隐私计算产品
通用安全分级白皮书
(2024 年)

2024年7月

版权声明

编制说明

编写指导组（排名不分先后）：

王小云　中国科学院院士、国际密码协会会士（IACR Fellow）

韦　韬　蚂蚁集团副总裁兼首席技术安全官

任　奎　浙江大学计算机科学与技术学院院长、区块链与数据安全全国重点实验室副主任

李肇宁　交通银行金融科技创新研究院、网络金融部总经理

何宝宏　中国信息通信研究院云计算与大数据研究所所长

钟　剑　深圳国家金融科技测评中心有限公司董事长

主编单位（排名不分先后）：蚂蚁科技集团股份有限公司、中国通信标准化协会大数据技术标准推进委员会、深圳国家金融科技测评中心、清华大学。

参编单位（排名不分先后）：北京银联金卡科技有限公司、招商银行股份有限公司、中国银行股份有限公司、交通银行股份有限公司、浙江网商银行股份有限公司、深圳前海微众银行股份有限公司、杭州高新区（滨江）区块链与数据安全研究院、中国人民大学信息学院、杭州数据交易所、天翼电子商务有限公司、中移信息技术有限公司、联通数字科技有限公司、深圳市洞见智慧科技有限公司、华控清交信息科技（北京）有限公司、北京冲量在线科技有限公司、深圳市纽创信安科技开发有限公司。

参编人员（排名不分先后）：潘无穷、王磊、吴莹、彭晋、廖威、徐基明、刘双、季雨洁、李宏宇、洪澄、周启贤、李婷婷、张晓蒙、黄琳、方文静、李漓春、翁海琴、沈桢天、段然、王明华、胡树伟、李超、许晓琦、刘焱、昌文婷、杨冰然、肖俊贤、姜春宇、闫树、袁博、王思源、白玉真、贾轩、杨靖世、童锦瑞、杨晓芸、董晶、许晋元、罗丰、叶晓聪、袁荣婷、刘强军、

黄榴勇、钱智超、王安宇、冉德龙、丛天硕、杨波、邱晓慧、谭亦夫、胡师阳、佟冬、傅杰、葛明嵩、张翼飞、石新蕾、谢谨、钱菲、张园超、谢宗华、陆茂斌、李辉忠、李贺、李昊轩、陈宇杰、王朝阳、杨萌、贾祥娟、张秉晟、毛应波、刘健、薛峰、黄科满、林洋、潘凯伟、周宇、贺伟、朱江、张金杰、郭叶、茹志强、降鑫磊、刘洋、孙林、王项男、贾晓芸、何浩、姜峰、王煜坤、靳晨、陈浩栋、宋雨筱、刘尧、朱凯。

序　言

党的十九届四中全会指出：“健全劳动、资本、土地、知识、技术、管理、数据等生产要素由市场评价贡献、按贡献决定报酬的机制”，首次将数据增列为新的生产要素。数据作为关键的生产要素之一，通过跨地域、跨行业、跨领域、跨机构的数据流通释放价值。然而，近几年数据泄露和滥用的上升趋势表明,数据流通仍存在诸多问题。如何在安全风险可防控的前提下，实现数据的高效流通和交易成为当前数据价值释放面临的首要任务。

数据流通具有双面性：数据价值越高，流通过程中的风险越大。如何让大规模高价值数据进行可信流通，成为数据要素市场发展的核心议题。**传统的数据安全问题是数据流通内循环问题：数据持有方，也是系统的运维管理方，对自己的数据安全负全责。随着数据要素市场的发展，数据流通外循环是更为关键的问题。外循环指的是，数据要素离开了数据持有方的安全域进行流通，此时数据持有方和系统运维方不同。这给数据流通带来全新的挑战：流通链路上相关运维管理方有窃取数据的动机和可能，则原有的、依赖运维管理方构建的安全防御措施全部失效，数据持有方难以有效保护和管控自己的数据。这也是导致数据持有方不敢提供数据、不愿参与数据流通的重要原因。**

隐私计算有多个起源,目前在工业界广泛使用的隐私计算 特指 隐私保护计算（Privacy-preserving computing, PPC），也称为 隐私增强计算（Privacy-enhancing computing, PEC），在工业界习惯性简称为隐私计算。广义隐私计算是面向隐私信息全生命周期保护的计算理论和方法，涵盖信息所有者、信息转发者、信息接收者，在信息采集、存储、处理、发布（含变换）、销毁等全生命周期过程的所有计算操作，是在保护隐私安全的前提下，实现数据

安全共享的一系列技术。**隐私计算为数据外循环提供全流程可信保障，并已经成为促进数据要素跨域流通和应用的核心技术领域，**广泛应用于金融、政务、医疗、能源、制造等诸多行业。2020 年 4 月，《工业和信息化部关于工业大数据发展的指导意见》提出激发工业数据市场活力，支持开展数据流动关键技术攻关，建设可信的工业数据流通环境。2021 年 5 月印发的《全国一体化大数据中心协同创新体系算力枢纽实施方案》提出，促进数据有序流通，试验多方安全计算、区块链、隐私计算、数据沙箱等技术模式，构建数据可信流通环境，提高数据流通效率。2022 年 10 月国务院办公厅印发的《全国一体化政务大数据体系建设指南》提出探索利用核查、模型分析、隐私计算等多种手段，有效支撑地方数据资源深度开发利用。

隐私计算技术可以在保护隐私安全的前提下实现数据可信流通，最小化数据泄露的可能性，从而极大地促进数据流通的发展和降低安全风险，实现整体社会价值最大化。然而，在实际应用中，各场景中的参与方信任程度不同、数据类型不同，各方在不同场景里需要达到的数据可控程度也是不同的，一味追求绝对安全或者忽视安全都是不可取的。所以，隐私计算产品需要安全分级方法，为实际产品选型提供指导。一方面，隐私计算技术路线众多，且不断有新的技术涌现，使用方难以评估这些技术的安全程度。另一方面，由于部分隐私计算技术性能较低，市场上存在牺牲安全性换取性能的产品。在不明实况的情况下，使用方可能会更青睐这些产品，从而出现劣币驱逐良币的现象。目前，虽然针对单一技术路线已经有一些安全分级标准，但是不同技术路线的分级标准完全无法对应，用户无法对所有的产品进行横向比较，这些标准也不适用于新出现的技术路线。因此，适用所有技术路线的通用安全分级思路亟需明确，来引导数据跨域流通全领域的安全评估工作，进而推

进更宽广的隐私计算运用，加速数据可信流通的发展。

建立统一的安全分级体系来评估数据流通链路的安全性需要多行业各方参与，久久为功。以前针对单一隐私计算技术路线进行分级时，可以依靠技术特征来分级。但是，隐私计算包含的分支技术各自有不同的安全根基，技术特征完全不同，无法沿用该思路进行通用安全分级。如何对这些原理完全不同的技术进行统一分级，是十分有挑战的。更进一步，中间变量泄露如何量化、半诚实密码协议的实际安全效果如何、侧信道攻击对可信环境的安全影响有多大，这些内容的量化仍然缺乏有效方法或者存在争议。从另一个方面考虑，目前的一些分级方法侧重于评估技术原理和方案设计。但是，从“设计安全”到“真正安全”仍然有较大距离，即，实现环节的安全性。若分级标准缺乏对“实现环节的安全性”的度量，产品获得的等级与实际攻防效果将不匹配，导致安全隐患或成本过高。此外，除了隐私计算核心技术模块，通信框架、访问控制系统、主机系统、管理系统、部署系统等都会影响产品整体的安全性，安全分级应该包括这些内容，系统整体的安全性应由其中最薄弱的环节来界定。本《白皮书》旨在逐一讨论隐私计算安全分级面临的诸多难点，包括技术路线特征不同难以进行统一分级、部分重要的安全能力难以被分级和量化、安全是系统性问题涉及的维度多、范围广。本文针对以上挑战，给出通用安全分级的设计思路，包括按照攻防效果分级来屏蔽不同技术路线差异，在“可证安全”和“不安全”之间增加一个“抵御已知攻击”的分级水位，引入软件信誉度等更多维度量化“实现安全”，明确所有技术特征与安全分级的对应关系。

当前，围绕隐私计算安全标准化与安全分级问题，中国通信标准化协会 CCSA 已推动完成了包括 YD/T 4690-2024《隐私计算 多方安全计算产品安

全要求和测试方法》、YD/T 4691-2024《隐私计算 联邦学习产品安全要求和测试方法》、YD/T 4947-2024《隐私计算 可信执行环境产品安全要求》三项针对隐私计算细分技术安全的行业标准，对各分支技术路线的安全风险进行了全面梳理。基于隐私计算安全分级标准需求的紧迫性，由深圳国家金融科技测评中心牵头，联合北京银联金卡科技有限公司、蚂蚁科技集团股份有限公司、杭州高新区（滨江）区块链与数据安全研究院、北京冲量在线科技有限公司、深圳市洞见智慧科技有限公司、天翼电子商务有限公司、深圳市纽创信安科技开发有限公司、杭州超限数动科技有限公司共同编制的Q/NFEC 0001—2024《隐私计算产品安全能力分级要求》标准于2024年2月正式发布实施，该标准是国际上首个适用于不同隐私计算技术路线的通用安全分级标准，根据攻防效果结合不同应用场景安全需求对安全等级进行了五级划分并提出了具体要求。

本文将在《隐私计算产品安全能力分级要求》标准的基础上，对隐私计算产品通用安全分级涉及的一系列问题与解决思路展开介绍。

一、隐私计算技术背景与概览

- 数据要素流通为社会经济创造巨大价值，也对数据安全与隐私保护等方面提出了更高要求。隐私计算技术作为保障数据可信流通的有效方式，已逐渐成为促进数据要素跨域流通和应用的核心技术。
- 隐私计算主要分为算法类、可信类、融合类三大类技术，本章对隐私计算主流技术路线进行简要介绍。

二、隐私计算产品通用安全分级的挑战与思路

- 介绍制定隐私计算安全分级面临的诸多难点，包括技术路线特征不

同难以进行统一分级、部分重要安全能力难以被分级和量化、安全是系统性问题涉及的维度多、范围广等。

- 针对以上挑战，本章介绍了通用安全分级设计思路，包括按照攻防效果分级来屏蔽不同技术路线的差异，在“可证安全”和“不安全”之间增加一个“抵御已知攻击”的分级水位，引入软件信誉度等更多维度来量化“实现安全”，明确所有技术特征与安全分级的对应关系。

三、隐私计算产品通用安全分级介绍

- 本章分不同维度详细介绍了各个等级的安全要求，包括所有隐私计算技术产品都要遵守的通用要求，以及算法类技术产品和可信类技术产品各自要遵守的扩展要求。
- 同时将中间结果细分为直接中间结果和间接中间结果，提出基于自由度和基于熵两种量化中间结果泄露程度的方法。

四、隐私计算产品通用安全的场景应用

- 通用安全分级在实际应用中，各场景对应的参与方信任程度不同、数据类型不同，各参与方需要达到的数据可控程度也是不同的。来自金融、通信、教育等行业的数据安全应用需求方参与了本《白皮书》的编制，并结合实践阐述针对具体应用场景的安全需求。本章对基于不同业务场景的安全水位要求应该采用的安全等级提出了建议。

五、总结和展望

- 总结通用安全分级在分级思路、涵盖范围、特定问题的度量方法上都有突破创新，实现对不同技术路线、不同产品形态进行统一分级的目标，便于使用方根据实际的应用场景进行产品选型。

附录

- 对联邦线性回归算法和联邦 SecureBoost 算法的中间变量泄露进行了详细分析，最终给出了泄露程度与样本数、特征数、训练轮数之间的数学关系。

目 录

一、隐私计算技术背景与概览 1
（一）数据要素流通面临的安全风险 1
（二）隐私计算核心技术介绍 1
二、隐私计算产品通用安全分级的挑战与思路 9
（一）隐私计算产品通用安全分级的挑战 9
（二）隐私计算产品通用安全分级设计思路 11
三、隐私计算产品通用安全分级介绍 18
（一）通用安全分级的安全要求 18
（二）约定可泄露信息 34
四、隐私计算产品通用安全分级的场景应用 41
（一）隐私计算典型应用场景与发展情况 41
（二）应用场景安全等级建议 42
五、总结与展望 45
附录：约定可泄露信息分析参考示例 46
（一）联邦线性回归 46
（二）联邦 SecureBoost 56

一、隐私计算技术背景与概览

（一）数据要素流通面临的安全风险

数据流通能够为企业和社会创造更多价值，有效推动数字经济快速发展，但同时，也对数据安全与隐私保护等方面提出了更高要求，主要体现在以下几点。

一是，数据流通具有双面性：数据价值越高，流通过程中的风险越大。数据具有易复制、非竞争、非排他等属性，能够快速地以近乎零成本的方式进行复制，且同一个数据可被不同主体重复采集、存储，数据被一个主体使用时并不影响其他主体的使用。这些特性为数据带来了更普遍的使用效益与更大的潜在经济价值。然而，当拥有数据持有权的机构或个人控制数据的手段被恶意攻击者攻破时，这些原先被管控的数据将很快被复制传播，且数据持有者拥有的数据门槛会顷刻丧失，导致数据价值大幅度贬值甚至归零，对高价值数据影响巨大。特别是当数据涉及到众多第三方时，其被泄露或者滥用时还会造成巨大的外部损害。

二是，数据流通场景是数据外循环，相较数据内循环，数据流通面临的安全威胁发生本质变化，且参与主体多、涉及的环节多、面临的安全风险范围广。当前，数据流通已经从企业内部延伸至外部共享、交易和使用，数据流通链路增长，流通环境更加开放，参与主体更加多样，在大规模流通利用过程中将显著增加数据暴露面，各环节存在的安全隐患也将被放大，导致数据流通的安全保障工作变得更为复杂。**除了来自外部环境的恶意攻击者外，还需全面考虑来自参与数据加工和流通活动的各类主体以及相关内部人员作恶的风险。**尤其是，数据在处理过程中需要位于数据加工方的信息系统内。此时，数据加工方如果主动发起攻击，即，系统的最高权限方（运维管理方）

发起攻击，安全威胁较一般的黑客攻击显著增大且发生了本质变化。同时，需要防范的安全问题也贯穿于数据流通利用的各个环节，例如，在数据传输过程中，截获、篡改或重放攻击导致的数据泄露或完整性受损问题；数据存储过程中，数据未授权访问、内部窃取或拆卸硬盘等问题；认证和鉴权过程中，跨组织数据授权管理和数据流向追踪问题；程序验证和更新过程中，运维管理方更新恶意软件等问题；动态脚本的研发和使用过程中，因程序调试导致的数据泄露或者故意编写恶意脚本等。

隐私计算技术作为保障数据可信流通的有效方式，已逐渐成为促进数据要素跨域流通和应用的核心技术。

（二）隐私计算核心技术介绍

1.算法类

（1）多方安全计算

多方安全计算（Secure Multi-party Computation，MPC）是指在无可信第三方的情况下，多个参与方共同计算一个目标函数，并且保证每一方仅获取自己的计算结果，无法通过计算过程中的交互数据推测出其他任意一方的输入数据。多方安全计算是多种密码学基础工具的综合应用，其实现依赖于混淆电路、秘密分享、不经意传输等技术。多方安全计算技术能够保证数据提供方的原始数据不离开本地，通信和计算过程均在密文状态下进行，仅依靠算法就能提供安全保障能力，不需要依赖额外的硬件。但是大量的密码操作与通信交互，也限制了其性能表现。

多方安全计算涉及的各类算法协议大部分存在特定的安全假设，在现实应用过程中，应重点考虑需要保护的安全属性、支持的敌手类型以及合谋阈值等指标。

安全属性方面，主要包括隐私性（Privacy）、正确性（Correctness）等。隐私性是指参与方无法从计算过程中获得除最终输出之外的任何额外信息，即使在有敌手存在的情况下，敌手也不能通过观察计算过程来推断出其他方的输入数据。正确性是指保证所有诚实的参与方都能获得正确的计算结果，即使存在敌手，协议的设计也能够确保计算结果不会被篡改或错误地影响。

敌手类型方面，主要包括半诚实敌手（Semi-honest Adversary）、恶意敌手（Malicious Adversary）、隐蔽敌手（Covert Adversary）等。半诚实敌手会遵循协议的规则，但可能会尝试从协议执行过程中泄露的信息中推断其他参与方的私有输入，也被称作被动攻击者。恶意敌手可以任意偏离协议的规则，尝试通过发送错误信息或不遵循协议步骤来获取其他参与方的私有输入或影响计算结果，也被称作主动攻击者。隐蔽敌手在某种程度上可以偏离协议，但必须小心行事以免其恶意行为被检测到。如果攻击被检测到，敌手将不会获得任何额外信息。但近年来业界也逐渐形成共识，无法有效审计检测的半诚实敌手或者是隐蔽敌手也会引入巨大的实际恶意攻击风险。

合谋阈值方面，一个安全的 MPC 协议需要能够确保即使部分参与方腐败（成为攻击者），他们也无法单独或联合起来推断出其他方的私有输入。在安全假设中，当诚实参与方数量超过腐败参与方时，存在合谋情况下也能安全执行的协议，称为诚实大多数（Honest majority）；诚实参与方数量小于腐化参与方时，称为不诚实大多数（Dishonest majority）。

（2）联邦学习

联邦学习（Federated Learning，FL）是指一种多个参与方在保证各自原始私有数据不出数据方定义的私有边界的前提下，以保护隐私数据的方式交换中间计算结果，从而协作完成某项机器学习任务的模式。根据参与计算的

数据在数据方之间分布的情况不同，可以分为横向联邦学习、纵向联邦学习和联邦迁移学习。联邦学习是解决数据孤岛和数据安全问题的重要框架，其强调的核心理念是“数据不动模型动”。为有效保护数据提供方的原始数据，联邦学习需要对各参与方的模型信息交换过程增加安全设计，常采用如差分隐私、同态加密等技术对中间结果进行保护，若对中间结果保护不当，也存在数据泄露隐患。

联邦学习与多方安全计算技术类似，同样存在特定的安全假设，考虑到性能与安全的平衡，当前大部分联邦学习算法基于半诚实假设，在此基础上扩展对于部分特定攻击的检测和抵御能力。

（3）同态加密

同态加密（homomorphic Encryption，HE）可以分为半同态加密和全同态加密等技术分支。半同态加密能够支持在密文上进行加法或乘法运算，一般作为多方安全计算和联邦学习的辅助技术，单独使用的数据流通场景较少。全同态加密既能支持密文上进行加法运算，也能支持进行乘法运算。2009 年，有实用化可能的全同态加密算法首次被提出，并在之后得到快速发展。目前该技术距离规模化应用仍有较大提升空间，其不需要额外的通信交互，但计算量、密文膨胀倍数均高于多方安全计算与联邦学习。

在同态加密技术的实际应用中，还需额外考虑以下两点。一是使用同态加密进行数据流通计算时，数据提供方无法控制加工方的计算逻辑。二是当参与方数量超过两方以上时，需要额外考虑解密密钥的分配问题。

2.可信类

（1）可信执行环境与机密计算

可信执行环境（Trusted Execution Environment，TEE）通过软硬件方法

在中央处理器中构建隔离的安全区域，从技术上实现了对其内部加载的程序和数据的机密性与完整性保护，不受到外部软硬件和运维人员的影响。将多个参与方的数据经安全信道汇聚到可信执行环境的隔离区域内进行融合计算，能够实现数据的可信流通和利用。与算法类技术相比，除了计算过程近似明文计算外，可信执行环境在算子类型方面几乎没有限制，能够支持所有的算子和复杂算法，上层业务表达性更强。利用 TEE 提供的计算度量功能，还可以对外提供身份、数据、算法等全流程的一致性证明。TEE 通常包括可信根、远程证明、内存加密等关键技术组成。

可信根是可信类系统的信任基点，主要提供密码计算、可信基准值存储、策略存储等服务，通常以专用芯片、固件等形态实现在硬件可信根模块中。固件形式的硬件可信根由于运行在固件系统内，可以隔离用户进程的访问，从而防范恶意进程的攻击。芯片形式的硬件可信根一般可以抵御更多类型的攻击，包括侵入式、半侵入式、非侵入式攻击等。根密钥的管理和保护是硬件可信根的基本功能，除了可信根自身的安全管控能力之外，可以基于根密钥进行密钥派生、设备身份识别、通信加密等操作，从而实现基于可信根来派生 TEE 系统的信任链，有效防范恶意管理员等潜在攻击者对 TEE 应用内容的威胁。

远程证明为 TEE 提供完整性和真实性方面的安全保障，它允许一个节点远程验证另一个节点（TEE）的可信状态和安全属性。远程证明的核心目的是确保两个节点之间的通信安全，并验证其中一方的平台和运行环境的完整性和真实性，通常包括完整性度量、完整性证明、完整性报告、挑战响应协议等关键功能。远程认证通常是通过可信根进行数字签名来实现的，因此可信根要确保相关密钥不会被窃取、随意调用等。

内存加密为 TEE 中的敏感数据提供机密性方面的安全保障。TEE 一般通过硬件手段对内存中的数据进行加密，以保护数据免受未授权访问和各种攻击，确保即使在系统遭受攻击，且攻击者能够物理接触到硬件的情况下，内存中的敏感数据也不会被轻易解密和读取。

机密计算指的是利用硬件保护数据的计算过程，机密计算目前实现的主要方式为可信执行环境。

（2）可信计算

可信计算（Trusted Computing）通常以可信平台模块（Trusted Platform Module，TPM）或可信平台控制模块（Trusted Platform Control Module，TPCM）等硬件芯片为可信根，依次验证 BIOS、操作系统、应用软件等软件启动链条，确保启动的软件没有被恶意篡改。TPM（TPCM）一般还提供远程证明功能，通过该功能远程客户端可以确认与其交互的计算机平台是否使用了安全的（预期的）软件。与 TEE 技术类似，可信计算技术中也包括可信根和远程证明等技术组成，但是通常并未采用内存加密等技术，以及禁止系统管理员查看系统状态等。因此，可信计算技术如果要抵御运维者攻击，还需要实施额外的安全保障措施。

3.融合类

当前，隐私计算虽然处于快速发展阶段，各主流技术路线的创新与突破层出不穷，但是在短时间内仍然难以从本质上解决单一技术的瓶颈限制，各技术路线都存在着不同的局限性。行业内普遍开始尝试探索多技术融合的实践路径以解决隐私计算技术面临的各类技术问题。在一些场景下，隐私计算多技术融合往往能够产生“1+1>2”的效果。在实际的隐私计算产品中，技术融合往往呈现多种形态，不同技术之间的融合深度也有所不同。

一类是“横向”的融合方式，以某种隐私计算技术作为主线，在部分关键步骤中引入其他技术辅助配合，实现安全方面的完备性。例如，在联邦学习过程中，借助多方安全计算技术用于模型参数的安全汇聚，增强对中间数据的保护，实现更加安全的联邦学习聚合算法；在多方安全计算过程中，将部分需要借助可信第三方的操作（如乘法三元组生成）在可信执行环境内完成，通过技术手段解决了信任依赖的问题，进而增强了整个系统的安全性。

另一类则是“纵深”的融合方式，将多种隐私计算技术深度融合，由多重技术手段共同作用，形成崭新的产品形态，以突破单一隐私计算技术瓶颈。例如可信密态计算（Trusted-Environment-based Cryptographic Computing，TECC），通过将数据以密态形式在高速互联的可信节点集群中进行计算、存储、流转，实现数据持有权保障、使用权出域可控，支撑任意多方大规模数据安全、可靠、高效地融合与流转。TECC 具有可信节点内进行密态计算、数据持有方与计算方解耦、域外可控的数据密态封装等基本特征，可以通过安全编程语言、形式化验证、多级别可信节点等进一步提升安全性和适用性。

4.广义隐私计算其它概念介绍

沙箱技术原本是指为不可信进程或不可信代码提供隔离的运行环境，目的是防止这些程序影响主机环境，例如虚拟机或容器。在数据流通场景下，沙箱技术也泛指一般意义的隔离环境，既包括“防止沙箱环境内的程序影响主机”，也包括“防止主机访问沙箱环境内的程序”。要指出的是，这是两种不同的安全能力，部分沙箱技术只具备其中一种安全能力。此外，沙箱技术也不要求其提供的隔离性能够抵御管理员作恶，是否具备该能力跟沙箱技术的具体实现有关。

差分隐私是指通过添加噪声的方法，防止攻击者从统计信息中反推出个

体信息的技术。差分隐私一般具有可调参数，通过参数的调整可以大幅度变化安全保护能力。需要指出的是，差分隐私提供安全能力的同时，一般也伴随着数据精度的损失，并且提供的安全能力越大损失越大。

数据脱敏是指通过脱敏规则，对敏感信息进行变形或屏蔽，降低数据敏感级别，扩大数据可共享和被使用范围。在数据共享流通过程中，需要量化分析其敏感度或保护程度要求，按需采取适当的数据脱敏措施。

在数据流通领域，数据空间技术通常指对数据流通的整体设想，包括组成结构、各个元件的作用机制、安全要求等。数据空间一般会表明各个元件的能力要求，以及当各个元件满足该能力要求时，整体达到的效果。数据空间一般未指明各个元件应该采用何种安全技术来实现这些能力要求，各个元件主要还是依靠前述的隐私计算等技术来实现。

二、隐私计算产品通用安全分级的挑战与思路

（一）隐私计算产品通用安全分级的挑战

不同技术路线、不同产品形态、不同应用场景下的隐私计算产品所面临的隐私数据泄露风险及安全需求可能差异巨大，在没有统一安全分级标准的情况下，产品开发方及使用方难以评估和衡量安全与性能之间的平衡，做到既相对安全又高效好用。在“序言”中我们了解到隐私计算产品通用安全分级的意义和必要性，但要在异构的技术体系下建立通用统一又合理可行的安全分级维度，并在确定的分级维度下，结合实际情况进行合理的安全等级划分，以匹配大多数应用场景的安全需求，使得厂商在满足特定应用场景的安全要求前提下能够更加注重效率的提升，有着诸多挑战需要面对和解决。

1.挑战一：技术路线多，技术特征显著不同

从第一章的“技术介绍”中我们可以看到隐私计算现有的技术路线众多，每个技术路线的实现方式、应用场景、技术特征及所面临的安全风险存在显著差异。虽然不同的技术路线单独进行安全分级时，可以按照技术路线上的特征进行分割，使得在特定技术路线上制定安全分级规则具备可行性，但这种分级方式不得不面临两个困境：一是各技术路线上的安全分级维度和基准不一，最终难以拉到统一的安全水位上进行横向比较，使得产品使用方在面对异构产品选型时，将难以做出客观的安全比较；二是只能针对单一技术路线的产品进行系统评估，随着越来越多融合类隐私计算产品的出现，单一技术路线的分级和评估难以保障产品的整体安全性。

虽然构建统一的安全维度能够解决以上问题，但不同技术路线的安全关注点不一样，例如多方安全计算要面对的是算法、协议等安全问题，而可信执行环境更多的是要考虑安全机制的完备性、硬件安全性（如侧信道攻击）

等。两条路线面对的安全问题不同，评判的角度也完全不同。如何找到统一的、可量化的安全分级维度，是安全分级的首要前提和一大难点。

2.挑战二：技术特征的最小可量化粒度，安全性跨度可能很大

中间结果泄露的数据越多，安全性就越差，但往往性能越高。通过调整中间结果的泄露程度，获得安全性和性能不同的算法，是隐私计算产品设计的一个重要手段。例如，一些联邦 LR 算法通过增加安全聚合协议，减少梯度的泄露数量。我们在设计安全分级标准时，必须要考虑到这一点，而不能强制要求所有产品升级为不泄露中间结果、或直接忽略这一隐患。

中间结果泄露的多少对安全性影响至关重要。但目前产业界却很少对其进行进一步展开分析，往往是一概而论，以中间结果是否泄露作为隐私计算安全与否的一个分界点，使得隐私计算的安全性评判标准跨度过大，要么过于安全，要么不安全。但事实上，依据获取的中间结果信息，所能推导还原出的隐私数据和造成的安全风险程度可能相差巨大，例如在仅有 ID 信息被泄露的情况下，并不会由此推导出参与方的其他隐私数据，但梯度信息的重复泄露，将有可能被利用从而恢复出关键信息甚至整个数据集，在安全分级上如何进一步考虑各类中间结果泄露所造成的安全风险，成为了安全分级标准制定无法忽视却需要重新梳理研究的内容之一。

3.挑战三：实现环节的安全性难以被量化，但又很重要

产品的实现安全一直难以被量化。因成本原因，评测一般重点检查关键逻辑的实现情况。但是其他逻辑或模块带来漏洞的可能性，与关键逻辑是相同的。这部分的代码量非常大，例如包含大量第三方开源代码，很难被有效检测。隐私计算由于运维管理方潜在的作恶的可能，以及其在使用过程中可能执行来自其他参与方的指令或代码片段，使得其安全风险比一般信息系统

更加复杂。如何应对难以量化的实现安全成为了安全分级标准制定的难点和考量点。

4.挑战四：安全性评估涉及的范围广、维度多

隐私计算产品安全性仅仅考虑计算过程中的泄漏是远远不够的，一个完整的隐私计算产品可能涉及数据的上传、存储、授权、融合计算、联合研发、数据治理、结果交付等多个数据流通环节。一方面安全性评价应该综合考虑多个因素，包括但不限于数据的保密性、完整性、可用性以及访问控制等。例如，除了中间信息的泄漏，还需要考虑数据的输入、输出以及处理过程中可能存在的安全威胁，如数据篡改、未经授权的访问等。另一方面，由于隐私计算的实现技术路径不同，可能涉及硬件、操作系统、容器、镜像、算法、应用等多个层次，每个层次存在的安全风险差异巨大且内容庞杂。

总的来说，作为需要融合多技术路线的通用安全分级标准，需要考虑的安全性评估范围极大，如何在跨度极大的安全风险面上找出核心重要的安全评估要点，做好取舍，成为了安全分级标准编制过程中需要反复考量的内容。

（二）隐私计算产品通用安全分级设计思路

安全性度量的本质在于，评估攻击者需要付出多大的成本、克服多大的不确定性，才能攻破给定的安全防护保障，造成信息泄露或有相关风险。我们依照这个思路，评判那些有争议的观点，开展安全分级工作。例如，对于“无法在理论上证明安全，但是现实中难以被攻破”的产品，仍然给予相对较高的等级；对于“在理论上进行了部分论证，但是对真实攻击者抵御能力较弱”的产品，给予较低的等级；一些确定性的攻击手段，需要在较低的等级进行抵御；一些减少系统漏洞的安全措施，能够增加实际攻击的难度，也是安全分级依据的重要参考。

1.按照攻防效果分级以适应不同的技术路线

按照攻防效果进行分级是解决上述挑战的最主要思路，它比按照技术特征分级更贴近实际需求。评价一个产品的安全性，最重要的是它能够抵御什么样的攻击，而不是它采用什么样的技术进行实现。用户也更关注攻击的抵御情况，而不是具体的实现方法。最重要的是，按照攻防效果进行分级可以跨越不同的技术路线，为不同的技术路线提供一个公平的、统一的度量尺度。

当我们确定以攻防效果进行分级之后。下一个目标就是制定每一个等级具体的安全程度。有两个原则，一个就是每个等级都对应着一些典型的场景需求或产品形态；另一个，就是不同等级之间的跨度尽量均匀。

例如，第三级（一共五级）作为中间等级，定位为安全性和性能相对均衡的等级，即在保障没有实质性安全问题的情况下，尽可能的让渡安全性提升性能。第三级是既有实质安全保障，又能够大规模普及的等级。高于三级则定位为对安全性有高度需求的特定场景，低于三级则定位为可以适度牺牲安全性的场景。要想满足该定位，第三级产品应该至少达到“没有已知攻击”可以攻破，即目前对此类产品没有可实施的攻击方法。需要强调的是抵御参与方作恶是隐私计算的核心目标，所以第三级要抵御参与方主动发起的上述攻击。

要指出的是，以下两种情况也符合第三级要求：一是如果仅仅是有安全隐患，但是离发生实质性攻击还有较大距离；二是如果有攻击方法，但是攻击方法仅能获得一些不重要的信息，对产品的实际安全性影响不大。

第三级以下，我们的制定思路为：第三级要求能够抵御“已知攻击”。但并不是所有的“已知攻击方法”都是容易实施的。一些“攻击方法”需要非常专业的密码学知识、硬件知识、专业硬件设备和时间成本，一般的运维

人员、程序开发人员无法实施上述攻击，实际实施这类专业攻击的门槛很高、成本代价很大，部分普通数据应用场景只需要抵御来自一般人员的攻击即可。所以，我们针对这类场景制定了第二级。具体地，第二级只需要抵御一般水平攻击，而不需要抵御所有已知攻击。

第一级我们针对的场景是“平台方有较高的信誉度，其他参与方相信平台方不会作恶”。产品只需要抵御外部的攻击者，不需要考虑参与方作恶的情况。此时，产品可以采用传统的安全防护手段达到目的，例如认证和鉴权等。第一级不需要防备参与方作恶，安全防护更加容易，安全要求也是最低的。

第三级以上，我们的制定思路为：第三级要求能够抵御“已知攻击”，但是未要求能够抵御安全隐患。隐患在一些因素的诱发下，可能会转变为实质性攻击。所以，第四级要求能够抵御主要的隐患也叫“未知攻击”。第四级不要求抵御所有的“未知攻击”，而是要求抵御主要的、最有可能发生的攻击。

第五级是本标准的最高级别，安全水平是最高的。一至四级的安全性都是通过专家论证、渗透测试等方法进行验证。这些方法可能因为人的检验疏忽，漏掉一些小的安全问题。为此，第五级设定为需要“通过形式化方法进行论证”。即通过数学推导的方法证明产品的安全。因为数学推导具有高度的严密性，所以一旦被论证，就可以认为产品在理论上是安全的。目前的技术还做不到对整个产品进行形式化验证，如果要求对整个产品进行形式化验证将失去现实意义。所以第五级只要求对关键位置进行形式化验证。但是，这仍然能够从理论上排除主要风险，安全性较第四级有显著提升。

2.为技术特征打造新的分界点，解决现有度量方法颗粒度过大的问题

我们在“挑战”章节中讲述了，目前“中间结果泄露”没有合适的分界点，将安全的算法和不安全的算法分开。本标准的思路是打造一个新的分界点。具体地，我们将中间结果分为两类，一类是，在目前的攻击水平下，泄露后对数据方利益损失不大，叫做“约定可泄露信息”；另一类，认为其泄露后对数据方利益损失较大，叫做“非约定可泄露信息”。“约定可泄露信息”在大部分安全等级中都允许泄露，使得那些实际在用的、安全性较高的产品，能够获得合理的等级。

下面说明一下如何将新的分界点应用到五级安全分级中。第三级只需要抵御“已知攻击”。“约定可泄露信息”的泄露，并不会造成实质性的攻击。所以第三级产品允许泄露“约定可泄露信息”。“约定可泄露信息”只是在目前的攻击水平下，无法利用其反推数据方原始信息，但是理论上仍然有可能从中反推出原始信息。所以第五级不允许泄露“约定可泄露信息”。

在“非约定可泄露信息”中，有这么一部分信息，虽然能够推出数据方原始信息，但是推导的过程需要复杂的数学方法。因为第二级不需要抵御复杂的攻击方法，所以这些信息在第二级也是允许泄露。第二级要防止的是有价值的信息的泄露，或者泄露的信息能够通过一般攻击方法推导出有价值信息。

下图是上述内容的一个总结。“有价值信息”不允许在二级泄露，容易被利用的信息，即，基于一般攻击方法可以从中推导出有价值的信息，在二级也不允许泄露。不容易被利用的信息，即，通过复杂的已知攻击才能从中推导出有价值信息，在二级允许泄露但是在三级不允许泄露。“约定可泄露

信息”，目前没有已知方法能够从中推导出有价值信息，在三级允许泄露但是在五级不允许泄露。

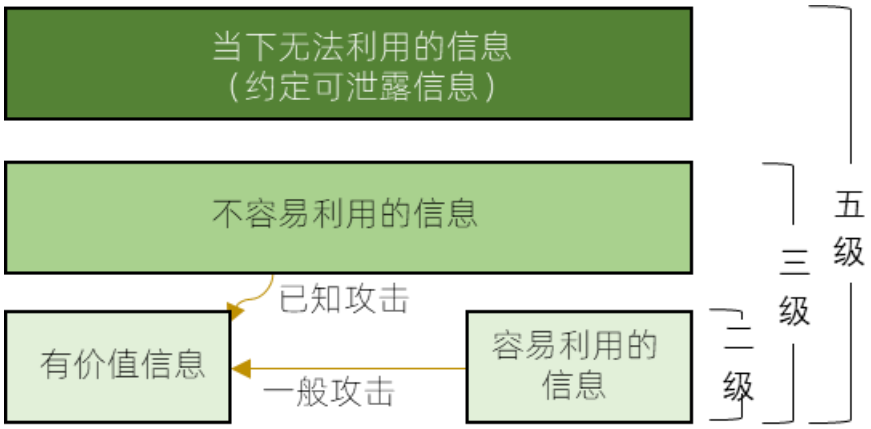

图 1 各安全等级保护的数据范围

3.提出更多维度量化实现安全

为了提升对实现环节安全性的评估，我们要引入了更多的、有效的、易量化的维度。

一是“软件信誉度”，即软件历史上出现过多少漏洞。漏洞产生的原因跟研发人员、设计人员的技术水平和严谨程度相关。这些因素在整个产品研发过程中相对稳定，因此，一个经常被发现漏洞的产品，未来被发现漏洞的概率也非常大。我们将这一思想应用到安全分级中。

- 这只是一种概率层面的推测，所以在安全要求相对宽松的第二级没有增加这方面要求。
- 如果产品的某个组件(比如产品引入的开源软件)经常有高危漏洞，未来发生高危漏洞的可能性就很大。这不符合第三级 “要求没有确定性较高的攻击方法”。所以，第三级要求不能使用安全信誉度低的组件。另外，这样的要求条款也促使行业更多的选择那些安全性好的组件，有助于行业的良性发展。

- 第四级产品要求能够抵御“未知攻击”。那么第四级产品被实质性攻破的可能性应该非常低。因此第四级产品要求在实际运行过程中不能被实质性攻破。

二是采用的纵深防御措施是否足够。纵深防御是系统安全设计的重要理念和手段，所以我们把它引入到安全分级的要求当中。纵深防御主要是用于抵御潜在的攻击隐患，所以我们把它做为四级的要求之一。为了使得四级较三级有个全面的提升，四级要求所有主要的安全隐患都要有相应的纵深防御机制。四级产品的单个组件允许出现漏洞，但要保证基于漏洞的攻击能够被纵深措施阻止，攻击不能真正穿透所有防御措施，对产品造成实质性危害。

三是实际的攻防表现如何。这里的“实际攻防表现”指的是由专业的攻防人员，对产品进行长时间的攻击，通过防守表现判断产品安全性。攻击者视角与一般检测人员不同，他们可能可以发现意想不到的攻击途径。另外，这一措施也保证了通过检验的产品，在面临实际黑客时，具有较强的抵御能力。这一措施主要是利用专家经验去挖掘产品的未知漏洞，所以作为第四级的要求。要说明的是，在一至三级的时候，检测人员仍然可以使用攻击者视角对产品进行攻击，来判断产品是否符合特定条款。

四是防护措施是否进行了形式化验证。形式化验证也是保障实现安全的重要手段。因为它的严格程度要求“产品能够从数学上证明是安全的”，所以它被作为第五级的要求。另外，我们鼓励产品采用安全编程语言。采用安全编程语言在研发时需要遵循更多的规则，但是可以从源头上消除内存安全等高危隐患。使用安全编程语言研发的产品，能够更容易进行形式化验证。

4.评测范围覆盖整个产品和所有技术特征

本标准涵盖的范围是整个隐私计算产品，不仅仅是关键技术部分，也包括机构的注册、鉴权，程序的更新、动态脚本的研发，以及数据的传输、存储、鉴权、交付等多个环节。因运维管理员潜在的作恶可能，这些环节与一般的信息系统有着不同的安全要求，这一点容易被忽视，进而导致产品整体安全性显著降低。

为了适应不同的技术路线，本标准按照攻防效果去分级。虽然解决了一大部分问题，但仍然遗留了“对某些具体实现方式的评判带有一定的模糊性”的问题。因此，我们提出从攻防效果和技术特征两方面进行约束。攻防效果是总的划分标尺，保证各个等级要求的安全程度是合理的。业内专家再依据该要求，共同商议“什么的技术特征能够达到该安全程度”，并将商议的结论也做为分级标准的一部分。有了这些内容，实际检测中，就可以根据产品的技术特性进行检测，既保证了安全等级划分的合理性、又减少了检测过程中的二义性。因为这些内容跟具体的技术点有关，我们会在后面的章节逐步讨论。主要分为这三个部分：

- 通用安全要求：与一般信息系统要求的维度相同，但会根据隐私计算的实际情况，调整这些维度上的具体要求，以及加强与“实现安全”相关内容。
- 算法类扩展要求：与密码算法相关的特色要求。
- 可信类扩展要求：与可信类产品相关的特色的要求。

所有的产品都需要遵守通用安全要求，而算法类和可信类产品还需要遵守一种相应的扩展要求。

三、隐私计算产品通用安全分级介绍

（一）通用安全分级的安全要求

1.各级别抵御攻击能力概述

第一级（基础防护级）要求抵御非参与方角色（也不能操纵参与方）攻击。第一级中主要考虑外部攻击者从互联网上发起的常见攻击行为，如：1）利用已有扫描工具从公网进行命令执行、SQL 注入、未授权、SSH 弱口令等漏洞的发现和利用；2）对常见传输协议中传输的明文数据进行窥探。

第二级（中等防护级）要求抵御非参与方角色的攻击以及参与方角色的一般水平攻击。相较于第一级，第二级需要增加考虑恶意参与方从内部网络发起的一般水平攻击行为（这也包括外部攻击者已经攻破参与方的外部防护，进入到内部网络发起攻击的情形），如：1）恶意参与方利用已有扫描工具从办公网或生产内网进行命令执行、SQL 注入、通信数据的篡改或窥探，以及对未授权、SSH 弱口令等漏洞的发现和利用；2）恶意参与方利用其拥有的合法身份和权限对数据库中数据、日志文件进行窥探和篡改，或者其他窥探参与方数据的方法；3）恶意参与方伪造可信环境；4）恶意敌手通过简单数学方法攻破算法、篡改本地协议进行攻击；5）同一法律主体以及该法律主体控制的其他主体机构进行的合谋攻击等。6) 基于硬件已有功能进行的攻击，例如拔插没有防拆保护的硬盘。

第三级（安全设计级）要求抵御非参与方角色以及参与方角色的所有已知攻击。相较于第二级，第三级需要增加考虑恶意参与方从内部网络发起的复杂攻击行为，如：1）三方及少于三方的合谋攻击，包括合谋攻击密码算法、合谋绕过审批/认证鉴权/访问控制等安全限制；2）针对算法类产品，所有已知的攻击方法，以及基于专家能力能够组合出来的新的攻击方法；3）

针对可信类产品，开盖、篡改电路板、窃听硬件信号等硬件攻击，以及侧信道攻击、重放攻击等。

第四级（攻防检验级）除了要求抵御上述攻击外，还要求能够对重点隐患进行纵深防御。

第五级（安全证明级）除了要求抵御上述攻击外，还要求能够对重点隐患的防御效果进行形式化验证。对合谋攻击的抵御能力要求升级为“抵御除己方外所有其他方的合谋”。

2.通用安全要求介绍

（1）安全漏洞

安全漏洞指的是系统中存在的因系统设计缺陷、代码不安全实现、配置不当等原因形成的脆弱点，攻击者可以利用这种脆弱点绕过系统正常的功能设计，对系统进行非预期的数据访问、数据窃取、数据破坏或服务拒绝等恶意行为。需要提醒的是，安全漏洞既可能是产品自身产生的，也可能是产品所依赖的一些组件或系统框架带来的，这两部分的安全漏洞都需要关注。另外，在测评后还应该持续关注产品的漏洞情况，如果产品被外部发现漏洞数量过多，导致其不符合当前等级要求，则产品应进入整改期，授予的等级暂时失效，待修复完成并经过相关检查后才能恢复当前等级。

在第一级中，攻击者不是合法参与者，一般位于互联网中（非参与方内网）。从互联网攻入到办公网或内网环境有一定的难度。因此第一级要求产品不应该存在公网可以直接利用的高危安全漏洞。

第二级假设攻击者可能是合法参与者而且具备一般的安全专业能力，可以自己设计攻击方法或通过多个安全漏洞组合进行攻击。因此在这个等级中，

我们扩大了需要修复的漏洞等级范围，要求产品要修复公网和内网可利用的中危以上漏洞。

第三级的产品应该达到理论上没有已知攻击途径的水平，是一个相对比较成熟的安全水位。因此，第三级要求产品修复所有的公网和平台方内可利用的安全漏洞。根据实际安全工作经验，评判产品的安全性不仅要考虑当下产品的安全漏洞情况，还要考虑未来出现漏洞的可能性以及是否能够及时有效的处理。所以，第三级要求产品所使用的组件安全信誉度不能过低，安全信誉度可以根据历史数据来推断，即，根据过去一年这个组件所被爆出的漏洞数量来衡量。未来可能会出现新的漏洞、攻击方法，安全需要持续运营。所以，第三级要求产品要有一个持续的漏洞修复运营机制，新发现的漏洞要能够在一定期限内被有效修复。

第四级要求产品能够抵御“未知攻击”，由于攻击手段是未知的，产品方没办法提前去预见和修复。但是因为产品方的有利位置，在实施恰当的措施后，有可能先于攻击者发现并修复安全漏洞。例如：建立有效的安全运营机制、不断增强安全编码能力、内部定期进行漏洞巡检以及及时获取漏洞情报等。因此，第四级要求产品真正被外部发现的漏洞数量每年不应该超过一个，并且这个漏洞不能被用于突破所有纵深防御措施，造成实质性信息泄露。

第五级同样是要求产品能够抵御“未知攻击”，但要通过“形式化证明”的手段在关键隐患部分证明不存在安全漏洞。

（2）访问控制

第一级由于不考虑参与方作恶的情况，因此安全要求与一般信息系统类似，包括基本的身份认证、授权鉴权以及公网通信链路加密等。身份认证是为了验证参与方的合法身份，避免非参与方仿冒参与方的身份直接获取数据，

最小化的授权和鉴权也是保障了基本的数据隔离。需要提醒的是，这里的授权鉴权不仅包括应用层面，还包括网络层面。公网通信链路加密是为了避免攻击者从传输通道中窃取数据。

第二级要考虑参与方以及平台方作为攻击者的情况。隐私计算产品的特殊性之一为身份认证、授权鉴权、密钥管控系统都可能需要跨域进行（在数据方域外进行），因此要重点考虑平台方作为攻击方，利用其权限和内部网络环境绕过安全机制发起攻击的情况。为了防止这种情况，建议关键的认证、授权鉴权、密钥管控系统在可信的环境中（或者在数据方本地）实现。另外不仅是公网的通信链路需要加密，内网的也同样需要，以避免平台方拿到非己方明文的传输数据。

第三级要防范所有已知攻击途径，在第二级的基础上要重点要考虑平台方（如果有平台方的情况下）和部分参与方合谋作恶的情况。即使平台方和部分参与方合谋，产品的认证、授权鉴权和密钥管控等安全机制也不应该被绕过。从实际攻击难度出发，第三级对于合谋攻击的数量定为不超过 3 方（注意这不是传统拜占庭将军问题中的比例要求）。

第四级要求和第三级保持一致，不做赘述。

第五级在合谋攻击场景上考虑的阈值增强，即使除自己外其他方都是恶意且合谋的情况，产品的认证、授权鉴权和密钥管控等安全机制也不应该被绕过。

（3）镜像及代码安全

第一级进行了基本的安全要求，比如应默认只开放必须要开放的端口、操作系统符合行业内安全基线等。另外产品部署过程中使用的账户和密码、

进程权限等也应该是符合最小化原则的。避免因不安全配置导致外部攻击者直接利用进行安全攻击。

从系统安全角度看，隐私计算产品与传统 web 服务的一个不同点是在隐私计算产品中某个参与方有可能会下发自己编写的算法或代码执行逻辑到另外一方的机器中执行，而第二级起参与方也是可能作恶的，第二级要考虑动态下发的算法或代码是恶意的情况。由于第二级不要求产品能完全做到事前防护，但需要做事后可审计、可追溯。因此第二级要求接收方需要详细记录发送方的身份信息和下发的代码或指令信息以供事后审计追责。

第三级中针对上文提到的动态下发的恶意指令或代码，要求可以做到事前防护。目前，行业内针对于这种攻击常用的防护手段是安全隔离。实践中，可以使用系统级、语言级或沙箱级的安全容器实现安全隔离。要指出的是，这种安全隔离虽然可以防止代码下发方作恶，但对代码执行方作恶（例如平台方窃取数据）一般没有抵御作用。针对后者需要采用其他防御措施，例如密码协议或 TEE。另外第三级要求镜像具备防篡改能力，以减少镜像在供应链环节遭遇的安全风险。

在第四级中，虽然无法预判“未知攻击”，但产品可以增加一些安全检查和隔离措施降低攻击的成功率和减弱攻击后果。事前，建议产品在高风险功能点，如可能注入恶意代码的风险入口做额外的安全校验，如白名单。事中，建议产品对不同参与方发起的计算进程在网络、可访问资源、运行环境等方面进行安全限制，如利用 iptables 规则、数据源帐密隔离等。

第五级要求对系统安全中风险较高的部分从理论上证明安全性，而不是像三级的“没有已知攻击路径”或四级的“基于纵深思路降低未知风险”。

（4）日志安全

第一级对日志相关提出了基本要求，包括应用层和系统层的日志记录和查询，以及日志的保存时间。

第二级不要求抵御高水平的已知攻击，但要求这些攻击发生后具有一定的溯源能力。因此，第二级重点加强了日志方面的要求。包括：1）增加了日志需记录的内容范围，即除了应用层和系统层的日志，还要求记录网络层、主机层、数据层以及算法层的日志，只有各个层面的日志都持久化的有记录，后续才有足够的日志数据以进行完整攻击链路的审计溯源；2）考虑到参与方作恶的可能性，日志以及审计信息的完整性需要额外的技术手段去保证，如可信存证系统。

第三级要求和第二级保持一致，不做赘述。

在第四级中，虽然未知攻击难以预测，但在高风险的功能点上可以基于日志数据和可配置的安全预警规则实现安全预警功能。例如，针对动态代码下发功能，可以通过对日志配置预警规则实现恶意代码下发时发出告警。

第五级要求和第四级保持一致，不做赘述。

（5）数据使用策略

本节主要介绍参与方之间如何协商数据的使用方式，以及确保实际使用不会超过约定的范围。

第一级对数据安全提出了基本要求，比如基本的数据权限隔离等。

第二级要求完整的审计溯源，这个级别中强调了数据使用过程中的日志记录，不仅事前需要各参与方对数据的使用方式进行约定，实际的使用方式和使用范围也需要持久化记录下来，以供事后审计追责使用。

第三级的重点在于即使是平台方和部分参与方合谋的情况下，数据相关的审核、授权鉴权机制不应该被绕过。对于非研发模式，不能有“非约定可泄露信息”泄露。对于研发模式，因程序调试的需求，可能需要暴露更多的明文数据。参与方应提前约定用于调试的数据的范围、量级、是否允许下载等，并且平台方和部分参与方合谋也不能可以违背这些约定。另外，进入研发模式也应经过各方同意。

除合谋阈值外，第四、五级要求跟第三级保持一致。不做赘述。

3.算法类扩展要求

（1）“抵御半诚实敌手”不符合现实要求

在密码协议领域中，半诚实敌手指的是恶意参与方（同时也是攻击者）会严格遵守协议；恶意敌手指的是参与方会篡改自己本地的协议，攻击能力远大于半诚实敌手。现实中，攻击者大多数情况下都是攻击能力更强的恶意敌手。这是因为，攻击者本身就是在做违法行为，他们不会遵守任何假定，他们会进行条件允许的任何攻击，篡改本地协议对于他们来说是十分简单的事情。

仅仅抵御半诚实敌手的密码协议，在实际使用时有可能会被攻破。一些攻击案例也展示了，此类协议面临真实攻击时，可能会被窃取大量原始数据，并且受害者毫无察觉。目前很多研发和产品都是围绕半诚实密码协议展开的，主要原因有：1）半诚实密码协议可以作为研究其他密码协议的前序，在学术研究方面具有意义；2）半诚实密码协议性能更高，一些产品方故意忽略或淡化半诚实协议的安全隐患，在用户不了解真实安全性的情况下，半诚实密码协议更容易取悦用户。

我们认为,不具备审计追溯能力的半诚实协议，在现实中的意义很小。而且这些协议一旦被攻击成功，都无法回溯攻击者的数据窃取或滥用行为。所以，第二级重点增加了审计追溯能力。第二级虽然也假定了攻击者是恶意敌手，但是假设他们只是一般意义的工程人员，对算法的攻击能力有限（不是密码专家）。三级需要抵御来自专家级的攻击。

少量设计者陷入了另一个极端，认为抵御恶意敌手，不仅要保证恶意敌手无法破坏数据的机密性，还要保证结果正确性。实际上，保证结果正确性开销比较大，但是收益一般，因为数据提供者可以通过对数据投毒干扰结果正确性。结果正确性不是隐私计算产品的第一目标，也不是大部分产品具备的能力，所以不作为安全分级的要求。

（2）对合谋攻击的抵御能力要求

合谋攻击指的是多个攻击者合作进行攻击。如果多个攻击者属于同一个机构，它们在机构的胁迫下是很容易合谋的，这种情况的合谋需要在二级进行抵御。此外，如果多个攻击者受同一个机构的控制，例如，隶属于同一集团公司，那么他们也容易在胁迫、或共同利益的驱使下进行合谋，这种情况的合谋需要在二级进行抵御。如果多个攻击者隶属于不同的且相对独立的机构，他们之间要达成合谋，需要进行紧密的合作、协商利益分配等，有较大的暴漏风险，难度超过了一般水平攻击，在二级不需要抵御，但是在三级需要抵御。进一步地，参与合谋的机构越多暴露风险就越大，当参与合谋的机构超过一定数量时，在现实中很难达成，可以认为三级产品也不需要抵御。按照实际攻防经验，4 方及以上独立机构的合谋，在现实中就很难达成。因此，三级只要求抵御 3 个参与方合谋（受同一机构控制的多方算一方）。注

意这不是传统学术研究中拜占庭将军问题中的比例要求，而是更加贴合实际风险的数量要求。

（3）算法评估方面的要求

第三级要求能够抵御“已知攻击”。要想达到该目标，产品方应该对所有的攻击形式、产品所采用的防护措施有全面的掌握。因此，从第三级检测开始，要求产品方能够提供详尽的安全性分析说明，以证实自己的产品能够抵御“已知攻击”。虽然第三级并不要求从数学上证明产品能够抵御所有已知攻击，但是，该“安全性分析”，要足够的详尽、逻辑合理，使得检测人员能够容易的判断出“产品是否能够抵御所有已知攻击”。

在划分“约定可泄露信息”时，有这么一种特殊的情况：一些中间结果的泄露，会造成少量的原始信息的泄露，但泄露的量跟原始数据的规模相比占比并不大。如果把这些中间结果划为“约定可泄露信息”，即在第三级的时候允许泄露这些信息，与第三级给用户的整体印象—“没有已知攻击”有出入；但如果将这些中间结果划为“非约定可泄露信息”，即在第三级的时候不允许泄露这些信息，这些泄露造成的后果并不算严重。如何对这种情况进行规定，是有一定的弹性空间的。第三级的倾向是“在安全性变化不大的情况下，允许产品追求更高的性能”。所以分级标准的规定为：第三级允许这类中间结果的泄露，但要求将相关隐患告知数据方并获得数据方的显式确认。“显式确认”指的是数据方通过点击按钮等人工的方式进行显性确认，确保数据方知晓此事。

（4）对纵深防御机制的要求

第四级要求产品具备纵深防御能力。算法类产品，一般被部署在各个数据提供方内部。各个数据提供方的信息系统一般都具备外围防御体系（防火

墙、入侵检测等）。这些防御措施也可以被视为一种纵深防御措施，能够对外部攻击者起到防御效果。但是，有一类隐患这些防御措施是保护不到的。即，数据方在密码协议交互中，主动泄露给其他方的内容（比如为了提升性能等）。第三级要求只允许泄露“约定可泄露信息”。“约定可泄露信息”是在当前的技术水平下，无法被攻击者利用。但是仍然有一定隐患：攻击者如果掌握了更先进的攻击技术，有可能会利用这些信息。因此，算法类产品的纵深防御机制主要是防备“从约定可泄露信息反推有价值信息”。本分级标准中提议了三种可行的措施，产品方可以选择其中一种：

- 中间结果的泄露量本身就非常少，且为常数级。这种情况，自然隐患也非常低。
- 算法能够检测并阻断恶意攻击行为。这种情况下，攻击者一旦篡改本地协议，数据方就会中止协议。攻击者的攻击窗口就会非常小，能够造成的危害十分有限。
- 通过可信技术防止或“检测并阻断”其他参与方的恶意行为。与上一点的原理类似，不过是通过可信技术来实现的。要指出的是，此时，可信技术是用来防备恶意参与方的，即需要保证即使参与方作恶，也不能关闭或者绕过可信技术的防御。

（5）对形式化验证的要求

密码协议背后的数学原理十分复杂，依靠人工审查有可能会出现考虑不周的情况，因此，第五级要求对密码协议进行形式化验证。

在密码协议工程实现的过程中，出于性能和实现复杂度的考虑，研发人员有可能会省略部分密码协议过程，并通过恰当的修改使得协议仍能自洽。麻烦的是，一般的测试往往只能覆盖到用户可见的功能，对内部安全性的变

化很难发现。因此，第五级产品要求对“算法实现和设计”是否一致进行形式化验证。

4.可信类扩展要求

可信类隐私计算产品指的是通过提供一个安全的隔离运行环境，来达到隐私计算目标的产品。典型的技术有可信执行环境（TEE）、可信计算（TPM、TPCM）等。此外，一些一体机、板卡以特殊的外壳设计构建隔离环境，具有类似的安全能力，也属于可信类产品。为了方便统一表述，后文以可信环境代指上述技术提供的隔离运行环境，以可信应用代指运行在可信环境中的应用。

（1）硬件可信根

在可信类隐私计算产品中，可信环境的目标是抵御参与方作恶，即系统运维管理人员的攻击。在已经完全掌控操作系统权限的情况下，系统运维管理人员攻破基于软件的防护措施难度不高。所以从二级起，就要求可信类隐私计算产品具备硬件级的防护措施，包括硬件可信根。

硬件可信根可以管理和保护根密钥，提供基于根密钥的密钥派生、设备身份识别、通信加密等功能。用户可以通过带外的方式（例如权威机构担保）与硬件可信根建立互信。大部分的可信类产品都可以对系统（软硬件）进行度量，并由可信根以可靠的方式（例如数字签名）将度量值传递给用户，即下一节中的远程认证。用户、权威机构、可信根、可信应用构成了一个信任链，用户依赖这个信任链相信可信应用。第二级要求，这个信任链在安全性上是完备的，且即便系统运维管理方是恶意的，也无法破坏或伪造上述信任链。要达到上述目标，至少要保证，系统管理员不能导出、篡改或任意调用根密钥。尤其是一些产品对外提供可信根的应用接口，更要注意这一点。

硬件可信根可以基于芯片、固件等形态实现，这些形态可以将其与用户进程隔离，从而防范恶意进程的攻击。业界针对芯片等有专门的安全评测标准，全面考虑了侵入式、半侵入式、非侵入式等多种攻击形式。例如，面向安全芯片的《GM/T 0008-2012 安全芯片密码检测标准》，面向通用芯片 CPU（搭载固件可信根）的 3C 认证证书、《GM/T 0028-2014 密码模块标准》等。产品方可以通过相应的测评报告，说明产品对这些攻击的抵御能力。

（2）远程证明

可信类应用通过远程证明技术，允许一个参与方向执行计算任务的另一个参与方发送挑战，要求对方对计算平台上的代码等资产进行完整性度量，对度量结果进行签名。挑战发起方通过验证签名，可以确认平台上的代码等内容是否被恶意篡改。计算执行平台如果存在恶意管理员，可能在植入恶意程序后通过窃取密钥等方式来伪造签名信息，因此需要通过硬件可信根等对密钥进行保护。

在系统运维管理人员潜在恶意的情况下，如果可信应用的代码被篡改了，安全性很难保障。所以，从第二级开始，要求能够远程验证代码未被篡改。第三级要求更为完善，要求能够通过远程认证对可信应用的身份进行确认，例如，用户可以获得可信应用的度量值等更详细的信息，进而更好地判断可信应用的合法性。第三级还要求，用户可以基于上述可信应用身份与可信应用建立端到端的安全传输信道，避免传输过程被管理员窥探。

（3）隔离要求

隔离的运行环境是抵御运维管理人员窥探的主要措施，所以，从第二级开始，就要求可信类产品提供硬件的隔离措施，以抵御来自运维管理人员的窥探或篡改。这里的“硬件隔离措施”，指的措施本身包含硬件机制，能够抵

御软件层面的攻击,即,即便攻击者掌握了操作系统等软件的最高运维权限,也无法打破该隔离性。

考虑到攻击者可能会以物理的方式破坏隔离性,并对其中残余的信息进行分析。硬盘等静态存储中的信息容易被这种方式攻击。因此,第二级要求可信环境内存储的敏感信息要进行加密存储。内存等易失性存储,通过物理攻击窃取其中的信息有一定难度,所以二级不要求对内存中的数据进行加密(但仍要具备硬件隔离措施)。

三级要求对可信环境的内存进行保护,以防止攻击者将内存替换成有恶意输出接口的内存,或者冷启动攻击等。在防护形式方面,三级允许产品采用内存加密的方式,或采用可自证安全的防拆机制(外壳),两种选择一种即可。

在目前的技术水平下,外壳防拆机制的防御效果仍显著低于单芯片防御方案。所以对于等级更高的四级,要求可信环境的内存必须进行加密。

(4)交互安全

可信环境主要是为可信应用提供一个隔离的环境,防止从外向内的攻击。但是,可信应用如果自身有安全问题,主动向外泄露数据,可信环境一般不能防御。可信应用除非漏洞似乎没有理由产生这样的行为。但是,大部分的软件模块并不是为可信应用研发的,它们可能会往日志、其他模块发送敏感数据。可信应用在引入这些模块时,如果未进行良好的检查,或者研发人员未重视这类情况,可能会导致上述隐患。

这种泄露一旦发生,很容易被外部的攻击者利用。所以,从第二级开始,就要求“可信应用程序设计层面应无主动数据泄露逻辑,包括内存泄露、外部存储泄露、日志泄露、网络传输泄露等”。

同时，第二级也要求可信应用对与外部的交互进行严格检查，避免外部通过接口交互攻入可信应用内部。具体包括输入参数合法性检查，如：数据指针必须指向普通内存；以及必要的访问权限控制，如：部分可信接口仅允许可信环境内部调用，部分可信接口仅支持从可信环境外部调用等。

（5）防重放攻击

可信环境一般是信息系统的一个组件，系统运维管理人员一般是拥有调用可信环境的权力。如果系统运维管理人员是恶意的，他就有可能重复调用可信环境。因为整个外围环境都是他运维的，他甚至可能无限次的调用。这种攻击形式，一方面，会突破数据方原本设定的调用次数限制；另一方面，可能会将单次调用的泄露进行放大，通过蚂蚁搬家的方式逐步窃取数据方的数据。

重放攻击的发现和实施具有一定难度，所以在二级没有相关要求。重放攻击的防御成本较高，所以三级未强制要求实施重放攻击防御机制，但是要求“不能有重放攻击可以造成非约定信息的泄露”。在未实施重放攻击防御机制的情况下，产品仍然有相关隐患，所以四级要求关键流程具备防重放机制。

（6）防侧信道攻击

侧信道攻击是指通过采集安全运算在执行时的泄漏特征，反推出敏感值的攻击方式。依照侧信道泄漏源分类，通常可以分为硬件侧信道和软件侧信道。其中软件侧信道一般包括对内存读写数据、内存访问特征、网络数据包特征等进行分析，通常可以通过恶意进程软件、管理员监控软件进行采集。硬件侧信道一般包括对芯片功耗、电磁辐射、散热特征、执行时间等物理性质进行分析，通常通过传感器、探头、示波器等物理设备进行采集，在攻击

之前通常还需要对目标设备进行开盖、磨片、逆向等操作，对于采集到的物理信号也需要进行预处理和数学分析。

在系统运维管理人员是恶意的情况下，他容易采集到可信环境向外泄露的信息。所以侧信道攻击也是可信类产品重点考虑的攻击形式之一。但是因为侧信道的分析和实施门槛较高，所以二级不做要求。

三级要求能够抵御典型的侧信道攻击。例如，无法从时间、功耗曲线中分析出敏感信息。如果产品凭借良好的设计习惯、系统噪声，能够满足上述要求，也可以不实施专门的侧信道防护方案。

四级要求实施专门的侧信道防护方案，在设计层面保证可以大概率抵御侧信道攻击。例如，尽量让可信应用在不同的输入下保持相同的执行流，以免从执行流反推出数据信息；对于会泄露访问地址的可信环境，要避免可信应用的地址访问模式与敏感信息相关（例如利用敏感信息查找预计算表）；安全芯片可以通过增加特定噪音、扰乱系统时钟或增加掩码，使得攻击者难以获得目标模块真实的运行时间、功耗等。

（7）供应链安全

可信应用软件一旦有漏洞，就有可能会导致数据泄露，所以，对其安全性、软件质量有着更高的要求。第三级要求可信应用要有完善的供应链管理。包括只使用有可靠来源的软件组件，对采购、交付、运维、废止等环节进行记录和管理。另外，产品方应具备软件技术风险管理能力，包括但不限于漏洞检测分析能力与漏洞修复能力、漏洞扫描工具、软件部署前的完整性校验、软件防篡改机制保护等。

为了能够更好的对软件进行溯源，第三级要求可信应用程序通过签名机制担保其来源。

（8）封装和解封

可信环境中的封装（Sealing）和解封（Unsealing）是一种安全存储机制。封装是一种将数据与特定可信环境（包括环境内运行的可信应用）相关联的过程，只有该环境才能解封这些数据。该机制可以更加方便、安全的支持数据的持久化，第三级产品宜具备该机制。

这一机制通常依赖于平台的度量值来实现。以下是封装和解封过程的一般描述：

- 封装：1）将敏感数据与可信环境特定的属性（如 PCR 值、CPU 安全版本号、硬件唯一密钥等）绑定；2）封装后的数据被称为“密封数据”，只有当可信环境的特定的属性与封装时相匹配时，才能解封这些数据。
- 解封：1）解封是将之前封装的数据解密回原始形式的过程；2）为了成功解封数据，可信环境必须达到与封装时相同的安全属性（大多数情况下意味着执行代码与封装时相同），如相同的 PCR 值、CPU 安全版本号、硬件唯一密钥；3) 解封过程需要使用与封装时相同的密钥或密码学材料，这通常存储在可信环境的安全存储中。

（9）日志安全

通用安全虽然对日志进行了详尽的要求，但，可信类产品还有一些特殊的要求。一个是可信应用程序应有主动日志输出，记录关键计算行为和关键资源访问情况等（但不能输出敏感信息）。另一个是对可信应用与非可信应用的交互进行记录。这些都是与安全性有较大关系的信息，一旦发生泄露事件，可以通过这些信息回溯攻击过程。因为审计追溯是第二级的主要技术特征，所以这部分是从第二级开始要求的。

（10）环境安全和纵深防御

第四级要求产品具备纵深防御，以应对未知攻击。很多针对可信类产品的攻击，攻击者首先攻占操作系统，然后再进一步攻击可信环境。所以第四级要求可信环境所在的操作系统等软硬件环境，要具备启动链验证，以防止被篡改。要指出的是，可信环境内部的防篡改要求，在第二级的远程认证中就已经提出了，这里指的是可信环境所在的主机系统。

但是，启动链验证不足以应对运行时的漏洞、侧信道攻击等。所以第四级产品还需要实施更多的纵深防御措施。这里，建议可以从以下两种措施中选择一种：

- 硬件防护:外壳具备防拆保护并且在遭遇入侵时能够自动销毁敏感信息。这种方式能够大幅减少攻击面，进而提升安全性。要指出的是，如果可信环境本身就是依靠外壳防拆来构建的，这一层外壳不能作为满足纵深防御要求的措施。
- 算法防护：通过密码学机制，使得单一可信环境被攻破后，不会有敏感信息泄露。这种情况下，可信环境内不能出现明文的敏感信息，需要使用 MPC、HE 等算法进行加密后再传入可信环境。

（二）约定可泄露信息

1.约定可泄露信息分析流程概述

在隐私计算过程中，各参与方之间会传递一些中间结果。这些中间结果，可能会暴露原始数据的一些信息。然而，不同隐私计算算法所泄露的信息量有大有小，泄露信息的价值有高有低。因此，为了对隐私计算算法的安全性进行细粒度的分级，我们对信息泄露情况进行更加深入的分析，明确算法在各种威胁场景下，会泄露哪些中间结果、泄露的程度如何；这些泄露会暴露

原始数据的哪些信息、是否会引发严重的利益损失或安全风险。如果分析认为算法泄露的信息量不多或泄露的信息价值不大，我们同样可以认为该算法满足对应的分级要求。但是，这些信息暴露情况需要得到数据方的显式确认，称为约定可泄露信息。

我们的评估框架主要分为三个阶段：首先，我们分析面对各个分级的安全威胁时，各个参与方对外发送的信息是否会发生泄露。此时泄露的信息称为直接中间结果。接下来，我们分析结合多个直接中间结果，能否推导出更多更明确的中间结果，此时泄露的信息称为间接中间结果。然后，我们分析直接中间结果和间接中间结果的泄露，会造成多少原始数据信息量损失。从而决定隐私计算产品是否满足对应分级的要求。通过这套分析流程，我们就可以识别出泄露信息较高的不安全协议，也可以分析出满足分级的协议中，包含有哪些约定可泄露信息。

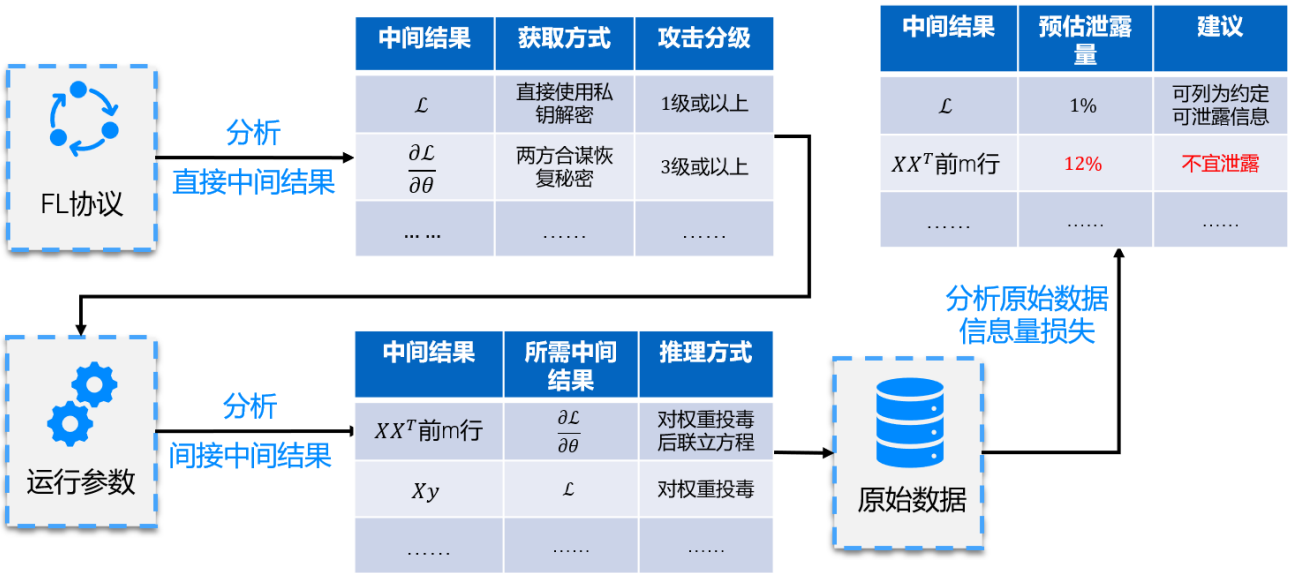

图 2 约定可泄露信息分析流程

2.直接中间结果泄露分析

直接中间结果是参与方在隐私计算的过程中，直接对外发送的数据信息。虽然隐私计算产品往往会保护这些中间结果在传输过程中的机密性，但当接收者本身也是攻击者的时候，自然能够获得这些中间结果。因此，我们首先根据各个安全分级对应的威胁场景，定义攻击者所拥有的能力，并列出一些典型的直接中间结果泄露。

第二级安全标准要求抵抗参与方发起的简单攻击。因此，在这一级别的分析中，应额外考虑攻击者作为一个计算方、协调方或平台方，在保证训练正常进行的前提下对少量参数或流程进行简单修改时，能够获取的直接中间结果。这通常包括根据配置文件、标志位等控制条件，选择性发送的直接中间结果。

第三级安全标准要求算法能够抵抗更高级别的主动攻击，同时还允许参与者的合谋。因此在这一级别的分析中，应额外考虑攻击者控制多个计算方、协调方或平台方，大幅度修改算法或破坏模型训练性能，发动较高水平的已知攻击时能够获取的直接中间结果。具体来说，当参与方数量 n<5 时，攻击者可以控制其它 n-1 个参与方；当参与方 n>=5 时，攻击者可以控制任意 3 个参与方。这一级别下，攻击者可获取的直接中间结果通常包括可以通过合谋解密的中间结果、注入伪造数据能够得到的中间结果等。

第五级的敌手可以和所有可能的参与方合谋（除被攻击方以外的所有其他方）。此外，第五级不允许中间结果泄露，除非能够从数学上证明没有安全隐患（例如，攻击难度等价于数学难题）。

示例：对于如图 3 所示的纵向联邦线性回归算法，其直接中间结果及对应的泄露等级如图 4 所示。

	计算方 A	计算方 B	平台方 C
第 1 步	计算 $\boldsymbol{u}_{(t)}^{A}, \mathcal{L}_{(t)}^{A}$。向 B 发送 $[[\boldsymbol{u}_{(t)}^{A}]]_C, [[\mathcal{L}_{(t)}^{A}]]_C$	计算 $\boldsymbol{u}_{(t)}^{B}, [[\boldsymbol{d}_{(t)}]]_C$。向 A 发送 $[[\boldsymbol{d}_{(t)}]]_C$，向 C 发送 $[[\mathcal{L}_{(t)}]]_C$	C 解密 $[[\mathcal{L}_{(t)}]]_C$。如果达到中止条件则告知 A、B
第 2 步	计算 $[[\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{A}}]]_C$。生成随机掩码 $\boldsymbol{R}_{(t)}^{A}$。向 C 发送 $[[\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{A}} + \boldsymbol{R}_{(t)}^{A}]]_C$	计算 $[[\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{B}}]]_C$。生成随机掩码 $\boldsymbol{R}_{(t)}^{B}$。向 C 发送 $[[\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{(t)}^{B}} + \boldsymbol{R}_{(t)}^{B}]]_C$	C 解密 $[[\cdot]]_C$。向 A 发送 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{A}} + \boldsymbol{R}_{(t)}^{A}$。向 B 发送 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{B}} + \boldsymbol{R}_{(t)}^{B}$
第 3 步	根据 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{A}}$ 计算 $\boldsymbol{w}_{(t+1)}^{A}$	根据 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{B}}$ 计算 $\boldsymbol{w}_{(t+1)}^{B}$	

图 3 纵向联邦线性回归第 t 步迭代

直接中间结果	获取方式	攻击分级
$\mathcal{L}_{(t)}$	C 直接解密	2+
$\boldsymbol{u}_{(t)}^{A}$	B 与 C 共谋，或	3+
$\mathcal{L}_{(t)}^{A}$	B 修改协议第 2 步，借助 C 解密（影响模型训练）	3+
$\boldsymbol{d}_{(t)}$	A 与 C 共谋，或 A 修改协议第 2 步，借助 C 解密（影响模型训练）	3+ 3+
$\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{A}}$	传输时被掩码保护，其它参与方永远无法获取	
$\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}_{(t)}^{B}}$		

图 4 纵向联邦线性回归第 t 步迭代直接中间结果分析

3.间接中间结果泄露分析

敌手根据若干泄露的直接中间结果，可能间接地推测出更多中间结果。更进一步，如果敌手拥有偏离协议的能力，可以通过恶意行为干扰直接中间结果的计算过程，使其泄露更多信息。

在第二级中，由于敌手可以发动简单攻击，因此可以通过修改参数、简单修改流程等手段，影响直接中间结果的计算过程，使间接中间结果更加清晰。

在第三级及以上，敌手可以发动较为复杂的攻击，对直接中间结果的计

算进行投毒，并使用大量算力和各种已知攻击推演间接中间结果。

示例：在图 5 所示的横向联邦线性回归算法中，如果攻击者修改协议参数，将初始全局模型设为$\boldsymbol{w}=\boldsymbol{0}\in\mathbb{R}^{m\times1}$（简单攻击，2 级或以上），就可以从直接中间结果$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}}$中获得间接中间结果$\boldsymbol{X}^T\boldsymbol{y}\in\mathbb{R}^{m\times1}$。将初始全局模型设为$\boldsymbol{w}=\boldsymbol{c}$（简单攻击，2 级或以上），并结合间接中间结果$\boldsymbol{X}^T\boldsymbol{y}$，就可以获得间接中间结果$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}\in\mathbb{R}^{m\times1}$。当$\boldsymbol{c}_1,\ldots,\boldsymbol{c}_m$线性无关时，联立$m$个间接中间结果$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}_1,\ldots,\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}_m$，可以解出间接中间结果$\boldsymbol{X}^T\boldsymbol{X}\in\mathbb{R}^{m\times m}$。一种$\boldsymbol{c}_1,\ldots,\boldsymbol{c}_m$的选取方式是$\boldsymbol{c}_i=\boldsymbol{e}_i=(0,\ldots,0,1,0,\ldots,0)\in\mathbb{R}^{m\times1}$（简单攻击，2 级或以上），则对应的间接中间结果为$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{e}_i\in\mathbb{R}^{m\times1}$，即$\boldsymbol{X}^T\boldsymbol{X}\in\mathbb{R}^{m\times m}$的第$i$列。

因此，当该算法面对二级或以上的威胁场景时，运行第 1 轮后，泄露的间接中间结果是$\boldsymbol{X}^T\boldsymbol{y}$；运行第$1<t\leq m$轮后，泄露的间接中间结果是$\boldsymbol{X}^T\boldsymbol{y}$和$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}_t$（$\boldsymbol{c}_t\in\mathbb{R}^{m\times1}$由敌手任意指定）；运行第$m+1$轮后，泄露的间接中间结果是$\boldsymbol{X}^T\boldsymbol{y}$和$\boldsymbol{X}^T\boldsymbol{X}$。

Algorithm 1 客户端 A FedSGD 第 t 轮迭代

从服务器接收初始全局模型 w

计算梯度 $\frac{\partial\mathcal{L}}{\partial w}(\boldsymbol{X},\boldsymbol{y},\boldsymbol{w})=\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}-\boldsymbol{X}^T\boldsymbol{y}$

将梯度 $\frac{\partial\mathcal{L}}{\partial w}$ 上传到服务器

图 5 FedSGD 迭代算法

4.原始数据信息量损失度量

在进行安全分级时，我们不仅需要考虑隐私计算过程中泄露了哪些中间结果，还应该考虑这些泄露的中间结果对原始数据的信息量损失程度。这样，才能更全面地决定这些中间变量是否可以作为“约定可泄露信息”。为此，我们提供两种量化分析的思路：基于自由度的量化分析和基于条件熵的量化分析。

5.基于自由度的量化分析

在攻击者不知道原始数据的任何信息时，原始数据的取值对攻击者来说是完全自由的。然而，由于攻击者从隐私计算的过程中获取到了一些泄露的中间变量，所以原始数据可能的取值就会受到额外约束。因此，我们将原始数据中的自由变量个数定义为自由度，并用来量化原始数据的信息量变化：通过分析中间变量泄露前的自由度DoF_0和中间变量泄露后的自由度DoF_1，就可以量化出中间变量泄露后对原始数据的造成的信息量损失为$r = 1 - \frac{\mathrm{DoF_1}}{\mathrm{DoF_0}}$。更进一步，我们还可以从几何意义等角度出发，为泄露的原始数据信息提供明确的语义解释。

示例：若原始数据为 n 个 m 维向量$\boldsymbol{X} \in \mathbb{R}^{n \times m}$，则原始数据有$n * m$个自由变量。如果协议泄露的中间变量为$\boldsymbol{Xc}$，其中$\boldsymbol{c} \neq \boldsymbol{0} \in \mathbb{R}^{m \times 1}$为一常数向量，则原始数据在$\boldsymbol{Xc}$的约束下，有$n$个变量无法再自由变换，即自由变量降至$nm - n$。因此，中间变量$\boldsymbol{Xc}$引发的$\boldsymbol{X}$的泄露为$1 - \frac{\mathrm{nm-n}}{\mathrm{nm}} = \frac{1}{\mathrm{m}}$。

我们使用常见的隐私计算场景数据规模做分析：对于$m = 10$维的原始数据$\boldsymbol{X}$，中间变量$\boldsymbol{Xc}$对原始数据的泄露程度为 10%，对于$m = 100$维的原始数据$\boldsymbol{X}$，中间变量$\boldsymbol{Xc}$对原始数据的泄露程度为 1%，即特征数量越多，该中间变量对原始数据的泄露比越小。还可以从几何意义出发，将此约定可泄露信息的语义解释为：$\boldsymbol{Xc}$泄露了原始的n个向量中，每个向量$\boldsymbol{x}_i$与常数向量$\boldsymbol{c}$的内积信息。

如果同时有多个中间变量$\boldsymbol{Xc}_j$泄露，则泄露的总量为所有中间变量泄露的总和。假设最后泄露的总量小于一定阈值，例如 10%，则可以将这些中间变量视为“约定可泄露信息”，并向数据方明示此泄露的具体危害。

6.基于熵的量化分析

基于自由度的量化分析主要用于数据分布尚不明确时的分析。对于直接拥有数据的算法使用者，可以结合自己的实际数据，分析约定可泄露信息对原始数据的熵损。在这里，我们使用中间变量的条件熵作为量化指标：计算原始数据的熵为$H(X)$，已知中间变量泄露后的条件熵为$H(X|Y)$，则熵损为$\Delta H = H(X) - H(X|Y)$，泄露比为$r = \frac{\Delta H}{H(X)}$。

例：某隐私计算协议的一个中间变量为每个数据向量的模长。

对于某金融数据集，计算原数据集的熵$H(X) = 20bit$和已知模长时该数据集的条件熵$H(X|Y) = 13bit$。则此中间变量对数据集的熵损为$\Delta H = H(X) - H(X|Y) = 7bit$，泄露比为$r = \frac{\Delta H}{H(X)} = 35\%$，较为严重，故此应用场景下，该中间变量需加以保护，不宜作为约定可泄露信息。

对于某人脸向量数据集（向量已归一化到单位模长），计算原数据集的熵$H(X) = 32bit$和已知模长时该数据集的条件熵$H(X|Y) = 32bit$。则此中间变量对数据集的熵损为$\Delta H = H(X) - H(X|Y) = 0bit$，泄露比为$r = \frac{\Delta H}{H(X)} = 0\%$，故此应用场景下，该中间变量可以作为约定可泄露信息。

四、隐私计算产品通用安全分级的场景应用

（一）隐私计算典型应用场景与发展情况

1.金融领域

金融领域涉及大量敏感和个人化的客户数据，金融机构客户数据隐私保护要求也越来越高。其次，金融机构也面临更为严格的合规性要求，如 GDPR（通用数据保护条例）、反洗钱（AML）等规定，要求对个人数据进行适当的保护，并限制数据的跨境传输和处理。同时，金融行业中更多的机构和部门需要共享数据以完成风险评估、反欺诈和客户洞察等任务，然而高价值的数据共享往往涉及数据安全和隐私泄露风险。隐私计算允许不同金融机构之间在保持数据加密的同时进行计算和分析，多方可以同时参与计算，解决了数据共享和协作的痛点问题。同时，通过使用隐私计算技术可以在不直接暴露客户数据的情况下进行风险评估、客户画像分析等操作，使得金融机构可以在满足合规性要求的同时进行数据价值开发利用。此外，借助隐私计算技术，金融机构可以与监管机构共享有关客户的信息，同时保护客户隐私，这使得金融机构能够更好地理解客户需求、改进风险管理，并提供个性化的金融服务。

2.通信领域

在通信领域中，运营商拥有海量数据，具有开展隐私计算业务的数据资源，隐私计算在运营商应用也已逐步推动。基于当前数字化转型的趋势以及数据安全管控要求，各地方运营商迫切希望基于安全合规的数据智能技术赋能业务推广，如基于横向联邦模式集合多个地方运营商数据推广产品套餐，解决单一地方运营商模型数据样本不足的问题；同时运营商与公安部门拥有的数据大多为个人数据或敏感数据，利用隐私求交技术完成电诈黑名单的更

新，结合运营商的反诈预警机制予以拦截或干预。基于运营商的海量数据，通信行业常以数据提供方为其他行业客户提供数据服务。由于隐私计算行业平台所用的框架和协议各有差异，目前不同厂商、不同技术选型的平台还无实质性的互联互通能力。因此隐私计算业务的开展通常会在运营商侧部署各类隐私计算平台，以满足不同客户、不同厂商以及不同业务场景的需求。

3.教育领域

在教育领域中，教育机构及相关研究部门在校园管理、课堂建设和科学研究的过程中往往需要采集和管理大量带有个人敏感数据的信息。这些信息包含各级教育单位管理过程的私密数据，蕴含学术经济价值的科学实验数据，还涉及多人的隐私信息，如学生家庭信息、学业成绩、就业和财务状况等。教育机构或研究单位在进行系统决策和学术研究时往往需要联合不同的数据源，数据源的联合涉及数据的隐私与安全问题，数据面临泄露风险。针对这些问题，通过隐私计算技术可以提供一种安全、高效的数据联合分析方案，在保护学生和教职工隐私信息前提下，为教育部门及相关研究部门提供更加可靠、安全的数据管理和服务。

（二）应用场景安全等级建议

1.场景对应安全分级的主要影响因素

一是合作机构间的信任度。隐私计算涉及多机构的相互协同，不同类型机构之间的信任度也有所不同。一方面，合作机构间的关系对信任度有着较大影响，例如，集团公司内部、联盟组织内部之间往往具备较强的信任基础。另一方面，合作机构自身的单位属性、人员规模、信誉声誉、专业能力、财务稳定性等因素也会影响信任度。在开展数据流通利用合作前，应对相关机构的信任度进行初步评估，选择适用的安全等级，对于信任度较低的机构间

合作需要采用更加严格的安全防控手段。通常，参考上述相关维度，可以将合作机构间的信任度划分为高度信任、一般信任、无信任基础 3 个级别。

二是流通数据的重要程度。在国家标准 GB/T43697-2024 中，根据数据遭到泄露、篡改、损毁或非法获取、非法使用、非法共享，对国家安全、经济运行、社会秩序、公共利益、组织权益、个人权益造成的危害程度，将数据分为了核心数据、重要数据、一般数据（国标话语体系中，一般数据有四种可供参考的分级方法，本文参考其中第一种分四级的分级方法）。当前阶段，隐私计算产品的应用场景大多是面向企业间数据流通利用，范围和数据规模相对较小，涉及的数据等级主要集中在一般数据，仅部分少数场景可能涉及重要数据范畴。因此，根据国标建议，基于风险影响对隐私计算可能涉及的流通数据细化拆分为 4 个级别，分别是一般数据的 2、3、4 级以及重要数据，以适配特定应用场景下的产品安全等级选择（不适用于一般数据 1 级和核心数据）。

一般数据（一般危害），此类数据的相关风险会对个人权益、组织权益造成一般危害。组织权益危害方面，包括导致个别诉讼事件、或在某一时间造成部分业务中断，使组织的经济利益、声誉、技术等轻微受损。个人权益危害方面，包括导致个人信息主体可克服的困扰，如付出额外成本、无法使用应提供的服务等。

一般数据（严重危害），此类数据的相关风险会对个人权益、组织权益造成严重危害。组织权益危害方面，包括导致组织遭到监管处罚，或影响部分业务无法正常开展，造成较大经济或技术损失，破坏组织声誉。个人权益危害方面，对个人信息主体产生较大影响，克服难度高，消除影响代价较大，如遭受诈骗、资金被盗用、信用名誉受损等。

一般数据（特别严重危害），此类数据的相关风险会对个人权益、组织权益造成特别严重危害，或对经济运行、社会秩序、公共利益造成一般危害。组织权益危害方面，包括导致组织遭受监管严重处罚，或影响关键业务无法开展，造成重大经济或技术损失，严重破坏机构声誉等。个人权益危害方面，对个人信息主体造成重大、不可消除、无法克服的影响，如遭受无法承担的债务、失去工作能力等。

重要数据，指特定领域、特定群体、特定区域或达到一定精度和规模的，一旦被泄露或篡改、损毁，可能直接危害国家安全、经济运行、社会稳定、公共健康和安全的数据。仅影响组织自身或公民个体的数据一般不作为重要数据。

2.场景对应安全分级的建议

以场景涉及合作机构间的信任度、待流通数据的重要程度作为主要参考维度，我们建议满足不同级别的产品使用的业务场景如下。

	一般数据（一般危害）	**一般数据（严重危害）**	**一般数据（特别严重危害）**	**重要数据**
高度信任	第一级	第一级	第二级 *可根据信任度酌情降级	第三级 *可根据信任度酌情降级
一般信任	第一级	第二级	第三级	第四级
无信任基础	第二级	第三级	第四级	第五级

五、总结与展望

隐私计算作为数据流通的基础核心技术，在促进数据要素流通上发挥着重要作用。由于隐私计算技术路线的多样性且彼此之间的安全特征相差巨大，此前的隐私计算分级标准均是在特定技术路线下提出的，但随着多技术路线融合产品的不断涌现，产品形态的不断丰富以及同一应用场景的多技术路线解决方案的出现，在没有统一的安全基准前提下，使得产品使用方在进行产品技术选型时难以在众多隐私计算产品之间进行安全性的横向比较，而更加关注性能指标上的差距，影响了隐私计算的可信应用和良性发展。2024 年 2 月深圳国家金融科技测评中心牵头，联合 8 家机构，发布了 Q/NFEC 0001—2024《隐私计算产品安全能力分级要求》标准，首次以攻防效果作为不同技术路线、不同产品形态的统一安全考量维度，从算法、硬件、漏洞、访问控制、镜像及代码等方面保障产品的整体安全，并通过引入攻击水平、约定可泄露信息、软件信誉度等概念，进一步量化安全等级，结合不同应用场景的安全需求，构建出了通用的安全分级基准，便于产品开发方和使用方根据实际应用场景的安全需求进行设计和选型，达到安全和性能之间的平衡，为隐私计算的有序发展提供重要支撑。

附录：约定可泄露信息分析参考示例

（一）联邦线性回归信息泄露分析

1 预备知识

1.1 线性回归 [1]

给定由 n 个样本组成的数据集 $D=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^n$。$D$ 中的每个样本 $(\boldsymbol{x}_i,y_i)\in D$ 由 m 个属性值 $\boldsymbol{x}_i\in\mathbb{R}^{m\times 1}$ 和 1 个观测值 $y_i\in\mathbb{R}$ 组成。将所有样本的属性值和观测值堆叠，记为属性矩阵 $\boldsymbol{X}=(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)^T\in\mathbb{R}^{n\times m}$ 和观测值向量 $\boldsymbol{y}=(y_1,\dots,y_n)^T\in\mathbb{R}^{n\times 1}$。

线性回归将属性值与观测值之间建模为线性关系 $\hat{y}=f_{\boldsymbol{w}}(\boldsymbol{x})=\boldsymbol{x}^T\boldsymbol{w}$，并基于数据集 D 估计最优权重 $\boldsymbol{w}^*\in\mathbb{R}^{m\times 1}$，使得模型预测值向量 $\hat{\boldsymbol{y}}=\boldsymbol{X}\boldsymbol{w}$ 与观测值向量 $\boldsymbol{y}$ 的损失函数 $\mathcal{L}$ 最小，即

$$\boldsymbol{w}^*=\arg\min_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w},D)$$

如果取均方误差作为损失函数，则可定义 $\mathcal{L}(\boldsymbol{w},D)=\sum_{i=1}^n\frac{1}{2}(y_i-f_{\boldsymbol{w}}(\boldsymbol{x}_i))^2=\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})$。损失函数 $\mathcal{L}$ 关于权重 $\boldsymbol{w}$ 的导数为

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}}(\boldsymbol{w},D)=\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}-\boldsymbol{X}^T\boldsymbol{y}\in\mathbb{R}^{m\times 1}$$

当 $\boldsymbol{X}^T\boldsymbol{X}\in\mathbb{R}^{m\times m}$ 可逆时，可根据 $\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}}=\boldsymbol{0}$，使用解析法求得最优权重为 $\boldsymbol{w}^*=(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$。

在实际训练时，最优权重 $\boldsymbol{w}^*$ 常使用梯度下降法求得：首先选择初始权重 $\boldsymbol{w}_{(0)}$，接下来，反复选取一批 b 个样本 $D_{\text{batch}}=(\boldsymbol{X}_{\text{batch}},\boldsymbol{y}_{\text{batch}})\in\mathbb{R}^{b\times m}\times\mathbb{R}^{b\times 1}\overset{\$}{\leftarrow}D$，并基于这批数据将权重从 $\boldsymbol{w}_{(t)}$ 更新为

$$\boldsymbol{w}_{(t+1)}=\boldsymbol{w}_{(t)}-\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}}(\boldsymbol{w}_{(t)},D_{\text{batch}})$$

直到收敛条件 $|\mathcal{L}_{(t+1)}-\mathcal{L}_{(t)}|<\epsilon$ 满足时停止迭代，得到最优值 $\boldsymbol{w}^*=\boldsymbol{w}_{(t)}$。

1.2 联邦学习

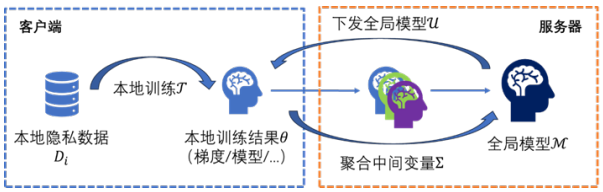

图 1: 联邦学习示意图

联邦学习是一种联合多方本地数据，共同训练全局模型的分布式机器学习技术。联邦学习的训练流程如图 1所示：各个客户端使用训练算法 $\mathcal{T}$，在本地的隐私数据集 D_i 上进行训练，得到本地模型；接下来，服务器（或各方联合模拟出的服务器）使用安全的聚合算法 Σ 整合所有客户端的训练结果，以此更新全局模型 $\mathcal{M}$；最后，服务器将更新后的全局模型下发给各个客户端，开始新一轮迭代，直到全局模型 $\mathcal{M}$ 收敛。其中，服务器聚合客户端训练结果的方式 Σ 是联邦学习算法的核心，关系到联邦学习算法的通信量、安全性、准确率。

按照数据在各个客户端的分布方式，联邦学习主要可以分为横向联邦学习和纵向联邦学习两类，其区别如图 2所示。在横向联邦学习中，各方拥有来自不同实体的数据。这些数据具有相同的特征，联合后将扩充数据集的规模；在纵向联邦学习中，各方拥有来自相同实体的数据。这些数据具有不同的特征，联合后将扩展数据集的维度。在横向联邦学习中，各方用户规模、用户分布可能有差异，因此设计时主要考虑如何平衡非独立同分布数据的异质性。在纵向联邦学习中，用户的待预测标签只分布在一端，因此设计时主要考虑多方如何联合完成梯度计算。

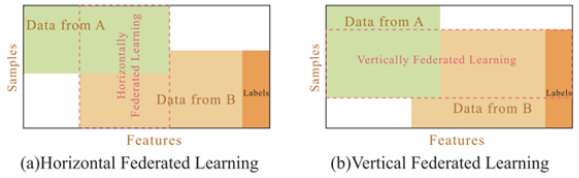

图 2: 横向联邦与纵向联邦数据划分

1.3 横向联邦线性回归

在横向联邦学习算法 [2] 中，各方每一轮训练都会得到本地数据集的梯度。由于各个数据集的梯度可以线性累加，因此，可以将各个本地模型或其梯度直接进行聚合，用于更新全局模型。不失一般性，我们记其中一位数据持有者为 A，其拥有的本地隐私数据集为 $D^A = (\boldsymbol{X}, \boldsymbol{y})$，样本量为 $n^A = |D^A|$，每个样本都包含 m 个属性值和 1 个观测值。在第 t 轮迭代时，A 首先从服务器获得此轮初始权重 $\boldsymbol{w}_{(t)}$，再基于本地数据 D^A 计算更新后的权重 $\boldsymbol{w}^A_{(t+1)}$ 并上传到服务器。服务器收集参与方集合 P 上传的权重，聚合后得到新的全局模型 $\boldsymbol{w}_{(t+1)} = \mathrm{Agg}_{i\in P}(\boldsymbol{w}^i_{(t+1)})$。根据全局梯度聚合的频次差异，常用的算法有 FedSGD 和 FedAvg 两种。前者会在每次梯度下降后均进行一次全局梯度聚合更新，其流程如算法 1所示。后者则是在本地进行多次梯度下降后才进行一次模型更新，其计算过程如算法 2所示。一般来说，FedAvg 算法的通信开销更小。

Algorithm 1 客户端 A FedSGD 第 t 步迭代

从服务器接收初始全局模型 $\boldsymbol{w}$

计算梯度 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y}$

将梯度 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$ 上传到服务器

Algorithm 2 客户端 A FedAvg 第 t 步迭代

ClientUpdate($\boldsymbol{w}_{(t)}$): // 客户端 A 接收到的全局模型为 $\boldsymbol{w}_{(t)}$

$\boldsymbol{w} = \boldsymbol{w}_{(t)}$

for each local epoch i from 1 to E **do**

  for batch $D_{\text{batch}} \leftarrow D^A$ **do**

    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}(\boldsymbol{w}, D_{\text{batch}})$

$\boldsymbol{w}^A_{(t+1)} = \boldsymbol{w}$

return $\boldsymbol{w}^A_{(t+1)}$ to server

1.4 纵向联邦线性回归

在纵向联邦学习 [3] 中，各个数据持有者分别持有样本的部分属性或观测值。不失一般性，我们仅对两方的场景进行分析。记被动方 A 仅拥有样本的 m_A 个属性，其属性矩阵为 $\boldsymbol{X}^A \in \mathbb{R}^{n\times m_A}$，主动方 B

拥有样本的其余 m_B 个特征和样本的观测值，其属性矩阵为 $\boldsymbol{X}^B \in \mathbb{R}^{n\times m_B}$，观测值向量为 $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$，A 方和 B 方的属性权重分别为 $\boldsymbol{w}^A \in \mathbb{R}^{m_A\times 1}, \boldsymbol{w}^B \in \mathbb{R}^{m_B\times 1}$。记 $\boldsymbol{u}^A = \boldsymbol{X}^A\boldsymbol{w}^A \in \mathbb{R}^{n\times 1}$ 为关于 A 方属性的预测向量，$\boldsymbol{u}^B = \boldsymbol{X}^B\boldsymbol{w}^B \in \mathbb{R}^{n\times 1}$ 为关于 B 方属性的预测向量。因此，所有样本的残差向量可以表示为 $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{u}^B - \boldsymbol{u}^A \in \mathbb{R}^{n\times 1}$。

根据以上符号，损失函数可以表示为 $\mathcal{L} = \frac{1}{2}\boldsymbol{d}^T\boldsymbol{d}$，$\mathcal{L}$ 关于 A、B 两方系数的梯度分别为：

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A} = (\boldsymbol{X}^A)^T\boldsymbol{d}$$
$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^B} = (\boldsymbol{X}^B)^T\boldsymbol{d}$$

因此，在第 t 轮迭代时，A、B 两方只需联合计算出 $\boldsymbol{d}_{(t)} \in \mathbb{R}^{n\times 1}$，就可以在本地完成参数更新

$$\boldsymbol{w}^A_{(t+1)} = \boldsymbol{w}^A_{(t)} - \frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^A_{(t)}} = \boldsymbol{w}^A_{(t)} - (\boldsymbol{X}^A)^T\boldsymbol{d}_{(t)}$$
$$\boldsymbol{w}^B_{(t+1)} = \boldsymbol{w}^B_{(t)} - \frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^B_{(t)}} = \boldsymbol{w}^B_{(t)} - (\boldsymbol{X}^B)^T\boldsymbol{d}_{(t)}$$

根据以上推导，为了计算 $\boldsymbol{w}^A_{(t+1)}, \boldsymbol{w}^B_{(t+1)}$，一种纵向联邦线性回归协议算法如表 1所示。

表 1: 纵向联邦线性回归第 t 步迭代

	计算方 A	计算方 B	平台方 C
第 1 步	计算 $\boldsymbol{u}^A_{(t)}, \mathcal{L}^A_{(t)}$。向 B 发送 $[[\boldsymbol{u}^A_{(t)}]]_C, [[\mathcal{L}^A_{(t)}]]_C$	计算 $\boldsymbol{u}^B_{(t)}, [[\boldsymbol{d}_{(t)}]]_C$。向 A 发送 $[[\boldsymbol{d}_{(t)}]]_C$，向 C 发送 $[[\mathcal{L}_{(t)}]]_C$	C 解密 $[[\mathcal{L}_{(t)}]]_C$。如果达到中止条件则告知 A、B
第 2 步	计算 $[[\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^A_{(t)}}]]_C$。生成随机掩码 $\boldsymbol{R}^A_{(t)}$。向 C 发送 $[[\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^A_{(t)}} + \boldsymbol{R}^A_{(t)}]]_C$	计算 $[[\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^B_{(t)}}]]_C$。生成随机掩码 $\boldsymbol{R}^B_{(t)}$。向 C 发送 $[[\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^B_{(t)}} + \boldsymbol{R}^B_{(t)}]]_C$	C 解密 $[[\cdot]]_C$。向 A 发送 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^A_{(t)}} + \boldsymbol{R}^A_{(t)}$。向 B 发送 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^B_{(t)}} + \boldsymbol{R}^B_{(t)}$
第 3 步	根据 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^A_{(t)}}$ 计算 $\boldsymbol{w}^A_{(t+1)}$	根据 $\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^B_{(t)}}$ 计算 $\boldsymbol{w}^B_{(t+1)}$	

2 威胁模型与分析框架

本分析框架以联邦学习中的每一个数据持有者作为分析对象。我们认为，分析对象主要包含数据集、算法、状态三类信息，其中数据集和算法是固定的，而状态可以在运行时改变。分析对象的数据集和状态是保密的，而算法是公开的。我们主要关注分析对象隐私数据集的信息泄漏情况，其威胁模型如图 3所示。

在联邦学习实际场景中，参与方之间互不信任。即使采用了密码学工具，数据持有者的信息仍然有可能被恶意的参与方探知 [4]。因此，本框架认为，只分析对象会忠实地执行本地的算法，其它参与方会在一定程度上被攻击者控制，并在一定程度上偏离算法。在联邦学习的过程中，分析对象可能会与这些攻击者产生数据交换。因此，恶意的攻击者可以结合数据分布等先验知识，以及算法等初始信息，适应性地构造发送给分析对象的数据，并收集分析对象发出的数据，最终综合这些数据推测分析对象隐私数据集的相关信息。

为了方便工程上评估联邦学习算法的安全性，我们提出了一套基于中间结果的信息泄露评估框架。该框架主要分为以下几个步骤：

1. 根据算法的传入传出变量语义，识别数据持有者在运行协议时直接向攻击者泄露的中间结果

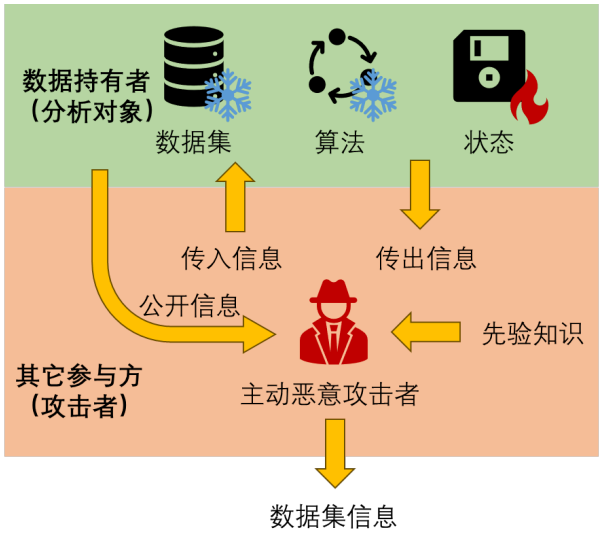

图 3: 中间结果泄露分析模型

2. 结合多个直接中间结果，分析如何推导出更多的间接中间结果

3. 分析每个中间结果对原始数据的泄露情况。根据需求，使用自由度或熵对泄露情况进行量化。

3 识别直接中间结果

直接中间结果是数据持有者在协议运行过程中，直接对外发送的信息。虽然这些信息在传输的过程中可能受到了同态加密、秘密共享等安全技术的保护，但当攻击者能力高于安全技术的保护能力时，这些中间结果就会泄露。这些直接泄露的中间结果是我们分析的基础。

在横向联邦学习的 FedSGD 和 FedAvg 算法中，客户端会 A 向服务器发送 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$ 和 $\boldsymbol{w}^A_{(t+1)}$。因此，在攻击分级大于等于 1 级的场景下，服务器可以直接获得这些中间结果。如果隐私计算产品在横向联邦学习梯度聚合时使用了更安全的聚合算法 [5]，则获取这些中间变量需要依靠服务器和其它客户端的共谋：其它客户端发送零梯度，使得服务器获得的聚合结果仅包含目标受害者上传的中间变量。此时，直接中间结果 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$ 和 $\boldsymbol{w}^A_{(t+1)}$ 只有在攻击分级大于等于 3 级的场景下才会泄露。综上所述，横向联邦学习的直接中间结果泄露如表 2所示。

表 2: 横向联邦线性回归第 t 步迭代的直接中间结果分析

联邦学习算法	直接中间结果	获取方式	攻击分级
FedSGD	$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$	服务器直接解密	2+
FedAvg	$\boldsymbol{w}^A_{(t+1)}$	服务器直接解密	2+
带安全聚合的 FedSGD	$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$	其他客户端发送零数据，服务器直接解密	3+
带安全聚合的 FedAvg	$\boldsymbol{w}^A_{(t+1)}$	其他客户端发送零数据，服务器直接解密	3+

采用相似的分析思路，我们可以对表 1所述的纵向联邦学习算法进行分析，得到直接中间结果泄露情况如表 3所示。

表 3: 纵向联邦线性回归第 t 步迭代的直接中间结果分析

直接中间结果	获取方式	攻击分级
$\mathcal{L}_{(t)}$	C 直接解密	2+
$\boldsymbol{u}^A_{(t)}$	B 与 C 共谋，或	3+
$\mathcal{L}^A_{(t)}$	B 修改协议第 2 步，借助 C 解密（影响模型训练）	3+
$\boldsymbol{d}_{(t)}$	A 与 C 共谋，或	3+
	A 修改协议第 2 步，借助 C 解密（影响模型训练）	3+
$\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^A_{(t)}}$	传输时被掩码保护，其它参与方永远无法获取	
$\frac{\partial \mathcal{L}_{(t)}}{\partial \boldsymbol{w}^B_{(t)}}$		

4 识别间接中间结果

攻击者通过结合若干直接中间结果，可以求解出更多更明确的中间结果关系。更进一步，如果敌手拥有违反协议的能力，可以通过恶意行为干扰直接中间结果的计算过程，使其泄露更多信息。经过这些分析得到的信息称为间接中间结果。为了确定间接结果的攻击分级，我们既要分析推断所需中间结果的攻击分级，也要分析间接中间结果获取方式对应的攻击分级。最终，间接结果的攻击分级为两个攻击分级的最大值。

4.1 横向联邦线性回归

在 FedSGD 中，客户端会从服务器接收 $\boldsymbol{w}_{(t)} \in \mathbb{R}^{m\times 1}$，并上传每次梯度下降的结果 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_{(t)} - \boldsymbol{X}^T\boldsymbol{y} \in \mathbb{R}^{m\times 1}$。因此，如果恶意的服务器传入 $\boldsymbol{w}_{(t)} = \boldsymbol{0} \in \mathbb{R}^{m\times 1}$，就可以从 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$ 中得到中间结果 $\boldsymbol{X}^T\boldsymbol{y} \in \mathbb{R}^{m\times 1}$。当攻击者得到中间变量 $\boldsymbol{X}^T\boldsymbol{y}$ 后，攻击者就可以继续任意选取感兴趣的 $\boldsymbol{w}_{(t)} \in \mathbb{R}^{m\times 1}$，从 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$ 中得到中间结果 $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_{(t)}$。其中一种较优的选取方式是选择 $\boldsymbol{w}_{(i)} = \boldsymbol{e}_i = (0,\ldots,0,1,0,\ldots,0) \in \mathbb{R}^{m\times 1}$，此时 $\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}\}_{i=1}^m$ 对应的中间结果为 $\boldsymbol{X}^T\boldsymbol{X} \in \mathbb{R}^{m\times m}$ 的第 i 列。这类攻击需要修改服务器的协议，但修改的难度不大，故此间接中间结果只会在 2 级或以上的威胁场景中泄露。

FedAvg 算法的直接中间结果语义较为复杂。因此，我们可以考虑发动客户端分离攻击：服务器将客户端前一轮上传的模型 $\boldsymbol{w}^A_{(t)}$ 直接作为新一轮的初始全局模型 $\boldsymbol{w}_{(t+1)}$ 传回客户端。客户端使用本地数据多次迭代直至收敛。此时，最终上传的模型权重 $\boldsymbol{w}^A_{(\text{conv})}$ 将会接近于最优权重 $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \in \mathbb{R}^{m\times 1}$。这类攻击需要修改服务器的协议，但修改的难度不大，故此间接中间结果只会在 2 级或以上的威胁场景中泄露。

综上所述，横向联邦线性回归的间接中间结果如表 4所示。随协议运行，各个算法的中间结果泄露情况如表 5所示。

表 4: 横向联邦线性回归的间接中间结果分析

算法	间接中间结果	所需中间结果	获取方式	攻击分级
FedSGD	$\boldsymbol{X}^T\boldsymbol{y}$	$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$(2+)	服务器发送全局模型为 $\boldsymbol{w} = \boldsymbol{0}$ (2+)	2+
	$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}$	$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^A}$ (2+),$\boldsymbol{X}^T\boldsymbol{y}$ (2+)	服务器发送全局模型为 $\boldsymbol{w} = \boldsymbol{c}$ (2+)	2+
	$\boldsymbol{X}^T\boldsymbol{X}$	$\{\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}_i\}_{i=1}^m$，各 $\boldsymbol{c}_i$ 线性无关 (2+)	解线性方程组 (2+)	2+
FedAvg	$(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$	$\boldsymbol{w}^A_{(\text{conv})}$ (2+)	在训练时分离客户端，直至收敛 (2+)	2+

表 5: 随协议运行时横向联邦线性回归的间接中间泄露情况

算法	攻击分级	迭代轮数 t	泄露中间结果
FedSGD	2+	1	$\boldsymbol{X}^T\boldsymbol{y}$
		2∼m	$\boldsymbol{X}^T\boldsymbol{y}, \{\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}_i\}_{i=1}^{t-1}$
		≥m+1	$\boldsymbol{X}^T\boldsymbol{y}, \boldsymbol{X}^T\boldsymbol{X}$
FedAvg	2+	收敛	$(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

4.2 纵向联邦线性回归

纵向联邦学习中有主动方、被动方两方的参与。因此，我们对两方分别进行分析。

我们首先分析 A 的间接中间结果泄露。注意到 $\boldsymbol{u}_{(1)} = \boldsymbol{X}^A\boldsymbol{w}_{(1)}^A$。因此，如果 $\boldsymbol{w}_{(1)}^A = \boldsymbol{c} \neq \boldsymbol{0}$，攻击者就可以在第 1 轮迭代中获取到中间变量 $\boldsymbol{X}^A\boldsymbol{w}_{(1)}^A$。另外，攻击者根据 A 在第 $t+1$ 轮和第 t 轮两次迭代时发送的中间结果 $\boldsymbol{u}_{(t+1)}^A, \boldsymbol{u}_{(t)}^A$，可以计算出

$$\Delta\boldsymbol{u}_{(t+1)}^A = \boldsymbol{u}_{(t+1)}^A - \boldsymbol{u}_{(t)}^A = \boldsymbol{X}^A(\boldsymbol{w}_{(t+1)}^A - \boldsymbol{w}_{(t)}^A) = -\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{d}_{(t)}$$

因此，如果攻击者在第 t 轮迭代时，作为 B 向 A 发送指定的 $\boldsymbol{d}_{(t)}$，就可以根据 $\boldsymbol{u}_{(t+1)}^A, \boldsymbol{u}_{(t)}^A$ 计算出间接中间变量 $\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{d}_{(t)}$。例如攻击者选取 $\boldsymbol{d}_{(i)} = \boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{R}^{m\times 1}$，可以获得对应的中间结果为 $\boldsymbol{X}^A(\boldsymbol{X}^A)^T \in \mathbb{R}^{n\times n}$ 的第 i 列。这类攻击需要修改 B 的协议，但修改的难度不大，故此间接中间结果只会在 2 级或以上的威胁场景中泄露。当攻击者收集到 n 个 $\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{d}_{(i)}$，其中 $\{\boldsymbol{d}_{(i)}\}_{i=1}^n$ 线性无关时，就可以联立方程组解出中间结果 $\boldsymbol{X}^A(\boldsymbol{X}^A)^T \in \mathbb{R}^{n\times n}$。

B 的中间结果泄露分析与 A 基本相同。如果 $\boldsymbol{w}_{(1)}^B = \boldsymbol{c} \neq \boldsymbol{0}$，攻击者就可以在第 1 轮迭代中获取到中间变量 $\boldsymbol{y} - \boldsymbol{X}^B\boldsymbol{w}_{(1)}^B$。攻击者根据 B 在第 $t+1$ 轮和第 t 轮两次迭代时发送的中间结果 $\boldsymbol{d}_{(t+1)}, \boldsymbol{d}_{(t)}$，可以计算出

$$\Delta\boldsymbol{d}_{(t+1)} = (\boldsymbol{y} - \boldsymbol{u}_{(t+1)}^A - \boldsymbol{u}_{(t+1)}^B) - (\boldsymbol{y} - \boldsymbol{u}_{(t)}^A - \boldsymbol{u}_{(t)}^B) = \boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(t)} - \Delta\boldsymbol{u}_{(t+1)}^A$$

因此，攻击者可以根据 $\boldsymbol{d}_{(t+1)}, \boldsymbol{d}_{(t)}$，和自己任取的 $\boldsymbol{u}_{(t)}^A, \boldsymbol{u}_{(t+1)}^A$，获得中间变量 $\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(t)}$。由于 $\boldsymbol{d}_{(t)}$ 的计算还涉及 B 方的数据，因此 A 方无法精确控制 $\boldsymbol{d}_{(t)}$ 的取值。因此，攻击者完全可以不修改协议，而是被动接受 B 方计算得到的 $\boldsymbol{d}_{(t)}$。故此间接中间结果会在 1 级或以上的威胁场景中泄露。不过，当攻击者收集到 n 个 $\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(i)}$，其中 $\{\boldsymbol{d}_{(i)}\}_{i=1}^n$ 线性无关时，仍然可以联立方程组解出中间结果 $\boldsymbol{X}^B(\boldsymbol{X}^B)^T \in \mathbb{R}^{n\times n}$。

综上所述，纵向联邦线性回归的间接中间结果如表 6所示。随协议运行，各个算法的中间结果泄露情况如表 5所示。

表 6: 纵向联邦线性回归的间接中间结果分析

间接中间结果	所需中间结果	获取方式	攻击分级
$\boldsymbol{X}^A\boldsymbol{w}_{(1)}^A$	$\boldsymbol{u}_{(1)}^A$ (3+)	$\boldsymbol{w}_{(1)}^A = \boldsymbol{c} \neq \boldsymbol{0}$ 已知 (2+)	3+
$\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{c}$	$\boldsymbol{u}_{(t+1)}^A$ (3+), $\boldsymbol{u}_{(t)}^A$ (3+)	B 向 A 发送 $\boldsymbol{d}_{(t)} = \boldsymbol{c}$ (2+)	3+
$\boldsymbol{X}^A(\boldsymbol{X}^A)^T$	$\{\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{c}_i\}_{i=1}^m$，各 $\boldsymbol{c}_i$ 线性无关 (3+)	解线性方程组 (2+)	3+
$\boldsymbol{y} - \boldsymbol{X}^B\boldsymbol{w}_{(1)}^B$	$\boldsymbol{d}_{(1)}$ (3+)	$\boldsymbol{w}_{(1)}^B = \boldsymbol{c} \neq \boldsymbol{0}$ 已知 (2+)	3+
$\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(t)}$	$\boldsymbol{d}_{(t+1)}$ (3+),$\boldsymbol{d}_{(t)}$ (3+)	A 与 B 正常交互并计算 $\Delta\boldsymbol{d}_{(t+1)}$ (2+)	3+
$\boldsymbol{X}^B(\boldsymbol{X}^B)^T$	$\{\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(i)}\}_{i=1}^m$，各 $\boldsymbol{d}_{(i)}$ 线性无关 (3+)	解线性方程组 (2+)	3+

表 7: 随协议运行时纵向联邦线性回归的间接中间泄露情况

参与方	攻击分级	迭代轮数 t	泄露中间结果
A	3+	1	$\boldsymbol{X}^A\boldsymbol{w}_{(1)}^A$
		2∼n	$\boldsymbol{X}^A\boldsymbol{w}_{(1)}^A, \{\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{c}_i\}_{i=1}^{t-1}$
		≥n+1	$\boldsymbol{X}^A\boldsymbol{w}_{(1)}^A, \boldsymbol{X}^A(\boldsymbol{X}^A)^T$
B	3+	1	$\boldsymbol{y}-\boldsymbol{X}^B\boldsymbol{w}_{(1)}^B$
		2∼n	$\boldsymbol{y}-\boldsymbol{X}^B\boldsymbol{w}_{(1)}^B, \{\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(i)}\}_{i=1}^{t-1}$
		≥n+1	$\boldsymbol{y}-\boldsymbol{X}^B\boldsymbol{w}_{(1)}^B, \boldsymbol{X}^B(\boldsymbol{X}^B)^T$

5 中间结果泄露对数据集的影响

识别出联邦学习算法泄露的中间结果后，我们再考虑这些中间结果将会为数据集降低多少不确定性。

在攻击者不知道原始数据的任何信息时，原始数据的取值对攻击者来说是完全自由的。然而，由于攻击者从隐私计算的过程中获取到了一些泄露的中间变量，所以原始数据可能的取值就会受到额外约束。因此，我们将原始数据中的自由变量个数定义为自由度，并用来量化原始数据的信息量变化。

5.1 一些关于自由度的引理

5.1.1 $\boldsymbol{WW}^T$ 型中间结果的泄露影响

在横向、纵向联邦线性回归中，很多中间变量都是形如 $\boldsymbol{WW}^T \in \mathbb{R}^{w\times w}(\boldsymbol{W}\in\mathbb{R}^{w\times r}, w\neq r)$ 的形式。因此，我们先对 $\boldsymbol{WW}^T$ 矩阵的性质作简要补充，并分析 $\boldsymbol{WW}^T$ 泄露对 $\boldsymbol{W}$ 的自由度损失。

对于 w 个 r 维行向量组成的矩阵 $\boldsymbol{W}\in\mathbb{R}^{w\times r}$。矩阵 $\boldsymbol{WW}^T\in\mathbb{R}^{w\times w}$ 称为 $\boldsymbol{W}$ 的 Gram 矩阵，具有如下性质：

- $\boldsymbol{WW}^T$ 是一个对称矩阵
- 该矩阵无法确定唯一的 $\boldsymbol{W}$[1]，但可以在 r 维空间上确定这 w 个向量的模长和这 w 个向量间的余弦值

我们可以从 Gram 矩阵的第二条性质入手，分析 $\boldsymbol{WW}^T\in\mathbb{R}^{w\times w}$ 矩阵泄露对矩阵 $\boldsymbol{W}$ 的隐私威胁。

$\boldsymbol{WW}^T\in\mathbb{R}^{w\times w}$ 矩阵泄露前，w 个 r 维行向量组成的矩阵 $\boldsymbol{W}\in\mathbb{R}^{w\times r}$ 中所有元素均可自由取值，因此其自由度为 $\mathrm{DoF}(\boldsymbol{W})=wr$。$\boldsymbol{WW}^T\in\mathbb{R}^{w\times w}$ 矩阵泄露后，将固定住这 w 个向量的模长和这 w 个向量间的余弦值。

对于这 w 个 r 维向量中的第一个向量，由于 Gram 矩阵确定了其模长，所以其自由度仅剩 $(r-1)$；对于其中的第二个向量，由于 Gram 矩阵确定了此向量的模长和此向量与第一个向量的余弦值，所以其自由度仅剩 $(r-2)$……对于其中的第 r 个向量，由于 Gram 矩阵固定了此向量模长和此向量与前 $r-1$ 个向量的余弦值，所以其自由度为 0[2]。其余的第 $r+1, r+2, \ldots, w$ 个向量将完全由这 r 个向量所决定，因此它们的自由度为 0。

因此，如果 $\boldsymbol{W}\in\mathbb{R}^{w\times r}$ 的 Gram 矩阵 $\boldsymbol{WW}^T\in\mathbb{R}^{w\times w}$ 泄露，则 $\boldsymbol{W}\in\mathbb{R}^{w\times r}$ 的自由度将变为

$$\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{WW}^T)=\sum_{i=\max(r-w,0)}^{r-1} i=\begin{cases}(rw-\frac{w^2+w}{2}) & (w<r)\\ \frac{r^2-r}{2} & (w\geq r)\end{cases}$$

即 $\boldsymbol{WW}^T$ 对 $\boldsymbol{W}$ 的自由度泄露比为 $\alpha(\boldsymbol{W}|\boldsymbol{WW}^T)=1-\frac{\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{WW}^T)}{\mathrm{DoF}(\boldsymbol{W})}$。

[1] $\boldsymbol{WW}^T$ 确定了一族矩阵 $\{\boldsymbol{W}\}\subset\mathbb{R}^{w\times r}$。对于矩阵族中的任意矩阵 $\boldsymbol{A}\in\{\boldsymbol{W}\}$ 都有 $\forall\boldsymbol{U}\in\mathbb{R}^{m\times m}, \boldsymbol{UU}^T=\boldsymbol{I}_m\Rightarrow\boldsymbol{B}=\boldsymbol{AU}\in\{\boldsymbol{W}\}$。

[2] 部分情况下，该向量可能有两个取值。此处为方便表示忽略这种情况。这一简化我们的估计不会产生太多影响。

5.1.2 $\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k$ 型中间结果的泄露影响

在横向、纵向联邦线性回归中，还有一些中间变量仅泄露了 $\boldsymbol{W}\boldsymbol{W}^T \in \mathbb{R}^{w\times w}$ 的部分信息，例如仅泄露了与 k 个线性无关向量 $\{\boldsymbol{c}_i\}_{i=1}^k$ 的乘积 $\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k$。因此，我们进一步考虑 $\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k$ 泄露对 $\boldsymbol{W}\boldsymbol{W}^T \in \mathbb{R}^{w\times w}$ 的信息量损失。

由于 $\boldsymbol{W}\boldsymbol{W}^T \in \mathbb{R}^{w\times w}$ 是对称矩阵，因此 $\boldsymbol{W}\boldsymbol{W}^T$ 的自由度为 $\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T) = \frac{(w+1)w}{2}$。为简单起见，我们仅考虑 $\boldsymbol{c}_i = \boldsymbol{e}_i$ 的情况。此时，$\boldsymbol{W}\boldsymbol{W}^T$ 的其中 k 列被完全确定，仅余 $w-k$ 列可以自由取值。因此 $\boldsymbol{W}\boldsymbol{W}^T$ 的自由度将变为

$$\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = \sum_{i=0}^{\max(w-k,0)} i = \begin{cases} \frac{(w-k)(w-k+1)}{2} & (k<w) \\ 0 & (k\geq w) \end{cases}$$

即 $\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k$ 对 $\boldsymbol{W}\boldsymbol{W}^T$ 的自由度泄露比为 $\alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = 1 - \frac{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)}{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T)}$

进一步，$\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k$ 对 $\boldsymbol{W}$ 的自由度泄露比为

$$\alpha(\boldsymbol{W}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k) \approx \alpha(\boldsymbol{W}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = \alpha(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) * \alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)$$

5.2 分析具体算法的自由度损失

接下来，我们结合数据规模、训练轮数等设定，代入上述推导结果，计算数据泄露程度。

5.2.1 横向联邦学习示例场景

以 FATE 横向联邦学习的示例场景为例：每个客户端使用 1024 个样本，以 64 的 batch size，对 19 维特征和 1 维标签，进行 10 个 epoch 的 FedSGD 学习。由于数据是分 batch 进行学习的，所以我们只需对每个 batch 进行分析，即 $m = 19, n = 64, t = 10$。每个 batch 原自由度为 $\mathrm{DoF}(\boldsymbol{X},\boldsymbol{Y}) = (m+1)n = 1280$。

根据表 5，由于 $t < m$，所以此过程泄露的中间变量为 $\boldsymbol{X}^T\boldsymbol{y} \in \mathbb{R}^{m\times 1}, \{\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}_i \in \mathbb{R}^{m\times 1}\}_{i=1}^{t-1}$。

其中，$\boldsymbol{X}^T\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ 泄露后，将增加 m 个约束条件，使自由度变为 $\mathrm{DoF}(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{X}^T\boldsymbol{y}) = (m+1)n - m$，则此中间信息对原数据 $(\boldsymbol{X},\boldsymbol{Y})$ 的泄露比为 $\alpha(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{X}^T\boldsymbol{y}) = 1 - \frac{(m+1)n-m}{(m+1)n} = 1.48\%$

中间变量 $\{\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{c}_i \in \mathbb{R}^{m\times 1}\}_{i=1}^{t-1}$ 泄露后，根据5.1.2节，有（取 $\boldsymbol{W} = \boldsymbol{X}^T, w = m = 19, r = n = 64, k = t-1 = 9$）

$$\begin{aligned}
\mathrm{DoF}(\boldsymbol{W}) &= wr = 1216 \\
\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) &= (rw - \frac{w^2+w}{2}) = 1026 \\
\alpha(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) &= 1 - \frac{\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T)}{\mathrm{DoF}(\boldsymbol{W})} = 15.63\% \\
\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T) &= \frac{(w+1)w}{2} = 190 \\
\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= \frac{(w-k)(w-k+1)}{2} = 55 \\
\alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= 1 - \frac{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)}{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T)} = 71.05\% \\
\alpha(\boldsymbol{W}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k) &= \alpha(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) * \alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = 11.10\% \\
\mathrm{DoF}(\boldsymbol{W}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= \mathrm{DoF}(\boldsymbol{W}) * (1 - \alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)) = 1081.02
\end{aligned}$$

则原数据的自由度变为 $\mathrm{DoF}(\boldsymbol{X},\boldsymbol{Y}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = \mathrm{DoF}(\boldsymbol{X}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) + \mathrm{DoF}(\boldsymbol{Y}) = 1081.02 + 64 = 1145.02$，泄露比为 $\alpha(\boldsymbol{X},\boldsymbol{Y}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = 1 - \frac{1145.02}{1280} = 10.55\%$。

综上，此场景下，各中间变量总共泄露 $1.48\% + 10.55\% = 12.03\%$ 的原始数据信息。

5.2.2 纵向联邦学习示例场景

以 FATE 纵向联邦学习的示例场景为例，每个客户端使用 570 个样本，以 100 的 batch size，进行 10 个 epoch 的纵向联邦学习。其中 A 拥有 7 个特征，B 拥有 4 个特征和 1 个标签。

考察 A 方的每个 batch，其参数为 $m=7, n=100, t=10$，即原自由度为 $\mathrm{DoF}(\boldsymbol{X}^A)=mn=700$。根据表 7，泄露的中间变量有 $\boldsymbol{X}^A\boldsymbol{w}^A_{(1)}, \{\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{c}_i\}_{i=1}^{t-1}$。其中 $\boldsymbol{X}^A\boldsymbol{w}^A_{(1)}\in\mathbb{R}^{m\times 1}$ 的泄露将增加 n 个约束条件，故 $\alpha(\boldsymbol{X}^A|\boldsymbol{X}^A\boldsymbol{w}^A_{(1)})=1-\frac{mn-n}{mn}=14.29\%$。中间变量 $\{\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{c}_i\}_{i=1}^{t-1}$ 泄露后，根据5.1.2节，有（取 $\boldsymbol{W}=\boldsymbol{X}^A, w=n=100, r=m=7, k=t-1=9$）

计算得到

$$
\begin{aligned}
\mathrm{DoF}(\boldsymbol{W}) &= wr = 700\\
\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) &= \frac{r^2-r}{2} = 21\\
\alpha(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) &= 1-\frac{\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T)}{\mathrm{DoF}(\boldsymbol{W})} = 97.00\%\\
\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T) &= \frac{(w+1)w}{2} = 5050\\
\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= \frac{(w-k)(w-k+1)}{2} = 4186\\
\alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= 1-\frac{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)}{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T)} = 17.11\%\\
\alpha(\boldsymbol{W}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k) &= \alpha(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T)*\alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = 16.60\%
\end{aligned}
$$

则原数据的泄露比为 $\alpha(\boldsymbol{X}^A|\{\boldsymbol{X}^A(\boldsymbol{X}^A)^T\boldsymbol{c}_i\}_{i=1}^{t-1})=16.60\%$。

综上，此场景下，对于参与方 A，各中间变量总共泄露 $14.29\%+16.60\%=30.89\%$ 的原始数据信息。

对于 B 方，其训练参数为 $m=4, n=100, t=10$，即原自由度为 $\mathrm{DoF}(\boldsymbol{X}^B, \boldsymbol{Y})=(m+1)n=500$。根据表 3，泄露的中间变量有 $\boldsymbol{y}-\boldsymbol{X}^B\boldsymbol{w}^B_{(1)}, \{\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(i)}\}_{i=1}^{t-1}$。其中 $\boldsymbol{y}-\boldsymbol{X}^B\boldsymbol{w}^B_{(1)}\in\mathbb{R}^{n\times 1}$ 的泄露将增加 n 个约束条件，故 $\alpha(\boldsymbol{X}^B|\boldsymbol{y}-\boldsymbol{X}^B\boldsymbol{w}^B_{(1)})=1-\frac{(m+1)n-n}{(m+1)n}=20\%$。中间变量 $\{\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(i)}\}_{i=1}^{t-1}$ 泄露后，根据5.1.2节，有（取 $\boldsymbol{W}=\boldsymbol{X}^B, w=n=100, r=m=4, k=t-1=9$）

$$
\begin{aligned}
\mathrm{DoF}(\boldsymbol{W}) &= wr = 400\\
\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) &= \frac{r^2-r}{2} = 6\\
\alpha(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T) &= 1-\frac{\mathrm{DoF}(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T)}{\mathrm{DoF}(\boldsymbol{W})} = 98.50\%\\
\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T) &= \frac{(w+1)w}{2} = 5050\\
\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= \frac{(w-k)(w-k+1)}{2} = 4186\\
\alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= 1-\frac{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)}{\mathrm{DoF}(\boldsymbol{W}\boldsymbol{W}^T)} = 17.11\%\\
\alpha(\boldsymbol{W}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{c}_i\}_{i=1}^k) &= \alpha(\boldsymbol{W}|\boldsymbol{W}\boldsymbol{W}^T)*\alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) = 16.85\%\\
\mathrm{DoF}(\boldsymbol{W}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k) &= \mathrm{DoF}(\boldsymbol{W})*(1-\alpha(\boldsymbol{W}\boldsymbol{W}^T|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)) = 332.6
\end{aligned}
$$

则原数据的自由度变为 $\mathrm{DoF}(\boldsymbol{X}^B, \boldsymbol{Y}|\{\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(i)}\}_{i=1}^{t-1}) = \mathrm{DoF}(\boldsymbol{X}^B|\{\boldsymbol{X}^B(\boldsymbol{X}^B)^T\boldsymbol{d}_{(i)}\}_{i=1}^{t-1})+\mathrm{DoF}(\boldsymbol{Y}) = 332.6+100=432.6$，泄露比为 $\alpha(\boldsymbol{X}^B, \boldsymbol{Y}|\{\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{e}_i\}_{i=1}^k)=1-\frac{432.6}{500}=13.48\%$。

综上，此场景下，对于参与方 B，各中间变量总共泄露 $20\%+13.48\%=33.48\%$ 的原始数据信息。

5.3 分析数据集熵损

在自由度的分析中，我们没有对数据的分布作任何假设。如果我们对数据集的分布有一定了解，可以结合数据集的熵特征，对信息泄露做更精确的刻画。

自由度泄露比反应了中间变量泄露后，自由变量的损失个数。在实际中，这些变量有的取值范围较广（如收入）、有的取值范围较小（如性别）。因此，敌手如果适应性地选取熵最高的几个变量作为泄露数据，可以获得更大的优势。

根据这种估算方法，我们首先根据数据集，统计每个特征或标签的熵，并从大到小记为 $E_1, \ldots, E_m$。接下来，根据自由度的泄露比 α，折算特征损失数为 αm。此时，我们可以认为熵最大的 $\alpha \times m$ 个特征先被泄露，数据集的熵损可以估算为

$$\Delta E = \frac{\sum_{i=1}^{\alpha \times m} E_i}{\sum_{i=1}^{m} E_i}$$

例如，某数据集有 $m = 7$ 个特征，每个特征的熵分别为账单金额 $E_1 = 10bit$，年龄 $E_2 = 6bit$，职业 $E_3 = 6bit$，收入 $E_4 = 4bit$，教育程度 $E_5 = 3bit$，还款情况 $E_6 = 3bit$，性别 $E_7 = 1bit$。当自由度泄露比为 $\alpha = 16.60\%$ 时，折算泄露特征数为 $\alpha m = 1.162$ 个特征。对应的熵损为 $\Delta E = \frac{\sum_{i=1}^{\alpha \times m} E_i}{\sum_{i=1}^{m} E_i} = \frac{10+6*0.162}{10+6+6+4+3+3+1} = 33.25\%$。

参考文献

[1] 周志华, 机器学习. 清华大学出版社, 2016.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.), vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR, 20–22 Apr 2017.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, jan 2019.

[4] D. Pasquini, D. Francati, and G. Ateniese, "Eluding secure aggregation in federated learning via model inconsistency," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, (New York, NY, USA), p. 2429–2443, Association for Computing Machinery, 2022.

[5] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, (New York, NY, USA), p. 1175–1191, Association for Computing Machinery, 2017.

（二）SecureBoost 信息泄露分析

1 预备知识

1.1 SecureBoost

SecureBoost[1, 2] 是一种在纵向数据联邦的场景下构建决策树的学习算法。该算法需要两种角色：一个既拥有数据特征 $\boldsymbol{X}^A$ 也拥有数据标签 $\boldsymbol{y}$ 的主动方，和一个仅拥有数据特征 $\boldsymbol{X}^B$ 的被动方。在算法开始前，双方各自对本地的每个特征进行等额分桶，其中 j 号特征 b 号分桶的最大值为 $s_{j,b}$。之后，双方各自将本地每个数据的每个特征映射为分桶值 $\widetilde{x}_{i,j} = \arg\min_b(x_{i,j} \leq s_{j,b})$。这些数据分桶值将作为 SecureBoost 训练的原始数据。在初始状态，所有节点均位于 0 号叶子节点、预测值均为 0。接下来，SecureBoost 逐层对叶子节点进行分裂。其训练算法如表 1所示。

表 1: SecureBoost 节点 t 分裂算法

	主动方 A	被动方 B
第 1 步	根据 t 号节点中的每一个样本 $\boldsymbol{x}_i^A$ 的预测值 $\hat{y}_i$ 和标签 y_i，计算梯度 $g_i^{(t)}, h_i^{(t)}$，并向 B 发送 $[[(g_i^{(t)}, h_i^{(t)})]]_A$	B 计算位于 t 号节点的所有样本根据每个本地特征进行分桶的情况。特征 j 的第 b 个分桶拥有的样本序号记为 $Buk_{j,b} = \{i\|\widetilde{x}_{i,j} = b\}$。之后，计算每个特征 j 的第 b 个分桶作为子树分裂点时，划分到左子树的节点 $L_{j,b} = \cup_{b'=1}^{b} Buk_{j,b'}$ 及其梯度和 $[[LGH_{j,b}]]_A = \sum_{i \in L_{j,b}} [[(g_i^{(t)}, h_i^{(t)})]]_A$。向 A 发送乱序的 $[[LGH_{[[j,b]]_B}]]_A$ [1]。
第 2 步	A 计算本地特征分裂产生的 $LGH_{j,b}$、解密 B 方传回的 $LGH_{[[j,b]]_B}$，从中选择效果最好一个分裂点。如果属于 B，将分裂点 $[[j,b]]_B^{\text{best}}$ 告知 B	若最优分裂属于 B，将收到的 $[[j,b]]_B^{\text{best}}$ 转为本地特征分桶 $(j^{\text{best}}, b^{\text{best}})$
第 3 步	如果分裂点属于 A，计算节点分裂后每个样本的位置 $p_i = \mathbb{I}[x_{i,j^{\text{best}}} \leq s_{j^{\text{best}},b^{\text{best}}}]$，并告知 B	如果分裂点属于 B，计算节点分裂后每个样本的位置 $p_i = \mathbb{I}[x_{i,j^{\text{best}}} \leq s_{j^{\text{best}},b^{\text{best}}}]$，并告知 A
第 4 步	A 根据所有样本的新位置，分配新子树的节点权重，并更新每个样本的预测值 $\hat{y}_i$	

[1] 乱序后，每个 $LGH_{j,b}$ 的真实下标对外已不可知。为方便起见，这里借用符号 $[[\cdot]]_B$，表示仅有 B 知晓。

2 直接中间结果分析

在 SecureBoost 算法的第 1 步，主动方 A 的梯度 $g_i^{(t)}, h_i^{(t)}$ 被同态加密保护，被动方 B 无法获得。但被动方 B 使用 A 的密钥加密了 $LGH_{[[j,b]]_B}$ 并直接发送给了 A，因此主动方 A 可以直接解密并获取这一中间结果。此过程不需要修改协议，因此在一级及以上的威胁场景中均可泄露。

在 SecureBoost 算法的第 2、3 步，A 有可能把最优分裂点发送给 B，而被动方 B 可以直接解密并获取这一中间结果。之后，拥有最优分裂点的一方将会把每个样本与最优分裂点的大小关系发送给对方。这些信息都是依照协议明文传输的，因此在一级及以上的威胁场景中均可泄露。

综上所述，SecureBoost 算法的直接中间结果如表 2所示。

表 2: SecureBoost 节点 t 分裂时的直接中间变量

直接中间结果	获取方式	攻击分级
$LGH_{[[j,b]]_B}$	A 直接解密	1+
$(j^{\text{best}}, b^{\text{best}})$	当 B 为最优分裂点时，B 直接解密	1+
p_i	根据协议直接获得	1+

3 间接中间结果分析

主动方可以借助这些直接中间结果，推断出有关原始数据分桶的相关信息。我们将依次介绍梯度逆向、子树匹配、特征分裂三种攻击，来说明如何利用这些直接中间结果获得与数据分桶值 $\widetilde{x}_{i,j}$ 有关的间接中间结果。

3.1 梯度逆向攻击

由于每个 $LGH_{[[j,b]]_B}$ 是对应分裂下左子树所有样本的 $(g_i^{(t)}, h_i^{(t)})$ 的简单加和，而所有样本的 $(g_i^{(t)}, h_i^{(t)})$ 由 A 提供、对 A 已知。因此，主动方 A 可以求解子集和问题，获得乱序后的各个候选左子树 [3]，即

$$L_{[[j,b]]_B} = \{i | \sum (g_i^{(t)}, h_i^{(t)}) = LGH_{[[j,b]]_B}\}$$

然而，子集和问题是一个 NP 完全问题，求解难度随着样本规模迅速增加。为了降低逆向梯度和的难度，恶意的主动方可以向被动方发送精心构造的梯度信息，使得每个样本的 $(g_i^{(t)}, h_i^{(t)})$ 相互正交，且可以快速地从和式中分解。这样，主动方就能够以很小的代价，恢复参与 $LGH_{[[j,b]]_B}$ 计算的每一个元素下标，即 $(g_i^{(t)}, h_i^{(t)}) = Encode(i)$。$(g_i^{(t)}, h_i^{(t)})$ 的一种构造方案是逐比特编码样本，即 $(g_i^{(t)}, h_i^{(t)}) = \boldsymbol{e}_i$。另一种更紧致的构造方案可参见 [4]。总之，无论是被动求解子集和，还是主动进行梯度投毒，主动方都可以获得被动方乱序后的分裂候选左子树 $L_{[[j,b]]_B}$ 这一中间结果。

在实际攻击中，攻击者无法进行大规模样本的梯度逆向攻击。因此，攻击者可以均匀选取一些样本进行梯度逆向，把其他样本梯度置为 0。此时，攻击者获得的是被动方乱序后的分裂候选左子树 $L_{[[j,b]]_B}$ 的一个子集。

我们认为，进行梯度逆向攻击需要消耗较高算力或需要对协议进行大量修改，因此只有在三级及以上的威胁场景中才会泄露。

3.2 子树匹配攻击

通过梯度逆向攻击，主动方可以获得乱序后的 $L_{[[j,b]]_B}$。为了获得顺序的 $L_{j,b}$，主动方还需要推断出 $L_{[[j,b]]_B}$ 对应的特征 j 和分桶 b。

通过观察分桶及分裂候选左子树的特征（如表 3），我们可以发现，每个特征各个分桶的分裂候选左子树具有如下性质[2]：(1) $\forall j_1, j_2, b, |L_{j_1,b}| \approx |L_{j_2,b}|$ (2) $\forall j, b, L_{j,b} \subset L_{j,b+1}$。我们可以利用这两个特征，恢复每个 $L_{[[j,b]]_B}$ 对应的下标。

表 3: 示例数据特征分桶及其分裂候选左子树

分桶 $Buk_{j,b}$	分桶 1	分桶 2	分桶 3
特征 1	$Buk_{1,1} = \{x_3, x_4, x_5\}$	$Buk_{1,2} = \{x_6, x_7, x_8\}$	$Buk_{1,3} = \{x_1, x_2, x_9\}$
特征 2	$Buk_{2,1} = \{x_1, x_2, x_4\}$	$Buk_{2,2} = \{x_6, x_7, x_8, x_9\}$	$Buk_{2,3} = \{x_3, x_5\}$

分裂候选左子树 $L_{j,b}$	分桶 1	分桶 2	分桶 3
特征 1	$L_{1,1} = Buk_{1,1}$	$L_{1,2} = Buk_{1,1} \cup Buk_{1,2}$	$L_{1,3} = Buk_{1,1} \cup Buk_{1,2} \cup Buk_{1,3}$
特征 2	$L_{2,1} = Buk_{2,1}$	$L_{2,2} = Buk_{2,1} \cup Buk_{2,2}$	$L_{2,3} = Buk_{2,1} \cup Buk_{2,2} \cup Buk_{2,3}$

首先，我们对每个分裂候选左子树的分桶进行推断：以元素个数作为排序依据，对所有分裂候选左子树集合进行排序。根据分裂候选左子树的第一个分布特征，可以推测排好序的候选左子树序列中，每 j 个分裂候选左子树具有相同的分桶号。对示例数据的推断过程如图 1所示。

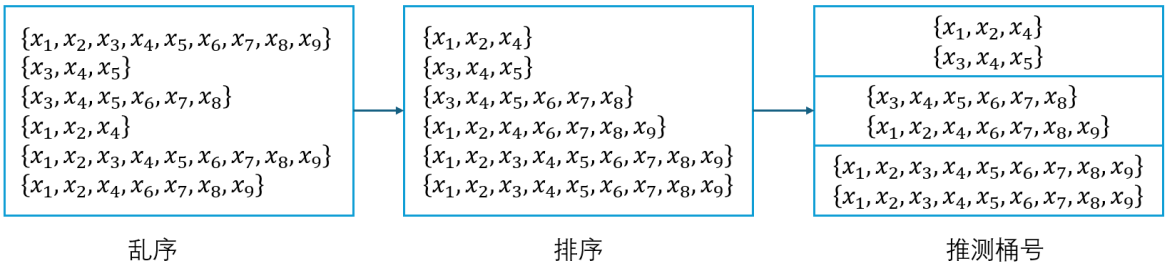

图 1: 候选左子树对应桶号推测过程示例

接下来，我们对每个分裂候选左子树的特征进行推断：以相邻两个桶号对应的候选左子树作为源点和汇点，对满足包含关系的候选左子树建边，并计算这个二分图的最大匹配。根据候选左子树的第二个分布特征：桶号相同的样本分属于不同的特征，而相同特征的候选左子树之间是包含关系。因此，匹配边关联的两个结点间具有相同的特征号。对示例数据的推断过程如图 2所示。

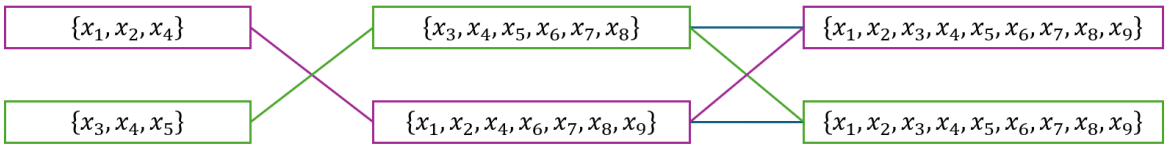

图 2: 候选左子树桶间特征匹配过程示例

经过以上操作，我们就可以恢复出每个候选左子树的特征等价类和和分桶号这一中间结果 $L_{\sigma(j),b}$。每个分桶可以通过 $Buk_{\sigma(j),b} = L_{\sigma(j),b+1} - L_{\sigma(j),b}$ 计算得到。进而每个样本的分桶值为 $\widetilde{x}_{i,\sigma(j)} = \arg_b i \in$

[2]这两个性质对3.1节得到的各个分桶分裂候选左子树子集也成立

$Buk_{\sigma(j),b}$。对示例数据的推断如表 4所示。虽然这些样本对应的具体特征语义仍不明确，但这些信息已经可以反应出关于数据的绝大部分信息。

表 4: 推测的分裂候选左子树及分桶值结果示例

推测分裂候选左子树 $L_{*,*}$	分桶 1	分桶 2	分桶 3
特征 j_1	$\{x_1, x_2, x_4\}$	$\{x_1, x_2, x_4, x_6, x_7, x_8, x_9\}$	$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$
特征 j_2	$\{x_3, x_4, x_5\}$	$\{x_3, x_4, x_5, x_6, x_7, x_8\}$	$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$

推断分桶值	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
特征 j_1	1	1	3	1	3	2	2	2	2
特征 j_2	3	3	1	1	1	2	2	2	3

当然，如果我们能获得一个样本的分桶值，即可再次进行未知特征和已知特征的二分图匹配，获得未知特征的语义。对示例数据的推断过程如图 3所示。

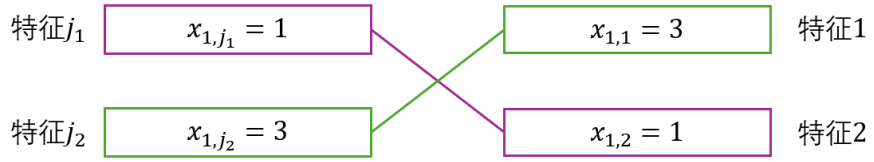

图 3: 特征语义匹配示例

综上，子树匹配攻击的流程如算法 1所示。我们对 FATE 的 SecureBoost 的默认数据集进行攻击。被动方的数据集包含 569 个样本、20 个特征，每个特征有 32 个分桶。恢复准确率可达到 78% ∼ 98%。

Algorithm 1 子树匹配

Require: 分桶数 B，特征数 J，乱序分裂候选左子树 $\{L_{*,*}\}$

对 $L_{[[j,b]]_B}$ 按照集合大小从小到大进行排序，得到序列 $L'_0, \ldots, L'_{b*j-1}$

for b=0 **to** B-1 **do**

$Buk_b = \{L'_i | \lfloor i/B \rfloor = b\}$ {推断候选左子树的分桶}

for b=0 **to** B-2 **do**

$V_l = Buk_b, V_l = Buk_{b+1}, E = \{(v_l \in V_l, v_r \in V_r) | v_l \subset v_r\}$ {建立特征匹配二分图}

$E_{b+1} = BipartiteMatching(V_l, V_l, E)$ {进行特征匹配}

for $L'_j \in Buk_0$ **do**

$Feature_j = \{L'_j\}$ {赋予特征号}

for b=1 **to** B-1 **do**

for $(L'_l, L'_r) \in E_b$ **do**

$j = \arg_j(L'_l \in Feature_j)$

$Feature_j = Feature_j \cup \{L'_r\}$ {扩展等价类}

3.3 特征分裂攻击

当参与训练的样本数超过梯度逆向攻击的能力范围时，梯度逆向攻击和子树匹配攻击只能推测投毒样本的分桶值信息。但是，子树匹配攻击让我们获得了顺序的 (j, b) 这一中间变量。因此，我们可以借助样

本分裂位置 p_i 这一中间变量，推测样本分桶值的相关信息。

注意到，攻击者可以在第 2 步，发送虚假的最优分裂点，强制被动方 B 根据特征 j 的第 b 个特征对样本进行划分。因此，攻击者可以根据推断出的 $L_{j,b}$，选择能二分特征 j 的分桶 b，将其作为最优分裂点发给 B。根据协议，攻击者就可以在第三步获得 $p_i = \mathbb{I}[x_{i,j^{\text{best}}} \leq s_{j^{\text{best}},b^{\text{best}}}]$。此中间变量泄露了每个样本第 j 个特征与 $s_{j^{\text{best}},b^{\text{best}}}$ 的大小关系。

3.4 量化中间结果泄露对数据集的信息损失

由于 SecureBoost 的中间变量只会泄露被动方数据的分桶信息，因此，我们使用基于熵的度量方法。

当被动方拥有 n 条数据，每条数据有 m 个特征，每个特征被划分为 B 个分桶，则原始数据的熵为 $E(\boldsymbol{X}) = n * m * log(B)$。

对于梯度逆向攻击和子树匹配攻击，如果攻击者每次可以对 k 条数据进行梯度逆向攻击，则可以完全确定 k 个样本所有特征的分桶信息，因此每一轮的泄漏量为 $k * m * \log(B)$。如果算法一共构建了 T 棵树、每棵树的深度为 D，则泄漏量为 $T * 2^D * k * m * \log(B)$。

对于特征分裂攻击，攻击者每次可以获得此节点所有样本某个特征的 1bit 信息。因此，每进行一层的特征分裂，泄漏量为 n。如果算法一共构建了 T 棵树、每棵树的深度为 D，则泄漏量为 $n * T * D$。

综合两攻击，其中有 $T * 2^D * k$ 个样本所有特征的所有分桶值完全泄露，有 $n - T * 2^D * k$ 个样本泄露了 T 个特征的部分信息。因此综合泄漏量为 $T * 2^D * k * m * \log(B) + (n - T * 2^D * k) * T * D$。

例如，如果被动方拥有 $n = 100,000$ 条数据，每个数据拥有 $m = 20$ 个特征，每个特征分为 $B = 32$ 个桶。双方构建了 $T = 2$ 棵决策树，每棵树的深度 $D = 3$。攻击者每次可以对 $k = 512$ 个样本进行梯度注入攻击。则原始数据的熵为 $n * m * log(B) = 10,000,000$，使用梯度逆向攻击和子树匹配攻击，泄漏量为 $T * 2^D * k * m * \log(B) = 819,200$，约泄露 8.192% 的信息。使用特征分裂攻击，泄漏量为 $n * T * D = 600,000$，约泄露 6% 的信息。综合考虑两攻击，则 SecureBoost 算法对原始数据的泄露量为 $T * 2^D * k * m * \log(B) + (n - T * 2^D * k) * T * D = 8192 * 100 + (100000 - 8192) * 2 * 3 = 819,200 + 550,848 = 1,370,048$，约泄露 13.7% 的信息。

4 可能的防御

被动方可以通过随机交换左右子树等方式，干扰子树匹配的成功率。

参考文献

[1] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, vol. 36, no. 6, pp. 87–98, 2021.

[2] W. Chen, G. Ma, T. Fan, Y. Kang, Q. Xu, and Q. Yang, "Secureboost+: A high performance gradient boosting tree framework for large scale vertical federated learning," *arXiv preprint arXiv:2110.10927*, 2021.

[3] J. G. Chamani and D. Papadopoulos, "Mitigating leakage in federated learning with trusted hardware," 2020.

[4] H. Weng, J. Zhang, X. Ma, F. Xue, T. Wei, S. Ji, and Z. Zong, "Practical privacy attacks on vertical federated learning," 2022.