的批量计算框架(如MapReduce)、流式计算框架(如Spark Streaming、Storm、Flink)和机器学习框架(如

Google TensorFlow、Facebook PyTorch①、Microsoft DMTK②)等,对比如下:

(1)批量计算框架.批量计算的特点是数据存储后再集中离线计算,即计算跟着数据走,适合延时较高的静态数据处理场景.

(2)流式计算框架.流式计算的特点是数据流到来后直接在内存中实时计算,不再对流式数据进行永久化存储,即数据跟着计算走,适合低延时或实时的流数据处理场景.

(3)机器学习框架.它支持基于大规模数据集的模型训练和预测服务,适合集中式学习或者联邦学习场景.其中,集中式学习需要将数据收集到中央服务器集中进行模型训练,而联邦学习中训练数据分布在各客户端,他们在本地训练出局部模型,并依赖中央参数据服务器更新全局模型.

尽管上述大数据计算环境提供了极大的便利,但是也存在许多隐私泄露问题,近年来引起了学术界和工业界的广泛关注.例如,苹果iOS系统[18]、谷歌Chrome Web浏览器[19]和微软Windows Insiders[20]等纷纷采取了一些隐私保护措施;学术界也开展了一系列相关研究.

2.2 隐私问题、挑战与主要研究方向

本小节首先从敌手角色和敌手能力两个角度分析大数据计算环境下的敌手模型,进一步分析存在的隐私问题与挑战,然后根据不同的隐私挑战,将大数据计算环境下隐私保护的相关研究工作主要分为5个研究方向.

2.2.1 敌手模型

根据敌手角色的不同,敌手可以分为不可信的云服务提供商(见图2中的敌手1)和不可信的数据消费者(见图2中的敌手2).当敌手是不可信的云服务提供商时,虽然CSP出于商业利益承诺会按照隐私规范存储和计算数据,但是云内部攻击者可以在输入阶段或者计算过程中(如shuffle过程)窃取个体隐私数据[5,8];当敌手是不可信的数据消费者时,虽然数据持有者信任CSP,并将数据托管到云平台,但是云外部攻击者可以伪装成数据消费者,通过云平台开放的服务接口执行恶意程序,观察返回结果试图窃取原始数据隐私[21-22].

无论敌手是不可信的云服务提供商还是不可信的数据消费者,根据敌手所具有的不同攻击能力,通常分为恶意敌手和半诚实敌手两种.恶意敌手(malicious adversary)是主动攻击者,一般是指恶意的数据消费者.通过安装恶意软件或者执行恶意程序,它可以主动地篡改或者删除其控制的数据,偏离期望的计算行为,试图恶意地窃取受保护数据;半诚实敌手(semi-honest adversary)也称为诚实但好奇型敌手(honest-but-curious adversary),这种类型的敌手是被动攻击者,大多是指CSP.按照计算框架和通信协议要求,它诚实地在计算单元之间传递数据,同时好奇地观察计算过程以推测出隐私数据.

2.2.2 问题与挑战

通过对敌手角色和敌手能力的分析,从计算过程涉及的数据输入、计算和输出等三个环节出发,大数据计算环境下的隐私泄露问题可以分为三类:输入隐私、计算隐私和输出隐私.

(1)数据输入阶段的原始数据隐私泄露,简称输入隐私问题.在数据输入阶段,如果对数据所有者的敏感信息不采取标记和去隐私处理,那么有关个体的隐私数据将可能被不可信的云服务提供商恶意窃取,造成输入阶段隐私泄露.

(2)计算过程中的机密数据被攻击者窃取,简称计算隐私问题.许多大数据计算框架在设计之初并没有考虑数据的安全与隐私保护,数据往往被明文计算,中间结果在多个服务器间被明文传输和存储.因此,隐私数据很容易被攻击者窃取.尽管为了保护数据的机密性,数据被加密且以密文形式通信传输,例如采取传统的对称或者非对称加密算法,但是数据在计算之前仍然需要被解密,那么计算节点上的攻击者依然可以窃取明文数据以及隐私信息,造成计算过程隐私泄露.

(3)不可信的数据消费者在结果输出阶段试图推断出隐私数据,简称输出隐私问题.对于一些具有背景知识的恶意敌手,如果他们对执行的作业类型甚至输入数据的统计信息等背景知识具有一定的了解,那么这类特殊的恶意敌手可以借助背景知识和输出结果获知其中可以追溯到特定个体的敏感信息,造成输出阶段隐私泄露[23].

从隐私问题和隐私保护需求出发,我们总结出了大数据计算环境下实施隐私保护的5个挑战问题,并梳理了相应的隐私保护解决技术,如下:

(1)考虑到数据输入阶段发生隐私泄露,数据所有者希望敏感数据保留在本地私有云进行存储及计算,而非敏感数据可以选择是否上传到公有云.基于

① PyTorch, an open source machine learning framework. http://pytorch.org/

② DMTK, distributed machine learning toolkit. https://www.dmtk.io/

该隐私保护需求，如何保证敏感数据或者全部原始数据安全地在本地私有云环境被处理是个挑战，相应的隐私保护技术是数据分离技术.

(2)考虑到云服务提供商不可信时，数据所有者希望原始数据在本地去隐私即脱敏之后再到云平台进行计算；考虑到云服务提供商可信但是数据消费者不可信时，为了防止输出隐私泄露，计算结果需要在脱敏之后再发布给数据消费者. 基于该隐私保护需求，如何保证数据去隐私后再发布到不可信的一方是个挑战，相应的隐私保护技术是数据匿名和差分隐私等数据干扰技术.

(3)考虑到计算过程中不可信的云服务提供商会窃取隐私数据，数据所有者希望数据加密传输并被密文计算. 基于此隐私保护需求，如何在加密的数据集上执行密文计算是个挑战，相应的隐私保护技术是依赖密码学手段实现安全多方计算.

(4)基于隐私挑战(3)，密文计算给大数据处理带来了计算开销和通信开销的性能瓶颈. 出于隐私性和高效性的权衡，更倾向于数据被加密传输但在CSP受信任的硬件环境高效地执行明文计算. 基于此隐私保护需求，如何在加密前提下借助可信硬件支持的隔离环境对关键代码和数据执行安全高效的明文计算是个挑战，相应的隐私保护技术是Intel SGX或者其它TEE等硬件增强技术.

(5)基于隐私挑战(3)和(4)，即使数据被加密传输且仅在可信环境下执行明文计算，有研究工作表明攻击者也可以观察内存层的访问模式和网络层的访问模式，进一步地推测出数据隐私. 基于此隐私保护需求，如何在计算过程中防止访问模式泄露是个挑战，相应的隐私保护技术主要是基于ORAM或者不经意混洗等技术隐藏访问模式.

2.2.3 主要研究方向

我们调研了近年来发表在信息安全顶级会议/期刊(例如USENIX Security、CCS、S&P、NDSS、和TDSC等)上关于大数据计算环境下隐私保护技术的国内外研究进展，总结了5个研究方向. 表1对比分析了不同研究方向的主流隐私保护技术、主要研究内容、抵抗的敌手模型、解决的隐私问题、技术途径、局限性和代表性研究工作等.

(1)基于数据分离的隐私保护：考虑到敏感数据或者全部原始数据在本地或者私有云环境被处理的隐私保护需求，该类研究方向主要借助数据分离技术抵抗不可信的CSP，解决了输入隐私问题. 主要研究工作包括基于敏感数据标记的分布式计算[24-30]和基于数据分离的联邦学习[31-41]，为了保持处理结果的一致性，私有云与公有云之间或者本地与第三方服务器之间的交互会带来较高的通信开销. 具体的研究现状与进展见第3节.

(2)基于数据干扰的隐私保护：考虑到数据需要去隐私后发布到不可信第三方的隐私保护需求，为了抵抗不可信的CSP或者不可信的数据消费者，该类研究方向主要是在数据输入或者计算结果发布之前，利用数据匿名或者差分隐私技术泛化、压缩或者随机扰动真实数据，解决输入隐私和输出隐私问题. 有关数据匿名技术的主要研究工作包括多维数据的匿名发布[42-48]和高效的大数据匿名并行处理[49-58]；有关差分隐私技术的主要研究工作包括基于LDP的隐私保护[18-20,59-66]和基于CDP的隐私保护[21-23,67-74]. 由于数据失真会降低数据的可用性，因此在实际应用中需要权衡隐私性、可用性以及高效性. 具体的研究现状与进展见第4节.

(3)基于安全多方计算的隐私保护：考虑到云服务提供商不可信，需要数据加密上传后并密文计算的隐私保护要求，该类研究方向主要借助安全多方计算协议允许互不信任的参与方安全地执行联合计算，同时不泄露计算数据的隐私. 根据实现安全多方计算协议的底层密码学原语不同，主要研究工作包括基于混淆电路的安全多方计算[75-83]、基于秘密共享的安全多方计算[23,83-88]以及基于同态加密的安全多方计算[89-97]等. 考虑到大数据计算环境下的实际应用，侧重于实现高实用的安全多方计算协议. 具体的研究现状与进展见第5节.

(4)基于硬件增强的隐私保护：由于密文计算在实际应用中面临性能瓶颈，考虑数据被加密传输但在TEE环境下执行明文计算的隐私保护需求. 该类研究方向主要是利用Intel SGX技术提供安全隔离执行环境，保护关键数据和代码的机密性，能够抵抗不可信的CSP在计算过程中窃取隐私数据，解决计算隐私问题. 主要研究工作包括基于硬件增强的大数据计算框架[5-6,8,98-99]以及计算性能优化[100-103]. 具体的研究现状与进展见第6节.

(5)基于访问模式隐藏的隐私保护：由于Intel SGX技术在实际应用中面临侧信道攻击，例如切换内存页会暴露内存层访问模式[7]，计算节点间的通信流量会暴露网络层访问模式[8]. 因此，需要考虑隐藏访问模式的隐私保护需求. 该类研究方向主要利用不经意的访问模式隐藏技术实现隐私计算，主要研究工作包括在大数据环境下实现基于ORAM的

表1 大数据计算环境下隐私保护的五类研究方向

隐私挑战	研究方向	主流的隐私保护技术	主要研究内容	抵抗的敌手模型	解决的隐私问题	技术途径	局限性	代表性研究工作
敏感数据或原始数据在本地私有云存储并计算	基于数据分离的隐私保护	数据分离	(1)基于敏感数据标记的分布式计算 (2)基于数据分离的联邦学习	不可信的CSP	输入隐私	根据数据的敏感性将敏感和非敏感数据，及相关计算分离	本地(私有云)与第三方(公有云)之间的通信开销较高	HybrEx[24], Sedic[25], Tagged-MapReduce[26], SEMROD[27], Hogwild[38], MOCHA[41]等
数据去隐私后发布	基于数据干扰的隐私保护	数据匿名	(1)多维数据的匿名发布 (2)高效的大数据匿名并行处理	不可信的CSP，不可信的数据消费者	输入隐私 输出隐私	通过压缩、泛化、随机响应或Laplace加噪等手段干扰数据	数据失真，降低数据的可用性，影响数据处理的准确性	RAPPOR[19], PINQ[21], Airavat[22], MRBUG[49], MRTDS[51], Spark-subtree[56], DPSGD[73], DPFL[74]等
		差分隐私	(1)基于LDP的隐私保护 (2)基于CDP的隐私保护					
加密传输，密文计算	基于安全多方计算的隐私保护	安全多方计算	(1)基于混淆电路的安全多方计算 (2)基于秘密共享的安全多方计算 (3)基于同态加密的安全多方计算	不可信的CSP	计算隐私	允许在加密的数据集上执行计算和查询等操作	密文计算带来计算开销和通信开销等性能瓶颈	Fairplay[77], SecureTPC[81], SMCQL[82], Conclave[83], Sharemind[85], PrivateMR[86], MrCrypt[95], Crypsis[96], SecureMR[97]等
加密传输，可信硬件支持明文计算	基于硬件增强的隐私保护	Intel SGX	(1)基于硬件增强的大数据计算框架 (2)基于硬件增强的计算性能优化	不可信的CSP	计算隐私	提供安全隔离执行环境，保护关键数据和代码的机密性	内存加密存在高昂的性能开销，面临侧信道攻击威胁等	M2R[5], VC3[6], ObserMR[8], Opaque[99], Haven[100], SCONE[102], Ryoan[103]等
加密传输，计算过程暴露访问模式	基于访问模式隐藏的隐私保护	不经意计算	(1)基于ORAM的不经意计算 (2)基于不经意混洗的不经意计算	不可信的CSP	计算隐私	隐藏访问模式，例如页面访问和网络流量，抵抗侧信道攻击	不经意计算带来的通信开销较高，执行效率较低	M2R[5], Controlled-channel[7], ObserMR[8], Opaque[99], ObliVM[106], GraphSC[107]等

不经意计算[104-107]和基于不经意混洗的不经意计算[5,8,99]. 具体的研究现状与进展见第7节.

3 基于数据分离的隐私保护

随着数据持有者的数据不断增长，对数据的维护成本越来越高，导致数据处理的部分或者全部任务从本地(或私有云)迁移到公有云. 虽然数据处理的位置发生了变化，但是对敏感数据的隐私要求没有改变. 假设云服务提供商作为不可信的第三方，一旦存在内部攻击者或者软件脆弱性等潜在的安全风险，将直接造成数据隐私泄露.

为了解决上述问题，研究者们提出根据数据的敏感性分离存储和计算数据的解决思路，即基于数据分离的隐私保护方法. 相比其他的隐私保护方法，该方法保证了敏感数据在本地或者私有云环境被安全高效的处理，而不会被迁移到不可信的公有云环境. 目前，基于数据分离的隐私保护相关研究工作主要包括基于敏感数据标记的分布式计算和基于数据分离的联邦学习，下面展开详细的介绍.

3.1 基于敏感数据标记的分布式计算

主要思想是借助混合云环境，首先对原始数据中包含的敏感数据进行标记，将数据划分为敏感数据集和非敏感数据集，然后将在不同数据集上的相

关计算任务也进行划分，并将非敏感数据及其相关的计算任务外包到公有云存储并计算，而小规模的敏感数据及其相关的计算任务保留在本地或者安全的私有云执行.因此分布式计算即保证了隐私性又具有弹性.基于敏感数据标记的分布式计算研究工作中最主要的两个挑战点分别为：

(1)数据的敏感性标记：一方面需要依据数据所有者的隐私保护需求对原始数据集打标签，另一方面，对于大规模数据集而言，标记效率也是值得思考的问题.例如，是否支持自动地标记敏感数据，以及是否支持动态地更新数据敏感性.

(2)混合云中的分离执行：以MapReduce为代表的大数据计算框架最初仅适用于部署到单个云，它们没有考虑到数据具有不同的安全等级.由于缺失支持混合云环境的计算框架，迫使编程人员需要手动分割并提交每个计算任务到公有云或私有云，这严重地妨碍了数据代码的重用.因此，在混合云中分离执行时需要考虑数据分离、代码重用以及跨云聚合等问题.

针对上述挑战问题，学术界和工业界提出了很多解决方案，相关的研究工作梳理和总结如下.

3.1.1 数据的敏感性标记方法

Ko等人[24]首次在MapReduce计算框架下提出了一种数据分离策略，使用私有标签(private label)和公有标签(public label)两种标签将输入数据划分为敏感和非敏感两个部分.类似于信息流控制技术中使用标签来控制和追踪信息在系统组件之间的流动[108-109]，受此启发，分布式计算框架和底层的文件系统能够根据这些标签确定数据的敏感性和计算任务的位置，将非敏感数据和相关计算调度到公有云，敏感数据和相关计算调度到私有云.但是它假设数据所有者决定了数据敏感性，在作业执行之前标记敏感数据，并手动地分配计算任务到公有云和私有云，这限制了大规模数据标记的效率.

为了解决这个问题，Zhang等人[25]提供了一种隐私感知的数据密集型计算框架Sedic.它集成了数据标记工具，根据所处理的数据安全等级在混合云中自动地划分和分配计算任务，同时它也支持对小规模数据集的手动标记方式.主要区别在于：(1)手动方式：在私有云环境，根据数据的敏感性策略，敏感数据被打上私有标签，非敏感数据被打上公有标签.(2)自动方式：在私有云环境，通过运行一个简单的字符串扫描器，在给定的数据集中搜索关键字或其他描述个体敏感信息的文本模式.一旦找到这些目标，创建一个安全标签：元组(＜filename，offset，length＞)来记录信息的位置.

但是上述文献只考虑作业执行之前的数据敏感性，忽略了计算过程中数据的敏感性变化.文献[110]指出Sedic[25]中所采用的清理方法(sanitation approach)披露了敏感数据的相对位置和长度，会泄露重要信息.为了解决这个问题，文献[26]提出了一种元组级别(tuple-level)的数据标记方法，并采用非升级策略(non-upgrading policy)和降级策略(downgrading policy)动态地识别每个键值对的敏感性，更细粒度地控制公有云与私有云之间的数据传输.其中，非升级策略是指如果某个map/reduce计算的所有输入元组是非敏感的，那么所有的输出元组也应该被标记为非敏感的；降级策略是指在某些情况下，即使某个map/reduce计算的输入元组是敏感的，输出元组也可能被降级为非敏感.由于经过降级策略后的非敏感元组可以传输到公有云，因此对于输入数据敏感但经过简单预处理后只有少数数据敏感的应用程序，该策略非常有用.

3.1.2 混合云中的分离执行模型

混合云中的分离执行模型主要研究工作是将数据分离技术用于大数据计算框架中，代表研究工作包括HybrEx[24]、Sedic[25]、Tagged-MapReduce[26]、SEMROD[27]以及HKF-SEMROD[28]等.其中，HybrEx[24]和Sedic[25]没有考虑到迭代计算，仅支持单层MapReduce作业(简称MR作业)；后续的研究工作考虑到更细粒度的敏感数据标记，也支持多轮迭代计算和链式计算.

HybrEx[24]基于Bigtable分布式存储系统和MapReduce分布式计算框架，首次提出了支持四种分离执行策略的安全计算框架，分别是Map混合(map hybrid)、水平分割(horizontal partitioning)、垂直分割(vertical partitioning)和混合(hybrid)等四种MapReduce执行模型.

(1)Map混合：如图3(a)所示，敏感数据的map阶段在私有云中执行，非敏感数据的map阶段在公有云中执行.然而，公有云中的map输出要发送到私有云中，reduce阶段只在私有云中执行.

(2)水平分割：如图3(b)所示，数据存储在私有云，map阶段在私有云中执行，然后数据混洗后加密传输到公有云，执行reduce阶段，该执行模型适用于长期归档数据的应用场景.

(3)垂直分割：如图3(c)所示，敏感数据和非敏感数据分别独立地在私有云和公有云中执行map/

reduce阶段，不允许跨云传输数据.

(4)混合：如图3(d)所示，分别独立地在私有云和公有云中对敏感数据和非敏感数据执行map阶段，混洗后跨云传输数据，然后在私有云和公有云中分别执行reduce阶段.

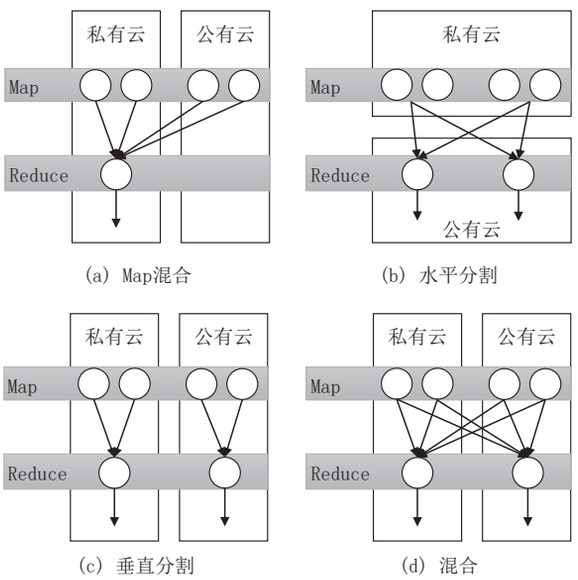


图3 HybrEx支持的四种MapReduce执行

虽然HybrEx[24]提出的分离执行模型实现简单，但是其存在两个缺陷.一方面是在作业运行之前，根据隐私保护需求，敏感数据需要被静态地标识，计算任务需要被手动地划分到混合云，这限制了大数据处理的效率；另一方面是它没有考虑到跨云传输元组带来的云间(inter-cloud)通信开销，它可以通过度量跨云传输的数据量来衡量.

为了解决HybrEx[24]存在手动分配计算任务的缺陷，Sedic[25]采用自动化分析和代码转换技术，为Map混合执行模型中公有云map与私有云reduce之间的数据传输提供了隐私感知的数据移动策略.通过修改分布式文件系统HDFS的副本机制，该策略要求仅含敏感数据的块被复制到私有云的私有数据节点(DataNode).对于仅含非敏感数据的块，首先被发送到公有云，然后复制副本数据块到公有云的公有数据节点.而对于包含敏感数据和非敏感数据的块，名称节点(NameNode)首先启动私有数据节点复制该数据块，然后将块中的非敏感数据作为新数据块移动到公有数据节点.该策略不仅保证了非敏感数据及相关计算任务尽可能多地外包给公有云，而且也保证了敏感数据及相关计算任务仅在私有云执行.此外，为了降低公有云和私有云之间的通信开销，Sedic[25]聚合公有云上的map输出后，再将其传输到私有云进行最终的reduce操作，它实现了如图3(a)所示的Map混合执行模型.由于它要求所有的reduce计算必须在私有云执行，这限制了Sedic框架的扩展性，尤其不适用于处理复杂的分布式计算任务，如链式或迭代计算.

基于此，Tagged-MapReduce[26]采用更细粒度的元组标记策略，支持包含敏感信息的数据可以被后续作业执行.它不仅兼容单阶段(single-phase)调度，而且也实现了两阶段交叉(two-phase crossing)、两阶段非交叉(two-phase non-crossing)和切换(hand-off)等三种调度模式.在执行作业时，根据数据的敏感等级灵活地启动不同的调度模式.实验结果表明，当原始数据集中敏感数据的比例为20%，且以两阶段交叉模式调度链式的人脸检测作业时，Tagged-MapReduce[26]相比Hadoop MapReduce，云间通信开销增长了95%；以两阶段非交叉模式调度时，云间通信开销增长了73%.

然而，以上研究工作都没有考虑map阶段在公有云和私有云生成相同键的情况.虽然敌手无法对私有云发起任何攻击来获取敏感信息，但是它可以窃听公有云和私有云之间的通信.通过观察shuffle阶段跨云传输的键值对，推测出哪些键和敏感信息相关联，从而确定敏感键，这间接地造成了键推理暴露(key-inference exposure).

为了抵抗这种攻击，SEMROD[27]提出了一种高效的MapReduce隐私保护框架.通过追踪变得敏感的中间键(intermediate key)动态地调度任务，敌手既不能从存储在公有云的数据中，也不能从公有云与私有云之间的通信中获取有关敏感数据的任何额外信息.为了防止键推理暴露，混洗阶段不仅从公有云拉取包含敏感键的键值对到私有云，也拉取公有云上的所有map输出.但是该方法带来了混洗阶段的冗余计算，特别是当运行多层MapReduce作业时，冗余计算会导致通信开销增加和负载不均衡.通过测试单词统计、排序和中位数等不同作业的混洗开销，结果表明相比Hadoop，大部分作业带来的云间通信开销增长超过1倍，其中单词统计作业带来的云间通信开销增长了高达1.77倍.针对SEMROD存在的两个问题，HKF-SEMROD[28]将其扩展为更加灵活高效的分离执行模型，要求在第二个或者后续的作业中继续保持第一个作业中的键；Oktay等人[29]利用结构化查询SQL语义实现分区执行，从而降低云间通信开销.Zhou等人[30]采用自动安全的数据分区技术和具有隐私保护的作业追踪器，实现了混合云上

隐私保护的数据检索，同时优化云间通信开销.

通过以上分析，混合云中采用数据分离技术能够有效地防止敏感数据泄露，同时在主流的大数据计算框架下能够实现灵活且可扩展的分离执行模型. 表2对比了基于敏感数据标记的分布式计算相关研究工作，未来还需要进一步地考虑输入数据集中敏感信息比较密集的情况. 在这种情况下，敏感数据标记将面临一定的困难，并且也会进一步加重私有云负载，本身算力受限的私有云将成为制约大数据高效处理的瓶颈.

表2　基于敏感数据标记的分布式计算相关研究工作

分类	挑战	核心方法	代表性工作	主要优点	主要缺点	云间通信开销
数据的敏感性标记方法	如何有效地标记数据	文件级别的标记	HybrEx[24]	容易部署实施	手动标记效率低	--
		元组级别的标记	Tagged-MapReduce[26]	细粒度的标记便于追踪	私有云负载加重	1.73～1.95×
混合云中的分离执行模型	如何有效地分离数据和调度任务	单层MR作业执行Map混合	Sedic[25]	隐私感知的数据移动策略	可能暴露敏感信息位置	1.17～1.80×
		多层MR作业追踪执行过程中敏感的中间键	SEMROD[27]	防止键推理暴露	云间通信开销较高	1.66～2.77×
			HKF-SEMROD[28]	调度策略灵活高效	没有在分布式环境验证	--

注：--表示没有测试.

3.2　基于数据分离的联邦学习

为了解决输入隐私问题，避免数据输入阶段隐私泄露，学术界提出了原始数据全部在本地存储及计算的思路. 特别是对于敏感信息比较密集，且不太容易被标记和划分的原始数据集，例如医疗数据集. 出于隐私保护需求，基于数据分离的联邦学习允许在远程设备或者孤立的数据中心（例如移动终端或医院）来训练机器学习模型. 如图4所示为通用联邦学习架构，多个本地设备（数据持有者）与中央参数服务器之间经过本地训练、上传本地更新、服务器端安全聚合以及下载全局模型等步骤保证联合训练模型的一致性. 由于联邦学习将训练数据集与训练中的模型参数分离，保证训练数据集在本地进行训练. 因此，基于数据分离的联邦学习能够有效地保护训练数据集的隐私.

尽管如此，在训练过程中本地设备与中央参数服务器之间更新的模型参数也有可能会泄露模型信息[31-32]，因此还需要进一步地保护模型参数的隐私. 近年来，联邦学习隐私保护是学术界的热门研究方向. 为了保护训练数据隐私和模型隐私，主要利用差分隐私、同态加密或者安全多方计算等技术进一步地增强隐私保护. 但是相比集中式学习，联邦学习通常会降低模型精度或训练效率[33-34]. 如何权衡隐私性、模型精度和收敛性能是联邦学习隐私保护中需要解决的问题，相关研究工作将在后续隐私保护技术章节进行详细地梳理和总结（见第4.2节和第5.2节）.

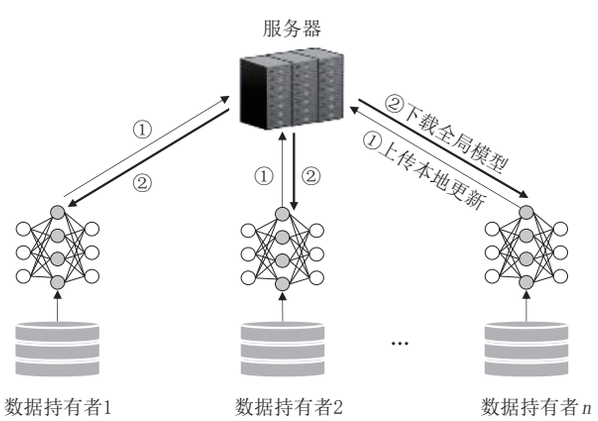


图4　通用联邦学习架构

目前，工业界已经部署联邦学习方法到实际应用中，并在隐私敏感的应用中发挥关键作用[35]. 虽然联邦学习可以保护训练数据集的隐私，但是在实际应用中面临着各种性能挑战，挑战问题及研究进展总结如下：

（1）通信开销：由于本地设备的带宽、电力和功率等资源有限，使得联邦学习中的通信可能比本地计算慢许多数量级[111]. 目前，学术界主要从减少通信轮数以及每轮传输的消息大小两个方面提高通信效率. Stich等人[36]采用最小批优化方法改进本地更新策略，可以减少通信轮数；Wang等人[37]采用稀疏化和量化的模型压缩方法，显著地减少每轮传输的消息大小. 但是设备的低参与性和非独立同分布（Non-Independent Identically Distribution，Non-IID）的本地数据对模型压缩方法提出了新的挑战，特别是在低带宽或高延迟的网络环境中.

（2）系统异构：由于联邦学习中每个设备的硬件及网络等配置不同，因此系统特性存在很大差异. 目前，学术界主要采用异步通信、主动采样设备和容

错等方式来解决系统异构性挑战. Recht等人[38]指出异步通信是一种用于缓解延迟的有效方式；Nishio等人[39]基于系统资源提出了一种新颖的设备采样策略，让服务器在预定的时间窗口内聚集尽可能多的设备更新；Bonawitz等人[40]采用忽略故障设备的简单容错机制，偏向在具有良好网络条件的设备上训练模型. 但是，这些方法都假定系统的网络特性为静态模型，如何更好地扩展这些方法以处理实时的计算和通信仍然是一种挑战.

(3)数据异构：由于本地设备的不同，它们经常生成和收集Non-IID数据，导致数据中包含设备之间的关系及相关分布. 这违背了主流机器学习算法基于独立同分布的假设，因此使得联邦学习下的建模更加复杂. 目前，学术界主要从异构数据建模和相关训练过程的收敛分析两个方面展开研究. Smith等人[41]提出了一种适用于凸目标模型训练的优化框架MOCHA，允许每个设备单独学习相关的模型来实现个性化，理论上保证了可证明的收敛结果. 但是MOCHA框架不适用于大规模网络环境，后续研究者们提出了可以处理非凸函数且可扩展到大规模网络的异构数据建模方案[112-114].

3.3 小结

通过本节的梳理总结，可知基于数据分离的隐私保护方法中最主要的两种技术手段分别是借助混合云实现基于敏感数据标记的分布式计算，以及基于数据分离的联邦学习. 对比分析如下：

(1)共同点是通过敏感数据集在本地或者私有云环境存储和计算，保证了原始数据集中敏感数据的输入隐私. 但是，基于数据分离的隐私保护面临着通信开销较高的局限，如前面提到的在低带宽或高延迟的网络条件下，混合云中的分离执行模型带来的云间通信开销，以及联邦学习中本地设备与中央参数服务器之间的通信开销，未来还需要进一步地降低通信成本.

(2)不同之处在于基于敏感数据标记的分布式计算将非敏感的数据分配到公有云环境，充分利用公有云的计算能力提高大数据处理效率，更适合处理敏感数据不密集的原始数据集；而基于数据分离的联邦学习要求所有数据在本地存储并训练，更适合处理敏感数据密集的原始数据集.

4 基于数据干扰的隐私保护

近些年发生的隐私泄露事件都表明未经过脱敏处理的数据在云上发布或者共享很容易泄露数据隐私，特别是个体敏感信息. 在基于云的大数据计算环境下，随着大数据分析的普及，云服务提供商越来越热衷于聚合来自数据持有者的数据，以获取更有价值的结果. 但是假设云服务提供商或者数据消费者不可信，对数据具有访问权限的云平台内部攻击者可能窥探其中的个体敏感信息，造成输入隐私泄露. 具有背景知识的外部攻击者可能提交恶意程序获取特定的输出结果，试图推测出能够关联到特定个体的敏感信息，造成输出隐私泄露.

为了解决上述问题，研究者们提出基于数据干扰的隐私保护. 相比其他的隐私保护方法，该方法会造成数据失真，通过牺牲数据的精度来增强隐私保护水平，因此隐私性和可用性的权衡问题一直是该研究方向的热点问题. 目前，基于数据干扰的隐私保护方法中最主要的两种隐私保护技术是数据匿名和差分隐私，本节重点围绕大数据计算环境介绍相关研究工作进展.

4.1 数据匿名技术

数据匿名是较早提出的隐私保护技术，代表性算法包括k-匿名[115]、l-多样化(l-diversity)[116]和t-紧密度(t-closeness)[117]等. 主要采用泛化、抑制、解剖(anatomization)、置换和扰动等五种类型的匿名化操作，限制准标识符与敏感信息之间的链接关系[118-119]. 在大数据计算环境下，针对大规模数据的多维特性以及并行且可扩展的高效处理需求，利用数据匿名技术保护大数据隐私面临着新的挑战：

(1)多维数据的匿名发布：在多维的大规模数据集上直接执行上述匿名化操作会严重地降低数据可用性，带来大量的信息损失. 如何设计适用于多维数据的匿名算法，包括静态数据匿名和数据流匿名，并在匿名数据的隐私保证下提高发布数据集的可用性是一个重要的挑战.

(2)高效的大数据匿名并行处理：在大数据计算环境下，将上述匿名化操作直接应用到大规模数据匿名处理时，会带来扩展性和效率的挑战. 因此需要研究高效的大数据匿名并行处理方法，提高数据的匿名效率.

针对上述挑战问题，学术界和工作界提出了很多解决方案，下面梳理和总结相关代表性工作.

4.1.1 多维数据的匿名发布

在大数据计算环境下，考虑到大规模数据的收集状态主要为静态批数据和数据流形式，下面从静态数据的匿名化和数据流的匿名化两个角度总结有

关多维数据匿名发布的研究进展.

多维静态数据的匿名化研究工作主要基于k-匿名算法展开，研究如何对原始数据集进行有效的匿名化，尽可能地实现匿名效果最好、数据可用性最高且时间空间开销最小的匿名算法. 所提出的基于k-匿名的子树泛化方法大致分为两类，一类是从属性分类树(taxonomy tree)的底部向顶部泛化数据，称为自底向上泛化(Bottom-Up Generalization，BUG)[42]；另一类是从属性分类树的顶部到底部遍历，称为自顶向下规范(Top-Down Specialization，TDS)[43]. 但是，上述匿名化算法不适用于数据流匿名. 由于数据流是连续的，需要及时匿名发布，并且任意时刻发布的匿名数据都是局部的，因此匿名算法的多次非独立应用会引起信息披露.

针对这些挑战，多维数据流的匿名化主要基于扰动、树状结构、伪造值和聚类等方面展开研究. Li等人[44]提出了一种基于随机扰动的数据流匿名方法，该方法在给定可用性的情况下最大限度地保护数据流隐私. 但是该方法只能处理数值数据，并且过多添加的噪声会降低匿名数据的可用性. Zhou等人[45]提出了基于树状结构的数据流匿名方法，时间复杂度为$O(|S|\delta\log\delta)$，空间复杂度为$O(\delta)$，其中$|S|$表示数据流的大小，δ表示可接受的树数量. Kim等人[46]提出了一种基于伪造值的无延迟数据流匿名方法，通过泛化准标识符和添加伪造值到敏感属性保证l-多样化. 但是，在匿名发布大规模数据流时，上述方法会带来较高的信息损失. 为了提高匿名数据的可用性，Cao等人[47]利用匿名簇和非匿名簇泛化元组，提出了一种基于聚类的数据流连续匿名发布方法. 虽然它有效地降低信息损失，但是会带来较高的时间复杂度$O(S^2)$. Guo等人[48]提出了一种改进的基于聚类的数据流匿名算法，该算法的时间复杂为$O(|S|)$，与数据流大小成线性关系，空间复杂度为$O(C)$，即受常数C的约束.

4.1.2　高效的大数据匿名并行处理

随着需要匿名的数据集规模急剧增加，如何高效地匿名大规模数据集成为传统匿名算法面临的挑战. 学术界已对大数据匿名并行处理的可扩展性和效率问题展开了较多研究，相关研究工作主要有两种解决思路：一种是借助主流的大数据计算框架实现分布式的匿名并行处理，例如MRBUG[49-50]、MRTDS[51]、TPTDS[52]、混合子树[53]和SparkDA[54-56]等；另一种是将已有的匿名算法并行化，以适应大数据计算环境下的分布式处理特性，例如采用可伸缩的决策树和采样[120]、R数索引[121]或者并行数据流[122]等. 其中，大多数研究工作基于第一种思路展开，它们充分利用了计算框架本身的扩展性和高效优势，在大数据计算环境下更加实用，下面重点总结这方面的代表性研究工作.

Irudayasamy等人[49]借助MapReduce计算框架首次实现了并行BUG方法，简称MRBUG，它仅支持任务级的并行化，即多个mapper/reducer任务在数据分区上并行执行；Pandilakshmi等人[50]提出了在更小的分区数据集上运行MRBUG，支持作业级和任务级两个级别的并行化，但是存在邻近记录的隐私披露问题；Balusamy等人[51]借助MapReduce计算框架实现了可扩展的TDS方法，简称MRTDS；但是该方法没有解决负载均衡调度问题，两阶段的TDS方法(Two-Phase TDS，TPTDS)[52]被提出；虽然该方法能够通过索引匿名数据记录提高TDS的可扩展性，但是两阶段处理带来的计算复杂度问题还需要进一步优化；Zhang等人[53]组合TDS与BUG技术提出了一种混合子树匿名方案，降低了计算复杂度和执行时间；但是该方案存在邻近记录的隐私披露问题，Zhang等人[57]采用局部编码策略提出了可扩展的两阶段聚类方法；为了抵抗多个数据源的非独立发布所造成的身份披露，Mehta等人[58]首先将输入数据集划分为较小的等价类，然后再多次迭代逐步实现最小失真的k-匿名和l-多样化.

除了以上基于MapReduce计算框架实现可扩展的匿名方法之外，一些研究工作基于Spark计算框架实现了更高效的数据匿名并行处理方法，简称SparkDA. Sopaoglu等人[54]借助Spark计算框架改进了TPTDS算法的可扩展性和匿名效率，Ashkouti等人[55]借助Spark计算框架改进了分布式的Mondrian算法以满足l-多样化；虽然它们在应用层实现了特定的匿名化方法，但是却缺少通用的框架以支持多种匿名算法；Bazai等人[56]从框架层提出了基于Spark弹性分布式数据集的子树泛化策略，使用有效的数据分区、改进的内存管理和增强的迭代计算提高了匿名效率.

表3对比分析了大数据计算环境下基于数据匿名技术的隐私保护相关研究工作，从主要优点、主要缺点、抵抗的攻击类型、以及时间和空间复杂度等角度评测了有关多维数据的匿名发布和高效的大数据匿名并行处理方面的代表性工作. 分析可知，虽然数据匿名技术能够抵抗链接攻击和身份披露，但是它总会在新的背景知识假设下遭受新型攻击，例如同质攻击. 并且它无法抵抗最大背景知识下的差分攻击，也无法量化隐私保护水平.

表3 基于数据匿名技术的隐私保护相关研究工作

分类		核心方法	代表性工作	主要优点	主要缺点	链接攻击	身份披露	同质攻击	差分攻击	时间复杂度	空间复杂度
多维数据的匿名发布	静态数据匿名化	基于k-匿名的子树泛化	BUG[42]	从属性分类树的底部向上	匿名效率较低	√	√	×	×	$O(\|S\|\delta\log\delta)$	$O(\delta)$
			TDS[43]	从属性分类树的顶部向下	匿名效率较低	√	√	×	×	$O(\|S\|\delta\log\delta)$	$O(\delta)$
	数据流匿名化	基于树状结构	Stream-tree[45]	支持数据流的实时匿名	带来较高的信息损失	√	√	×	×	$O(\|S\|\delta\log\delta)$	$O(\delta)$
		基于聚类	Castle[47]	利用两个簇降低泛化失真	带来较高的时间复杂度	√	√	√	×	$O(S^2)$	$O(S^2)$
高效的大数据匿名并行处理	借助主流的大数据计算框架	基于MapReduce	MRBUG[49]	BUG支持任务级的并行化	可能带来邻近隐私披露	√	√	×	×	$O(\|S\|\log\delta)$	$O(\delta)$
			MRTDS[51]	TDS支持数据匿名并行化	没有解决负载均衡调度	√	√	×	×	$O(\|S\|\log\delta)$	$O(\delta)$
			TPTDS[52]	利用两阶段TDS提高效率	计算复杂度有待优化	√	√	×	×	$O(S^2)$	$O(S^2)$
			Hybrid-sub[53]	灵活地支持BUG和TDS	可能带来邻近隐私披露	√	√	×	×	$O(\|S\|\log\delta)$	$O(\delta)$
		基于Spark	DI-Mondrian[55]	改进分布式Mondrian算法	仅支持特定的匿名算法	√	√	√	×	$O((S^2)/\delta)$	$O(C)$
			Spark-subtree[56]	通用的框架支持子树泛化	没有考虑递归操作	√	√	×	×	$O(\|S\|)$	$O(C)$
	匿名算法并行化	决策树和采样	Workload[120]	支持工作负载感知的匿名	没有解决过拟合问题	√	√	√	×	$O(\|S\|\log\delta)$	$O(\delta)$
		R树索引	Indexing[121]	建立空间索引提高效率	对索引的维护成本较高	√	√	×	×	$O(\|S\|\log\delta)$	$O(\log\delta)$

注：√表示可抵抗，×表示不可抵抗.

4.2 差分隐私技术

差分隐私技术[123-124]主要通过添加噪音干扰真实数据，能够抵抗攻击者实施的背景知识攻击和差分攻击. 目前，学术界对差分隐私的数据发布和数据挖掘已经展开了较多的研究，而工业界更倾向于采用本地化差分隐私技术保护数据隐私，已经被应用到苹果iOS[18]、谷歌Chrome Web浏览器[19]和微软Windows[20]等软件系统中.

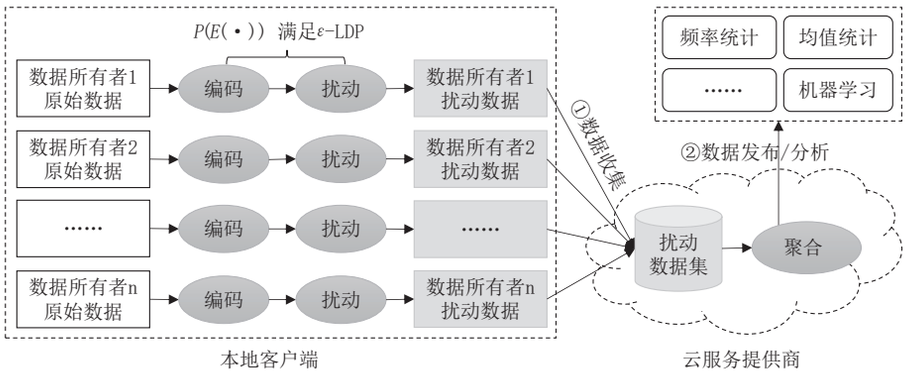
考虑到大数据计算环境下差分隐私保护的隐私性、可用性和性能等要求，根据本地客户端与云服务提供商之间的信任关系，如图5所示，展示了大数据计算环境下的差分隐私保护模型. 本小节主要围绕差分隐私技术在大数据计算环境下的应用，重点介绍关于基于LDP的隐私保护和基于CDP的隐私保护两个方面的研究工作进展.

4.2.1 基于LDP的隐私保护

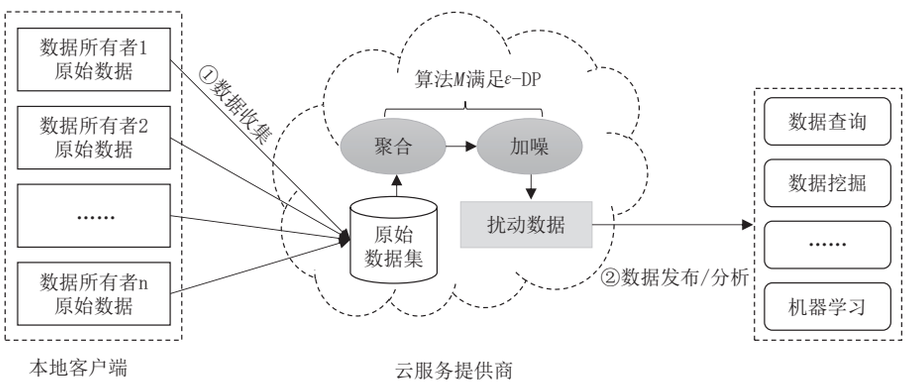
在大数据计算环境下，假设CSP是不可信的，为了防止输入隐私泄露，学术界和工业界开展了适合该场景的LDP研究. 如图5(a)所示为大数据计算环境下基于LDP的隐私保护模型，原始数据在本地编码和扰动后，扰动数据被收集到云端进行聚合. 由于本地客户端执行的随机扰动函数$P(E(\cdot))$满足ε-LDP，因此无论CSP内部攻击者具有怎样的背景知识，它都无法区分扰动元组v^*的原始元组是元组v还是另一个高置信度的元组v'.

目前，学术界对基于LDP的大数据隐私保护开展了广泛地研究，相关工作中涉及到的扰动机制要根据特定的应用类型而设计以保证隐私性. 因此，我们按照应用类型分类介绍有关隐私保护的统计类查询和机器学习等应用的研究工作. 下面在基于LDP的大数据隐私保护模型基础上，展开介绍相关应用中的代表性研究进展.

关于本地化差分隐私保护的统计类查询，云端聚合器旨在收集数据所有者的扰动数据以应答用户的特定查询，主要包括离散分类数据的频率统计和连续数值数据的均值统计. 其中，为了保证数据隐私性，目前学术界主要采用以Warner-RR模型[59]为代表的随机响应(Randomized Response，RR)扰动机制. 为了提高应答结果的可用性，学术界主要采用哈希、转换和子集选择等方式进一步地实现无偏估计[18-20,60-61].



(a) 基于LDP的大数据隐私保护模型



(b) 基于CDP的大数据隐私保护模型

图5　大数据计算环境下的差分隐私保护模型

例如，Erlingsson等人[19]采用哈希函数对单属性值编码后随机响应，提出了一种针对分类数据的高可用频率统计方法，但是会带来较高的通信成本$O(h)$，h为本地与服务器之间传输的向量长度. 除了上述简单的数据类型，少数工作关注大数据计算环境下的复杂键值型数据. Ye等人[62]首次针对键值型数据的频率和均值估计问题，基于现有LDP技术中的“扰动-校准”范式提出了PrivKV方案，并逐步迭代改善估计精度. 但是在加噪阶段，该工作忽略了键值对的相关性，Gu等人[63]基于抽样填充协议提出了更优的预算分配和组合策略，并进一步地改善评估精度.

关于本地化差分隐私保护的机器学习，需要在模型训练之前，先对本地训练数据进行随机扰动. 目前，学术界主要采用LDP技术保证集中式学习中数据收集阶段的训练数据隐私，以及联邦学习中上传至服务器的中间参数隐私. 在大数据计算环境下，由于训练数据的高维性和多轮迭代，训练出具有高隐私保障的精确模型是一个挑战点. Yilmaz等人[64]通过划分数据所有者或者属性集节省了隐私预算，继而提高了分类模型的精度. Wang等人[65]考虑到大数据计算环境下的分布式评估问题，设计出优化的k-子集机制保证本地训练数据的隐私，通信成本为$O(m)$，m为子集最大值. Wei等人[66]假设联邦学习框架中存在外部攻击者，提出在上载信道加入噪音来保护局部参数的隐私，并发现存在一个最优的客户端参与数$|N|$，能够在固定的隐私保护水平下使得多层感知训练达到最佳的收敛性能，但是通信成本与通信轮次r正相关.

4.2.2　基于CDP的隐私保护

中心化差分隐私适用于数据所有者信任云服务提供商的场景，如图5(b)所示为大数据计算环境下基于CDP的隐私保护模型. 在该模型下，原始数据被集中聚合后再添加适当的噪音，最终返回给用户隐私保证的聚合结果. 因此，它能够抵抗恶意敌手的差分攻击，避免输出隐私泄露.

相比LDP技术，基于CDP的隐私保护可以应用到更复杂的聚合任务，而不局限于简单数据类型的统计. 在大数据计算环境下，中心化加噪方式一般与具体的聚合算法是松耦合的，研究工作更侧重于权衡CDP方案的隐私性和可用性，下面主要从这两个度量指标展开介绍与大数据计算环境相关的中心化差分隐私研究工作.

在保证隐私性方面，隐私预算ε和敏感度Δf共同

决定加入噪音的大小. McSherry等人[21]利用隐私预算的序列组合特性为每次查询分配局部隐私预算以保证隐私性,但是由不可信的云用户请求局部隐私预算会造成噪音加入不合理,导致隐私预算浪费. 为此,Roy等人[22]提出由可信的噪音生成组件管理隐私预算,并在MapReduce计算框架下利用强制访问控制和差分隐私技术设计并实现了隐私保护系统. Haeberlen等人[67]指出如果查询用户与云平台内部的半诚实敌手勾结,通过观察执行完成时间和执行状态等隐蔽信道,能够实施计时攻击和状态攻击. 为了抵抗这两种攻击,Pettai等人[23]结合安全多方计算和差分隐私技术同时保证计算隐私和输出隐私. Ding等人[68]首次考虑隐私预算选取的正确性问题,设计了一种基于统计假设检验理论的反例生成器. Bichsel等人[69]考虑隐私预算选取的合理性问题,基于采样和优化等方法提出了一种能够自动生成隐私预算下界的工具. 为了避免隐私预算浪费,Lécuyer等人[70]基于主流的TensorFlow机器学习框架,设计出隐私预算消耗的度量模块,以满足多种机器学习算法的并行训练.

为了提高数据可用性,研究工作主要在满足一定隐私保护水平下尽可能地提高大数据分析结果的可用性. Chen等人[71]针对实时的流式大数据,采用滑动窗口和平滑优化手段实现了支持连续查询应答的高可用差分隐私保护系统. Avent等人[72]在混合信任假设下结合LDP和CDP技术实施灵活的隐私保护方案,明显地提高分析结果的可用性. 对于模型精度要求较高的机器学习算法,虽然通过扰动目标函数、中间参数或者模型输出可以提供隐私保证,但是对于复杂的深度学习模型往往需要多次迭代才能获取局部最优解,这将增加隐私预算才能达到预期的模型训练精度. 例如,Abadi等人[73]添加高斯噪音到随机梯度下降,并在TensorFlow机器学习框架上训练手写数字图像数据集MNIST,结果显示在$(8, 10^{-5})$的隐私预算保证下,神经网络模型才能达到97%的训练精度. Geyer等人[74]从客户端的角度出发,将上述工作扩展到联邦学习场景,并在相同的隐私预算下增加客户端数量来提高模型精度,但是带来了较高的通信成本.

表4对比分析了大数据计算环境下基于差分隐

表4 基于差分隐私技术的隐私保护相关研究工作

分类			代表性工作	主要优点	主要缺点	计时攻击	状态攻击	隐私性vs.可用性			通信成本
								数据集	隐私预算	精确度	
基于LDP的隐私保护	按大数据应用分	统计类查询	RAPPOR[19]	哈希编码后随机响应,可用性高	通信成本较高	√	√	分类数据	0.5	85.0%	$O(h)$
			PrivKV[62]	支持逐步迭代改善评估精度	忽略键值对的关联性	√	√	键值型数据	0.4	92.5%	$O(1)$
			PCKV[63]	利用抽样填充协议干扰键值	未优化填充长度l	√	√	键值型数据	0.4	99.3%	$O(l)$
		机器学习	k-SM[65]	优化的k-子集机制保护数据隐私	不适用非凸函数	√	√	分类数据	0.5	99.8%	$O(m)$
			NbAFL[66]	对上传至服务器的中间参数加噪	通信成本高	√	√	图数据	$(100, 10^{-2})$	85.5%	$O(\lvert N\rvert r)$
基于CDP的隐私保护	按度量指标分	隐私性	PINQ[21]	利用ε序列组合性分配隐私	隐私预算恶意消耗	×	×	数值数据	--	--	$O(1)$
			Airavat[22]	由可信的噪音生成组件管理ε	不适用复杂链式计算	×	×	数值数据	0.4	88.2%	$O(1)$
			Violation[68]	基于统计假设检验理论生成反例	无法约束预算上界	--	--	数值数据	--	--	--
			Sage[70]	采用块组合支持多训练任务	数据集独立同分布	×	×	分类/数值数据	$(1.0, 10^{-6})$	74.3%	$O(1)$
		可用性	PeGaSus[71]	采取平滑器优化分析结果	没有考虑递归操作	--	--	分类数据	0.01	66.7%	$O(1)$
			Blender[72]	采用混合策略灵活地加入噪音	没有解决过拟合问题	√	√	AOL log	1.0	96.2%	$O(h)$
			DPSGD[73]	随机抽样地添加高斯噪音	训练数据被集中训练	--	--	图数据	$(8.0, 10^{-5})$	97.0%	$O(1)$
			DPFL[74]	添加高斯噪音到全局梯度更新	通信成本较高	--	--	图数据	$(8.0, 10^{-5})$	92.0%	$O(\lvert N\rvert r)$

注:√表示可抵抗,×表示不可抵抗,--表示不考虑.

私技术的隐私保护相关研究工作，从主要优点、主要缺点、抵抗的攻击类型、隐私性、可用性和通信成本等角度评测了有关基于LDP的隐私保护和基于CDP的隐私保护方面的代表性工作.经过分析可知，差分隐私技术能够量化隐私保护水平，但是会牺牲分析结果的精确度.特别是对于模型精度要求比较高的深度学习模型，为了能够达到可观的精度一般要选取较大的隐私预算，导致隐私性和可用性的权衡问题变得困难.因此，如何合理地选取隐私预算值对差分隐私技术的实际应用非常重要.

4.3 小结

通过以上分析，基于数据干扰的隐私保护方法中最主要的两种隐私保护技术分别是数据匿名技术和差分隐私技术，它们在大数据计算环境下广泛地被应用.两种技术对比分析如下：

(1)数据匿名技术思想简单易理解，匿名算法更容易应用到主流的大数据计算框架中，执行高效的大规模数据匿名处理.但是，相比差分隐私技术，数据匿名技术在安全性方面较弱，仍然面临着背景知识攻击和差分攻击.

(2)差分隐私技术基于更严格的隐私定义能够量化隐私保护水平，更适用于对隐私保护需求比较严格的大数据应用场景，例如数据共享.相比其他的隐私保护技术，差分隐私技术应用到大数据计算环境时一般不会给复杂的计算任务带来过多额外的计算开销和通信开销，因此该技术具有较高的研究价值和应用前景.但是对于可用性要求比较高的机器学习应用，目前影响其在产业界实际应用的瓶颈是隐私性和可用性的合理权衡问题，未来还需要进一步地改善.

5 基于安全多方计算的隐私保护

在基于云平台的大数据计算环境下，假设多个数据持有者之间互不信任，但是出于商业合作的目的，他们需要共享数据以联合分析出更有价值的信息.如果对于共享的数据不进行加密或去隐私处理，那么将会直接破坏共享数据的机密性和隐私性.要么在数据共享之前对其进行干扰，但是会严重制约联合分析的任务类型和数据可用性，不适用于复杂的联合计算任务.要么对数据集进行加密后传输，因此，需要采取一种能够在敏感数据集上进行安全计算的隐私保护技术.

目前，学术界对安全多方计算协议有较多的理论研究[125-127]，它允许互不信任的各方在不泄露隐私数据的情况下进行联合计算.但是，其在实际的大数据应用中扩展性较差，一方面是在密文上执行复杂计算任务时其执行效率非常低，一般用执行时间或计算成本来衡量；另一方面是多方联合计算会带来较高的通信开销，一般用通信成本衡量.本节主要展开介绍大数据计算环境下高实用的安全多方计算研究工作进展，根据实现安全计算协议的底层密码学原语不同，分别从基于混淆电路、秘密共享和同态加密等三个角度重点归纳当前高实用的相关研究工作.

5.1 基于混淆电路的安全多方计算

基于姚氏混淆电路[75]的安全多方计算主要关注两方的场景，它使用布尔电路(boolean circuit)表述待计算函数，结合不经意传输(Oblivious Transfer，OT)技术设计安全多方计算协议.自姚氏混淆电路被提出以来，为了使它们扩展到大数据计算中，许多研究工作从降低通信成本[76]、缩减执行时间[77-80]和减少电路门数[81-82]等角度在一定程度上优化通信、优化执行和优化电路，下面从这三个优化角度出发，介绍具体的研究工作进展.

关于通信优化，研究工作旨在减少计算过程中参与方之间必须要传输的数据量，达到降低通信成本的目的.在基于姚氏混淆电路的安全多方计算工作中，各参与方之间按照传输的混淆电路执行以实现隐私计算.在评估过程中，混淆电路的大小远远大于输入数据大小，这使得在大数据计算的所有通信步骤中混淆电路的通信成本是最高的.因此，有效地压缩混淆电路大小可以在一定程度上降低通信成本.Goyal等人[76]提出了一种针对隐蔽敌手(covert adversary)的安全两方计算方案，该类型的敌手可能会偏离协议的执行以试图欺骗其他参与方.它采用剪切和选择技术减少两方之间传输的数据量和通信步骤数，并扩展到n个参与方，通信复杂度由$O(t|C|+tsm)$降为$O(|C|+sm+t)$.其中，$|C|$表示电路大小，m表示输入大小，s表示安全参数，t表示安全参数的多项式阶数.

关于执行优化，研究工作旨在减少执行相同数量的电路门所需要的计算时间，保证高效的大数据隐私计算.Malkhi等人[77]首次利用姚氏混淆电路实现了Fairplay优化方法，通过引入更快的查找表来减少混淆电路估值所需要的时间，并进一步扩展到多方计算[78].Huang等人[79]引入流水线电路执行的思想，通过并行处理部分电路的混淆步骤和估值步

骤，实现了更快速的安全两方计算协议.但是它假设存在半诚实敌手，区别于Fairplay方法的恶意敌手威胁模型，选择用更低的安全性换取更快的执行速度.基于已有的研究工作，Kreuter等人[80]结合查找表和流水线执行的思路提出了更快速的混淆电路实现方法，并且在恶意敌手模型下，它能够用更少的计算产生比Fairplay更大的电路.

关于电路优化，研究工作旨在降低某个计算操作的电路门数，从而降低隐私计算的复杂度.Pinkas等人[81]通过增加预处理步骤来优化混淆之前的电路，精简影响最终输出结果的电路门以达到电路优化的效果.实验结果表明该方法可以对通用电路生成器生成的电路进行优化，相比优化之前的电路大小，它缩小了60%的电路大小.基于此，Bater等人[82]执行本地预处理，将整个多方计算分为几个较小的多方计算以减少输入数据的大小，并设计了支持大数据分布式查询的安全系统SMCQL.

综上分析可知，目前主要采取剪切和选择、并行处理以及预处理等手段从不同的角度优化基于混淆电路的安全多方计算协议，使其能应用到实际的大数据计算场景.Volgushev等人[83]组合以上优化手段，通过数据并行、局部明文处理和较小的多方计算等步骤加速查询，并应用到Spark计算框架进一步地提高大数据查询的可扩展性.

5.2 基于秘密共享的安全多方计算

基于秘密共享[84]的安全多方计算主要关注三方及以上的场景，它将每个敏感值分割为多个“秘密共享”，这样每个秘密共享都不能泄露任何有关原始值的信息，但是当重组时原始值被重构.如果使用一个(k, n)门限秘密共享模型，n表示参与方总数，k表示门限，那么至少需要用k个秘密共享来重构敏感值.目前，大多数基于秘密共享的SMC研究工作基于Bogdanov等人[85]提出的Sharemind SMC框架展开.在该框架下，输入数据和指令被发送到多个计算参与方，数据库和堆栈中的数据在计算参与方之间秘密共享，即使存在妥协且合谋的参与方，仍然能够提供安全计算和数据隐私保护.

后续研究工作基于Sharemind SMC框架展开，使其适用于大数据计算环境下的隐私保护应用.与同态加密的技术思路相比，基于秘密共享的安全多方计算在计算成本上更低，而且它也支持更多的安全计算类型.例如，Dolev等人[86]基于秘密共享的思想提出支持计数、查找、提取、相等连接(equijoin)和范围查询等五种类型的MapReduce计算，同时保证公有云中数据和计算隐私.但是它带了较高的通信成本$O(m^2)$，m表示数据输入大小.Pettai等人[23]考虑到多个组织同时维护及拥有包含个体隐私的数据集场景，使用Sharemind SMC框架在敏感数据集上实现了支持浮点数操作和整数操作的安全计算.但是安全计算的输出结果无法解决恶意数据消费者实施的输出隐私问题，它进一步地提出了一种差分隐私和安全多方计算相结合的敏感数据隐私保护方法.但是它无法保证参与方的输入数据是真实的，后续有研究工作提出采用区块链技术保证参与方之间计算和数据的可验证性.

然而，随着参与方数量的增加，基于秘密共享的安全多方计算在计算过程中会带来较高的通信成本，这限制了方案的可伸缩性.为了能够扩展到多个参与方，一些研究工作借助主流的大数据计算框架实施并行处理，这能够有效地减少通信开销.Volgushev等人[83]将可扩展的Spark大数据计算框架与安全的Sharemind SMC框架相结合，提出了新的混合安全多方计算协议.与文献[85]相比，它将连接和聚合操作的执行效率提高了7倍之多，通信成本由$O(m^2)$降低为$\tilde{O}(m+o)$.其中，m表示输入大小，o表示输出大小.Dahl等人[87]提出了一个建立在TensorFlow机器学习框架之上的安全多方计算开源库，后续许多工作基于该开源库进一步地研究高效且高可用性的机器学习隐私保护.Kumar等人[88]从框架层提出将TensorFlow代码转换为安全多方计算协议支持的机器学习系统，它构建了端到端的编译器，支持代码转换为抵抗半诚实敌手的安全多方计算协议，并进一步地提出安全的三方协议以提高机器学习算法的训练效率.

5.3 基于同态加密的安全多方计算

基于同态加密[89]的安全多方计算允许计算参与方在加密数据上执行计算，同时能够保证公有云中数据隐私，其安全性和执行效率取决于底层同态加密方案的安全假设和加解密效率.为了使其能够应用到大数据计算环境下，一方面要设计能够支持大数据复杂操作的安全方案，另一方面需要重点解决加解密带来的高计算成本问题.

目前，学术界主要基于已有的同态加密方案构造支持大数据计算环境的安全多方计算协议，主要挑战在于如何在多方之间安全地共享和管理密钥.Asharov等人[90]基于带门限的全同态加密构建了支持隐私计算的多方计算协议.基于此协议，在每次计算阶段，所有计算参与方生成秘密密钥、公共密钥和

评估密钥等系统参数，并以较低的通信成本进行多方计算和交互. 为了满足大数据实时计算的要求，López-Alt等人[91]采用多密钥参与的全同态加密提出了支持云上实时多方计算的安全协议. 在安全计算的过程中，每个计算参与方都有他们自己的长期公私密钥对. 但是，当云服务提供商不可信时，密钥参数有可能会被恶意敌手获取，降低协议的安全性与隐私保证. 为了解决该问题，Martin等人[92]利用安全的匿名代理处理元数据和身份验证操作，提出了一种保护MapReduce计算隐私的全同态加密优化方案. 为了增强分布式计算过程的安全性，该方案利用默克尔(Merkle)哈希树结构来管理元数据认证和数据动态. 为了降低多项式时间的密文计算带来的效率较低问题，该方案使用一个优化的三叉树算法库处理元数据.

虽然全同态加密具有较高的安全保证，但是其计算成本较高. 目前，学术界主要采用并行处理、部分同态加密或者硬件加速等手段降低通信开销和计算开销. Peter等人[93]假设大数据计算环境下存在两个非竞争性云服务器，一个用于存储底层加密方案的主密钥，另一个用于存储所有计算参与方生成的密文，提出了支持加法同态加密的实用协议. Popa等人[94]假设云服务提供商内部存在半诚实敌手，提出了支持用户在加密数据上进行结构化查询的密文检索系统. 基于此，Tetali等人[95]提出了一种支持在加密数据上执行MapReduce计算的隐私保护机制，先在私有云中静态分析map/reduce代码以选择一个能够支持所有必要操作的最小同态方法，然后使用该加密方法对计算程序进行转换. 虽然该机制容易实现且通信成本较低，通信复杂度为$O(mwk)$，其中m表示输入数据大小，w表示模式长度，k表示属性数. 但是它仅支持密文上的相等和比较操作，采用的全同态加密技术限制了查询范围. 为了解决该问题，Stephen等人[96]提出一种支持数据流编程语言(Pig)的部分同态加密方案；Dong等人[97]利用部分同态加密技术设计并实现安全的MapReduce计算框架，称为SecureMR. 它通过并行处理提高计算效率，并将大数据应用的明文计算到密文计算的转换问题形式化为最小切割问题，以确保最优转换，进而尽可能地降低服务器与客户端之间的通信开销. 尽管如此，在实际测评中，相比执行基准的MapReduce作业，SecureMR还是增加了高达1.39倍到3.85倍的性能开销.

5.4　小结

通过以上分析，基于安全多方计算的隐私保护主要利用混淆电路、秘密共享和同态加密等密码学原语保证了大数据计算环境下的数据隐私和计算安全. 表5对比分析了基于安全多方计算的高实用隐私保护代表性工作，从核心技术、优化目标、参与方、攻击者模型、通信复杂度和时间复杂度等角度评测了有关代表性工作. 三类安全多方计算协议之间的对比分析如下：

表5　基于安全多方计算的高实用隐私保护工作比较

分类	代表性工作	核心技术	优化目标	参与方	攻击者模型	通信复杂度	时间复杂度
基于混淆电路的安全多方计算	GMS[76]	利用剪切和选择技术替代零知识证明	通信成本	两方/多方	隐蔽敌手	$O(\|C\|+sm+t)$	$\tilde{O}(n^3st)$
	Fairplay[77]	引入更快的查找表，减少电路估值的时间	执行时间	两方	恶意敌手	$O(m\log m)$	$O(m\log^2 m)$
	fasterSMC[79]	并行处理部分电路的混淆和估值步骤	执行时间	两方	半诚实敌手	$O(m\log m)$	$O(m\log^2 m)$
	SMCQL[82]	本地预处理以及划分较小的多方计算	电路门数	两方	半诚实敌手	$O(\log m)$	$O(m\log m)$
基于秘密共享的安全多方计算	Conclave[83]	基于Spark和Sharemind SMC框架	执行时间	多方	半诚实敌手	$\tilde{O}(m+o)$	$O(m)$
	PrivateMR[86]	基于MapReduce框架实现隐私计算	通信成本	多方	半诚实敌手	$O(m^2)$	$O(m)$
	Cryptflow[88]	基于TensorFlow框架设计端到端的编译器	通信成本	多方	半诚实敌手	$O(m^2)$	$O(m)$
基于同态加密的安全多方计算	MrCrypt[95]	静态分析和转换技术实现部分同态加密	计算成本	多方	半诚实敌手	$O(mwk)$	$O(mwk)$
	Crypsis[96]	支持查询操作的部分同态加密，安全性低	计算成本	多方	半诚实敌手	$O(mwk)$	$O(mwk)$
	SecureMR[97]	基于MapReduce框架实现部分同态加密	计算成本	多方	恶意敌手	$O(mwk)$	$O(mwk)$
	Op_FHE_SHCR[92]	利用匿名代理增强安全性，优化全同态加密	计算成本	多方	恶意敌手	$O(mwk)$	$\tilde{O}(m\sqrt{m})$

(1)基于混淆电路的安全多方计算其安全性和复杂性在于设计混淆电路,它能够在参与方之间互不串通的假设下保证安全计算.如果混淆电路设计简单那么安全性较低,同时复杂的混淆电路将带来较高的通信成本和计算成本.为了实现高实用性,既要从通信开销上降低乱码电路的大小,又要从执行效率上降低电路混淆和估值的时间,甚至从电路门数上对电路结构本身进行优化.

(2)相比混淆电路,基于秘密共享的安全多方计算能够更好地扩展到多方,即使大多数参与方被妥协或者参与方之间存在合谋时,它仍然能够执行安全计算并有效地保证数据的隐私性.相比同态加密,基于秘密共享的安全多方计算在执行时间上更加快速.但是在进行乘法计算时,各参与方之间需要多次交互大量数据,数据量与输入数据大小和输出数据大小有关,这带来了较高的通信开销.

(3)相比以上两种方式,基于同态加密的安全多方计算的架构简单,其应用到大数据计算环境的实用性依赖底层同态加密方案的执行效率.虽然利用全同态加密的安全多方计算方案能够有效地保证计算过程中的安全性,但是,全同态加密存在较高的计算开销.如果基于部分同态加密在加密数据上完成特定操作类型,它能够实现实用的协议.但是,其支持的计算类型有限,不能适用于复杂的大数据计算任务.因此,未来仍然需要研究效率更高且适用于大数据复杂计算的同态加密方案.

6 基于硬件增强的隐私保护

在基于云平台的大数据计算环境下,采用密码学手段加密数据并在其上执行安全计算存在计算开销和通信开销的性能瓶颈.出于隐私性和高效性的权衡,研究学者提出了基于硬件增强的"加密传输,明文计算"思路,即数据被加密传输但在可信硬件支持下高效地执行明文计算.特别地,当云平台部署的操作系统被妥协时,如何有效地抵抗具有根访问权限的攻击者通过执行恶意程序窃取数据隐私是一个值得被研究的问题.

目前行之有效的隐私保护手段从硬件增强的角度提供隐私保护,许多研究工作在大数据计算环境下借助Intel SGX技术的加密内存来保护关键代码和数据的机密性.相比其他的TEE技术,Intel SGX基于安全硬件的最小可信计算基(Trusted Computing Base, TCB)提供了用户空间的安全隔离执行环境,同时能够兼容虚拟化和容器技术;而AMD(Advanced Micro Devices)硬件虚拟化技术基于可信的特权软件(hypervisor)提供了操作系统级别的安全隔离执行环境,其安全性依赖特权软件的安全性,特别是当特权软件被妥协时其安全性受到威胁;TrustZone技术通过CPU将系统划分为安全和非安全的两种隔离执行环境,其主要应用到嵌入式平台.因此,在基于云平台的大数据计算环境下,采用Intel SGX硬件增强技术保护计算过程中数据隐私是比较热门的,本节主要梳理该领域的大数据计算框架以及计算性能优化的研究工作.

6.1 基于硬件增强的大数据计算框架

2013年,Intel公司为x86家族添加了一套新的处理器结构Intel SGX,旨在通过一组新的指令集扩展和内存访问机制,依靠受信任的硬件来保护用户级的代码和数据不受潜在的恶意特权软件的攻击[98].它也支持在现有的地址空间内创建安全的内存区域(Enclave Page Cache, EPC),而其余的内存地址空间是不可信的,并允许将应用程序实例化为一个受保护的容器enclave.

Schuster等人[6]采用Intel SGX技术为云环境中大数据计算提供了一个安全可信的执行环境,并提出了首个基于Intel SGX保护MapReduce分布式计算的安全系统,简称VC3系统.如图6所示,在MapReduce计算过程中,enclave外的数据被加密传输和存储,除非敏感数据和代码被加载到mapper或reducer任务节点上的可信处理内存中,加密数据被解密后以明文形式处理.此外,VC3系统隔离每个计算节点的内存区域,并部署加密协议来保护

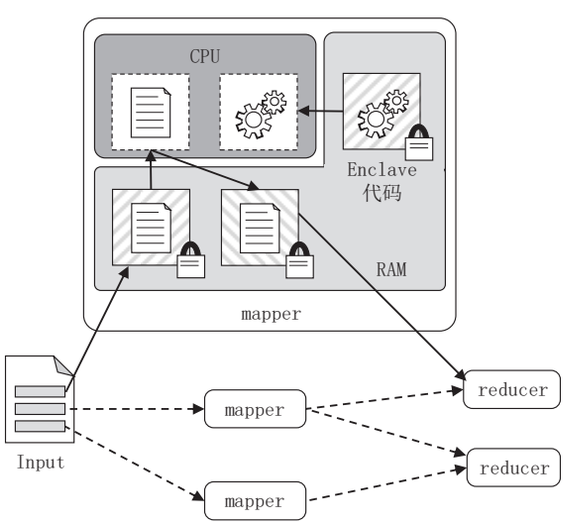


图6 VC3概述图

MapReduce分布式计算.为了防止不安全的内存读写造成内存攻击,该系统对加载到enclave内的代码强制执行区域自完整性检查,包括写完整性和读写完整性;不同于Haven[100]将库操作系统(library OS)和应用程序一起加载到enclave内,VC3系统只将用户指定且受保护的map/reduce函数加载到enclave内执行,将大数据计算依赖的应用软件(如Hadoop)和特权软件排除在TCB之外,从而构建最小TCB.因此,即使特权软件或者应用程序被妥协,该系统也能保证分布式计算过程中数据的机密性和完整性.

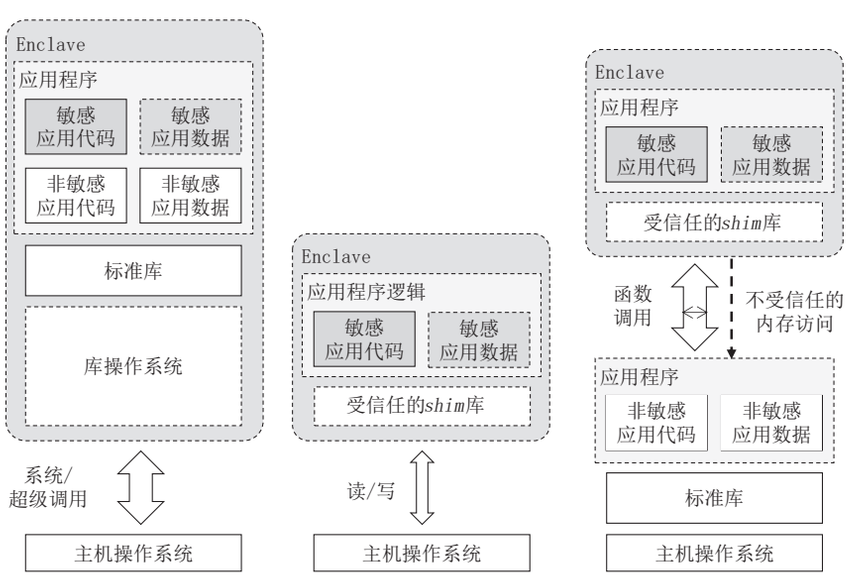
但是VC3系统没有解决侧信道攻击问题,攻击者可以观察计算过程中传输的加密数据流量,实施网络层的访问模式泄露[8];M2R[5]系统指出仅仅依靠硬件原语加密enclave内受保护的数据是不安全的,攻击者仍然可以进一步实施被动攻击和主动攻击.为了解决该问题,M2R系统采用不经意混洗技术隐藏了访问模式,具体细节下一节展开介绍.其中,被动攻击是指半诚实敌手能够利用数据流模式、执行顺序和访问时间等信道实施攻击;主动攻击是指恶意敌手通过篡改元组和错误路由元组等手段窃取数据的计算隐私.

Opaque[99]系统借助Intel SGX技术将Java代码划分为可信和不可信的部分,对于可信部分代码使用Intel SGX提供的应用程序接口将其重写为C/C++代码,同时采取不经意计算保护访问模式.但是代码重写对系统使用人员的编程能力提出更高要求,而且重写过程是容易出错的.为了解决重写代码带来的局限性,一些代码重用的方法被提出,例如SGX-Spark①.它使用专门线程在enclave中执行受信任的Java代码,并将加密的数据传递到enclave中进行解密和计算.尽管如此,现有基于Intel SGX的大数据计算框架仍然存在侧信道攻击的风险,并且在执行计算密集型作业时,EPC受限可能会带来严重的性能下降.

6.2　基于硬件增强的计算性能优化

由于Intel SGX需要对受保护的代码和数据提供机密性和完整性保证,因此数据加密和完整性验证过程增加了系统性能开销.在实际应用中,考虑到大数据隐私计算的高效性要求,需要对基于硬件增强的大数据计算进行性能优化.

当前,主要由三种可选的enclave安全接口设计方案,如图7所示.但是如何合理划分程序是当前的难点,Glamdring[101]是第一个用于自动划分Intel SGX应用程序的框架,可以将应用程序划分为可信和不可信两个部分.为了保护数据隐私性,它可以基于开发人员对于敏感数据的标记,执行静态数据流分析以检测所有访问敏感数据的函数;为了保证数据完整性,它采用静态向后切片(static backward



(a)完整的enclave接口　(b)预定义的enclave接口　(c)特定应用程序的enclave接口

图7　enclave安全接口设计方案

① SGX-Spark, this is Apache Spark with modifications to run security sensitive code inside Intel SGX enclaves. https://github.com/lsds/sgx-spark

slicing)来识别可能影响数据完整性的函数.在代码生成阶段,它在安全边界添加运行时检查和加密操作,保护enclave接口免受攻击.

Haven[100]基于微软提出的Drawbridge沙箱机制[128]提供了粗粒度地隔离执行应用程序的安全容器,并设计了如图7(a)所示的完整enclave接口.它将应用程序、标准库(standard library)以及Windows库操作系统都放入enclave,保护未修改的应用程序抵御外部特权软件攻击或者物理攻击.另外,Drawbridge沙箱机制也能够保证运行在enclave内的应用程序不对外界系统造成破坏.尽管Haven能够最大限度地减少外部接口,但是将库操作系统放入enclave会增加TCB大小.考虑到EPC内存大小有限时,执行输入/输出会增加性能开销,后续需要进一步地缩小TCB大小.

SCONE[102]基于Intel SGX和Docker容器[129]技术为多租户环境提供了一种安全容器机制,同样将全部应用程序放入enclave,实现了粗粒度地隔离.与Haven相比,SCONE也能保护未修改应用程序不受外部恶意特权软件的攻击,但是它在enclave内放置修改的C语言标准库缩小了TCB大小.另外,它采用用户级线程和异步系统调用,降低enclave内线程同步和系统调用带来的性能影响,满足Docker容器快速启动和高读写性能的要求.实验评估表明,相比不采取硬件增强保护,使用Intel SGX保护未修改的Linux应用程序增加了大约30%到40%的额外开销.但是,敏感和非敏感的代码都加载到enclave仍然会导致TCB过大.

Ryoan[103]利用Intel SGX和分布式沙箱技术保护数据所有者的数据免受不可信CSP或用户的窃取,即使分布式计算平台不可信,仍然能够提供沙箱实例保护机密数据的隐私性,保证分布式计算隐私.不同于Haven和SCONE的安全假设,Ryoan模型假设应用程序和CSP都不可信,它不依赖地址空间隔离,而是基于编译器技术限制不可信的代码.在具体实现时,每个enclave包含一个用户级的本地客户NaCl(native client)沙箱实例,加载和执行不受信任的模块,并通过控制输入/输出信道和隐蔽信道(如系统调用和数据大小)的安全来保护计算过程中的数据隐私.

6.3 小结

通过以上分析,基于硬件增强的隐私保护方法主要利用Intel SGX技术为大数据分析提供可信执行环境,保护计算过程中数据和代码的机密性和完整性.表6主要从核心方法、enclave接口类型、安全假设以及额外开销等角度对比分析了与Intel SGX应用相关的代表性研究工作,其中额外开销是指采用Intel SGX执行应用相比不采用Intel SGX的基准环境增加的额外开销.下面从两个角度总结如何扩展Intel SGX应用保证安全且实用:

表6 基于硬件增强的隐私保护代表性研究工作

代表性工作	核心方法	enclave接口	安全假设	额外开销
VC3[6]	受保护的map/reduce函数加载到enclave	预定义的enclave接口	CSP不可信	4.5%~8.0%
Haven[100]	容器中创建enclave保护未修改应用程序	完整的enclave接口	CSP不可信	31%~54%
SCONE[102]	采用用户级线程和异步系统调用	完整的enclave接口	CSP不可信	30%~40%
Ryoan[103]	利用enclave保护沙箱示例中的秘密数据	特定应用程序的enclave接口	应用程序与CSP不可信	1.4%~32.0%

(1)将受保护的大数据应用程序放入enclave中安全执行并非易事,在实际应用中仍面临着诸多问题,例如敏感代码的安全划分、TCB大小,以及网络层或者内存层的侧信道攻击等.特别地,对于复杂的大数据应用程序而言,划分敏感代码的开发量一般比较大,并且没有通用标准界定哪些代码应该被划分为敏感的代码.

(2)相比密文计算,在enclave中执行明文计算可以在一定程度上降低系统开销.但是目前Intel SGX支持的EPC内存大小受限,当受保护的应用程序规模比较大时,特别是主流的大数据计算框架都支持内存密集型计算,enclave需要频繁地换出/进页面以切换上下文,这也带来了较高的通信开销.因此实际应用中,需要进一步优化性能.

7 基于访问模式隐藏的隐私保护

尽管数据加密可以很好地隐藏数据的机密性,但是不能隐藏一些元数据,比如访问模式、数据来源和去向等.云平台内部攻击者可以利用这些元数据获得两种隐私信息,其一是根据访问模式推测出数据的相关属性,如果攻击者知道有关数据的背景知识,那么它可以推测出传输数据的明文信息;其二是根据数据来源和去向推测数据发送方和接收方的身

份，虽然已有一些元数据隐藏技术，但是它们无法抵抗能力更强的节点访问型攻击者[13].

Zheng等人[99]指出访问模式泄露攻击发生在内存层和网络层，当恶意操作系统通过监视应用程序的页面访问来推断有关加密数据的信息时，云平台会发生内存层的访问模式泄漏.而网络层的访问模式泄漏发生在分布式系统的任务调度和消息传输中，尽管通过网络发送的消息数据是加密的，但是某些分布式任务（例如排序、混洗或分区等）也会产生披露加密数据隐私的网络流量.严重地，攻击者可以通过分析计算过程中网络流量的特点实施流量分析攻击.

为了解决内存层和网络层的访问模式泄露，目前主要采用不经意随机访问机ORAM和不经意混洗技术，在云服务提供商不可信的安全假设下，实现不经意计算来隐藏访问模式.在实际应用中，基于访问模式隐藏的隐私保护方法一般不会单独使用，通常在数据加密或可信硬件支持等条件下采取该方法进一步地增强安全和保护隐私，本节重点总结在大数据计算环境下该方向的相关研究工作.

7.1 基于ORAM的不经意计算

现有工作指出ORAM协议具有一个对数级别的性能下界[104]，因此将其应用在数据频繁访问的大数据计算场景中，它依然面临着性能较低的瓶颈.目前，基于ORAM的不经意计算需要解决安全方案设计和性能优化两个方面的挑战问题，代表性工作有Oblivistore[105]、ObliVM[106]和GraphSC[107]等，但是它们都存在性能较低的瓶颈.

Xu等人[7]指出攻击者可以从一个拼写检查的应用程序中提取数百KB大小的机密数据，甚至可以从一个运行在enclave内部的图像处理应用程序中提取出可辨别的图像轮廓.为了解决网络层的访问模式泄露，Stefanov等人[105]采用ORAM协议隐藏读写访问模式.实验证实，相比不采用ORAM保护的计算任务，在云场景中利用ORAM保护网络访问模式时，会产生高达17倍的通信开销.

ObliVM系统[106]采用ORAM技术设计了通用的不经意计算平台，并提供了支持安全计算程序的编程框架.ObliVM旨在将程序编译成适用于安全计算要求的高效且遗忘式表示形式，并提供了一种功能强大且表现力较强的编程语言，以及用户友好型的隐蔽编程抽象.基于此，用户可以在云平台执行数据挖掘、流计算和图计算等应用.但是在实际应用中，不经意计算带来了较高的计算开销.例如相比基准的图计算任务，不经意计算带来了10^6倍的开销.GraphSC系统[107]基于ObliVM的研究工作降低了一个数量级的计算开销，在云服务提供商不可信的威胁模型下，利用混淆电路和安全两方计算实现了安全增强的不经意计算.虽然上述工作能够抵抗内存层的访问模式攻击，但是他们主要依赖密码学手段实现加密数据上的安全计算，并不支持代码的完整性验证.

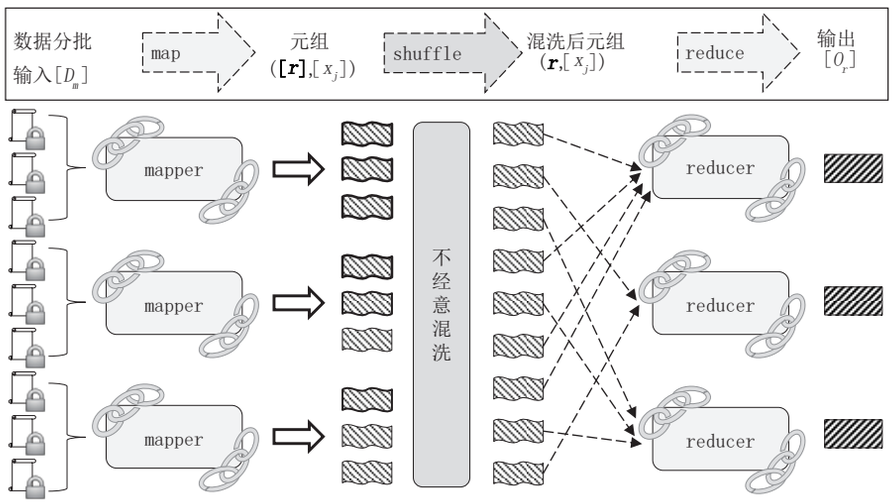
通过以上分析，从安全性的角度，ORAM技术能够保护读写操作的访问模式，并且具有可解释性.但是，在主流的大数据计算框架下，直接采用ORAM实现不经意计算将会带来非常高的性能开销，严重地影响大数据分析的高效要求.因此，未来急需设计一种优化的分布式ORAM解决方案.

7.2 基于不经意混洗的不经意计算

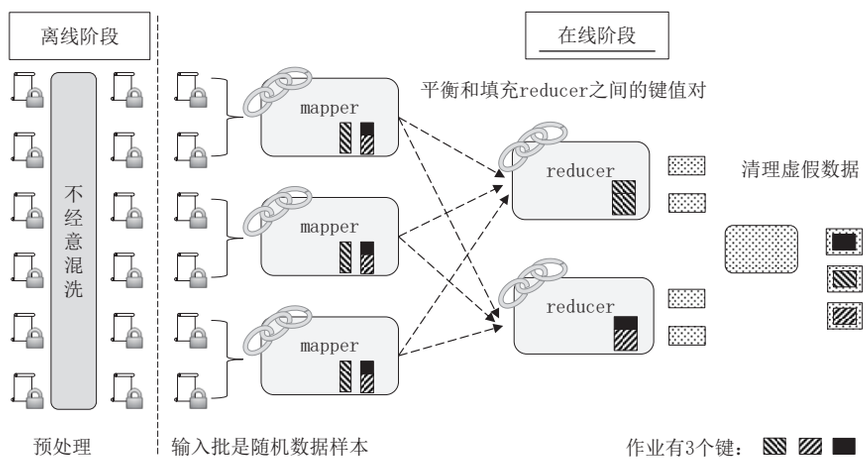
代替采用成本较高的ORAM技术，学术界提出了在大数据计算中采用不经意混洗技术来实现不经意的分布式计算.目前，实现不经意混洗的两种方式分别是不经意排序和级联（cascade）混合网络，它们试图产生具有足够可扩展性和效率的安全随机排列；其中，不经意排序主要以数据独立的方式选择不可预测的排列并对数据进行相应排序，其有效性取决于具体排序算法的复杂度；级联混合网络主要通过执行k个串级混合步骤实现任意随机排列，其有效性取决于执行混合步骤的轮数.

已有研究基于以上方式在Intel SGX可信硬件支持下对访问的数据执行不经意混洗，既能够隐藏访问模式实现不经意计算，也能够保证数据和关键代码的隐私性，下面展开介绍相关的研究工作.

Ohrimenko等人[8]利用一种可并行化的墨尔本混洗（melbourne shuffle）算法实现了不经意混洗，并应用到Intel SGX支持的MapReduce计算框架，实现了隐私数据分析的不经意计算.他们发现即使加密shuffle阶段的通信元组，云内部攻击者仍然可以观察并统计出shuffle阶段的流量矩阵，并利用背景知识分析出多个流量矩阵之间的对应关系，从而推测出原始输入数据集中的隐私信息.为了避免这种网络层的访问模式泄露，他们基于VC3系统实现了如图8所示的中间混洗（shuffle-in-the-middle）和混洗且平衡两种解决方案，其中不经意混洗切断了mapper与reducer任务之间数据传输的对应关系，使得攻击者无法通过统计流量获得准确的流量矩阵.另外，通过平衡和填充不同reducer任务之间的键值对，使得外部攻击者无法通过计算输出分析出数据隐私.在性能方面，虽然该工作中采用的墨尔本排序



(a) 中间混洗方案



(b) 混洗且平衡方案

图8 基于SGX硬件环境实现不经意计算示例

算法不需要完全排序而降低了不经意计算开销，但是它忽略了实际数据和计算的存储空间，带来了较差的可扩展性.

Opaque系统[99]利用基于Intel SGX的列排序(columnsort)算法实现了不经意混洗，并应用到Spark计算框架，将分布式的Spark SQL查询任务转换为隐藏访问模式的不经意关系运算符，可以同时防止内存层和网络层的访问模式泄漏. 与普通的Spark SQL相比，Opaque提供了数据加密、身份验证和计算验证等特性. 另外，Opaque提出了基于规则优化和成本优化的查询，进一步地提高不经意计算的性能. 相比采用ORAM实现的不经意计算系统GraphSC，Opaque运行网页排序作业的性能提升了2300倍. 尽管如此，相比不采取任何隐私保护的大数据计算，当enclave EPC内存大小受限时，它带来了高达46倍的计算开销. 因此，计算开销仍需要进一步地优化.

M2R系统[5]采用级联混合网络实现了不经意混洗，支持MapReduce计算框架的不经意计算，能够防止攻击者在不可信CSP的安全假设下实施网络层的访问模式泄露. 此外，该系统保证了最小的TCB，放入TCB的代码行数少于500行，不到Hadoop代码库的0.16%. 如表7所示测试及对比了不同隐私保护技术下7种MapReduce作业的执行时间，明文计算(第2列)是指不采用任何隐私保护的基准环境，同态加密(第3列)是指基于部分同态加密执行密文计算，Intel SGX(第4列)是指在数据加密传输的条件下提供硬件增强保护，不经意计算(第5列)是指在可信硬件支持下采用不经意混洗技术实现不经意计算以隐藏访问模式. 分析单词统计作业可知，相比基于同态加密的隐私保护方法，采用基于Intel SGX的隐私保护方法降低了69%的运行时间. 相比

表7　不同隐私保护技术下作业的运行时间(*s*)对比

测试作业	明文计算(基准)	同态加密(vs. 基准)	Intel SGX(vs. 基准)	不经意计算(vs. Intel SGX)
单词统计	221	1859 (8.41×)	570 (2.58×)	1156 (2.03×)
索引	423	2061 (4.87×)	666 (1.57×)	1549 (2.33×)
查询文字	48	1686 (35.12×)	70 (1.46×)	106 (1.51×)
聚合	80	9140 (114.25×)	125 (1.56×)	205 (1.64×)
连接	211	5716 (27.09×)	422 (2.00×)	510 (1.21×)
网页排序	334	1209 (3.62×)	521 (1.56×)	755 (1.45×)
k均值	71	6071 (85.51×)	123 (1.73×)	145 (1.18×)

基准环境下的明文计算，M2R增加了高达1.04倍到4.23倍的额外开销.

7.3　小结

通过以上分析，基于访问模式隐藏的隐私保护方法主要依靠ORAM或者不经意混洗技术实现不经意计算，防止攻击者观察到网络层或者内存层的访问模式. 如表8所示，主要从敌手模型、隐藏的访问模式、采用的不经意技术、是否Intel SGX支持、是否考虑数据加密和完整性验证、支持的计算框架以及计算开销等角度对比分析了大数据计算环境下该方向的相关研究工作. 两种实现不经意计算的技术对比分析如下：

(1)ORAM通过将每一次读或者写访问都随机地转换为一次读取和一次写回操作，使得攻击者无法区别数据的具体访问方式，从而隐藏了处理器访问内存的操作序列和地址序列访问模式. 然而，由此带来的额外操作在实际应用中往往带来较高的计算开销，严重制约了其实际应用. 因此，现有大多数研究工作更倾向于采用不经意混洗技术在大数据计算框架下设计专门的访问模式隐藏方案，以代替在方案中采用低效的通用ORAM协议.

(2)不经意混洗通过对数据的重排列使得攻击者无法关联输入数据和输出数据的分布，从而阻止攻击者判断计算过程中的访问模式. 在基于Intel SGX硬件支持的条件下，采用不经意排序或者级联混合网络手段实现不经意混洗方法要比ORAM高效的多. 现有很多研究工作尝试降低不经意排序算法的复杂度来提高处理效率，比如$O(n\log n)$复杂度的排序网络[130]和随机希尔排序[131]等，但是未来还需要进一步地结合大数据计算系统来实现.

表8　基于访问模式隐藏的隐私保护相关研究工作

代表性工作	敌手模型	隐藏的访问模式	不经意技术	是否有Intel SGX支持	是否考虑数据加密	是否考虑完整性验证	支持的计算框架	计算开销
ObliVM[106]	半诚实敌手	内存层	ORAM	否	是	否	未考虑	10^6×
GraphSC[107]	半诚实且非合谋敌手	内存层	ORAM	否	是	否	未考虑	10^5×
Observing[8]	半诚实或者恶意敌手	网络层	墨尔本混洗	是	是	是	MapReduce	1.07～4.55×
Opaque[99]	半诚实或者恶意敌手	内存层和网络层	不经意混洗 列排序	是	是	是	MapReduce	1.6～46×
M2R[5]	半诚实或者恶意敌手	网络层	级联混合网络	是	是	是	Spark SQL	2.04～5.23×

8　总结和展望

本文对大数据计算环境下的隐私保护技术研究进展进行了综述. 首先分析了大数据计算环境下的敌手模型、隐私问题与挑战，以及隐私保护的研究方向；接着，根据隐私保护技术的不同，分别总结分析了基于数据分离的隐私保护、基于数据干扰的隐私保护、基于安全多方计算的隐私保护、基于硬件增强的隐私保护、以及基于访问模式隐藏的隐私保护等研究方向的最新研究进展，并对比分析了不同隐私保护技术的优缺点；最后，展望了大数据计算环境下隐私保护技术的未来研究方向. 期望本文的工作，能给以后的研究者提供有益的参考与借鉴，为大数据隐私保护的进一步发展做出贡献.

综合分析可知，不同隐私保护技术具有不同的

技术特点、局限性和适用场景.在大数据计算环境下应用隐私保护技术时,数据分离和匿名技术侧重于在数据输入阶段保护原始数据的敏感信息,其中数据分离技术主要存在通信开销较高的局限性,适用于本地或私有云环境具有较强算力的隐私保护场景,匿名技术实现简单但是主要面临着更强背景知识攻击的困扰;差分隐私技术侧重于在数据输入和结果输出阶段扰动数据,在实际应用中计算效率较高,但是主要存在可用性不高的局限性,适用于计算节点算力较弱且对隐私保护水平有一定要求的场景;安全多方计算、Intel SGX和不经意计算等技术侧重于在数据计算过程中保护数据的隐私性和计算的安全性,在实际应用中,安全多方计算主要存在通信开销较高和执行效率较低的局限性,适用于多方分布式联合计算的隐私保护场景;Intel SGX技术需要可信硬件辅助以在安全隔离环境下执行明文计算,在应用中主要面临侧信道攻击的安全威胁;不经意计算主要依赖ORAM或不经意混洗手段隐藏访问模式,但是这些手段本身存在低效性和特殊性,特别是ORAM在实际应用中带来了较高的计算复杂度.因此,未来在大数据计算环境下应用这些隐私保护技术,仍然存在很多问题需要亟待解决,其中以下五个问题值得进一步地研究.

(1)研究低带宽网络环境下的高效数据分离保护:目前,数据分离技术主要存在通信开销较高的局限,不仅表现在混合云中跨云聚合时的通信数据量和通信总时耗(见第3.1.2节),也体现在联邦学习中达到预定模型精度时,本地客户端与云端服务器之间的通信数据量和通信轮次(见第3.2节).为了适用低带宽网络环境,特别是随着越来越多的通信带宽和电力有限的终端设备接入,降低数据分离方法中的通信开销提高通信效率显得十分重要.因此,需要研究低带宽网络环境下的高效数据分离保护,例如通过对键的独立划分降低公有与私有云之间传输的元组数量,压缩模型或者选择部分客户端参与更新降低联邦学习中传输的模型参数量,以及降低模型精度来减少通信轮次等方式.

(2)研究针对复杂数据类型的高可用差分隐私保护:目前,大部分研究工作重点关注简单数据类型的差分隐私保护,例如针对离散分类数据的本地化差分隐私地频率估计以及针对连续数值数据的本地化差分隐私地均值估计.而对于大数据计算中的半结构化或者非结构化数据研究较少,例如键值型数据或者图数据等复杂数据类型.在实际应用中,参与大数据计算的主要是这些复杂数据类型.另外,现有方案对键值数据进行扰动时忽略了键与值之间的对应关系[62],一方面将造成隐私泄露降低隐私性,另一方面将导致过多的噪音被添加,影响了可用性.因此,在实际应用场景下,权衡复杂数据类型扰动的隐私性和可用性,设计出高可用的差分隐私保护是未来这一类研究方向的重点.

(3)研究实用型的安全多方计算协议,进一步提升性能:虽然现有的安全计算协议能够保护计算过程中数据隐私,但是当真正应用到海量数据的安全计算时,它面临着较高的通信开销和计算开销瓶颈(见第5.4节).为了支持多用户并发访问且快速响应的大数据系统,设计高实用的安全多方计算协议,并兼容目前主流的计算框架,例如MapReduce、Spark和TensorFlow等,是促进安全多方计算应用于实际的关键.因此,性能优化问题一直是这一类研究亟待解决的问题.

(4)研究安全增强的Intel SGX应用,进一步提升性能:虽然Intel SGX技术能够有效地解决大数据计算环境下云平台上应用程序和敏感数据的安全计算问题,但是它在实际应用中也面临着诸多安全问题和性能瓶颈[16].正如第6.3小结提到的如何有效地解决敏感代码安全划分与验证,抵抗侧信道攻击以及内存攻击等安全问题,以及兼容容器与虚拟化技术减轻EPC内存大小受限引起的性能瓶颈.尤其在隐私计算方面,支持数据密集型计算的多任务并行处理以及大数据的安全审计等应用.因此,安全问题与性能优化是未来Intel SGX应用需要亟待解决的问题.

(5)研究高效的通用访问模式隐藏结构:一方面,基于ORAM实现的通用访问模式隐藏结构在实际应用中面临着较高的性能瓶颈,无法与大数据计算框架相结合同时满足高效计算和隐私保护的需求[17,132],因此未来有必要从ORAM协议设计本身入手,进一步地降低计算复杂度提升性能;另一方面,虽然针对特定计算专门设计的不经意混洗方法能够有利于与实用的大数据计算框架相结合,但是当把它们集成到对实时性要求更高的分布式流式计算框架时,如Storm、Flink和Spark Streaming等,它们具有不同的计算原语和执行模型,对现有工作提出了新的设计和性能挑战,因此未来需要研究一种高效的通用访问模式隐藏结构.

除了以上针对五个研究方向存在的问题值得进一步研究之外,随着大数据、物联网和人工智能等产

业的发展，各种隐私问题以及隐私保护技术越来越受到重视.未来的隐私保护研究工作应该重点关注以下几个新方向：

(1)适用于大数据计算各个环节的通用隐私保护方案

在基于云平台的大数据计算环境下，主要考虑数据输入、计算和输出等三个环节可能存在的隐私泄露风险，采取相对应的隐私保护技术保证数据隐私.但是正如前面提到的，每种隐私保护技术具有不同的优势和局限性.目前，越来越多的研究工作结合多种隐私保护技术解决多个环节的隐私泄露问题(见第5.2节).例如，结合安全多方计算和区块链技术构建去中心化场景下多参与方之间的信任关系.尽管如此，区块链共识机制的安全性和效率也需要满足实际应用的需要.因此，在大数据计算环境下，如何充分结合各隐私保护技术的优势，解决大数据计算各个环节的隐私问题，是设计通用隐私保护方案的关键点.

(2)针对端边云计算架构的可行隐私保护

随着物联网技术的发展，各种终端设备接入以及边缘与中心云之间的协作，形成了端边云的三级计算架构.边缘计算节点往往计算能力有限，适合于采用计算效率较高的匿名或本地化差分隐私技术保护数据隐私，但是干扰真实数据影响了可用性(见第4.2.1节).而云平台的计算资源比较充足，适合采用隐私性和可用性较高的安全多方计算技术保护数据隐私，但是其通信开销较高影响了执行效率(见第5.4节).因此，针对端边云计算架构的实用场景，未来需要权衡隐私性、可用性和效率等因素进一步地研究可行的隐私保护方法.

(3)面向多数据源协同训练的隐私保护框架

随着人工智能技术的发展，多数据源期望共享数据以学习更有价值的模型，即协同训练.另一方面，随着一系列信息保护法案的出台，个人隐私保护越来越受重视，数据持有者往往不愿意直接共享训练数据.已有研究工作[133-134]表明虽然能够依赖加密、匿名或者本地化差分隐私等手段保护实施集中式学习的训练数据(见第3.2节)，但是不能应对复杂的大数据协同训练环境.另外，联邦学习能够很好地解决协同训练与个体隐私之间的权衡问题，但是在一定程度上也限制了模型训练准确性.并且现有的联邦学习框架本身安全性不可解释，仍然存在着较多安全问题[31-32].因此面向多数据源协同训练的场景，仍需要进一步地完善隐私保护框架.

致 谢 在此，我们向对本文的工作给予支持和宝贵建议的评审老师和同行表示衷心的感谢！

参 考 文 献

[1] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2008, 51(1): 107-113

[2] Zaharia M, Das T, Li H, et al. Discretized streams: Fault-tolerant streaming computation at scale//Proceedings of the 24th ACM Symposium on Operating Systems Principles. Farmington, USA, 2013: 423-438

[3] Carbone P, Katsifodimos A, Ewen S, et al. Apache flink: Stream and batch processing in a single engine. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2015, 36(4): 28-38

[4] Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016

[5] Dinh T T A, Saxena P, Chang E C, et al. M2R: Enabling stronger privacy in mapreduce computation//Proceedings of the 24th USENIX Security Symposium. Washington, USA, 2015: 447-462

[6] Schuster F, Costa M, Fournet C, et al. VC3: Trustworthy data analytics in the cloud using SGX//Proceedings of the 36th IEEE Symposium on Security and Privacy. San Jose, USA, 2015: 38-54

[7] Xu Y, Cui W, Peinado M. Controlled-channel attacks: Deterministic side channels for untrusted operating systems//Proceedings of the 36th IEEE Symposium on Security and Privacy. San Jose, California, USA, 2015: 640-656

[8] Ohrimenko O, Costa M, Fournet C, et al. Observing and preventing leakage in mapreduce//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. Denver, USA, 2015: 1570-1581

[9] Zhou Shui-Geng, Li Feng, Tao Yu-Fei, Xiao Xiao-Kui. Privacy preservation in database applications: A survey. Chinese Journal of Computers, 2009, 32(5): 847-861 (in Chinese)
(周水庚，李丰，陶宇飞，肖小奎.面向数据库应用的隐私保护研究综述.计算机学报，2009，32(5)：847-861)

[10] Zhang Xiao-Jian, Meng Xiao-Feng. Differential privacy in data publication and analysis. Chinese Journal of Computers, 2014, 37(4): 927-949 (in Chinese)
(张啸剑，孟小峰.面向数据发布和分析的差分隐私保护.计算机学报，2014，37(4)：927-949)

[11] Binjubeir M, Ahmed A A, Ismail M A B, et al. Comprehensive survey on big data privacy protection. IEEE Access, 2020, 8: 20067-20079

[12] Feng Deng-Guo, Zhang Min, Li Hao. Big data security and privacy protection. Chinese Journal of Computers, 2014,

37(1): 246-258 (in Chinese)
(冯登国, 张敏, 李昊. 大数据安全与隐私保护. 计算机学报, 2014, 37(1): 246-258)

[13] Derbeko P, Dolev S, Gudes E, et al. Security and privacy aspects in MapReduce on clouds: A survey. Computer Science Review, 2016, 20: 1-28

[14] Tan Zuo-Wen, Zhang Lian-Fu. Survey on privacy preserving techniques for machine learning. Journal of Software, 2020, 31(7): 2127-2156 (in Chinese)
(谭作文, 张连福. 机器学习隐私保护研究综述. 软件学报, 2020, 31(7): 2127-2156)

[15] Liu Rui-Xuan, Chen Hong, Guo Ruo-Yang, et al. Survey on privacy attacks and defenses in machine learning. Journal of Software, 2019, 30(3): 866-892 (in Chinese)
(刘睿瑄, 陈红, 郭若杨, 赵丹, 梁文娟, 李翠平. 机器学习中的隐私攻击与防御. 软件学报, 2019, 30(3): 866-892)

[16] Dong Chun-Tao, Shen Qing-Ni, Luo Wu, Wu Peng-Fei, Wu Zhong-Hai. Research progress of SGX application supporting techniques. Journal of Software, 2019, 30(1): 137-166 (in Chinese)
(董春涛, 沈晴霓, 罗武, 吴鹏飞, 吴中海. SGX 应用支持技术研究进展. 软件学报, 2019, 30(1): 137-166)

[17] Wu Peng-Fei, Shen Qing-Ni, Qin Jia, Qian Wen-Jun, Li Cong, WU Zhong-Hai. Survey of Oblivious RAM. Journal of Software, 2018, 29(9): 2753-2777 (in Chinese)
(吴鹏飞, 沈晴霓, 秦嘉, 钱文君, 李聪, 吴中海. 不经意随机访问机研究综述. 软件学报, 2018, 29(9): 2753-2777)

[18] Differential Privacy Team, Apple. Learning with privacy at scale. Machine Learning Journal, 2017, 1(8): 1-25

[19] Erlingsson Ú, Pihur V, Korolova A. Rappor: Randomized aggregatable privacy-preserving ordinal response//Proceedings of the 21st ACM SIGSAC Conference on Computer and Communications Security. Scottsdale, USA, 2014: 1054-1067

[20] Ding B, Kulkarni J, Yekhanin S. Collecting telemetry data privately//Proceedings of the 31st Annual Conference on Neural Information Processing Systems. California, USA, 2017: 3574-3583

[21] McSherry F. Privacy integrated queries: An extensible platform for privacy-preserving data analysis//Proceedings of the 35th ACM SIGMOD International Conference on Management of Data. Providence, USA, 2009: 19-30

[22] Roy I, Setty S T V, Kilzer A, et al. Airavat: Security and privacy for MapReduce//Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation. San Jose, USA, 2010: 297-312

[23] Pettai M, Laud P. Combining differential privacy and secure multiparty computation//Proceedings of the 31st Annual Computer Security Applications Conference. Los Angeles, USA, 2015: 421-430

[24] Ko S Y, Jeon K, Morales R. The HybrEx model for confidentiality and privacy in cloud computing//Proceedings of the 3rd USENIX Workshop on Hot Topics in Cloud Computing. Portland, OR, USA, 2011: 1-5

[25] Zhang K, Zhou X, Chen Y, et al. Sedic: Privacy-aware data intensive computing on hybrid clouds//Proceedings of the 18th ACM Conference on Computer and Communications Security. Chicago, USA, 2011: 515-526

[26] Zhang C, Chang E C, Yap R H C. Tagged-MapReduce: A general framework for secure computing with mixed-sensitivity data on hybrid clouds//Proceedings of the 14th International Symposium on Cluster, Cloud and Grid Computing. Chicago, USA, 2014: 31-40

[27] Oktay K Y, Mehrotra S, Khadilkar V, et al. Semrod: Secure and efficient mapreduce over hybrid clouds//Proceedings of the 41st ACM SIGMOD International Conference on Management of Data. Melbourne, Australia, 2015: 153-166

[28] Kawamoto S, Kamidoi Y, Wakabayashi S. A framework for fast mapreduce processing considering sensitive data on hybrid clouds//Proceedings of the 44th IEEE Annual Computers, Software, and Applications Conference. Madrid, Spain, 2020: 1357-1362

[29] Oktay K Y, Kantarcioglu M, Mehrotra S. Secure and efficient query processing over hybrid clouds//Proceedings of the 33rd IEEE International Conference on Data Engineering. San Diego, USA, 2017: 733-744

[30] Zhou Z, Zhang H, Du X, et al. Prometheus: Privacy-aware data retrieval on hybrid cloud//Proceedings of the 32nd IEEE International Conference on Computer Communications. Turin, Italy, 2013: 2643-2651

[31] Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models//Proceedings of the 38th IEEE Symposium on Security and Privacy. San Jose, USA, 2017: 3-18

[32] Nasr M, Shokri R, Houmansadr A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning//Proceedings of the 40th IEEE Symposium on Security and Privacy. San Francisco, USA, 2019: 739-753

[33] Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning//Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 1175-1191

[34] McMahan H B, Ramage D, Talwar K, et al. Learning differentially private recurrent language models. arXiv preprint arXiv: 1710.06963, 2017

[35] Sheller M J, Reina G A, Edwards B, et al. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation//Proceedings of the 4th International MICCAI Brainlesion Workshop. Granada, Spain, 2018: 92-104

[36] Stich S U. Local SGD converges fast and communicates little. arXiv preprint arXiv: 1805.09767, 2018

[37] Wang H, Sievert S, Liu S, et al. Atomo: Communication-efficient learning via atomic sparsification//Proceedings of the 32nd Advances in Neural Information Processing Systems.

Montréal, Canada, 2018: 9850-9861

[38] Recht B, Re C, Wright S, et al. Hogwild: A lock-free approach to parallelizing stochastic gradient descent//Proceedings of the 25th Advances in Neural Information Processing Systems. Granada, Spain, 2011: 693-701

[39] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge//Proceedings of the 53rd IEEE International Conference on Communications. Shanghai, China, 2019: 1-7

[40] Bonawitz K, Eichner H, Grieskamp W, et al. Towards federated learning at scale: System design. arXiv preprint arXiv: 1902.01046, 2019

[41] Smith V, Chiang C K, Sanjabi M, et al. Federated multi-task learning//Proceedings of the 31st Advances in Neural Information Processing Systems. Long Beach, USA, 2017: 4424-4434

[42] Wang K, Yu P S, Chakraborty S. Bottom-up generalization: A data mining solution to privacy protection//Proceedings of the 4th IEEE International Conference on Data Mining. Brighton, UK, 2004: 249-256

[43] Fung B C M, Wang K, Yu P S. Top-down specialization for information and privacy preservation//Proceedings of the 21st International Conference on Data Engineering. Tokyo, Japan, 2005: 205-216

[44] Li F, Sun J, Papadimitriou S, et al. Hiding in the crowd: Privacy preservation on evolving streams through correlation tracking//Proceedings of the 23rd International Conference on Data Engineering. Istanbul, Turkey, 2007: 686-695

[45] Zhou B, Han Y, Pei J, et al. Continuous privacy preserving publishing of data streams//Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. Saint Petersburg, Russia, 2009: 648-659

[46] Kim S, Sung M K, Chung Y D. A framework to preserve the privacy of electronic health data streams. Journal of Biomedical Informatics, 2014, 50: 95-106

[47] Cao J, Carminati B, Ferrari E, et al. Castle: Continuously anonymizing data streams. IEEE Transactions on Dependable and Secure Computing, 2010, 8(3): 337-352

[48] Guo K, Zhang Q. Fast clustering-based anonymization approaches with time constraints for data streams. Knowledge-Based Systems, 2013, 46: 95-108

[49] Irudayasamy A, Arockiam L. Parallel bottom-up generalization approach for data anonymization using map reduce for security of data in public cloud. Indian Journal of Science and Technology, 2015, 8(22): 1-9

[50] Pandilakshmi K R, Banu G R. An advanced bottom up generalization approach for big data on cloud. International Journal of Computing Algorithm, 2014, 3: 1054-1059

[51] Balusamy M, Muthusundari S. Data anonymization through generalization using map reduce on cloud//Proceedings of IEEE International Conference on Computer Communication and Systems. Chennai, India, 2014: 039-042

[52] Zhang X, Yang L T, Liu C, et al. A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud. IEEE Transactions on Parallel and Distributed Systems, 2013, 25(2): 363-373

[53] Zhang X, Liu C, Nepal S, et al. A hybrid approach for scalable sub-tree anonymization over big data using MapReduce on cloud. Journal of Computer and System Sciences, 2014, 80(5): 1008-1020

[54] Sopaoglu U, Abul O. A top-down k-anonymization implementation for apache spark//Proceedings of the 2017 IEEE International Conference on Big Data. Boston, USA, 2017: 4513-4521

[55] Ashkouti F, Sheikhahmadi A. DI-Mondrian: Distributed improved mondrian for satisfaction of the *l*-diversity privacy model using apache spark. Information Sciences, 2021, 546: 1-24

[56] Bazai S U, Jang-Jaccard J, ScalableAlavizadeh H., high-performance, and generalized subtree data anonymization approach for apache spark. Electronics, 2021, 10(5): 589

[57] Zhang X, Dou W, Pei J, et al. Proximity-aware local-recoding anonymization with mapreduce for scalable big data privacy preservation in cloud. IEEE Transactions on Computers, 2014, 64(8): 2293-2307

[58] Mehta B B, Rao U P. Privacy preserving big data publishing: A scalable k-anonymization approach using mapreduce. Institution of Engineering and Technology Software, 2017, 11(5): 271-276

[59] Warner S L. Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association, 1965, 60(309): 63-69

[60] Wang T, Blocki J, Li N, et al. Locally differentially private protocols for frequency estimation//Proceedings of the 26th USENIX Security Symposium. Vancouver, BC, Canada, 2017: 729-745

[61] Wang T, Li N, Jha S. Locally differentially private heavy hitter identification. IEEE Transactions on Dependable and Secure Computing, 2021, 18(2): 982-993

[62] Ye Q, Hu H, Meng X, et al. PrivKV: Key-value data collection with local differential privacy//Proceedings of the 40th IEEE Symposium on Security and Privacy. San Francisco, USA, 2019: 317-331

[63] Gu X, Li M, Cheng Y, et al. PCKV: Locally differentially private correlated key-value data collection with optimized utility//Proceedings of the 29th USENIX Security Symposium. Boston, USA, 2020: 967-984

[64] Yilmaz E, Al-Rubaie M, Chang J M. Locally differentially private naive bayes classification. arXiv preprint arXiv: 1905.01039, 2019

[65] Wang S, Huang L, Nie Y, et al. Local differential private data aggregation for discrete distribution estimation. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(9): 2046-2059

[66] Wei K, Li J, Ding M, et al. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security, 2020, 15: 3454-3469

[67] Haeberlen A, Pierce B C, Narayan A. Differential privacy under

fire//Proceedings of the 20th USENIX Security Symposium. San Francisco, USA, 2011: 507-521

[68] Ding Z, Wang Y, Wang G, et al. Detecting violations of differential privacy//Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security. Toronto, Canada, 2018: 475-489

[69] Bichsel B, Gehr T, Drachsler-Cohen D, et al. DP-Finder: Finding differential privacy violations by sampling and optimization//Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security. Toronto, Canada, 2018: 508-524

[70] Lécuyer M, Spahn R, Vodrahalli K, et al. Privacy accounting and quality control in the sage differentially private ML platform//Proceedings of the 27th ACM Symposium on Operating Systems Principles. Huntsville, Canada, 2019: 181-195

[71] Chen Y, Machanavajjhala A, Hay M, et al. PeGaSus: Data-adaptive differentially private stream processing//Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 1375-1388

[72] Avent B, Korolova A, Zeber D, et al. BLENDER: Enabling local search with a hybrid differential privacy model//Proceedings of the 26th USENIX Security Symposium. Vancouver, Canada, 2017: 747-764

[73] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy//Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 308-318

[74] Geyer R C, Klein T, Nabi M. Differentially private federated learning: A client level perspective. arXiv preprint arXiv: 1712.07557, 2017

[75] Yao C C. How to generate and exchange secrets//Proceedings of the 27th Annual Symposium on Foundations of Computer Science. Toronto, Canada, 1986: 162-167

[76] Goyal V, Mohassel P, Smith A. Efficient two party and multi party computation against covert adversaries//Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Istanbul, Turkey, 2008: 289-306

[77] Malkhi D, Nisan N, Pinkas B, et al. Fairplay--A secure two-party computation system//Proceedings of the 13th USENIX Security Symposium. San Diego, USA, 2004: 287-302

[78] Ben-David A, Nisan N, Pinkas B. FairplayMP: A system for secure multi-party computation//Proceedings of the 15th ACM Conference on Computer and Communications Security. Taipei, China, 2008: 257-26

[79] Huang Y, Evans D, Katz J, et al. Faster secure two-party computation using garbled circuits//Proceedings of the 20th USENIX Security Symposium. San Francisco, USA, 2011: 35-35

[80] Kreuter B, Shelat A, Shen C H. Billion-gate secure computation with malicious adversaries//Proceedings of the 21st USENIX Security Symposium. Bellevue, USA, 2012: 285-300

[81] Pinkas B, Schneider T, Smart N P, et al. Secure two-party computation is practical//Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security. Tokyo, Japan, 2009: 250-267

[82] Bater J, Elliott G, Eggen C, et al. SMCQL: Secure querying for federated databases. arXiv preprint arXiv: 1606.06808, 2016

[83] Volgushev N, Schwarzkopf M, Getchell B, et al. Conclave: Secure multi-party computation on big data//Proceedings of the 14th EuroSys Conference. Dresden, Germany, 2019: 1-18.

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

[85] Bogdanov D, Laur S, Willemson J. Sharemind: A framework for fast privacy-preserving computations//Proceedings of the 13th European Symposium on Research in Computer Security. Málaga, Spain, 2008: 192-206

[86] Dolev S, Li Y, Sharma S. Private and secure secret shared MapReduce//Proceedings of the 30th IFIP Annual Conference on Data and Applications Security and Privacy. Trento, Italy, 2016: 151-160

[87] Dahl M, Mancuso J, Dupis Y, et al. Private machine learning in tensorflow using secure computation. arXiv preprint arXiv: 1810.08130, 2018

[88] Kumar N, Rathee M, Chandran N, et al. Cryptflow: Secure tensorflow inference//Proceedings of the 41st IEEE Symposium on Security and Privacy. San Francisco, USA, 2020: 336-353

[89] Rivest R L, Adleman L M, Dertouzos M L. On data banks and privacy homomorphisms. Foundations of Secure Computation, 1978, 4(11): 169-179

[90] Asharov G, Jain A, López-Alt A, et al. Multiparty computation with low communication, computation and interaction via threshold FHE// Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cambridge, UK, 2012: 483-501

[91] López-Alt A, Tromer E, Vaikuntanathan V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption//Proceedings of the 44th Annual ACM Symposium on Theory of Computing. New York, USA, 2012: 1219-1234

[92] Martin K, Wang W, Agyemang B. Optimized homomorphic scheme on map reduce for data privacy preserving. Journal of Information Security, 2017, 8(3): 257-273

[93] Peter A, Tews E, Katzenbeisser S. Efficiently outsourcing multiparty computation under multiple keys. IEEE Transactions on Information Forensics and Security, 2013, 8(12): 2046-2058

[94] Popa R A, Redfield C, Zeldovich N, et al. CryptDB: Processing queries on an encrypted database. Communications of the ACM, 2012, 55(9): 103-111

[95] Tetali S D, Lesani M, Majumdar R, et al. MrCrypt: Static analysis for secure cloud computations//Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications.

Indianapolis，Indiana，USA，2013：271-286

[96] Stephen J J，Savvides S，Seidel R，et al. Practical confidentiality preserving big data analysis//Proceedings of the 6th USENIX Workshop on Hot Topics in Cloud Computing. Philadelphia，USA，2014：1-7

[97] Dong Y，Milanova A，Dolby J. SecureMR：Secure MapReduce computation using homomorphic encryption and program partitioning//Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security. Raleigh，North Carolina，USA，2018：4-16

[98] Intel Corp. Software guard extensions programming reference. No. 329298-001，2013

[99] Zheng W，Dave A，Beekman J G，et al. Opaque：An oblivious and encrypted distributed analytics platform//Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation. Boston，USA，2017：283-298

[100] Baumann A，Peinado M，Hunt G. Shielding applications from an untrusted cloud with haven//Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation. Broomfield，USA，2014：267-283

[101] Lind J，Priebe C，Muthukumaran D，et al. Glamdring：Automatic application partitioning for intel sgx//Proceedings of the USENIX Annual Technical Conference. Santa Clara，USA，2017：285-298

[102] 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. Savannah，USA，2016：689-703

[103] Hunt T，Zhu Z，Xu Y，et al. Ryoan：A distributed sandbox for untrusted computation on secret data//Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. Savannah，USA，2016：533-549

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

[105] Stefanov E，Shi E. Oblivistore：High performance oblivious cloud storage//Proceedins of the 34th IEEE Symposium on Security and Privacy. Berkeley，USA，2013：253-267

[106] Liu C，Wang X S，Nayak K，et al. ObliVM：A programming framework for secure computation//Proceedings of the 36th IEEE Symposium on Security and Privacy. San Jose，USA，2015：359-376

[107] Nayak K，Wang X S，Ioannidis S，et al. GraphSC：Parallel secure computation made easy//Proceedings of the 36th IEEE Symposium on Security and Privacy. San Jose，USA，2015：377-394

[108] Efstathopoulos P，Krohn M，Vandebogart S，et al. Labels and event processes in the asbestos operating system. ACM SIGOPS Operating Systems Review，2005，39(5)：17-30

[109] Myers A C，Liskov B. A decentralized model for information flow control. ACM SIGOPS Operating Systems Review，1997，31(5)：129-142

[110] Lopresti D，Spitz A L. Quantifying information leakage in document redaction//Proceedings of the 1st ACM Workshop on Hardcopy Document Processing. Washington，USA，2004：63-69

[111] Van Berkel C H. Multi-core for mobile phones//Proceedings of the IEEE Design，Automation and Test in Europe Conference and Exhibition. Nice，France，2009：1260-1265

[112] Khodak M，Balcan M F F，Talwalkar A S. Adaptive gradient-based meta-learning methods//Proceedings of the 33rd Advances in Neural Information Processing Systems. Vancouver，Canada，2019：5917-5928

[113] Huang L，Yin Y，Fu Z，et al. Loadaboost：Loss-based adaboost federated machine learning on medical data. arXiv preprint arXiv：1811.12629，2018

[114] Eichner H，Koren T，McMahan H B，et al. Semi-cyclic stochastic gradient descent. arXiv preprint arXiv：1904.10120，2019

[115] Sweeney L. K-anonymity：A model for protecting privacy. International Journal of Uncertainty，Fuzziness and Knowledge-Based Systems，2002，10(5)：557-570

[116] Machanavajjhala A，Gehrke J，Kifer D，et al. L-diversity：Privacy beyond k-anonymity//Proceedings of the 22nd International Conference on Data Engineering. Atlanta，USA，2006：24-35

[117] Li N，Li T，Venkatasubramanian S. T-closeness：Privacy beyond k-anonymity and l-diversity//Proceedings of the 23rd International Conference on Data Engineering. Istanbul，Turkey，2007：106-115

[118] Samarati P，Sweeney L. Generalizing data to provide anonymity when disclosing information//Proceedings of the 17th ACM Sigact-Sigmod-Sigart Symposium on Principles of Database Systems. Seattle，USA，1998：188

[119] Sweeney L. Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty，Fuzziness and Knowledge-Based Systems，2002，10(5)：571-588

[120] LeFevre K，DeWitt D J，Ramakrishnan R. Workload-aware anonymization techniques for large-scale datasets. ACM Transactions on Database Systems，2008，33(3)：1-47

[121] Iwuchukwu T，DeWitt D J，Doan A H，et al. K-anonymization as spatial indexing：Toward scalable and incremental anonymization//Proceedings of the 23rd International Conference on Data Engineering. Istanbul，Turkey，2007：1414-1416

[122] Mohammadian E，Noferesti M，Jalili R. FAST：Fast anonymization of big data streams//Proceedings of the 3rd International Conference on Big Data Science and Computing. Beijing，China，2014：1-8

[123] Dwork C. Differential privacy//Proceedings of the 33rd International Colloquium on Automata，Languages and Programming. Venice，Italy，2006，26(2)：1-12

[124] Dwork C，McSherry F，Nissim K，et al. Calibrating noise to sensitivity in private data analysis//Proceedings of the 3rd Theory of Cryptography Conference. New York，USA，2006：265-284

[125] Yao A C. Protocols for secure computations//Proceedings of

the 23rd Annual Symposium on Foundations of Computer Science. Chicago, USA, 1982: 160-164

[126] Goldreich O, Micali S, Wigderson A. How to play any mental game//Proceedings of the 19th Annual ACM Symposium on Theory of Computing. New York, USA, 1987: 218-229

[127] Chaum D, Crépeau C, Damgard I. Multiparty unconditionally secure protocols//Proceedings of the 20th Annual ACM Symposium on Theory of Computing. Chicago, USA, 1988: 11-19

[128] Porter D E, Boyd-Wickizer S, Howell J, et al. Rethinking the library OS from the top down//Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems. Newport Beach, USA, 2011: 291-304

[129] Merkel D. Docker: Lightweight linux containers for consistent development and deployment. Linux Journal, 2014, 2014(239): 2

[130] Ajtai M, Komlós J, Szemerédi E. An $O(n\log n)$ sorting network//Proceedings of the 15th Annual ACM Symposium on Theory of Computing. Boston, USA, 1983: 1-9

[131] Goodrich M T. Randomized shellsort: A simple oblivious sorting algorithm//Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms. Austin, USA, 2010: 1262-1277

[132] Bindschaedler V, Naveed M, Pan X, et al. Practicing oblivious access on cloud storage: The gap, the fallacy, and the new way forward//Proceedings of the 22nd ACM Conference on Computer and Communications Security. Denver, USA, 2015: 837-849

[133] Mohassel P, Zhang Y. SecureML: A system for scalable privacy-preserving machine learning//Proceedings of the 38th IEEE Symposium on Security and Privacy. San Jose, USA, 2017: 19-38

[134] Zheng W, Popa R A, Gonzalez J E, et al. Helen: Maliciously secure coopetitive learning for linear models//Proceedings of the 40th IEEE Symposium on Security and Privacy. San Francisco, USA, 2019: 915-929

QIAN Wen-Jun, Ph. D. candidate. Her current research interests include big data security and privacy, and differential privacy.

SHEN Qing-Ni, Ph. D., professor, Ph. D. supervisor. Her current research interests include operating system and virtualization security, cloud computing and big data security and privacy, and trusted computing.

WU Peng-Fei, Ph. D. candidate. His current research interests include distributed system security, privacy protection, and big data security.

DONG Chun-Tao, Ph. D. candidate. His current research interests include distributed system security, big data security, and trusted computing.

WU Zhong-Hai, Ph. D., professor, Ph. D. supervisor. His current research interests include big data system and analysis technology, big data and cloud security, and highly dependable embedded system.

Background

Big data security and privacy issues have become hot topics in recent years. The continuous threats of private data theft have brought more and more serious impacts, especially from insiders and outsiders in cloud. From the perspective of government supervision, many laws and agreements regulating the collection and use of private data have been proposed. However, it is not enough to restrict the leakage of sensitive information from a legislative perspective. In the face of diverse scenarios and challenges, it is necessary to adopt privacy protection schemes from a technological standpoint.

With the development of privacy-preserving techniques, privacy-preserving data collection, processing and transmission in big data computing environment have achieved extensive attention and research from academia and industry, especially, how to effectively protect data privacy while ensuring the utility and efficiency of data processing. In the industry, some companies chose local differential privacy to implement their operating system (e. g. Apple iOS) and software (e. g. Google Chrome web browser). In academia, researchers from domestic and foreign have developed various privacy-preserving techniques to address privacy issues about personal data. Data anonymization methods attempt to protect personal identifiable attributes. However, some research work has shown that anonymized datasets can be de-anonymized. Based on this, differential privacy technique randomly perturbs raw data before publishing or adds noise to the computational process before sharing the result to the data consumer. Besides, data encryption is one primary technique that allows conducting computation on encrypted data. To enable privacy-preserving computation, there is a range of security primitives, including homomorphic encryption, secure multi-party computation,

Intel SGX for hardware-isolated computation, and oblivious computing against access pattern attacks.

This paper focuses on the privacy-preserving techniques in big data computing environment. According to the process of data computation, privacy issues in big data computing environment are divided into three categories, including privacy leakage of raw data during the data input stage, computational privacy stolen by untrusted attackers during the data computation process, and output privacy inferred by untrusted data consumers with the help of background and output. According to different privacy requirements and privacy-preserving techniques, existing research works on privacy protection are divided into five research directions, including privacy-preserving schemes based on data separation, data interference, secure multi-party computation, hardware enhancement, as well as access pattern hiding. The advantages and disadvantages of main privacy-preserving techniques are compared and the future research directions are discussed.

This work is supported by the National Natural Science Foundation of China under Grant No. 61672062, 61232005.