中国科学: 信息科学 2024 年 第 54 卷 第 3 期: 514–565
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
评述

面向外包服务的可验证技术研究进展

孙奕[1], 杨帆[1], 陈性元[1,2*], 杜学绘[1], 林玮[1]

1. 信息工程大学密码工程学院, 郑州 450001
2. 密码科学技术国家重点实验室, 北京 100094
* 通信作者. E-mail: chxy302@vip.sina.com

收稿日期: 2022–10–31; 修回日期: 2022–12–20; 接受日期: 2023–05–22; 网络出版日期: 2024–03–06

摘要 随着数字经济时代的到来, 数据已成为重要生产要素, 将数据外包给第三方存储、管理、分析、计算等应用越来越广泛, 随之而来的外包数据、外包计算等外包服务的可信性问题也日益突出, 可验证技术被视为判定外包服务可信的一种有效且具有广泛应用前景的技术. 该技术发展迅速、研究内容丰富、技术多样、种类繁多, 且随着区块链、人工智能、隐私计算等新型外包服务需求的出现, 为可验证技术带来了新的挑战和发展契机. 本文对面向外包服务的可验证技术研究进展进行了梳理与总结. 首先梳理总结给出了面向外包服务的可验证技术框架、分类方法、安全目标与评价体系. 接着从外包数据和外包计算两个角度给出可验证技术的形式化定义, 并分别论述其典型的可验证方案. 然后从外包数据的角度, 以验证数据类型为主线结合实现的功能和方法分别详细论述了不同数据类型、不同场景、不同实现方式的外包数据持有性验证和完整性验证技术; 从外包计算的角度, 根据外包计算类型的不同, 总结归纳了 5 种典型外包计算的可验证技术特点及研究进展, 并基于提出的评价体系对典型方案进行了详细的对比分析. 最后结合新兴技术和应用热点, 展望了可验证技术在前沿方向上的发展趋势、应用前景及其面临的挑战.

关键词 外包服务, 外包数据验证, 可验证计算, 可认证数据结构, 动态可验证结构

1 引言

随着数字经济时代的到来, 数据已成为重要生产要素, 数据量剧增, 数据分析算法更加复杂, 数据算力要求更高, 仅靠用户本地资源难以支撑, 考虑到成本和代价因素, 许多用户正在将本地数据以及一些复杂计算委托给第三方服务提供者 (如云平台) 进行存储、管理、分析与计算等. 用户在享受高效、方便、灵活的外包服务同时, 也带来了一些数据安全隐患.

将数据外包给第三方服务提供者的想法是由 Hacigumus 等 [1] 首先提出的, 由于数据外包给第三方服务提供者, 数据所有权与控制权分离, 数据拥有者不再具有外包数据的物理控制权, 而且外包到第

引用格式: 孙奕, 杨帆, 陈性元, 等. 面向外包服务的可验证技术研究进展. 中国科学: 信息科学, 2024, 54: 514–565, doi: 10.1360/SSI-2022-0360
Sun Y, Yang F, Chen X Y, et al. Research progress of verifiable technologies for outsourcing services (in Chinese). Sci Sin Inform, 2024, 54: 514–565, doi: 10.1360/SSI-2022-0360

三方存储和处理的数据往往处于公开的环境中, 面临诸多内部与外部的安全威胁. 外部安全威胁来自一些恶意攻击 [2] 和可能会导致数据丢失或服务中断的意外故障, 如自然灾害[1)]、电力中断[2)]等. 内部安全威胁来自使用外包服务的客户组织内部 (现有/前雇员、分支机构等) 以及软件本身存在的问题. 由存储系统软件错误、硬件故障或是操作不当意外删除[3)]等引起的不管是恶意还是无意的数据损坏都是不容忽视的. 因此, 为了保证数据没有被篡改或损坏, 需要对外包数据进行完整性验证. 对于外包计算, 指的是能力有限的用户将复杂的计算外包给服务提供者, 以云服务器为例, 由于云服务器的操作细节对用户不是完全公开透明的, 用户将计算任务委托给云服务器后, 无法确定云服务器是否正确地执行了计算过程, 云服务商可能为了节省计算成本而不去执行一些复杂的计算, 只是将一个与真实结果不可区分的数值返回给用户, 用户无法判断外包服务器返回的计算结果是否正确. 外包计算为了确保外包服务器确实按照要求正确执行了计算, 需要对计算结果的正确性进行验证. 同时还要保证的验证效率合理, 即用户的验证开销要小于用户自己计算这些数据的开销. 近年来, 随着分布式存储技术的快速发展, 越来越多的去中心化存储网络 (decentralized storage networks, DSNs), 如 Filecoin, Storj 和 Sia 等被广泛应用, 外包服务在 DSNs 中也有所体现, 数据拥有者将数据外包给 “矿工” 节点, 对区块链网络存储数据完整性如何检查, 区块链网络内部各节点是否确实提供了其承诺的服务等问题都给外包服务带来了新的挑战.

此外, 在政策法规方面, 我国相继出台的《网络安全法》《数据安全法》以及《个人信息保护法》等对数据安全与隐私保护提出了新要求. 对外包数据是否安全, 外包数据或计算过程是否存在隐私泄露, 外包计算结果是否可信、可靠等要求也上升到了更高的层面. 因此, 如何使用户在本地不存储数据的情况下, 又能够在轻量开销下验证不可信/半可信的外包服务返回结果的可信性和可靠性已成为亟待解决的问题, 也是近年来数据安全、云计算安全、人工智能安全、区块链安全等领域研究的热点和难点问题.

可验证技术被视为判定外包服务可信的一种有效且具有广泛应用前景的技术. 目前虽已有论文对其进行综述, 但该技术发展迅速、研究内容丰富、技术多样、种类繁多, 且随着区块链、人工智能、隐私计算等新型外包服务需求的出现, 对可验证技术带来了新的挑战和发展契机. 基于此, 本文通过追踪该技术国内外最新研究成果, 尝试对现有研究成果进行归纳与分析, 梳理总结给出了一种面向外包服务的可验证技术框架, 从基本原理、关键技术、功能实现到应用服务层层深入, 系统、全面的对现有外包服务可验证技术的研究现状进行归纳总结, 对其中的典型方案进行详细分析与比较, 对其未来发展趋势、研究热点、应用前景及其面临的挑战进行展望, 为后续可验证技术的研究与发展抛砖引玉.

本文组织结构如下, 第 1 节介绍了研究背景. 第 2 节描述了面向外包服务的可验证技术框架与分类方法、安全目标, 以及方案评价指标. 第 3 节对外包数据可验证技术研究进展进行了归纳与对比分析. 第 4 节对外包计算可验证技术研究进展进行了归纳与对比分析. 第 5 节在新应用背景下对未来研究趋势以及面临的挑战进行了分析和展望. 第 6 节总结全文.

1) Amazon Data Center Loses Power During Storm. 2012. https ://www.datacenterknowledge.com/archives/2012/06/30/amazon-data-center-loses-power-during-storm.

2) 亚马逊 AWS 再次宕机 Coinbase 和 Slack 等受影响. 2021. http://www.bianews.com/news/details?id=107250.

3) 腾讯云 “数据丢失” 事件引发的进一步思考. 2018. https://www.sohu.com/a/246805026_100175238.

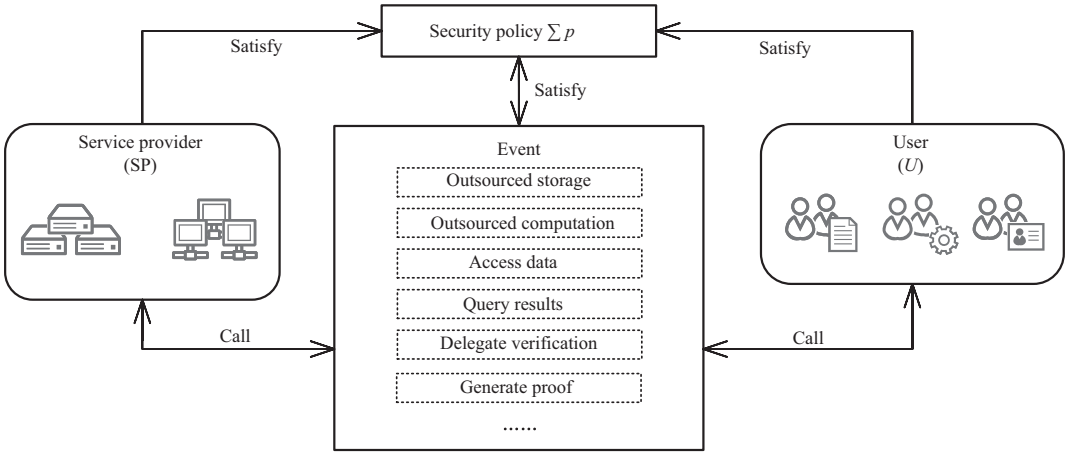

图 1　系统模型

Figure 1　System model

2　可验证技术研究框架

2.1　系统模型

为了进一步对面向外包服务的可验证技术进行深入研究, 我们形式化描述一个基本的可验证系统模型, 将其抽象为两个实体、一组事件集合和一组安全策略, 可以用四元组 $(\mathrm{SP}, U, E, \Sigma p)$ 表示, 如图 1 所示. SP 表示服务提供者 (service provider), 负责提供海量的存储服务和高效的计算服务. U 表示用户 (user), 可以将大量数据委托给 SP 维护, 或将复杂计算任务委托给 SP, 也可以根据相应的权限去查询使用数据或获取计算结果. E 表示 SP 和 U 调用的事件集合 (event), 主要包括: U 将数据集存储在 SP 上或将复杂的计算任务外包给 SP; SP 可以对数据进行维护与管理, 并利用本地存储的数据直接进行计算; 有需求的 U 可以向 SP 发起使用请求, 这类需求可以是访问数据也可以是委托计算任务; SP 根据需求查询数据或按照委托计算以及生成验证证明; U 可以自己验证, 也可以将验证任务委托给第三方; 验证者可以根据证明验证返回数据的完整性和计算结果的可信性等. Σp 表示安全策略集合 (security policy), 包括有无信任机构策略、密钥分发策略、访问控制策略等. 在整个面向外包服务的验证过程中, 实体 SP, U 及其之间的事件集合 E 都需要满足安全策略集合 Σp.

2.2　可验证技术框架

通过对现有外包服务可验证方案的研究, 可以从需求、场景、功能、外包服务类型以及实现方式等多种主题来研究可验证方案, 且一种可验证方案通常也会涵盖多个主题, 为了对现有外包服务可验证方案进行全面而清晰的分析, 我们试图将现有研究方案融合在一个研究框架下进行论述与比较, 按照从内在原理到外在应用的层次划分方法, 总结给出了一种可验证技术框架, 如图 2 所示. 根据可验证技术的实现原理主要分为软件和硬件两大类; 根据外包类型可以分为外包数据和外包计算; 根据实现功能主要包括支持动态更新、支持查询验证、可公开验证、可委托验证、隐私保护、访问控制等; 根据验证方式可以分为确定验证和概率性验证, 概率性验证多用于外包数据审计; 根据用户应用模式可分为: 一对一、一对多、多对一和多对多.

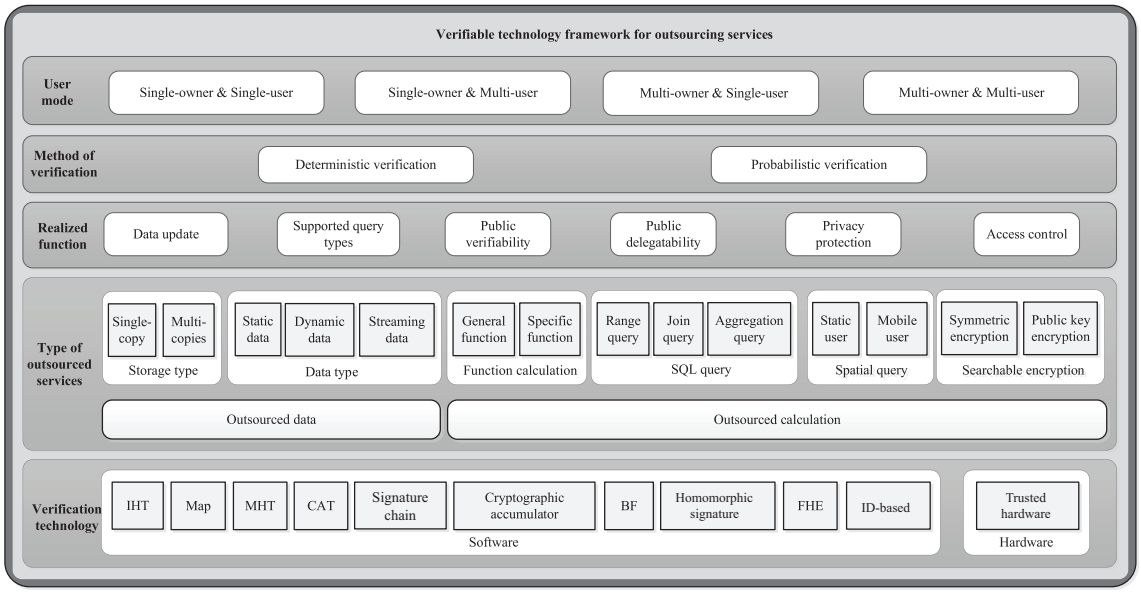

图 2 面向外包服务的可验证技术框架

Figure 2 Verifiable technology framework for outsourcing services

2.3 可验证技术分类

基于以上框架, 根据现有技术解决问题的不同以及关注的重点不同, 从关键技术、服务类型、实现功能以及在新技术中的应用等 4 个方面对现有可验证技术进行分类, 如图 3 所示.

从实现的关键技术角度进行分类, 可以分为基于索引哈希表 (index-Hash table, IHT) 的可验证技术、基于 Merkle 哈希树 (Merkle Hash tree, MHT) 的可验证技术、基于变色龙认证树 (chameleon authentication tree, CAT) 的可验证技术、基于同态签名 (homomorphic signature) 的可验证技术、基于全同态加密 (fully homomorphic encryption, FHE) 的可验证技术、基于身份 (ID-based) 的可验证技术, 以及基于可信硬件的可验证技术等.

从外包服务类型角度进行分类, 可分为外包数据完整性验证和外包计算可信性验证. 针对外包数据的完整性验证再从数据类型角度进行分类, 又可以分为针对静态数据、动态数据、流数据、空间数据的完整性验证技术. 静态数据是指在整个外包存储期间保持不变的数据. 动态数据是指支持对外包数据集进行动态更新操作, 例如修改、删除和插入. 流数据则是一种实时产生、连续、不可预测、数据量趋于无限的数据类型, DO 只能以流的方式将产生数据逐步外包给 SP, 与静态数据以及动态数据在初始时即可获取全部数据集不同, 流数据用户只能 “看到” 当前的元素, 以前的元素不能被存储, 未来元素不可预测, 动态数据与流数据可验证技术特点比较如表 1 所示. 空间数据指的是空间数据库中存储的地图数据等信息, 往往一段时间内并不需要更新. 此外, 为了解决中心服务器单点失效以及重要数据备份等问题, 现实中外包数据可能不止存储在一台服务器上, 而是在多个服务器中存储多个副本, 因此外包数据完整性验证又可分为单副本完整性验证和多副本完整性验证.

针对外包计算类型, 可以分为面向函数计算的可验证技术、SQL 查询验证技术、空间查询验证技术、可搜索加密验证技术和基于可信硬件的查询验证技术. 面向函数计算的可验证技术函数类型可以分为一般函数和具体函数, 一般函数指的是布尔函数, 具体函数主要的类型有序列比较、多项式计算、矩阵乘法和密码算法等. SQL 查询验证技术从支持的查询类型可以分为针对特定 SQL 查询验证的范

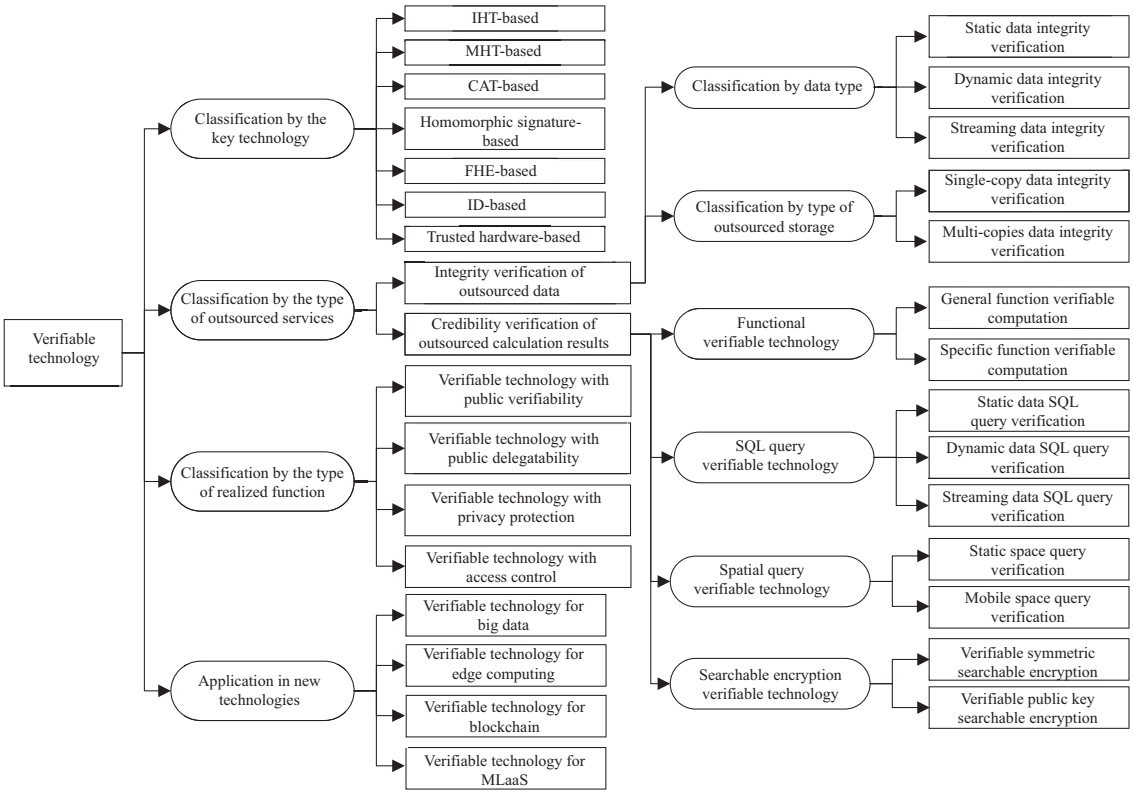

图 3　面向外包服务的可验证技术分类

Figure 3　Verifiable technology classification for outsourcing services

表 1　动态数据与流数据查询验证特点比较

Table 1　Comparison of dynamic data and streaming data query verification characteristics

Comparative item	Dynamic data	Streaming data
Update rate	Lower	High
Update method	Off-line arbitrary form	Real-time incremental
Data access	Random	Sequential
Query mode	Random	Continuously
Processing mode	Query driven (pull mode)	Data driven (push mode)
Storage location	Hard disk storage (unlimited)	Memory storage (limited)
Query status return	Non-real-time	Real-time (delay sensitive)

围查询验证、连接查询验证、聚合查询验证等, 以及通用 SQL 查询验证, 从查询的数据类型又可以分为静态数据 SQL 查询验证、动态数据 SQL 查询验证和流数据的 SQL 查询验证方案. 空间查询验证可以分为静态空间查询验证和移动空间查询验证. 可搜索加密验证技术可以分为对称加密和公钥加密的可搜索加密技术.

从实现的功能角度进行分类, 又可以分为具有公开可验证性 (public verifiability)、公开可委托性 (public delegatability)、具有隐私保护、具有访问控制的可验证技术. 公开可验证性是指任何拥有相关公开参数的用户都能够验证结果, 不仅支持用户本身对服务器输出进行验证, 而且可以在用户本地

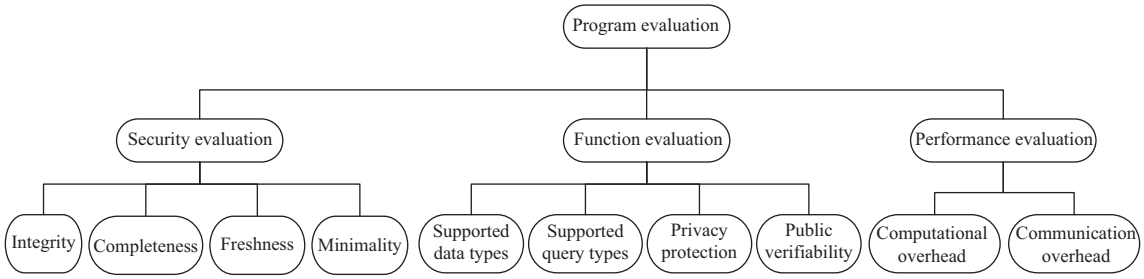

图 4 方案评价指标

Figure 4 Program evaluation indexes

资源不足时将验证过程委托第三方执行, 同时可以有效预防各参与方单方面的恶意抵赖. 公开可委托性是指用户只需要使用公开信息就可以将计算任务委托给服务器, 在多用户模式中应用更加广泛. 具有隐私保护的可验证技术是指在外包数据或外包计算结果的可信性验证过程中, 同时满足隐私保护的要求, 防止敏感数据或用户个人信息泄露. 具有访问控制的可验证技术是指查询验证返回的结果中不可以包含该用户无权访问的数据.

下面基于以上分类, 对可验证技术典型方案以及新技术背景下可验证技术的应用进行详细描述与分析, 并根据评价标准对比分析其优缺点.

2.4 可验证方案评价

对于可验证方案的优劣主要从 3 个角度评价, 安全性评价、功能评价和性能评价, 具体评价指标如图 4 所示.

一、安全性评价. 可验证方案安全性评价指标主要包括: 完整性、完备性、新鲜性和最小化. 根据不同的应用场景或安全目标, 一个可验证方案的安全性评价指标可以是以上指标的一种或多种的组合, 但完整性是必须满足的最基本的安全指标.

(1) 完整性. 完整性验证指标又可以进一步分为: 概率性验证和确定性验证. 所谓概率性验证是指通过随机抽样验证外包数据或外包计算结果的完整性. 确定性验证是指验证者能够依据外包服务器返回的验证证据对返回结果进行完整性验证. 概率性验证方案, 即验证通过也不能 100% 保证查询结果的真实性, 概率性方案往往会有较低的开销、更强的计算表达力, 却可能存在验证不完全的情形.

(2) 完备性. 在有些外包服务验证中, 由于验证返回的结果可能不只是一个值 (如, 范围查询验证), 此时在评价其可验证指标时还需考虑完备性. 完备性是指外包服务器所返回结果必须包含所有满足查询条件的所有数据记录. 敌手或者恶意的服务提供者不能随意删除返回结果中的数据记录或者是只在部分数据子集中执行查询. 对于外包数据库的可验证模型来说, 查询结果的完备性是非常重要的.

(3) 新鲜性. 新鲜性是指当外包数据有动态更新时, 可验证方案要能够验证外包服务器所返回的结果是来自最新的数据版本.

(4) 最小化. 最小化是指只返回满足查询条件的记录和属性值, 而不会返回多余的数据. 最小化评价指标是为了避免违反访问控制规则, 有些方案虽然会返回正确的查询结果, 但是一些额外的信息会作为辅助验证信息返回给查询用户, 最典型的如基于 MHT 的范围查询, 为了保证完备性, 就需要返回一些边界值作为辅助验证信息, 而这些边界值往往就造成了额外的数据信息泄露.

二、功能评价. 功能性评价主要是指对可验证方案能够实现哪些功能进行评价, 主要包括以下 4 个方面: (1) 是否具有公开可验证性. (2) 支持验证的数据类型. 如, 支持动态数据更新、支持流数据.

(3) 支持查询的种类. 如, 范围查询、聚合查询、关键字查询等. (4) 是否具有隐私保护.

三、性能评价. 性能评价主要从方案的计算开销与通信开销来考虑, 根据系统模型三方实体可归纳为以下 6 点: (1) 为了构造可验证结果的证据, 用户构造可认证数据结构的计算开销. (2) 用户向第三方服务器传输查询验证请求的通信开销. (3) 服务器在为验证请求生成证据时产生的计算成本. (4) 认证信息和签名在服务器上的存储成本. (5) 将证据从服务器传输到用户的通信成本. (6) 用户验证服务器返回证据花费的计算成本. 但在实际中, 衡量一个方案是否高效, 我们更多考虑数据拥有者与用户所花费的通信开销 ((2) 和 (5)) 与计算开销 ((1) 和 (6)).

随着新技术、新应用的发展, 可验证方案的评价指标也在不断的完善, 特别是近年来随着具有隐私保护的联邦学习、区块链下的存储验证、基于可信硬件的查询验证等新兴可验证方案的发展, 可验证方案的隐私保护强度、可部署性、是否支持用户组动态变化等都逐渐纳入到了方案评价指标中, 由于有些方案的评价指标还尚处于探索阶段, 因此在本篇综述中暂未将其纳入到评价体系中.

3 外包数据可验证技术研究进展

随着云计算、大数据、边缘计算等技术的快速发展与普及, 越来越多的学者致力于研究外包数据完整性验证技术, 提出了许多外包数据完整性验证方案. 为了便于研究与分析, 本节以验证数据类型为主线, 结合外包数据存储模式、实现的功能和关键技术, 对现有典型方案进行归纳与分类, 主要分为: 面向单副本数据的完整性验证技术、面向多副本数据的完整性验证技术和面向流数据的实时验证技术, 下面分别针对每一类可验证技术进行详细论述, 总结其工作原理与特征, 并从安全性、功能、性能等方面对现有典型方案进行对比分析, 为后续研究奠定基础.

3.1 外包数据可验证的形式化定义

一个基本的外包数据可验证方案是一个在用户和服务提供者之间执行的两方协议, 用户将数据存储在服务提供者中, 并生成相关元数据. 为了验证存储在服务提供者中的数据, 验证者向服务提供者提出挑战, 服务提供者将验证证据返回给验证者, 验证者根据证据进行验证并输出结果. 根据文献 [3] 给出外包数据可验证方案的形式化定义.

定义1 (外包数据可验证方案) 一个外包数据可验证方案 VD(Setup, Append, Query, Verify, Update) 由以下 5 个算法所组成.

• KeyGen$(1^\lambda) \to$ (pk, sk): 用户 U 运行, 以随机密钥生成算法和安全参数 λ 为输入, 生成一组公钥 pk 和私钥 sk, 并将公钥 pk 发送给服务提供者 SP.

• TagGen$(\mathrm{sk}, F) \to (\{\tau_x\}_{x\in F})$: 用户 U 运行, 利用私钥 sk 为数据文件 F 的每个数据块 x 生成标签 τ_x, 并将所有标签发送给服务提供者 SP.

• Callenge$(1^\lambda) \to$ chal: 用户 U 运行, 生成随机挑战 chal, 发送给服务提供者 SP.

• ProofGen$(\mathrm{chal}, \mathrm{pk}, \{\tau_x\}_{x\in F}) \to \sigma$: 服务提供者 SP 运行, 根据 chal 利用公钥 pk 和相应数据块标签 τ_x 生成证明 σ.

• Verify$(\mathrm{sk}, \mathrm{pk}, \mathrm{chal}, \sigma) \to$ ('success', 'failure'): 验证算法使用私钥 sk 或公钥 pk 根据 chal 验证 σ, 如果验证成功输出 'success', 否则输出 'failure'.

定义2 (外包数据可验证方案安全性) 对于一个外包数据可验证方案 VD, 我们定义攻击者 A 在以上实验中优势为: $\mathrm{Adv}_A^{\mathrm{Verif}}(\mathrm{VD}, f, n) = \mathrm{Prob}[\mathrm{Exp}_A^{\mathrm{Verif}}[\mathrm{VD}, f, n] = 1]$.

一个外包计算可验证方案 VD 对于数据文件 F 是安全的, 并且对于任意攻击者 A 运行在概率多项式时间.

$\mathrm{Adv}_A^{\mathrm{Verif}}(\mathrm{VD}, f, \lambda) \leqslant \mathrm{negli}(n)$, 这里 negli() 表示具有输入参数的可忽略函数.

更具体地, 如果恶意的 SP 无法让用户相信伪造或损坏的数据文件 F' 就是原始的上传的数据文件 F, 则认为外包数据可验证方案 VD 是安全的.

3.2 面向单副本数据的完整性验证技术

3.2.1 静态数据完整性验证方案

由于数据外包, 数据所有权与控制权分离, 数据拥有者在本地不再存储数据, 传统的验证数据完整性方法变得不可行. 为了解决远程不可信服务器中外包数据的完整性验证问题, 2007 年 CCS (Computer and Communications Security) 会议上, Ateniese 等 [3] 首先提出了可证明数据持有性模型 (provable data possession, PDP), PDP 模型的主要思想是利用 RSA (Rivest-Shamir-Adleman) 签名算法构建同态可验证标签 (homomorphic verifiable tags, HVT), 使得用户在不取回数据的情况下, 对远程不可信服务器 (如: 云) 中数据的完整性进行验证, 并通过随机抽样策略和聚合验证减少了计算开销和通信开销, 使模型应用于验证大型数据集的数据持有率成为可能, 但由于插入或删除新数据块时需要重新生成位于所操作块后面所有数据块相应的验证元数据, 更新成本较高, 因此该模型仅适用于验证静态数据 (如, 已归档数据、仓库数据) 或仅在最后添加新数据. 与此同时, Juels 和 Kaliski [4] 提出了数据可恢复性证明模型 (proof of retrievability, POR), POR 模型的主要思想是通过在文件中嵌入 “哨兵” 实现对文件持有型的概率性证明, 并利用纠删码技术实现部分出错数据的可恢复性. 这两类验证模型都是概率性验证, 且都是应用于静态数据的完整性验证. 所不同的是, PDP 模型基于公钥技术, 考虑了文件的公开审计问题, 可应用于公共数据库, 例如图书馆、存储库或档案, 交互可以重复任意次数. POR 的使用仅限于机密数据, 不支持公开审计, 而且由于在每次查询时, 必须向服务器显示 “哨兵” 及其在数据库中的位置 – 无法重用显示的 “哨兵”, 因此查询的数量是有限的, 并且需要预先确定.

后来, 针对是否支持动态数据验证 [5~18], 是否支持公开可验证性 [8~15,17,18]、是否支持具有隐私保护的第三方审计 [19~26], 以及多用户云数据共享存储审计 [27~30] 等问题, 研究者展开了一系列的深入研究. 一个可验证方案中往往可以同时具有以上多个功能, 本文重点关注可验证方案对动态数据验证问题的解决方法, 是否支持动态数据验证是外包数据服务能否进一步实用化的关键, 是外包数据可验证技术亟待解决的难点问题之一. 虽然 2008 年 Ateniese 等 [6] 基于 PDP 模型利用对称加密也实现了一个支持动态修改的数据完整性验证方案, 但该方案对查询数量施加了先验限制, 仅支持了一些简单的修改操作, 不能支持数据的插入、删除等操作且审计次数有限, 也不支持第三方公开审计. 文献 [31] 考虑了分布式场景中的动态数据存储, 提出的质询响应协议既可以确定数据的正确性, 也可以定位可能的错误, 但该方案也只支持部分动态数据操作, 不支持完全动态的数据操作. POR 模型由于预计算 “哨兵” 的引入阻碍了实现动态数据更新的发展, 且他们的方案也不支持公开审计. 为此, Shacham 和 Waters [32] 设计了一个改进的 POR 方案, 为文献 [4] 中定义的安全模型中提供了完整的安全性证明. 他们使用由 BLS (Boneh-Lynn-Shacham) 签名构建的可公开验证同态认证器 [33], 基于此, 可以将证明聚合为一个较小的认证器值, 并实现公开审计, 但该方案依然只针对静态文件, 不支持动态数据操作.

为了在有效地验证数据完整性的情况下, 同时能够支持数据的动态操作, 在后续的研究方案中引入了可认证数据结构 (authenticated data structures, ADS) 的概念. 目前常用的 ADS 的主要方法有索引哈希表 (IHT)、跳表以及 Merkle 哈希树 (MHT) 等. 下面重点分析几类典型的动态数据可验证方案.

3.2.2 基于索引哈希表的动态数据验证方案

经研究发现, 直接扩展当前的 PDP 模型和 POR 模型来支持动态数据操作可能会导致安全漏洞. 尽管研究人员面临许多困难, 但是支持外包数据的动态更新操作对外包存储数据服务的实际应用至关重要, 在实际应用中外包数据不会是一直静态不变的, 需要进行动态更新, 因此需要一种支持动态数据更新验证的方案来证明外包数据存储的完整性和可靠性. 2015 年, Erway 等 [7] 通过改进 IHT 构建一种分级认证跳表 (rank-based authenticated skip list, RASL), 基于 RASL 首次提出了一种支持所有动态操作的动态 PDP 方案 (DPDP), 用于支持不可信服务器上存储数据的可证明更新, 他们消除了 PDP 模型 "标记" 计算中的索引信息, 并采用 RASL, 在验证过程之前首先验证被质疑或更新块的标记信息. 为了实现对云平台下外包数据的动态审计, 文献 [5] 基于 IHT、碎片结构和随机抽样设计了一种支持外包数据动态更新的公开可验证方案, 实现审计外包数据的完整性. IHT 支持数据的修改, 但插入和删除数据时需要重新调整表的结构, 需重新计算部分受影响数据块的标记, 因此为了维护 IHT 将产生大量的计算开销. 虽然文献 [5, 7] 支持动态数据验证, 但频繁的更新操作仍然会给验证者带来高昂的计算和通信成本. 为此, Sookhak 等 [8] 提出了一种基于 IHT 的新型数据结构, 称为分治表 (divide and conquer table, D&CT), 并利用 D&CT 高效地实现了可更新数据完整性验证 (包括插入、追加、删除和修改) 并且以最小的计算成本应用于大规模数据存储. D&CT 由逻辑索引 (LI) 和版本号 (VN) 两部分组成, LI 表示服务器中块的物理位置, VN 根据更新次数表示数据块的当前版本. 在将数据块外包给 SP 之前, DO 必须创建 D&CT, 将新数据块的标识信息 (LI = I, VN = 1) 添加到 D&CT 中. 此外, 当 DO 为每个数据块计算标签时, 通过计算 $\tau = \text{fid}||\text{LI}||\text{VN}$ 将数据的抽象信息插入到标签中, 来防止恶意服务器伪造标签来欺骗验证者. 为了克服在大规模数据中插入或删除操作时, 移动数据块导致大量的计算开销, 将 D&CT 划分为 k 个数据结构, 其中每个数据结构都能够存储 $[m/k]$ 个数据块, 当在第 i 个块之后插入新块时, 由于数据块存储在 k 个不同的表中, DO 只需要移动 $[m/k] - i$ 数据块, 减小了额外的计算开销. 以上方案验证方案都是用于抽样审计, 是一种概率性验证.

3.2.3 基于向量承诺的动态数据验证方案

数据库即服务 (database as a services, DaaS) 的新型计算范例 [1] 使得缺乏专业知识或计算资源的用户可以将自身的数据和管理需求外包给第三方服务提供者. 带来的安全问题是如何保障外包数据库的完整性. 为了解决这个问题, 2011 年, Benabbas 等 [34] 首先提出了可验证数据库 (verifiable database, VDB) 的概念, 通过计算一个多项式来验证数据库的检索值, VDB 概念的提出使得资源受限的用户能够安全地将一个非常大的数据库外包给不受信任的服务器, 以便以后可以检索数据库记录并通过分配新值来更新记录. 如果服务器试图篡改数据库, 用户将极有可能检测到它. VDB 与 PDP/POR 的初始结构密切相关, 主要区别在于 PDP 和 POR 只关注数据集的完整性, 而不是庞大的数据库. 通常, 我们可以基于消息认证码或静态数据库的数字签名构建高效的 VDB 方案. 然而, 如果用户经常对数据库执行更新, 则必须有一个机制来撤销提供给服务器的旧签名. 否则, 恶意服务器可以利用以前有效的数据库记录和相应的签名来响应用户的当前查询. 这称为 VDB 上的向后替换更新攻击.

为此, Catalano 和 Fiore 等 [35] 在 2013 年 CRYPTO (International Cryptology Conference) 会议上第 1 次提出一种密码学原语向量承诺 (vector commitment, VC), 可以对一个由 q 个值组成的有序序列 $(m_1, \ldots, m_q)$ 进行承诺, 而之后只能在特定位置打开承诺 (例如, 证明 m_i 是第 i 个承诺信息). 他们利用向量承诺实现了一个支持数据动态更新的 VDB 方案. Chen 等 [10] 指出 Catalano-Fiore 的方案容易受到正向自动更新攻击 (forward automatic update, FAU), 即任何人都可以对数据库进行更新. 为

了解决这一问题他们在向量承诺方案的基础上提出了一种基于素数阶双线性群的承诺绑定 (binding commitment) 机制. 与文献 [35] 相比降低了一部分效率, 但方案支持公开验证且可以抵抗 FAU 攻击, 还具有可追责性, 即在验证失败时能够判断出是服务商篡改了数据还是用户诬陷服务商. 此外, Chen 等 [14] 还提出了一种新的应用情形, 当数据库发生频繁的小修改时, 用户必须随时在服务器上重新计算更新加密数据. 他们将向量承诺与先加密后增量的 MAC 加密模式 (encrypt-then-incremental MAC code of encryption) 相结合提出了一个通用的可验证的增量更新数据库框架 (verifiable database with incremental updates, Inc-VDB), 来减少用户的计算成本.

然而, Catalano-Fiore 的方案 [35] 只能支持替换操作, 不能支持插入和删除等常见的更新操作. Miao 等 [12] 通过使用承诺绑定和分层承诺 (hierarchical commitment) 技术支持替换、插入和删除等所有更新操作. 基于多级的分层承诺思想, 新插入的数据存储在分层承诺新的一级中, 删除操作可以看作是值为 0 的一种特殊的替换, 还引入了新标签 tag 来确定每个指标在某一层的位置. 由于插入 (或删除) 操作的开销几乎与替换操作的开销相同, 因此该方案与现有的向量承诺方案一样高效, 同时满足公开和私有可验证性.

此外, 当用户不断地在同一索引处插入数据记录时, 基于分层承诺的方案 [12] 中层数呈线性增加, 计算和存储开销也会线性增加, 在文献 [10,35] 中也具有同样的问题. 这完全违背了安全外包服务的效率规则, 即在整个外包算法过程中, 用户为了验证外包存储数据完整性的额外开销 (初始一次性预处理阶段除外) 要尽可能小, 不能超过直接计算或存储数据的开销. 为了支持各种更新操作, 同时考虑到方案计算存储开销过大的问题, Miao 等 [36] 基于 Merkle 算数哈希树 (Merkle sum Hash tree, MSHT) 构建了一种新的支持所有更新操作的 VDB 方案. MSHT 中的叶子节点仅存储数据记录数和更新操作数, 避免了 MHT 中昂贵的更新计算, 且所有的更新操作都可以被视为替换的特殊情况. 但是, 该方案不支持公开验证. 于是, Wang 等 [37] 将向量承诺与 Merkle 间隔哈希树 (Merkle interval Hash tree, MIHT) 相结合, 构造出一个既能抵抗正向自动更新攻击, 又支持各种动态更新操作的可公开验证 VDB 方案. 同时, 该方案基于素数阶双线性群的恒定大小密码假设, 具有更高的安全性. 同样, 为了解决文献 [12] 中方案计算存储开销过大的问题, Shen 等 [9] 将 BLS 签名和 IHT 相结合提出一种支持动态操作的可验证数据库方案, 支持数据所有者对数据库操作结果的可验证性, 该方案还利用多项式承诺 (polynomial commitment) 实现了在不泄露数据库信息的情况下支持数据公开验证. 还有, Chen 等 [13] 提出了一种新的原语承诺可逆 Bloom 过滤器 (committed invertible Bloom filter, CIBF), 融合了可逆 Bloom 过滤器和向量承诺两个概念, 将大型数据集上的插入和删除操作都看作是一种特殊的替换操作, 构造了一个高效的且支持任何动态操作的公开验证 VDB 方案.

Zhang 等 [11] 指出 Catalano-Fiore 的方案还存在着初始化计算开销较大的问题, 且不适合于当今大数据量环境下的存储, 因此他们提出了向量承诺树 (vector commitment tree, VCT), 一种类似于二叉查找树 (binary search tree, B-tree) 的结构. VCT 中的每个节点都是一个维度为 q 的向量承诺结构, 所有节点都通过向量承诺保证其更新可验证性, 其所有的向量承诺计算复杂度独立于数据的存储规模只与初始设定的 q 有关. 他们基于 VCT 设计了一种可扩展更新的分层可验证数据库方案 (hierarchical verifiable database, HVDB), 除了更新可验证和追责性还支持分层验证和篡改记录的定位. 该方案采用分层数据管理结构可以很方便地融入访问控制规则, 方便应用于分层管理的数据场景.

3.2.4 基于 MHT 的动态数据验证方案

另一个用于支持动态数据更新验证的可认证数据结构是 MHT. 为了同时支持公开审计和全动态数据操作, Wang 等 [15] 通过对 MHT 结构的变换, 改变现有验证证据的存储模型, 来实现全动态数据

操作, 并利用 BLS 算法构建数据块的同态认证标签, 实现公开审计. 虽然文献 [15] 能够支持数据块的完全动态更新, 但只支持固定大小的数据块更新, 每次细粒度更新都会导致整个数据块重新计算, 需要额外的存储和通信开销. 为了支持对云平台下大数据的细粒度更新验证要求, Liu 等 [17] 提出一种具有标记的默克哈希树方案. 该方案也是基于 BLS 算法为每一个分块数据计算同态线性验证标签, 与之前需要均分数据块大小的方案不同, 该方案中数据块大小是任意的, 在 MHT 叶子结点中除了保存数据分块信息以外, 还增加了标记信息, 该标记信息用于记录对每个数据分块再次细分后的数据块个数信息及位置信息, 从而利用该标记实现对每个数据块中数据的细粒度更新, 而无需更新整个数据块信息. 此外, 还在验证者和服务器之间添加了授权来确保查询挑战确实来自合法验证者, 防止恶意攻击者通过伪造查询挑战服务器来获得用户隐私数据. 与文献 [17] 不同, Chen 等 [18] 提出一种平衡更新树来支持细粒度数据更新验证, 平衡更新树消除了每次动态更新操作后复杂的验证, 减少了计算和通信开销. 此外, 该方案还采用基于 RSA 的同态加密算法构造标签, 在验证时无需发送数据块的任何组合, 使得验证者无法得到外包数据相关信息, 实现了具有隐私保护的公开验证. 此外, 还有 Anthoine 等 [38] 基于 POR 模型并结合 MHT 提出了第 1 个不需要对存储在服务器上的数据进行任何特殊编码的动态 POR 系统, 支持数据的任何小部分的有效检索和更新. 该方案还可以与任何数据库服务或现有的加密或冗余技术轻松组合, 减少了用户存储量和通信开销, 允许公开验证, 以及第三方审计可以是任何不受信任的实体.

综上, 这些方案的一个共同点是需要对要验证的数据进行分块, 然后将这些数据块作为 MHT 的叶子结点来构建 Hash 认证树, 不同方案中根据其所实现的功能不同, 叶子结点存储的附加信息不同. 在动态更新过程中, 通过变换 MHT 树型结构 (如, 合并某些结点、分裂某些结点) 来实现对数据的动态更新操作. 基于以上这些构建方法, 需要用户在构建 MHT 时, 提前确定所有叶子结点的信息, 才能生成 MHT, 但是当叶子节点信息无法提前确定时则无法进行验证.

3.2.5 方案对比分析

表 2 [3,5~15,17,18,35~37] 从关键技术、支持动态更新操作 (修改、插入、删除)、通信开销、计算开销、实现功能 (公开验证、隐私保护、访问控制) 几个方面对可更新数据验证方案进行比较, 其中, n 指的是数据块数, 每个数据块被分为 s 个部分, c 指的是完整性验证时被挑战的块数. 对于可更新数据验证方案, 通信开销是指随机挑战查询和验证证据的大小, 计算开销主要包括在用户或授权验证端验证证据的开销和在服务器端生成证据的成本. 在动态数据可验证技术的研究领域, 研究人员一直致力于提高效率和丰富功能, 他们追求的核心目标是高效地实现所有类型的动态更新操作. 研究发现, 几乎所有支持动态数据验证的 PDP 方案都是建立在 IHT, RASL, MHT 及其改进的基础上实现的, 在此基础上为了保证安全性和提高验证效率不断探索. 虽然 IHT 用于处理更新数据时成本最低, 但 MHT 更受学者欢迎, 因为它可以提供严格的验证, 其简单的结构也很容易适应各种场景. 此外, 还提出了平衡更新树等改进 MHT 的新动态结构, 用于大型外包数据的动态验证. 为了使方案更加完善, 许多研究方案还引入向量承诺实现公开验证, 基于属性签名实现访问控制等. 我们发现, 在满足公开验证时可能会舍弃一些其他性能, 一个好的解决方案是结合多种技术满足多功能需求. 由此可见, 在实现公开验证的同时高效地支持所有更新操作以及控制计算存储开销将仍然是研究人员广泛关注的问题.

表 2 中的计算开销主要考虑计算复杂度与数据块数量的关系. 为了进一步比较计算复杂度, 分析其中基于向量承诺的方案使用不同运算带来的开销, 我们对 Query 过程中服务器的计算开销、Verify 过程中验证者的计算开销以及 Update 过程中用户的计算开销, 且比较的每一类项目都是在同一安全级别下进行的, 如表 3 [10~14,35~37] 所示. 由于计算成本主要由双线性对、幂运算、群乘法和哈希运算

表 2 单副本数据完整性验证方案比较

Table 2 Comparison of single-copy data integrity verification schemes

Scheme	Deterministic/ probabilistic verification	Dynamic technique	Communication overhead	Computation costs		Dynamic operation			Function		
				User	Server	Modify	Insert	Delete	Public verification	Data privacy	Access control
Ateniese et al. [3]	Probabilistic	–	$O(1)$	$O(c)$	$O(c)$	–	–	–	✓	×	×
Ateniese et al. [6]	Probabilistic	Symmetric encryption	$O(c)$	$O(c)$	$O(c)$	✓	×	×	✓	×	×
Erway et al. [7]	Probabilistic	RASL	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	×	×	×
Zhu et al. [5]	Probabilistic	IHT	$O(c+s)$	$O(c+s)$	$O(c+s)$	✓	✓	✓	✓	✓	×
Sookhak et al. [8]	Probabilistic	D&CT	$O(c)$	$O(c)$	$O(c+s)$	✓	✓	✓	✓	✓	×
Shen et al. [9]	Probabilistic	IHT	$O(c+s)$	$O(c+s)$	$O(c+s)$	✓	✓	✓	✓	✓	×
Catalano and Fiore [35]	Deterministic	Vector commitment	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	×	×	✓	✓	×
Chen et al. [14]	Deterministic	Vector commitment	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	✓	✓	×
Wang et al. [15]	Deterministic	MHT	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	✓	✓	×
Liu et al. [17]	Deterministic	MHT	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	✓	×	×
Chen et al. [18]	Deterministic	Balanced update tree	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	✓	✓	×
Chen et al. [10]	Deterministic	Binding commitment	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	×	×	✓	✓	×
Zhang et al. [11]	Deterministic	VCT	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	×	×	✓	✓	✓
Miao et al. [12]	Deterministic	Hierarchical commitment	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	✓	✓	×
Chen et al. [13]	Deterministic	CIBF	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	✓	✓	×
Miao et al. [36]	Deterministic	MSHT	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	×	✓	×
Wang et al. [37]	Deterministic	MSHT	$cO(\log n)$	$cO(\log n)$	$cO(\log n)$	✓	✓	✓	✓	✓	×

表 3 计算复杂度

Table 3 Computational complexity

Scheme	Query (server)	Verify (verifier)	Update (user)
Catalano and Fiore [35]	$(q-1)(M+E)$	$1M+1E+1I+2P$	$1M+1E$
Chen et al. [10]	$(q-1)(M+E)$	$2M+1E+1I+4P$	$2M+2E+1I$
Zhang et al. [11]	$(q(Q-1)/(q-1)-n)(M+E)$	$n(M+E+I+2P)$	$(q(Q-1)/(q-1)+n)(M+E)+n(I+2P)$
Miao et al. [12]	$(q-2)M+(q-1)E$	$2M+1E+1I+4P$	$2M+(L+1)E+1I$
Chen et al. [13]	$(2q-5)M+(2q-3)E$	$3M1+3E+2I+6P$	$2M+3E$
Chen et al. [14]	–	$1M+1E+4P+1H+1H'$	$1E+1D+1H'$
Miao et al. [36]	$(q-1)M+2P$	$4M+3E+2F+3P+\log_q \cdot H$	$2M+4E+2F+1P$
Wang et al. [37]	$q \cdot M+1P$	$(\ell+1)H+1E+1M+1P$	$1E+q \cdot M$

决定, 其中双线性对和幂运算占主要开销, 因此我们通过计算此类运算的数量来评估其计算开销, 其中, P 表示双线性对的时间开销; E 表示幂运算的时间开销; M 表示 G_1 或 G_2 上乘法的时间开销; H 表示 Z_p 或 G_1 的哈希运算; I 表示 G_1 中取倒数运算; F 表示伪随机函数的运算; ℓ 表示为随机函数中参数的大小; Q 表示外包数据库的大小; q 表示每个划分的数据库记录单元的大小; n 等于 $\lceil \log_q Q \rceil$; L 表示当前数据库中的总级数.

3.3 面向多副本数据的完整性验证技术

存储多个副本是外包服务 (如: 云存储) 中提高可靠性和可用性的常用策略. 对于重要数据文件, 用户有时希望在服务器上存储多个副本, 来确保如果某些副本被损坏仍然可以从剩余的副本中恢复. 多副本外包数据验证的关键主要包含 4 点: 第一, 要确保服务器确实存储了多个副本; 第二, 当数据动态变化时, 每个副本也要实时变化, 逐个验证每个副本显然不符合实际, 为了保证数据的时效性和一致性, 要能够高效地验证多个副本数据的完整性; 第三, 当验证失败时, 能够准确定位损坏的副本; 第四, 能够对某些特定块的任意副本进行验证. 针对以上问题, 许多研究方案引入了 PDP 模型并基于同态签名、MHT、映射表等技术解决多副本外包数据的动态更新验证等问题. 此外, 为了进一步提高效率、降低证书管理的负担, 研究者还通过引入身份签名技术, 提出了基于身份的多副本数据完整性验证方案. 下面针对这些典型方案进行分类论述.

3.3.1 多副本可证明数据持有方案

为了验证多个副本外包数据的完整性, 许多研究方案引入了 PDP 模型. 在 PDP 模型中, 服务器只需访问少量不变的数据和随机采样的数据块集合生成概率性证明, 用户和服务器之间只需传输简短的数据完整性证明, 外包数据的用户可实现在不检索服务器上存储数据的情况下检查并验证服务器是否持有原始完整数据. 而 SP 为了获取更多的利益, 可能会多次出售相同的存储空间, 实际上只存储一个副本但却声称在多个服务器上已存储多个副本. 为了解决以上问题, Curtmola 等 [39] 基于 PDP 模型提出了首个实现多副本可证明数据持有方案 (multiple-replica provable data possession, MR-PDP). MR-PDP 除了继承 PDP 的优点, 还创新实现了: 用高效伪随机函数 (amortized closed-form efficient PRF) 按需为每个不同的服务器生成唯一加密副本, 以抵抗不同服务器间联合欺骗用户; 用户在预处理时根据原始数据块生成 HVT 并存储在服务器中; 将预处理计算成本分摊到多个副本上来降低总预处理成本. 但该方案只适用于静态场景而且需要大量计算开销逐个检查副本. 此外, MR-PDP 的验证只能由 DO 自己执行, 如果恶意攻击者获取该私有随机数可以很容易恢复原始数据. 为了能避免 DO 和 SP 之间发生纠纷, Hao 等 [40] 提出了一种公开可验证性的多副本远程数据持有检查协议. 该方案基于 BLS 签名的 HVT 实现了支持隐私保护的多副本完整性验证, 用户在用 PRF 生成的随机参数加密原始数据后将密钥分享给了第三方审计 (third party auditor, TPA), TPA 用该密钥还原随机参数验证服务器返回的证明实现了第三方公开验证. Barsoum 等 [41] 也提出了一种基于双线性对的多副本 PDP 方案 BP-PMDP, 实现了公开可验证性且生成数据副本成本更低, 还能通过二分法递归搜索识别损坏的副本. 该方案也基于 BLS-HLAs 为每个副本中相同索引的块生成一个聚合标签, 缩减了需要在 SP 中存储的标签数量, 并将每个副本的唯一指纹嵌入到块标签中, 防止 SP 使用来自不同块的标签进行欺骗.

然而, 基于 HLAs 为每个副本的相同索引块生成聚合标签的 MR-PDP 方案无法通过验证单个副本来识别副本中哪些数据块已损坏. 于是, Yi 等 [42] 将带有随机噪声的 FHE 与双线性对验证相结合, 实现了一个高效地支持块级操作的多副本完整性公开验证方案. 该方案能够从损坏的副本中识别出损坏的数据块, DO 只需重新生成损坏的数据块, 而不是整个副本, 同时可以抵抗伪造攻击、替换攻击和重放攻击. 此外, 不管是基于 HLAs 还是 FHE 的验证方法在生成标签时都需要进行复杂的群幂运算, 大大增加了系统的计算开销. 为此, 还有文献 [43] 采用向量点积代替昂贵的群幂运算来计算副本标签实现了多副本数据的完整性验证.

3.3.2 基于 MHT 的动态多副本 PDP 方案

在实际应用中数据是动态变化的, 逐个更新和验证每个副本将在通信和计算方面付出非常昂贵的代价. 为保证数据的时效性, 许多支持数据实时动态操作的多副本验证方案被推出. 在 2011 年, Barsoum 和 Hasan [44] 首次提出了基于 MHT 的动态多副本 PDP 方案 TB-DMCPDP. 该方案首先为每个副本构造一个 MHT, 然后用每个副本树的根构建目录 MHT, 分层验证所有副本的完整性. 此外, 为了支持全动态数据外包, 在插入新块之后无需重新计算移位的块, 数据块标签中不包含块索引, 但这样无法抵抗不诚实服务器的攻击. 于是, Liu 等 [45] 基于 MHT 设计了用于多副本验证的数据结构 MR-MHT, 并提出了多副本动态公开验证方案 MuR-DPA. MR-MHT 的节点中新增了 rank (节点包含的最大叶子节点数) 和 level (节点层级) 参数, 且 level 值是按照自顶向下的顺序分配的, 通过独立验证每个副本中的相同索引的数据块, 支持数据的全动态更新和块索引验证. 但是, MuR-DPA 需要逐个验证每个副本, 带来很大的通信和计算开销. 文献 [44, 45] 中的方案虽然支持所有更新操作, 但不支持可变大小的数据块, 而且验证计算开销太大. 为此, Zhang 等 [46] 提出了一个支持可变数据块大小的多副本公开可验证 PDP 协议, 为了简化计算不再使用双线性对验证, 仅进行简单的哈希, 提高聚合验证多个副本更新的效率.

此外, 文献 [45] 中的 MR-MHT 不是一个平衡的认证数据结构, 当在同一索引位置上进行大量连续插入或删除时, 会重复计算传输一些节点的哈希值, 这将消耗额外的计算资源. 为了解决上述问题, Guo 等 [47] 设计了一种隐式索引平衡 Merkle 树 (implicitly-indexed balanced Merkle tree, IBMT) 来实现动态多副本数据验证. IBMT 的每个内部节点可以有两个或三个子节点, 使叶子节点均在同一层, 能在同一个索引处进行大量连续插入或删除时始终是一棵平衡树, 还能批量验证, 无需重复传输和计算索引值, 节省了通信和计算开销. 然而, 该方案需要引入 TPA 来完成数据完整性验证. Yu 等 [48] 在 MHT 的基础上添加数据块索引设计了一种新的认证数据结构 IMHT 用于多副本外包数据完整性验证, 其叶子结点中不再存储数据块的哈希值而是存储数据块的逻辑位置和版本号的哈希元组, 其构建成本与副本的数量无关, 降低了用来维护 MHT 的存储、通信和计算成本.

3.3.3 基于映射表的动态多副本 PDP 方案

为了支持数据动态更新, 基于版本映射表 (map-version table, MVT) 多副本方案也得到了广泛研究. Barsoum 和 Hasan 在文献 [44] 中还提出了基于 MVT 的动态多副本 PDP 方案 MB-DMCPDP, 不仅支持数据动态变化还实现了可公开验证. MVT 存储在验证者端, 用于验证所有副本的完整性, 由序列号 (SN)、块号 (BN) 和版本号 (VN) 组成. SN 表示块在副本中的物理位置, BN 表示块的逻辑索引, VN 表示块的当前版本, 最初创建时所有块的 VN 为 1, 一个特定的块被更新后它的 VN 将增加 1. 为了适用于多服务器分布式协同存储系统, Zhu 等 [49] 基于交互式证明构建一个具有隐私保护的协作 PDP 方案用来验证多副本外包数据的完整性. 他们设计了一种类似于文件存储的分层结构来表示存储资源的所有块之间的关系, 并为之构造一个分层索引哈希 (Hash index hierarchy, HIH) 替代普通哈希函数. 在该方案中新增了一个聚合者用来聚合来自不同 SP 的多个响应, 为验证者提供透明的验证, 比非合作方案具有更低的通信和计算开销. 为了进一步完善副本验证方案, Barsoum 等 [50] 在 2015 年又提出了一种基于 MVT 的多副本动态 PDP 方案 MB-PMDDP, 以增加一些额外的通信、计算开销为代价, 利用二分法递归验证聚合标签来识别损坏的副本.

3.3.4 基于身份的多副本 PDP 方案

在大部分多副本完整性验证方案中, 能力有限的 DO 将验证任务委托给第三方验证者来实现公开

可验证. 然而, 这些方案大多依赖于复杂且消耗资源的公钥基础设施 (public key infrastructure, PKI), 面临着繁重的证书验证和管理. 且大多数验证者的计算能力不强, 为了提高效率许多基于身份的多副本完整性验证方案被提出, 无需复杂的证书管理. Wang [51] 设计了一个具体的多云存储场景下基于身份的分布式 PDP 方案 ID-DPDP, 实现了可公开的委托验证. 该方案中引入了根据身份生成私钥的生成器 (private key generator, PKG), 并采用分布式计算结合多个服务器的响应来应对验证者的挑战, 但该方案没有考虑数据动态更新的情况. 为了实现动态更新以及进一步降低计算和通信成本, Peng 等 [52] 提出了一种高效、动态的基于身份的多副本外包数据完整性验证方案 EDID-MRPDP. 他们设计了一种新的数据结构压缩认证阵列 (compressed authentication array, CAA) 来降低通信成本, 并基于 HVT 进行批量验证, 使得方案适用于全动态更新数据的大规模云应用. Li 等 [53] 也提出了基于身份的多云 PDP 协议, 不依赖于 PKI 管理用户的公钥证书, 还利用 HVT 聚合验证来提高效率. 但是, 该协议还是使用了传统的 BLS 签名, 验证者必须检查签名者的公钥与其身份之间的关系, 导致整个协议不是完全基于身份的, 仍会承担管理证书的负担. 因此, Chang 等 [54] 用基于身份签名替换文献 [53] 中的传统签名, 修改后的协议完全消除了对 PKI 的依赖和证书的管理, 是真正基于身份的跨云多副本完整性验证方案.

3.3.5 方案对比分析

如表 4 [39~41, 44~54] 所示, 根据方案实现的关键技术以及是否实现动态数据、聚合验证、定位损坏副本和公开验证等功能, 对多副本外包数据完整性验证典型方案进行对比分析. 我们发现, 为了支持动态数据验证, 同样也引入 MHT、映射表等数据结构, 同时还添加了块索引验证来提高动态数据验证效率. 针对多副本数据完整性验证的特点, 许多方案还基于聚合验证的思想来批量验证副本来提高多副本验证的验证效率, 实现了验证 t 个副本的成本远远低于运行 t 次的单个拷贝方案的成本, 同时验证开销与副本的数量无关. 此外, 在多副本外包数据完整性验证方案中还应具备一旦验证失败定位损坏的副本位置以及恢复备份等功能. 在未来的工作中, 找到一种更好的数据结构来实现全动态操作, 以及如何生成标签进行聚合验证来节省计算时间和存储空间仍然是一个值得研究的问题.

3.4 面向流数据的实时验证技术

流数据自身具有一定的独特性. 流数据实时验证基本原理如图 5 所示, 由于 DO 只能看到当前数据, 不可预知未来的数据, 因此 DO 仅能拥有当前缓冲区的部分数据, 为当前缓冲区部分数据生成本地认证信息, 然后以数据流方式将数据及认证信息外包给第三方服务器 SP, SP 根据到达的流数据及认证信息更新查询结果及验证证据信息. 基于外包流数据可验证基本原理得出, 传统的可认证数据结构不能直接应用于流数据, 需要针对流数据设计专用的流数据可认证数据结构, 不仅支持对流数据的实时验证, 还要确保流序列的可验证性. 此外, 流数据的查询验证往往具有连续性和不可预测性. 这些因素使外包流数据完整性验证问题变得更具挑战性. 目前构造可验证数据流的方法主要有 MHT、CAT 和密码累加器.

3.4.1 基于 MHT 的流数据验证方案

为了使传统 MHT 数据结构能适用于流数据, 必须要适应验证者只持有少量局部状态且不能与证明者交互的特点. 因此, 结合流数据的特点, Papamanthou 等 [55] 基于累加器与 MHT 的思想提出了一种无交互流数据验证结构, 称为广义哈希树 (generalized Hash tree, GHT). GHT 兼具 MHT 支持各种动态操作对数级别的复杂度和累加器无需交互即可完成更新的优点, 来确保数据值和范围查询结

表 4 多副本外包数据完整性验证方案比较

Table 4 Comparison of multi-copy data integrity verification schemes

Scheme	Key technology	Dynamic data	Aggregate verification	Locate damage copy position	Public verification
Curtmola et al. [39]	RSA sign	×	✓	×	×
Hao et al. [40]	BLS-HLAs	×	✓	×	✓
Barsoum et al. [41]	BLS-HLAs	×	✓	✓	✓
TB-DMCPDP [44]	MHT; BLS-HLAs	✓	✓	×	✓
Liu et al. [45]	MR-MHT	✓	×	×	✓
Zhang et al. [46]	MHT; HLAs	✓	✓	×	✓
Guo et al. [47]	IBMT	✓	✓	×	✓
MB-DMCPDP [44]	MVT; BLS-HLAs	✓	✓	×	✓
Zhu et al. [49]	HIH; HVR	✓	✓	×	✓
Barsoum et al. [50]	MVT; BLS-HLAs	✓	✓	✓	
Wang [51]	IBS; Distributed computing	×	✓	×	×
Peng et al. [52]	IBS; MR-PMT; CAA; HVT	✓	✓	×	✓
Li et al. [53]	IBS; BLS; HVT	×	✓	×	✓
Chang et al. [54]	IBS	×	✓	×	✓
Yi et al. [42]	FHE	✓	✓	✓	✓
Yu et al. [48]	IMHT	✓	✓	×	✓

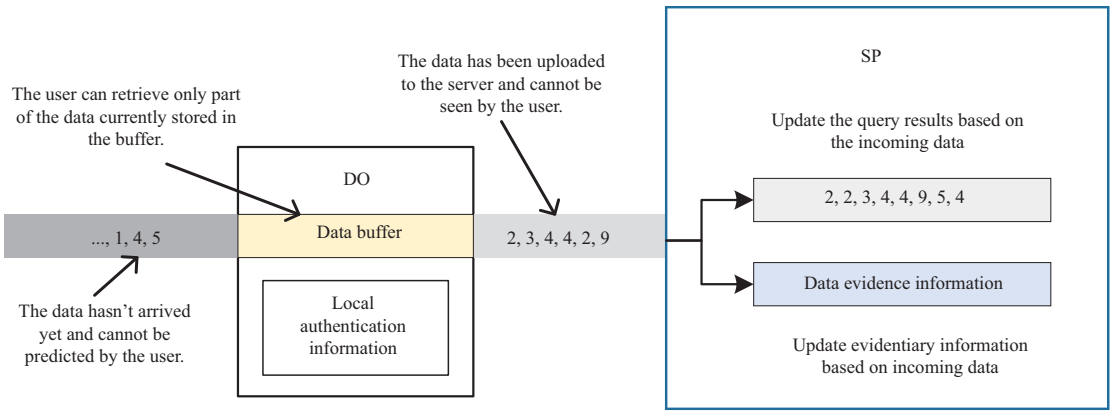

图 5 (网络版彩图) 外包流数据实时验证基本原理

Figure 5 (Color online) Fundamental of outsourcing streaming data real-time verification

果的正确性, 但是, 该方案过于复杂难以在实际应用中设计和实现. Yu [56] 结合 FHE 和 MHT 提出了 FHMT 来实现流数据的完整性验证. FHMT 利用 FHE 的性质几乎将所有的计算任务转移到服务器上, 使得用户的开销很小, 更适用于具有资源受限设备的系统, 如传感器、物联网设备和手机等. 此外, Xu 等 [57] 还提出了一种基于 FHMT 的轻量级适用于资源受限设备的隐私保护数据完整性验证模型. 但是, GHT 和 FHMT 都是静态的数据结构, 需要预先确定结构的规模, 不能在流数据场景中使用. 于是, 文献 [58] 通过改进 FHMT 提出了一种动态的流数据完整性验证结构 DFHMT, 而且更适合计算能力弱的用户, 同时还支持范围查询验证. 他们基于多目标优化算法来改进 FHMT, 使其在初始化时不需要确定 DFHMT 的规模, 并且在树中添加新数据时可以自适应扩展. 但由于 FHE 的效率较低且

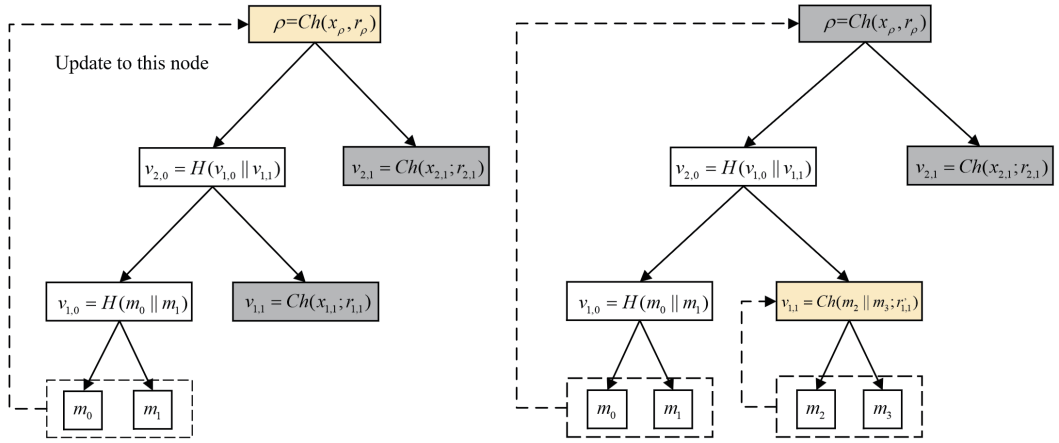

图 6 (网络版彩图) CAT 结构

Figure 6 (Color online) CAT structure

不支持范围查询, 这些基于 FHMT 的方案在目前的实际应用中是不可行的.

但是, 基于 MHT 及其改进的方法都需要提前确定所有叶子结点的信息才能生成数据认证结构. 而且, 在处理流数据时, 还需将流数据划分为很多个片段, 对每个流片段来构建 MHT, 频繁地建树以及生成的根节点签名产生较大的计算开销和存储开销, 同时由于一直缓存数据, 还会造成数据发送的延迟. 此外, 无论是更新旧的数据项, 还是增加新的数据项, 都需要立即更新验证密钥, 这给验证密钥的管理带来了负担, 特别是对于规模大、传输速率高的流数据.

3.4.2 基于 CAT 的流数据验证方案

为找到更适用于流数据完整性验证的数据结构, Schroeder 等 [59] 于 2012 年的 CCS 会议上引入了可验证数据流 (verifiable data streaming, VDS) 协议的概念. 他们构造了一种新的数据结构变色龙认证树 (chameleon authentication tree, CAT), CAT 本质上是一种广义的 MHT, 在 MHT 的构建过程中引入变色龙哈希函数 (陷门哈希). 陷门哈希具有抗碰撞性以及一个陷门, 在构建认证树时无需获取所有的数据, 在拥有陷门的情况下可以构造出碰撞值使得两组值拥有相同的陷门哈希值, 更新相关陷门节点的随机数值, 如图 6 所示, 虚线指示第 1 次插入与第 2 次插入时向上更新的情况. 与 MHT 方案相比, 基于 CAT 构造的完整性验证方案无需一次性接收全部数据来构造整棵认证树进行验证, 新增叶子节点时, 也无需重新计算先前所有的节点信息, 支持数据实时更新和验证, 其根节点是利用陷门哈希函数生成的, 只有持有陷门的用户才可添加叶子节点, 防止恶意用户新增或修改叶子节点. 此外, Backes 等 [60] 还将 CAT 应用到了实际网页访问的安全中, 将每个网页的内容分块后构造 CAT, 实现了实时网络中来自不可信服务器流形式的网页内容完整性验证.

CAT 结构几乎满足了 VDS 的所有要求, 但文献 [59] 中验证的数据量是有限的, 如果 CAT 结构层次预先设定过大, 会延长验证路径, 增加验证开销, 如果层次设定过小, 可处理的数据量就大大受限. 为了优化上述 VDS 协议, Schroder 和 Simkin 等 [61] 在 2015 年的 PKC (Public Key Cryptography) 会议上提出了第 1 个消除上界限制的全动态 f-CAT 方案, 并构造了第 1 个实用的可验证数据流框架 VeriStream. 在初始化时不需要指定认证树的上限, 也就是说可认证树的深度是不确定的, 在数据插入的过程中能够动态地扩展可认证树规模, 但其方案的安全性是基于随机预言 (random oracle) 模型, 相比于基于标准模型的 CAT 降低了安全级别. 因此, 又提出了第 2 种动态的 δ 界动态变色龙认证

树 (δ-CAT), 在初始化时便限制了 δ-CAT 的扩展规模上限为 δ, 将所有的后续动态生成根节点的信息提前生成存储于公钥中, δ-CAT 在深度 $d<\delta$ 时与 f-CAT 过程类似, 可以动态扩展认证树的规模, $d=\delta$ 时可认证树支持的数据量到达上限, 无法继续往后扩展. 虽然 δ-CAT 还是只支持有限的动态扩展, 但其安全性是基于标准模型的可证明安全. 为了同时支持无上限的数据流更新和在标准模型下的安全强度, Krupp 等 [62] 结合向量承诺 [35] 和 CAT 提出了一种新的 VDS 可认证数据结构, 称为变色龙向量承诺 (chameleon vector commitments, CVC). 在该方案中对每个节点使用变色龙向量承诺, 可以在不改变根节点的情况下向可认证树中增加新的层次, 处理的数据量与 f-CAT 类似没有上界限制, 而且是基于标准模型的可证明安全. 此外, Wu 等 [63] 也基于 CVC 实现了可公开验证的安全流数据外包方案 DSO-PVI.

后来, 研究者又根据不同的应用需求, 提出了许多基于 CAT 的改进方案. 为支持隐私保护和范围查询验证, 文献 [64] 提出了一种每个节点都使用 Bloom 过滤器计算的完整二叉树 – 隐私 Bloom 过滤器树 (privacy Bloom-filter tree, PB-tree), 具有索引不可区分性, 且搜索速度比普通二叉树快. 为了进一步提高验证效率, Tsai 等 [65] 在 PB-tree 的基础上进行改进, 增加了哈希函数和权值, 实现了高效的范围查询和验证, 但其数据上限仍然需要提前确定. 为了适应基于共享平台的流交换实时验证以及多用户验证权限控制问题, Sun 等 [66] 提出了基于属性加密的双陷门哈希认证树 AC-MTAT. 双陷门哈希函数解决了更新时可能出现的密钥泄露问题, 到数据存储上限时可以无交互地从第 1 个节点重新开始更新, 通过基于属性加密机制控制只有授权用户才能验证接收流数据, 防止流交换数据非授权接收与验证. 为了解决大数据流实时验证的同时防止审计用户带来的隐私泄露问题, 文献 [67] 在 f-CAT 的基础上进行改进, 提出一种具有隐私保护的自适应可认证数据结构, 该认证结构能够根据数据流规模变化而自适应拓展可认证树, 实现了对大数据流的边读边写边验证, 克服了以往可认证树的单点故障问题, 增强了可认证树完整性验证的健壮性和稳固性, 并基于双线性映射构造同态验证方案解决了第三方审计带来的隐私泄露问题. 2021 年, 文献 [68] 又结合 PB-tree 和 CAT 提出了前缀变色龙认证树 PCAT, 该方案不仅是轻量级的, 而且还支持数据流中的动态扩展和可验证的范围查询, 使其更适合资源受限的设备. 2022 年, Miao 等 [69] 也基于双陷门变色龙哈希树提出了一个 VDS 协议, 该方案中每个叶子的值都是用双陷门变色龙哈希计算, 并用计数器记录当前版本嵌入到根目录中, 来抵抗恶意服务器重放攻击. 此外, 每次更新数据时使用一次性陷门, 使得恶意服务器既不能恢复长期使用的陷门, 也不能计算出新的碰撞.

尽管已经提出了许多 VDS 协议来增强原始 VDS 协议的功能性和效率, 但它们都忽略了保存那些更新的数据项的问题. 事实上, 在 VDS 的各种应用场景中, 保存和存储先前更新的数据项实际上是必要的. 例如, 在 DNA 测序的设置中, 由于基因突变, 可能在同一位置存在多个版本的 DNA 片段. 显然, 为了更精确地利用这些 DNA 信息进行治疗, 需要保存 DNA 片段所有的历史版本. 为此, Miao 等 [70] 提出了一种 VDS 协议, 该协议的特点是能够查询每个数据项的更新历史. 具体来说, 首先提出了一种新的具有更新历史的 CAT, 它由两个 CAT (基本树和更新历史树) 组成. 更详细地, 基本树用于存储第 1 次附加到对应位置的数据项, 并且更新历史树用于保存对应数据项的每个更新版本. 此外, 他们还基于具有更新历史的 CAT 提出了一种支持更新历史查询的 VDS 协议, 该协议允许数据用户检索数据项的任何版本.

3.4.3 基于密码累加器的流数据验证方案

密码累加器也是常用的一类成员证明结构. 在文献 [62] 中, Krupp 等在数字签名的基础上引入双线性累加器, 构造了一个支持数据流动态可验证方案 ACC. 该方案用常规签名方案对数据集的每

个元素进行签名, 用加密累加器来撤销旧签名, 来支持流数据的更新, 用户只需存储签名的密钥对, 接收方只需存储 DO 的公钥. 但是该方案需要对每一个数据流片段进行签名, 与认证树相比计算开销更大. Zhang 等 [71] 通过对向量承诺进行拓展提出了一种维度增长的向量承诺 (dimension increasing vector commitment, DIVC) 用于解决数据流的可验证问题. 在 DIVC 中, 每个新到达的流元素都与之前的流元素在其单元内按时间顺序缓存在本地缓存区中, 直到其元素数量达到单个向量承诺元支持的数据上限, 再依据向量承诺的方法计算承诺值和验证, 随后将数据流和对应向量承诺元发送给服务器, 除了在向量承诺元的末端追加承诺值, 还附加了对承诺值的签名, 并在 DIVC 的基础上结合布隆过滤器 (Bloom filter) 和动态累加器构造了 P-CVDS 和 D-CVDS 两种方案用于流数据的验证. Wei 等 [72] 也基于累加器和 BLS 聚合签名构造了一个具有恒定成本的可验证数据流验证协议. 该方案是基于随机预言机模型可证明安全, 采用 BLS 签名作为底层的数字签名方案, 保证检索到的数据的完整性, 累加器用于在每次更新操作后使旧签名失效, 允许 DO 或其他第三方验证以前外包的数据项的完整性, 还要求 SP 返回那些受质疑的数据项的非成员身份证明以及这些数据项的标签集来抵抗重放攻击.

3.4.4 方案对比分析

通过以上对流数据实时验证方案的研究与分析, 可以看出, 为了适应流数据不能存储整个数据流, 也不能预测未来数据的特点, 该类技术研究重点在于构造一个连续性查询验证且具有一定自适应性的 ADS. 如表 5 [55,56,58,59,61~63,65~72] 所示, 从关键技术、是否有上界限制、安全性、支持查询类型、是否支持公开验证和隐私保护 6 个方面对流数据实时验证的典型方案进行比较分析, 可以看出基于 MHT 和 PB-tree 的方案能支持范围查询, 基于 CAT 的方案实现了流数据真正无上界的限制, 此外 FHMT 和 PB-tree 还能实现隐私保护.

如表 6 [58,59,61~63,66,71,72] 所示, 我们又从计算开销和存储开销两方面比较了几个重点方案的性能, 其中计算开销只考虑在 Append, Query, Verify, Update 4 个过程中主要操作 (如双线性对、指数运算) 的计算开销, 通信开销借助为了验证数据完整性而传递的验证证据的大小来体现, 且比较的每一类项目都是在同一安全级别下进行的. 其中, M 是允许存储到数据库中的最大数据条目数; N 是当前存储在数据库中的数据条目数; U 表示更新操作数; s 表示增维向量承诺的单元大小. 虽然已经做了很多有价值的研究工作, 但目前流数据完整性验证的结构还不是最优的, 还不能实现恒定的计算和通信成本. 此外, 以前的流数据完整性验证协议的复杂性是在单次查询下评估的. 也就是说, 对于一次检索多个数据项的并发查询, 这些协议的成本与并发查询的大小成线性关系. 针对这种情况, 如何设计最优的流数据实时验证协议仍是一个值得探索的问题.

4 外包计算可验证技术研究进展

外包数据验证领域的研究工作重点关注外包数据的持有性或者以挑战 – 应答方式检索验证某个具体数据的完整性, 没有考虑对外包数据进行任意或算术计算 (例如函数计算、统计分析、关键字查询、机器学习) 结果的完整性验证. 而提供外包计算也是外包服务非常普遍和重要的功能, 这样计算能力有限的用户可以把复杂的计算任务委托给外包服务商, 然后借助其强大的计算能力完成某些计算任务, 这种将复杂的计算任务委托给具有强大计算能力一方的行为被称作外包计算 (outsourcing computation). 为了实现对计算结果的正确性判定, Gennaro 等 [73] 提出了可验证外包计算 (verifiable computation, VC) 的概念, 其主要目的是验证计算结果的正确性和完整性.

由于篇幅限制, 我们这里主要针对目前典型的、或热门的可验证外包计算技术进行分类讨论, 首

表 5 流数据实时验证方案比较

Table 5 Comparison of streaming data real-time verification schemes

Scheme	Key technology	Data upper bound	Security model/assumption	Supported query types	Public verification	Data privacy
Papamanthou et al. [55]	GHT	Preset	Lattice assumption	Data value query; range queries	✓	×
Yu [56]	FHMT	Preset	FHE security	Data value query; range queries	×	✓
Xu et al. [58]	DFHMT	Adaptive extension	FHE security	Data value query; range queries	×	✓
Schroder et al. [59]	CAT	Preset	Standard	Data value query	✓	×
Schroder et al. [61]	f-CAT	Adaptive extension	ROM	Data value query	✓	×
Schroder et al. [61]	δ-CAT	Dynamic δ bound	Standard	Data value query	✓	×
Krupp et al. [62]	CVC	Adaptive extension	Standard	Data value query	✓	×
DSO-PVI [63]	CVC	Adaptive extension	Standard	Data value query	✓	×
VENUS [65]	HPB-tree	Adaptive extension	ROM	Data value query; range queries	×	✓
Sun et al. [66]	AC-MTAT	Preset	ROM	Data value query	✓	✓
Sun et al. [67]	p-ATHAT	Adaptive extension	Standard	Data value query	✓	✓
Xu et al. [68]	PCAT	Adaptive extension	CAT Security	Data value query; range queries	✓	✓
Miao et al. [69]	DCHT	Preset	Standard	Data value query	✓	×
Miao et al. [70]	UCAT	Preset	Standard	Update history query	✓	×
Krupp et al. [62]	Bilinear accumulator	Adaptive extension	Standard	Data value query	✓	×
Zhang et al. [71]	DIVC	Preset	q-SDH	Data value query	✓	×
Wei et al. [72]	RSA accumulator BLS	Adaptive extension	ROM	Data value query	✓	×

表 6 性能比较

Table 6 Performance comparison

Scheme	Communication overhead	Computation costs			
		Append	Query	Verify	Update
DFHMT [58]	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(U)$
CAT [59]	$O(\log M)$	$O(\log M)$	$O(\log M)$	$O(\log M)$	$O(\log M)$
f-CAT [61]	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$
δ-CAT [61]	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$
CVC [62]	$O(\log N)$	O(1)	$O(\log N)$	$O(\log N)$	$O(\log N)$
ACC [62]	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(U)$
DSO-PVI [63]	$O(1)$	$O(1)$	$O(\log N)$	$O(\log N)$	$O(1)$
Sun et al. [66]	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$
Zhang et al. [71]	$O(1)$	$O(\log N/s)$	$O(1)$	$O(1)$	$O(1)$
Wei et al. [72]	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$

先在 4.1 小节给出外包计算的一个基本的形式化定义, 接着在 4.2 小节针对工程和密码学等领域中大量的、计算成本较高的运算外包可验证技术进行分类讨论. 然后, 由于 DaaS 新型计算范例 [1] 的广泛

应用, 关注外包数据库完整性验证的方案层出不穷. 其中典型的 VDB 方案主要研究如何保证数据库的完整性, 特别是动态数据库的完整性, 即数据库在进行修改、删除等动态操作时如何进行完整性验证的问题. 因为如果数据库被篡改且未被发现则对数据库进行任何操作或对操作的验证都是毫无意义的. 除此之外, 在大多数外包数据库的现实场景中, 还需要对数据库进行各种类型的查询运算操作, 如, 范围查询、连接查询、关键字查询等. 而 VDB 方案只支持对某一索引的查询, 也就是说, 给定一个查询索引 x, 服务器返回相应的数据值 V_x 以及确保 V_x 有效性的证明. 因此, 还需要研究能够支持各种查询类型的可验证技术, 目前常用的查询方式主要有: 针对关系数据库的 SQL 查询、针对空间数据的空间查询、针对加密数据关键字搜索的可搜索加密查询等, 下面我们针对这些常用查询验证方式分别在 4.3∼4.5 小节论述其典型的可验证方案. 最后, 在 4.6 小节从实现方式的不同讨论基于可信硬件的外包计算验证技术的优势与不足.

4.1 外包计算可验证的形式化定义

一个基本的外包计算可验证方案是一个在用户和服务提供者之间执行的两方协议. 用户选择一个函数和一个输入给服务提供者. 后者希望在此输入上评估函数的执行结果, 因此服务提供者在响应输出的同时提供一个能够证明该输出是正确的证据. 然后用户根据这个证据验证服务提供者提供的输出结果, 证明该结果确实是函数在所提供输入上计算出的正确结果. 在外包计算中, 有两点最基本的要求: 一是安全性, 保证服务提供者返回给用户的计算结果是正确的; 二是有效性, 在执行计算外包时用户的本地计算开销要小于其独立执行该计算的计算开销. 下面结合文献 [73] 的定义, 给出可验证授权计算方案的形式化描述.

定义3 (外包计算可验证方案) 一个外包计算可验证方案 $\mathrm{VC} = (\mathrm{KeyGen}, \mathrm{ProbGen}, \mathrm{Compute}, \mathrm{Verify})$ 由以下 4 个算法所组成.

• $\mathrm{KeyGen}(f, 1^{\lambda}) \to (\mathrm{pk}, \mathrm{sk})$: 以外包计算函数 f 和安全参数 λ 为输入, 生成一组关于函数 f 的公私钥对, 公钥 pk 提供给服务提供者 SP, 私钥 sk 由用户 U 保留.

• $\mathrm{ProbGen}(\mathrm{sk}, x) \to (\sigma_x, \tau_x)$: 用户 U 利用私钥 sk 将输入 x 编码为公开值 σ_x 和私有值 τ_x, 并将公开值 σ_x 发送给服务提供者 SP, 私有值 τ_x 由用户 U 持有.

• $\mathrm{Compute}(\mathrm{pk}, \sigma_x) \to \sigma_y$: 服务提供者 SP 运行, 输入公钥 pk 和 τ_x 计算函数 $y = f(x)$ 的一个编码版本 σ_y.

• $\mathrm{Verify}(\mathrm{sk}, \sigma_y, \tau_x) \to (y, \perp)$: 使用私钥 sk、私有值 τ_x 和 σ_y, 输出 y 或者 $\perp$. 如果输出为 $y = f(x)$, 表示 σ_y 是 f 在 x 的有效输出, 如果输出 $\perp$, 表示 σ_y 不是 f 在 x 的有效输出.

定义4 (外包计算可验证方案安全性) 对于一个外包计算可验证方案, 我们定义攻击者在以上实验中优势为 $\mathrm{Adv}_A^{\mathrm{Verif}}(\mathrm{VC}, f, n) = \mathrm{Prob}[\mathrm{Exp}_A^{\mathrm{Verif}}[\mathrm{VC}, f, n] = 1]$.

一个外包计算可验证方案 VC 对于任意函数 $f \in F$ 是安全的, 并且对于任意攻击者 A 运行在概率多项式时间.

$\mathrm{Adv}_A^{\mathrm{Verif}}(\mathrm{VC}, f, \lambda) \leqslant \mathrm{negli}(n)$, 这里 negli() 表示具有输入参数的可忽略函数.

安全性即 SP 能够向用户证明关于函数 f 的计算结果是正确的. 此外, 需要说明的是这里定义的安全性是指满足可验证外包计算的基本安全要求, 根据不同的安全要求 (如: 隐私保护), 可以对安全性定义进行扩展.

定义5 (外包计算可验证方案有效性) 一个外包计算可验证方案 VC 是有效的, 如果对于任意的 x 和 σ_y, 计算 $(\sigma_x, \tau_x) \leftarrow \mathrm{ProbGen}_{\mathrm{SK}}(x)$ 所需时间加上验证 $\mathrm{Verify}_{\mathrm{SK}}(\tau_x, \sigma_y)$ 所需时间小于 $o(T)$, 这里 T

是计算 $f(x)$ 所需时间.

有效性即用户验证证据所需的时间小于计算函数 f 本身所需时间 (包括计算时间和通信时间). 也就是说在外包计算可验证中, 验证计算的代价要小于只是简单的让客户独自在本地执行计算的代价, 可验证授权计算才有意义.

4.2 面向函数计算的可验证技术

在工程和密码学等领域中大量存在一些计算成本较高的运算, 外包计算是解决此类问题的一种有效方法 [74], 如, 云计算可以凭借其强大的计算能力帮助人们处理繁重的计算任务. 在 Gennaro 等 [73] 提出可验证外包计算模型之后, 后续有许多研究者基于此模型针对不同计算类型提出了一系列可验证方案, 主要分为面向一般函数的可验证方案和面向具体函数的可验证方案.

4.2.1 一般函数的可验证计算方案

针对一般函数的可验证外包计算研究大大丰富了外包计算方案中函数的类型, 许多关于可验证外包计算的研究都以一般函数为基础展开. 一般函数 (general function) 指的是布尔函数 (Boolean function). 布尔函数是一个输入任意长度的字符串并只输出 1 比特的 “0” 或 “1” 的函数, 任意一个函数都可以转化为一系列具有逻辑结构的布尔电路来实现, 一个可以计算布尔函数的方案可以用来计算任意函数. Gennaro 等 [73] 构造了一个满足标准模型的适用于一般函数的非交互式可验证计算方案, 这也是第 1 个具体的可验证计算方案. 该方案首先利用混淆电路 [75, 76] 将输入数据关联成随机标签, 再利用 FHE 对标签进行加密保证输入数据的隐私性以及可计算性, 服务器利用同态性来计算加密后的标签并返回给客户, 客户解密并利用混淆表还原验证结果. 该方案是第 1 个可以保证输入数据和计算结果保密性的方案. 虽然基于 FHE 的方案能够保证数据机密性和密文数据的可计算性, 但其构造涉及到较复杂密码学计算, 计算开销过于昂贵, 不能满足实际应用的需求. 为了使适用于一般函数的外包计算不再只依赖于全同态加密, Parno 等 [77] 提出了基于属性加密 (attribute-based encryption, ABE) 的一般函数可验证外包计算方案, 该方案满足公开可验证性和公开可委托性, 且由于没有使用复杂的 FHE 和概率检验证明方法, 用户和 SP 的计算和验证效率都有大幅度提高. 但该方案是通过得到最终的计算结果来进行验证, 因此不具有结果保密性.

当实际情况有多个用户参与计算时, 往往是不同的用户将不同的输入数据外包给第三方服务器进行同一个计算, 而具有公开可委托性的外包计算方案 [77] 虽然可以为多用户提供使用, 但是每次计算只能有一个用户参加. 为了计算一个输入数据来自不同用户的多输入函数计算, Choi 等 [78] 将 Gennaro 提出的可验证外包计算方案 [73] 从单用户拓展到了多用户情形, 提出了适用于一般函数的非交互式多用户可验证外包计算. 该方案结合非交互式的密钥交换协议和代理不经意传输协议 (proxy oblivious transfer), 来实现用户之间数据保密的可验证计算方案, 能够抵抗半诚实用户间的合谋, 即半诚实用户相互合作也不能获得诚实用户的数据信息, 但是不能抵抗恶意用户间的合谋. 后来, Gordon 等 [79] 又构造出了能够抵抗恶意用户合谋行为的多用户可验证外包计算具体方案, 但是该方案中用户基于私钥验证计算结果正确性, 因此无法满足公开验证的要求. 2015 年, Li 等 [80] 基于双服务器模型利用混淆电路、全同态加密和代理不经意传输协议提出了一个用于多用户公开可验证一般函数外包计算方案. 在该方案中, 一个云服务器负责生成计算开销大的与目标函数对应的混淆电路, 另一个云服务器负责计算函数值, 减轻了用户的计算负担, 当两个服务器不合谋且生成混淆电路服务器是诚实的情况下是安全的. Yang 等 [81] 基于密钥策略的属性加密 (key-policy attribute-based encryption, KP-ABE) 设计了具有结果保密性和公开验证性的多用户一般函数可验证计算方案. 该方案同样也使用了双服务器模

式来减少用户计算开销, 为了实现结果保密性在计算私钥时进行混淆处理, 但用户把计算任务委托给云服务器时需要使用私钥, 使方案不再满足公开可委托性. Sun 等 [82] 也基于 KP-ABE 提出了一个用于布尔电路的可验证外包计算方案, 同时实现了公开可委托性、公开可验证性和结果保密性. 他们用随机数来盲化输出的计算结果, 在验证时基于合数阶的双线性群性质抵消掉用于盲化的随机数. 在方案中还存在一个拥有此次计算输入值的代理方, 用户在执行外包计算时, 可以直接委托代理方执行数据外包工作, 但是该方案没有考虑拥有输入数据的代理方和拥有被委托计算的云服务器共谋攻击时的安全性.

此外, 基于 Paillier 密码体系的内积函数加密 (inner product functional encryption, IPFE) 作为一个新的密码工具受到关注, 它可以像传统加密方案一样通过加密保证数据隐私, 还可以类似于同态加密方案计算内积, 利用内积计算可以实现多项式计算和矩阵乘积等计算. Yang 等 [83] 提出在数据共享情形下基于 IPFE 的高效可验证内积计算方案, 实现了只有授权用户才能获得根据指定数据计算的结果. 该方案采用内积函数加密来保护外包数据的隐私性, 将密钥转换为盲化形式实现密钥的隐私性和结果的隐私性, 通过让 SP 在计算外包数据之前执行认证程序, 实现对数据用户和外包数据的访问控制, 这样还可以消除由未经授权的数据用户和不需要的数据导致的大部分计算开销.

4.2.2 具体函数的可验证计算方案

在 Gennaro 等 [73] 的非交互式可验证计算模型提出以后, 有许多研究进一步考虑了具体函数的可验证计算, 用于解决各种实际问题. 在真实的应用场景中, 外包给服务提供者的计算往往由具体的函数或多个数学函数运算组成, 而不是简单的布尔函数运算. 目前, 针对具体函数的典型可验证方案主要包括序列比较 [84,85]、多项式计算 [34,86~89]、矩阵乘法 [90~94]、矩阵求逆 [95~98] 和密码技术运算 [99~110] 等.

(1) 序列比较. 序列比较以各种方式和形式出现在分子序列比较以外的许多应用中, 尤其是在文本编辑、语音识别、机器视觉等领域. 当涉及到巨大的序列时, 往往将计算外包给具有强大计算能力的服务提供者. 但如果序列数据是敏感的, 比较结果是保密的, 或者两者兼而有之, 那么将数据发送到这些远程服务器是有问题的. 在这种情况下, 有许多关于安全外包序列比较计算的研究, 使客户能够远程解决问题, 但不会向远程超级计算站点透露计算的输入或结果. 序列比较问题指的是给定长度分别为 n 和 m 的两个字符串 λ 和 μ, 找到将 λ 转换为 μ 的插入、删除和替换的最小成本序列. Blanton 等 [84] 提出了一种非交互式的解决方案, 用户拥有字符串 λ 和 μ, 并将计算外包给两个远程服务器, 而不会向其透露有关输入字符串或输出序列的信息, 利用乱码电路评估技术不再使用公钥, 使得方案更高效. Atallah 等 [85] 针对广泛应用的序列比较问题给出了一个有效的安全外包协议, 让用户将序列比较安全外包给两个远程代理, 保护用户的两个私有序列和比较结果.

(2) 多项式计算. 文献 [34] 提出了首个实用的高次多项式计算的可验证计算方案. 用户把多项式的系数作为数据编码后外包给服务器, 服务器根据不同输入数据计算相应的多项式计算结果, 用户能够根据返回的计算结果及证据有效的验证服务器端的计算结果并还原结果. 该方案利用高效的 PRF 来提升数据预处理阶段的计算效率, 减少用户验证的计算开销, 还能抵抗 “结果比特泄露” 攻击, 即恶意的服务器知道了用户上一次计算的验证结果, 用户也不必重新执行密钥生成过程来保证之后计算的安全性. 为了适用于大数据量的高效二次多项式计算, 2013 年 Backes 等 [86] 利用可摊销计算的高效伪随机函数 (amortized closed-form efficient PRF) 构造出了一个对具有线性结构的函数保持计算同态性的同态消息认证算法, 实现了验证效率与输入数据大小无关. 然而, 在文献 [34,86] 中, 用户在验证服务器返回的计算结果时使用的是用户在本地保存的私钥, 无法解决 SP 和用户之间因计算结果正确性所产

生的纠纷. 为了解决以上问题, Fiore 等 [87] 提出了公开可验证的多元高次多项式计算计算的安全外包方案. 该方案在文献 [34] 的基础上, 将高次多项式扩展到多元情况下, 并将具有闭合有效性的 PRF 扩展到处理一组伪随机值和一组任意输入上运行的任何计算, 并与双线性结合实现公开可验证性. 2016 年, Sun 等 [88] 提出了批量公开可验证的计算方案, 用户可以外包多个系数不同的同次多项式, 对于相同输入的不同多项式的计算结果, 能够一次性验证其正确性. 类似文献 [87] 的构造方法, Sun 将具有闭合有效性的 PRF 和双线性映射应用到批量可验证方案中, 计算结果可以由多个用户同时验证, 实现批量公开可验证性的方案. 2019 年, Yu 等 [89] 又利用 FHE 和多项式分解算法提出了一种公开可验证的外包多项式计算方案, 其核心思想是找到一个高效的验证等式, 将用户计算多项式的开销转化为计算验证项的开销, 并利用双线性对的性质实现公开验证.

(3) 矩阵乘法. 大型线性方程组 $Ax = b$ 是科学界最基本的代数问题之一. 在实践中, 有许多现实世界问题将形成具有多达数十万甚至数百万未知变量的大规模密集线性方程组. 系数矩阵的存储需求很容易超过客户计算设备的可用内存, 计算外包成为一种高效的解决方法. 因此, 为解决工程学科中最常见的大型线性方程组安全外包问题, 出现了许多针对外包矩阵乘法的可验证研究. Wang 等 [90] 利用迭代算法提出了一种适用于大规模线性方程组的可验证计算方案. 该方案通过对线性方程组逐次迭代求解将求解大规模线性方程组转化为求解矩阵向量的运算, 降低了用户计算开销, 解决了大规模数据的外包计算, 并基于 Paillier 加法同态加密对每次迭代结果进行验证, 同时基于矩阵向量运算的代数性质实现了批量验证, 但该方案需要用户与 SP 进行多轮交互. 为了进一步提高大规模线性方程组计算的可验证效率, Chen 等 [91] 基于稀疏矩阵 (sparse matrix) 构造了一个适用于任何非奇异矩阵的大型线性方程组进行高效的安全外包计算方案. 该方案利用稀疏矩阵用来盲化方程组系数矩阵, 使用户的原始数据具有保密性且在计算时具有同态性, 验证时只需用户与服务器间一轮通信, 大大提高了效率.

在多用户共享网络的环境下, 为了节省用户摊销模型下的计算开销, 当不同用户用不同输入数据进行同一个函数计算时, 希望不同的用户可以只进行一次密钥生成, 即能实现公开委托性. 为此, Elkhiyaoui 等 [92] 提出了一个满足公开可委托性和公开可验证性的适用于高次多项式计算和大型矩阵乘法的可验证外包计算方案. 他们利用欧几里得带余除法构造验证等式, 不用生成私钥且验证密钥可公开, 并利用矩阵环中乘法的结合律实现了公开可验证的矩阵乘法同时具有公开可委托性. 但是, 该方案中用户将明文矩阵直接发送给服务器, 存在用户隐私泄露的问题. 针对隐私保护问题, Zhang 等 [93] 重点研究了用户数据的隐私保护, 提出了一种支持公开可验证和公开可委托的多个用户的批量矩阵乘法可验证计算方案. 该方案利用隐私保护矩阵变换技术对用户的私有矩阵进行盲化, 从而保护用户私有矩阵中的敏感内容保密性, 还引入了矩阵摘要技术将二维矩阵转换为一维向量, 且转换过程是不可逆的, 在保证安全的同时大大降低了数据中心密钥生成阶段和 SP 矩阵乘法阶段的计算开销. 为了在实现公开验证的同时对计算结果保密, Sun 等 [94] 提出了关于多元多项式计算和矩阵向量乘法计算的具有结果保密性的公开可验证外包计算方案, 方案只输出一个比特表示验证结果是否正确而不是结果的值, 在验证通过后用户利用在预处理阶段保存的恢复密钥还原出计算结果. 因此, 只有用户能从服务器返回的消息中恢复出最终计算结果, 实现了计算结果的保密性, 同时服务器返回的消息也具有公开可验证的性质.

(4) 矩阵求逆. 矩阵逆计算 (matrix inverse computation, MIC) 是线性代数中的基本数学任务之一, 在科学和工程的许多领域都有应用. MIC 任务通常涉及大规模矩阵, 对于资源受限的用户计算成本过大, 因此将大规模 MIC 任务外包给一个计算能力强的 SP 是一种有效的手段. 为了保护外包 MIC 中包含的用户隐私信息, Lei 等 [95] 构建了一种安全外包矩阵逆计算的方法. 用户首先通过一般稀疏

矩阵隐藏原始矩阵的值, 然后将隐藏矩阵外包到云. 云计算逆矩阵后, 将结果返回给用户. 但是, 该方案中矩阵变换虽然隐藏了非零元素的值, 但它显示了原始位置, 同样会暴露用户隐私数据. Hu 等 [96] 提出了一种安全且可验证的外包方案来计算矩阵求逆. 在该方案中, 用户基于两个混沌系统生成两个用于创建两个稀疏矩阵的密钥集, 其置换矩阵用于矩阵加密和解密以保护输入和输出隐私. 服务器在密文矩阵上求逆, 并将结果返回给用户, 用户可以验证逆矩阵的有效性. Pan 等 [97] 用某些矩阵变换构造了 MIC 安全外包方案 SEMIC, 该方案在不降低效率的情况下还提供了隐私保护, 可以隐藏输入矩阵中零元素的数量和位置. Chen 等 [98] 提出了一种新的用于矩阵求逆的安全外包协议, 不仅满足安全性、可验证性和效率的要求, 而且可以检查矩阵的可逆性. 此外, 该协议是为一个通用矩阵设计的, 原始矩阵不一定必须是可逆的, 还可以隐藏矩阵的原始数据, 完全保护输入和输出的隐私.

(5) 密码技术运算. 针对密码应用, 一些计算成本较高的运算外包给能力强大的 SP 展现出了许多优势, 但也暴露出了一些安全问题, 因此研究如何构建安全高效的密码运算外包方案具有重要理论与实践意义, 目前已有的可验证密码运算外包方案主要包括模幂运算和双线性对运算.

在基于离散对数的许多公钥密码算法中, 模幂运算是最广泛使用的运算之一. 由于模数往往是很大的, 模幂运算是非常耗时的. 因此, 产生了许多关于如何安全地外包模幂运算的研究. 模幂外包方案可分为两类: 双服务器方案和单服务器方案. 主要目标是保护输入和输出数据的隐私, 并通过将模幂中的基数和指数分解为更小的部分 (称为逻辑除法) 来支持模幂外包方案中的验证过程. Hohenberger 等 [99] 首先研究了可变指数、可变基数模幂运算的安全外包. 然而, 该方案是基于两个不可信程序模型提出的, 客户需要将指数运算外包给两个不会串通的计算服务器, 实际情况很难满足. 于是, Wang 等 [100] 提出了第 1 个通用安全指数计算外包方案, 该方案允许安全地将模幂运算外包给一个不受信任的计算服务器, 只需用户和计算服务器之间较少的交互, 在计算效率和隐私保护方面具有优势. Chen 等 [101] 提出了一种新的模幂运算安全外包算法, 用于两个不可信程序模型的一个恶意版本, 在效率和可检查性方面性能更优越. 还将此算法用作子例程, 实现了外包 Cramer-Shoup 加密、Schnorr 签名和变色龙签名的安全算法. 在单一不可信程序模型下, Ding 等 [102] 提出了一种新的模幂安全外包方案, 设计了一种新的模幂逻辑分裂方法, 将模幂分成几个部分, 并保留一个随机值以供验证, 实现了指数和模指数的隐私保护, 使得用户在外包过程中的计算负担很小, 能够以高检查能力验证结果. Su 等 [103] 研究了如何安全有效地外包复合模指数, 并提出了两种基于单个不可信云的复合模指数的安全外包算法, 分别为用于外包单复合模幂运算的 MCExp 和用于外包联立复合模幂运算的 SMCExp. 与基于两个不可信服务器的算法不同, 该算法更实用, 避免了必须存在两个没有共谋服务器的强烈假设. 不仅同时保护指数和基数的隐私, 而且使用户能够以高概率验证云返回结果的正确性, 还可以用于构造 Shamir 基于身份的签名和基于身份的多重签名的安全外包算法. Ren 等 [104] 基于欧拉定理提出了基于单服务器和双服务器的两种具有完全可验证性的 MExp 安全外包算法, 用户能检测出不可信服务器所有的恶意行为. 但 Qi 等 [105] 指出由于在验证步骤中误用了欧拉定理, 文献 [104] 的算法是不正确的, 并提出了基于双服务器的外包算法补救措施.

近来, 双线性对的提出能够实现密码学中一些全新的加密原语, 例如基于 ID 的加密方案和短签名方案等. 与模幂运算一样, 双线性对计算也是一项非常昂贵的计算操作. 因此, 双线性对的安全外包是密码运算外包的另一个研究热点. Chevallier-Mames 等 [106] 提出了一种用于具有单个不可信服务器的外包双线性对的安全可验证算法. 资源有限的设备可以检测服务器的恶意行为, 但是用户仍然需要执行许多非常耗时的操作, 如标量乘法、点加法和模幂运算. Chen 等 [107] 首先针对两个不可信程序模型中的双线性对提出了一种有效且安全的外包算法, 资源受限的用户不需要执行任何昂贵的操作 (如, 点乘运算或指数运算), 并在此算法基础上实现了基于身份的加密和签名的安全外包. 此外, 还有 Dong

表 7 外包计算可验证方案比较

Table 7 Comparison of verification schemes for outsourcing computation

Scheme	Type of computation	Key technology	Public verification	Public delegatability	Privacy protection	Access control
Gennaro et al. [73]	Boolean function	Garbled circuits; FHE	×	×	×	×
Parno et al. [77]	Boolean function	ABE	✓	✓	×	×
Choi et al. [78]	Boolean function	Non-interactive key exchange; proxy oblivious transfer	×	×	×	×
Li et al. [80]	Boolean function	Garbled circuits; FHE; proxy oblivious transfer	✓	×	×	×
Yang et al. [81]	Boolean function	KP-ABE	✓	×	✓	×
Sun et al. [82]	Boolean function	KP-ABE	✓	✓	✓	×
Yang et al. [83]	Boolean function	IPFE	✓	×	✓	✓
Blanton et al. [84]	Sequence comparisons	Garbled circuit techniques	×	×	✓	×
Atallah et al. [85]	Sequence comparisons	Secure tabl lookup protocol	×	×	✓	×
Yu et al. [89]	High degree polynomial	FHE; Polynomial decomposition	✓	×	×	×
Elkhiyaoui et al. [92]	High degree polynomial; large matrix multiplication	Amortized model; Euclidean division	✓	✓	×	×
Zhang et al. [93]	Multi-user batch matrix multiplication	Privacy-preserving matrix transformation; matrix digest	✓	✓	✓	×
Sun et al. [94]	Multivariate polynomial; matrix multiplication	Amortized model	✓	×	✓	×
Lei et al. [95]	Matrix inversion	Sparse matrix	×	×	×	×
Pan et al. [97]	Matrix inversion	Matrix transformation	×	×	✓	×
Su et al. [103]	Modular exponentiation	Euler's theorem	×	×	✓	×
Chevallier-Mames et al. [106]	Bilinear pair calculation	Computational indistinguishability	×	×	×	×

等 [108] 通过引入预计算, 提出了两种基于单个不可信服务器的单双线性配对和 t 个联合双线性配对的高效外包算法. 但这些预计算需要用户配备强大的存储空间和计算能力, 资源有限的用户很难运行所有算法. 而文献 [109] 提出了一种面向物联网应用的基于双线性对加密的安全外包算法, 该算法中的用户不需要额外的存储空间来存储大规模的预计算结果, 也不需要进行任何耗时的计算操作.

4.2.3 方案对比分析

关于可验证计算研究进展的文献 [98] 提出了一个外包计算可验证协议应该具有以下基本性质: 正确性、安全性和高效性. 正确性就是指在正常运行协议时, 服务器端返回的计算结果能够被用户接受. 安全性是指即使存在敌手攻击的情况下, 服务器端返回的计算结果能够被验证通过. 高效性是指验证服务器端返回计算结果消耗的资源较少, 最差情况下不应高于用户本身计算相应结果所耗费的时间. 此外, 根据实际应用需求, 外包计算可验证协议还应该具有公开可验证性、公开可委托性、隐私保护、访问控制等功能. 如表 7 [73,77,78,80~85,89,92~95,97,103,106] 所示, 我们主要从支持计算类型、关键技术和实现功能 3 个方面归纳总结了关于布尔函数、序列比较、多项式计算、矩阵乘法、矩阵求逆、模幂运算和双线性对计算的外包可验证计算研究现状.

4.3 SQL 查询验证技术

SQL 查询是我们日常接触最多, 应用最广泛的一类查询方式, 主要用于查询和管理关系数据库系统, 它操作由二维矩阵集合 (称为表) 组成的数据库. SQL 查询验证是查询验证领域研究得最为广泛的一种可验证方案, 往往针对一个包含 n 个属性值的数据点 $v=(x_1,\ldots,x_n)$, 如 (年龄, 姓名, 学号, 身高), 这类数据每个属性值可以表示具体的数值或者是字符串, 它们的取值往往存在一定取值空间, 其查询结果都会基于数据点的属性值做一些查询运算, 如范围查询 "select age >18 and height >175 cm", 针对 SQL 查询处理数据类型的不同进一步将该类方案划分为静态数据 SQL 查询验证、动态数据 SQL 查询验证与流数据 SQL 查询验证. 与非关系数据库查询不同的是, 关系数据库查询类型多种多样, 且查询结果往往包含多个数据结果, 返回查询结果与验证证据附加信息 (verification object, VO) 时, VO 不仅要保证数据动态更新可验证, 还要能保证查询结果的完备性, 因此, SQL 查询验证的安全目标是要满足正确性、完备性和新鲜性.

4.3.1 静态数据 SQL 查询验证

Devanbu 等 [111] 是最早将 MHT 树方案应用于 SQL 查询结果完整性与完备性的工作之一. 如图 7 所示, 以范围查询为例, 若返回的数据为图中黄色线框标注的 l_3,l_4,l_5, 对应的验证证据为 $\pi=\langle v_{1,0},v_{1,3},\mathrm{sig}(R),\mathrm{bound}=(l_2,l_6)\rangle$. 可以利用图中虚线框中的节点 l_3,l_4,l_5 和 $v_{1,0},v_{1,3}$ 作为证据值中的信息计算出哈希值, 之后利用根签名验证查询结果, 同时根据返回的边界值 $\mathrm{bound}=(l_2,l_6)$ 判断查询结果的完备性. 在他们的方案中考虑了多种不同类型的查询方式包括投影、连接、集合并集和交集. 对于数据集 D, 若查询返回的结果为 t 个数据点, 则返回的 VO 的大小为 $O(\log_2|D|)$. 为了使得 VO 的大小能够独立于数据集 D 的大小, Pang 等 [112] 设计了一种可验证 B-tree (verifiable B-tree, VB-tree) 的认证数据结构. VB-tree 中每个内部节点都有一个经过签名的摘要或哈希相关联, 摘要中包含了以该节点为根节点所有子树中节点的信息. 验证的过程与 MHT 类似, 但是与之前一直计算到整个树的根节点不同的是, VB-tree 只需要计算到包络树的根节点即可, 使得认证证据的大小不再依赖于认证树的深度, 只与查询结果的范围有关. VB-tree 虽然节省了通信开销, 但是 VB-tree 无法保证查询结果的完备性, 同时存在计算开销较大的问题. 针对 VB-tree 的不足, Nuckolls 等 [113] 提出了一种新的 ADS, 名为混合认证树 (hybrid authentication tree, HAT), 它将快速哈希函数与高效的基于 RSA 的单向准交换累加器结合起来 [114]. HAT 是一棵二进制搜索树, 在 HAT 中, 叶子节点存储数据的哈希值, 内部节点的哈希值分别来自所有左、右相邻节点的最低共同祖先的哈希值. HAT 树的最终摘要值是由作为所有节点的哈希值的累加器与根哈希值一起被计算出来的. 但是以上方案主要考虑的是静态数据, 即数据所有者不再发布数据更新, 数据不再动态改变.

4.3.2 动态数据 SQL 查询验证

针对动态数据 SQL 查询验证方案无疑是 SQL 查询验证的主战场, 根据 SQL 查询验证所支持的查询类型不同, 可分为特定 SQL 查询验证和通用 SQL 查询验证.

一、特定 SQL 查询验证方案. 特定 SQL 查询验证指针对范围查询、连接查询和聚合查询等特定查询类型的一类验证技术, 特定 SQL 查询验证主要通过构造可认证数据结构来实现, 根据查询对象的不同可分为一维查询和多维查询, 一维查询验证方案的可认证数据结构主要基于可认证树和基于签名链两类方法, 虽然这两类可认证数据结构在查询验证中研究的最为广泛, 但也逐渐显示出了结构本身的不足. 为了支持多维查询验证, 引入集合运算验证 (set operation verification, SOV)、概率验证、可信硬件技术 (trust hardware) 等, 极大丰富了查询验证支持的功能与性能.

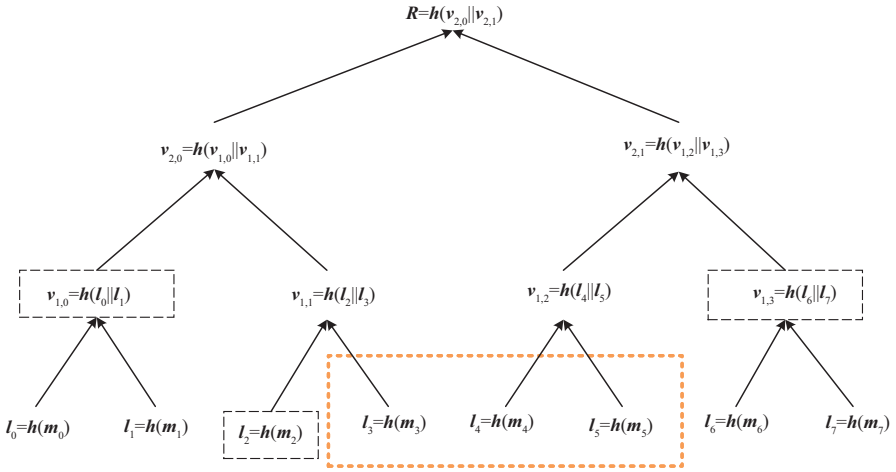

图 7 (网络版彩图) 利用 MHT 进行范围查询

Figure 7 (Color online) Use MHT for range queries

(1) 一维查询验证典型方案. 基于可认证树构造的一维查询验证典型方案是 Li 等 [115] 提出的支持范围查询的嵌入 Merkle B-tree (embedded Merkle B-tree, EMB-tree), EMB-tree 具有较高的查询效率和更新效率同时满足范围查询结果的正确性与完备性. Yang 等 [116] 基于 MB-tree 提出了支持连接查询验证的 3 种验证方案. Li 等 [117] 则针对聚合查询 (aggregate query) 验证, 提出前缀和认证树 (authenticated prefix sum tree, APS-tree) 方案和聚合 B-tree (authenticated aggregation B-tree, AAB-tree) 方案. AAB-tree 与 MB-tree 相同, 每个节点都与一个聚合值相关联, 作为其子节点的聚合值的和, 由子节点的散列值和聚合值连接而成哈希值. AAB-tree 支持数据的插入、删除和更新, 支持 COUNT, AVG, MIN 和 MAX 聚合查询验证等. 此外, 签名链也是一维查询验证中的常用 ADS, Mykletun 等 [118] 提出了第 1 个基于签名的范围查询认证方法. 该方法通过将多个元组签名组合到单个签名消息中, 极大地缩减了 VO. 在文献 [119] 中提出了第 1 个同时支持范围查询和连接查询的签名链查询验证方案, 而且该方案也是第 1 个支持数据动态更新的签名链方案, 它根据每个元组以及直接前导元组和后继元组的签名构造 VO. 但由于该方案要求元组根据其键属性值进行排序, 因此只支持对键属性的一维范围查询.

Wang 等 [120] 认为现有的方案无论是 Merkle 树还是签名链都没有考虑返回空集的情形, 即查询结果返回空集时, 现有的方案无法在不返回任何信息的情况下, 验证空集的有效性. 为此, 文献 [120] 中提出了一种布隆过滤器树 (Bloom filter tree, BFT) 方案, 该方案可以通过接收到的 VO 重构 Tuple-MHT 的根. 如果返回一个空集, 用户可以检查查询请求是否属于 BFT, 以确保搜索结果的正确性. 此外, 他们引入了一个第三方审计者, 通过存储的 BFT 可以消除由于 BF 的假阳性而引起的服务器与用户之间的纠纷. 但他们的方案在处理数据更新时效率较低, 每当发生一些更新操作时都需要执行计数过程, 不可避免地会导致大量的计算和通信开销. 针对上述挑战, Wang 等 [121] 提出了一种基于可逆布隆过滤器的查询验证方案, 该方案无需进行预计数即可实现搜索结果的可验证性. 通过结合多方可搜索加密将提议的方案扩展到多用户设置, 这不仅可以抵抗云服务器与任何恶意用户之间的共谋攻击, 还可以扩展查询验证方案完备性的研究范畴. 但是, 目前该方案只能处理一些特定查询返回空集的情形, 如, 聚合查询、连接查询, 依然无法解决对于任意查询返回空集的情形.

(2) 多维查询验证典型方案. 基于 SOV 的多维度查询验证方案最为代表性的是 Papamanthou

等 [122] 于 2011 年在 CRYPTO 会议上和 Canetti 等 [123] 在 2014 年的 PKC 会议上提出的两种方案. 文献 [122] 中支持对集合进行单一的运算方式 (如 $X_1 \cap \cdots \cap X_j$) 的验证. 文献 [123] 支持集合之间混合运算 (如 $(X_i \cup X_j) \cap X_k$) 的结果验证. 近年来一些研究工作发现现有的关系查询都可以通过集合之间的交并补运算来表达, 这种借助 SOV 强大的运算表达能力, 可以高效处理一些高维数据的查询验证.

Papadopoulos 等 [124] 于 2014 年 CCS 会议上指出现有的 ADS 都是基于单个属性值构造的, 不适合解决多属性间联合查询验证的问题, 当处理多属性间联合查询时, 需要为数据库中每个可能的维度组合构造一个 ADS, 这使得所需的 ADS 结构的数量与维数成指数关系. 为了解决以上问题, 他们将 SOV 引入查询验证, 将多维范围查询验证转化为基于数据库定义的集合上的集合运算操作, 这样对任意维度的范围查询验证可以通过集合之间的交、并、补运算来表达. 通过 SOV 与 MHT 结合实现了对任意维度的范围查询验证, 实现了方案复杂度仅与维度呈线性关系. 但该方案更新效率不高并且只能处理多维度之间范围查询. 为此, Zhang 等 [125] 提出了一种集合加法验证协议 (set operations with summation) 和一种叫做区间验证树 (authenticated interval tree, AIT-tree) 的认证数据结构, 并将其作为关键组件实现了一种验证系统 IntegriDB, 该方案不仅能处理范围查询, 还能处理连接查询和聚合查询, 但文献只能支持有限的 SQL 子集.

为了保障外包数据的安全和隐私性, 常采用数据加密方法, 但给查询验证带来了挑战. 为了支持对加密数据的多维范围查询验证, Wu 等 [126] 提出了一个安全高效可验证的多维范围查询方案 ServeDB, 这是第 1 个同时实现隐私、效率和可验证性的方法. 该方案使用基于分层立方体的编码方法, 分别对数据集和查询范围进行预处理, 将 d 维数据映射到 d 维空间, 并将空间划分为分层的 d 维立方体, 在立方体编码的数据之上建立一个树形结构加密索引, 并在树状结构中增加了验证信息, 以实现高效的查询处理以及查询结果的正确性和完整性证明.

二、通用 SQL 查询验证方案. 以上方案都是确定性的验证方案, 基于签名链或者 Merkle 树构建可认证数据结构, 往往一种结构只能高效处理一种类型的查询, 而概率验证思路的提出, 使得通用 SQL 查询验证成为可能, 与使用可认证数据结构的查询验证方法不同, 通用 SQL 查询验证方案不使用可认证数据结构进行认证, 只能返回概率性的完整性保证. 现有的通用 SQL 查询验证方案主要有两类: 基于 "检查点" 的可验证方案和基于电路的可验证方案.

典型的基于 "检查点" 的可验证方案是 Xie 等 [127] 最早针对 SQL 查询验证提出的一种概率验证框架, 用于检查外包加密数据库查询结果的可靠性、完备性和新鲜性. 主要思想是用户构造一些用于检测的假元组数据 T, 并且将其随机添加到原始数据库中的任意位置. 通过检查返回的结果是否包含 $Q(T)$ 来评估查询结果的正确性, 其中 $Q(T)$ 是在 T 上执行查询请求 Q 的结果. 伪元组方法在数据所有者端带来的验证开销比基于 MHT 树的确定性方法要低得多. 但为了保证较高检测概率, 需要插入较多的伪元组数量, 因此基于伪元组的方法在 SP 侧的计算成本比上述两种方法要高. 此外, 该构造中的一个隐含假设是所有用户都必须知道伪造的元组, 当 SP 与一些用户发起共谋攻击时可以欺骗其他诚实的用户而不被发现.

基于电路的通用可验证方案主要利用简洁的知识论证 (succinct non-interactive arguments of knowledge, SNARKs) [128, 129] 来处理任意的非确定性算术电路. SNARKs 为电路的正确评估提供了有效的、可验证的、恒定大小的证明. 早期通过 SNARKs 构建交互式可验证方案 [130~134], 虽然可以支持任意计算任务, 但需要一个特定于电路的预处理步骤, 把数据和查询程序都需要硬编码到证明密钥和验证密钥中, 从而需要一个昂贵的预处理步骤, 强加了极高的证明时间代价, 导致非常高的开销, 因此这些方法主要处于理论上的研究. 为了解决这个问题, Ben-Sasson 等 [135] 开发了一种 SNARKs 变

体, 其中预处理步骤仅取决于数据库和查询程序的上限大小. 最近, Zhang 等 [136] 于 2017 年 CCS 会议上提出第 1 个用于在动态外包数据库上的通用 SQL 查询可验证方案 vSQL, 该方案结合信息论的交互式证明系统 (interacive proof system) 和一种新的可验证多项式委托方案, 可以为证明提供辅助输入. vSQL 首先将 SQL 查询转化为运算电路, 然后基于一个信息理论交互式证明系统 CMT (Cormode-Mitzenmacher-Thaler) 协议 [137] 设计了一个新的多项式委托协议, 允许用户在任意点上能够评估某个取决于 x 的多元多项式 p_x. 与通用的基于交互式证明系统可验证计算方法 [128,129] 相比, 它在 SP 端带来了显著的性能提升. 而在查询验证方面的性能也可以与只支持受限制 SQL 查询子类的确定性方法 [125] 相当, 然而, 它仅限于具有固定模式的关系数据库.

4.3.3 流数据 SQL 查询验证

针对流数据的 SQL 查询验证一直以来都是查询验证领域的难点问题, 相比于动态数据, 可以在初始阶段就构造用于验证的 ADS, 流数据是逐步获取到的, 所以也就无法在初始阶段构造用于认证的数据结构, 目前流数据的 SQL 查询验证还处于初期发展阶段.

现有典型的方案主要有 Li 等 [138] 用改造的 MHT 处理数据流上单个属性上的聚合查询验证. 数据源基于连续的 b 个数据构造一个 MHT, 这将再产生 $\lceil N/b\rceil+1$ 个 MHT, 在每一个 MHT 中所有的元组都按照属性值 A 进行排序, 在 MHT 中不仅对根哈希值进行签名, 还对数据中的最老的时间戳 t_L (认证树中数据最早到达时间) 和最新时间戳 t_U (认证树中数据最晚到达时间) 进行签名 $\mathrm{sig}(h_{\mathrm{root}}|t_L|t_U)$. 此外, 他们还将方案拓展到多维属性的流数据聚合查询和验证. 为了用户能够验证位于公共属性 A 的属性值在特定选择范围内元组的新增、删除和更新, 即订阅查询 (subscription queries), Papadopoulos 等 [139] 基于 MHT 提出了临时 Merkle 哈希树 (temporal Merkle Hash tree, TMH-tree), 该方案的主要思想是将属性 A 定义域划分为多个分区, DO 将每个分区组织在一个 TMH-tree 中, 再为每个分区构建一个分域 Merkle 树 (domain-partition-Merkle-tree, DPM-tree), DO 将 DPM-tree、它的签名和 TMH-tree 分发到查询服务器. 此后, 每当出现元组更改时, 所有者将继续更新 DPM-tree、它的签名和 TMH-tree. 查询验证的过程与基于 Merkle 树方案类似, 查询结果的正确性由 DPM-tree 的根签名来保证, 完备性还是由边界来保证. PIRS 方案 [140] 第 1 个解决了对数据流的分组聚合查询验证问题, 但该方案不具有公开可验证性, 同时无法解决客户不可信的问题, 恶意的客户可能会与 SP 发起共谋攻击 [141]. 为了解决以上问题, Nath 等 [141] 提出了一种称为流式直方图摘要 (digest for streaming histograms, DiSH) 方案可以用于解决数据流的分组聚合, 且 DiSH 方案具有公开可验证性. 后来, Papadopoulos 等 [142] 对 DiSH 方案进行了拓展, 使其可以解决更加多样的数据流查询方案. 他们首先构建了解决数据流环境下线性代数查询运算, 包括: 一个或多个流产生的动态向量的和、不同流产生的动态向量的点积, 以及不同流产生的动态矩阵之间的积 3 种基本查询验证方案. 这些方案可用于验证流环境中的各种重要操作, 包括分组查询、连接、网络内聚合、相似度匹配.

4.3.4 私有信息检索验证

私有信息检索 (private information retrieval, PIR) 的概念最早由 Chor 等 [143] 在 1998 年提出, 是一种使用户能够从公共数据库检索记录, 同时保持所检索记录的私有标识的密码方案. 在 PIR 方案中, 允许客户机从拥有数据库的服务器检索条目, 而不透露检索的是哪个条目. PIR 在一些实际场景中, 服务器可能会由于恶意行为或意外故障而提供错误答案, 只要用户能够识别该答案, PIR 方案就可以有效的工作. 因此, 验证返回结果正确性是 PIR 中一个值得研究的重要问题.

目前主要存在两种类型的 PIR: 单服务器 PIR (single-server PIR, SPIR) 和多服务器 PIR (multi-

server PIR). Zhao 等 [144] 针对恶意服务器模型下的单查询情况, 给出了第 1 个实用的非交互式可验证单服务器 PIR 方案 VSPIR, 采用了有效的概率证明发现伪造的结果. 此外, 还扩展了单查询 VSPIR 方案, 构建了一个非交互的多查询解决方案. 相应的通信复杂度和计算复杂度与一些典型的 SPIR 方案相当. Zhang 等 [145] 构建了一个无条件且计算安全的 k-server 可验证多服务器 PIR 方案 (t-private k-server VPIR), 支持数据库的高效更新, 识别恶意服务器, 容忍轻微损坏的结果以及支持多个外包数据库. 但 VSPIR [144], t-private k-server VPIR [145] 的安全性是基于密码学假设, 不能提供信息论的安全性, 并且需要一个可信的第三方来预处理数据库并设置公钥进行验证. 此外, 这些可验证 PIR 协议由于使用非常慢的公钥操作速度较慢. 于是, Ke 等 [146] 构建了一个针对不诚实服务器的安全的 1-private 2-server PIR-RV 协议. 与文献 [144, 145] 相比, PIR-RV 协议只需要对有限域 Fp 进行计算, 速度更快, 同时提供信息理论安全, 不需要可信第三方.

4.3.5 方案对比分析

由于本小节不同方案其使用的技术以及面向的功能特点不同, 我们无法以统一的量化标准来比较其计算开销与通信开销, 因此我们主要从安全性评价、支持查询类型、实现功能、结构特点和主要贡献对比分析典型方案, 如表 8 [111, 112, 115, 124~126, 136, 138~141] 所示. 通过对现有的研究方案分析, 可以看出 SQL 查询验证中的数据每个属性值往往是具体的数值或者是字符串, 验证时需要对查询做运算, 且查询结果往往包含多个数据结果, 不仅要保证数据动态更新可验证, 还要能保证查询结果的完备性. 许多研究方案也是利用了 MHT 这种验证数据结构的特点, 并对其改进来支持数据流范围查询和订阅查询. 此外, 将 SOV 引入查询认证, 还能用于支持多维范围查询.

4.4 空间数据查询验证技术

随着 5G 网络、自动驾驶、车联网的迅速发展, 手机终端、智能穿戴设备在人们日常生活中扮演着越来越重要的角色, 嵌入定位功能的移动设备和基于位置服务 (location-based-services, LBS) 的出现, 多种类型的空间数据外包引起了人们的关注. 而现实中, 服务器可能只会执行部分查询操作以节省计算成本, 或者故意操纵查询结果. 因此, 有必要在空间数据查询中引入可验证技术, 确保查询结果的完整性. 空间数据查询验证的主要特点是验证对象是空间数据库中多属性的位置数据点, 空间数据库中存储的地图数据等信息往往一段时间内并不需要更新, 但验证用户是移动的, 需要查询验证的数据结果会受验证者所处的空间位置影响, 例如, 我们在行驶的汽车上用手机查询 “离汽车最近的医院”, 查询结果可能就会随着汽车位置的变动而发生变化. 因此, 根据查询验证者的位置是否发生变动将空间数据查询验证进一步分为静态空间查询验证与移动空间查询验证. 此外, 近年来人们对隐私保护问题越来越重视, 空间数据往往含有大量的隐私信息, 具有隐私保护的空间查询验证方法逐渐成为空间数据查询验证技术中需要重点研究的问题. 基于以上的分析, 本小节主要从静态空间查询、移动空间查询和具有隐私保护的空间查询 3 个方面对现有技术及典型方案进行论述.

4.4.1 静态空间查询验证方案

第 1 个针对空间数据的完整性验证方案是由 Cheng 等 [147] 提出的, 他们在 KD-tree [148] 和 R-tree 的基础上结合签名链 [119], 提出用于确保查询完整性的 VKD-tree 和 VR-tree. 其思想是对每个分区内的数据点以及数据空间中的所有分区构造签名链, 并将认证信息添加到空间数据结构中. 但是, LBS 的空间查询处理开销较大, DO 初始构建时签名计算成本较高, 用户的验证负担相当重. 为此, Yang 等 [149] 在 VR-tree 的基础上进行改进, 提出了适合空间数据外包验证的基于 Merkle 哈希的 R-tree

表 8 SQL 查询验证方案比较

Table 8 Comparison of SQL query validation schemes

Scheme	Safety evaluation	ADS	Types of queries			Function evaluation		The main contribution
			Range	Join	Agg	Public verification	Privacy protection	
MHT/MB-tree [111]	Deterministic	Tree	✓	✓	×	✓	×	First query verification scheme using MHT
VB-tree [112]	Deterministic	Tree	✓	×	×	✓	×	First scheme independent of dataset size
EMB-tree [115]	Deterministic	Tree	✓	×	×	✓	×	Embedding MB-tree
Papadopoulos [124]	Deterministic Minimality	Tree	✓	×	×	✓	×	Introduce SOV
IntergriDB [125]	Deterministic	Tree	✓	✓	✓	✓	×	Multi-attribute joint SQL query
ServeDB [126]	Deterministic	Tree	✓	×	×	×	✓	First multi-dimensional range query verification
vSQL [136]	Minimality	Tree	✓	✓	✓	✓	×	First general SQL query verification scheme
TM-tree [138]	Deterministic	Tree	✓	×	✓	✓	×	MHT supporting data stream range query
Papadopoulos [139]	Deterministic	Tree	✓	×	×	✓	×	MHT supporting subscription queries
PIRS [140]	Deterministic Minimality	Sig-chain	×	×	✓	×	×	Stream grouping aggregation queries
DiSH [141]	Deterministic Minimality	Sig-chain	×	×	✓	✓	×	Stream grouping aggregation queries with publicly verifiable

(Merkle Hash-based R-tree, MR-tree), 用于处理空间查询验证. MR-tree 可以看做是 MHT 在高维空间的拓展, 相比于 VR-tree, 其构建速度更快, 占用的空间也更少, 对查询处理和验证的效率更高, 还支持更新. 此外, 针对同一移动用户的连续查询都集中于数据空间的一小部分的特点, 他们还提出了一种同步缓存技术, 利用以前的查询结果来缩减 VO. 与签名链方案相比 MR-tree 方案节省了计算开销, 但在有些情况下需要返回查询结果之外的节点信息用于辅助验证, 造成额外的通信开销. 为了克服这些缺点, Yang 等 [150] 又改进了 MR-tree, 提出了 MR*-tree 进一步减少返回的非应答节点的数量. MR*-tree 采用了一种新的嵌入技术, 将节点内的空间划分为子分区, 在每个叶节点中的嵌入 KD-tree [148], 在内部节点中的嵌入 box-KD-tree [151], 验证时只需要返回重叠子分区中的点. 与 MR-tree 相比, 大大缩减了用于验证的附加信息, 但为 LBS 和用户带来了更多的 CPU 开销. 然而, MR-tree 和 MR*-tree 在实际应用中的维度都存在一定限制, 方案的性能和效率都随着维度的增加而降低. Mouratidis 等 [152] 也提出了一种改进的 KD-tree, 称为部分 KD-tree (partially materialized KD-tree, PMKD-tree), 它在查询响应时间、存储开销和构建索引结构成本等方面的性能显著优于 MR-tree. 该方案主要特点是将查询处理和生成证据任务分离, 委托给不同的服务器, 可以为只需要快速响应的查询提供未经验证的响应, 也可以直接对现有数据库进行验证, 而无需修改和重新上传.

为了加快验证过程的查询效率, 空间划分技术 Hilbert 曲线和 Voronoi 图在空间查询验证中得到了广泛应用. Hilbert 曲线能够将多维节点转为为一维节点同时维持节点邻近的性质, Voronoi 则具有非常高的临近节点搜索效率, 在地理信息系统数据分析中具有非常广泛的应用. Sharifzadeh 等 [153] 借

助 Voronoi 图的高效邻近搜索特性基于 MR-tree 提出了新的方案 VoR-tree, 使得其查询效率显著提高, 同时支持多种邻近查询, 包括 k–最近邻居查询 (k-nearest neighbors, k-NN)、反向 k 近邻查询 (reverse k-nearest neighbors, Rk-NN) 和 k 近邻聚合查询 kA-NN. 同样为了降低通信和查询运算的时间开销, Hu 等 [154] 将签名方案与 Voronoi 图相结合提出了 VN-Auth 方案, 支持 k-NN, Rk-NN, kA-NN 以及空间 skyline 查询. VoR-tree 和 VN-Auth 在初始化时需要构建 Voronoi 图, 同时每个节点都需要存储与其邻近的 Voronoi 生成元的位置信息, 相比于 MR-tree 和 MR*-tree 存储开销更大, 但提高了查询速率, 是典型空间换时间的方案.

随着边缘计算的发展, 计算任务从中心云向边缘云迁移, 而大多数空间数据查询验证研究都局限于集中式环境, Zhang 等 [155] 在分布式框架中提出了一种用于 k-NN 查询验证的数据结构, 称为分布式 MR-tree, 其索引结构包括指向每个分区中数据点的本地索引和含有每个本地 MR-tree 根哈希组成的全局索引. 由于全局索引很小, 可以被存储在所有节点中, 加快了 k-NN 查询处理的速度. 他们还使用网格分区的方法将整个数据集分区, 保证任何两个分区的最小边界框之间不重叠, 可以只使用几个分区来处理 k-NN 查询, 节省了计算资源.

4.4.2 移动空间查询验证方案

用户位置实时变化的空间查询称为连续空间查询 (continuous spatial queries, CSQ) 或移动空间查询, 实现了随时随地准确定位. 与传统的空间查询 (对象是静态的, 查询只计算一次) 不同, CSQ 必须在对象移动时针对用户实时变化的位置生成最新的查询结果, 而且查询效率对用户体验至关重要, 因此, CSQ 更具挑战性. 在早期 CSQ 方案中都是将用户运动简化为线性函数, 如, Tao 等 [156] 使用 R-tree 作为底层数据结构, 将处理连续单个邻居查询扩展到多个最近邻居查询; Iwerks 等 [157] 研究如何在用户移动速度改变时保证未来用户的 k-NN 查询. 然而很多时候用户未来的移动是未知的, 这种无法预测的 CSQ 主要分为基于缓冲区 [158] 和安全区域 [159~161] 两种方案.

在基于缓冲区的方案中, 对于用户的 k-NN 查询, LBS 可以基于以往静态查询方案返回 $k+\Delta k$ 个最近的点, Δk 的取值控制返回额外的位置点. 具体来说, 将 q_{last} 表示为用户最后一次查询时候的位置, d_i 为第 i 个最近邻居的位置, 缓冲区则定义为 $\odot(q_{\text{last}}, r)$, 表示为一个圆心为 q_{last} 半径为 $r = \frac{1}{2}(d_{k+\Delta k} - d_k)$ 的圆. Song 等 [158] 已经证明当用户在缓冲区 $\odot(q_{\text{last}}, r)$ 中移动时, 用户最新的查询结果可以从缓存的 $k + \Delta k$ 个位置点中获取到. 在缓冲区中存储精确预测的下次查询点候选位置可以降低初始搜索范围, 当用户离开缓存区时, 只需要向 LBS 发送一个 $(k + \Delta k)$-NN 查询请求. 然而, 在缓冲区中位置发生变化时, 用户需要根据返回的位置点自行计算出更新后的查询结果, 这给用户造成了额外的通信开销.

安全区域是指 LBS 为用户查询提供的一个区域, 客户在该区域移动时, 查询结果保持不变, 当用户离开安全区域时, 会向 LBS 发出一个新的查询. 与基于缓冲区的方法不同, 这种方法不需要客户机在本地计算结果, 不仅降低了 LBS 与用户之间的通信成本, 而且还降低了用户的计算开销. 然而, 安全区域是由 LBS 根据查询方式进行划分的, 基于认证树或 Voronoi 的针对静态用户的空间查询验证方案, 虽然能保证返回查询结果的完整性和完备性, 但是无法保证安全区域的完整性和完备性. 例如, 渗入攻击者的 LBS 服务器返回的查询结果中包含的都是攻击者指定的特定位置, LBS 可能会返回过大的安全区域来节省计算资源和通信带宽, LBS 也可能会选择返回过小的安全区域使得客户更频繁地请求新的安全区域. 因此基于安全区域的 CSQ 方案设计目标是为如何划分安全域, 以及如何为用户设计一种有效方法来验证 LBS 返回的安全区域的正确性. Yung 等 [160] 基于 MR-tree 提出了一种用于移动范围查询验证的方案, 利用几何知识划分安全区域, 远程服务器用查询结果和查询点当前所在的

相应安全区域回答用户, 用户根据返回的安全区域验证查询结果. 然而, 该方案在构建安全区域时需要额外的计算开销. 针对安全区域查询结果的安全验证, Papadopoulos 等 [161] 提出了一种解决高动态空间数据库中连续范围处理和验证的方案, 利用 Hilbert 曲线将多维数据转换为一维数据, 并在 MB-tree 的基础上设计了一个新的索引结构 DPM-Tree 处理数据验证, 减少了 LBS 和用户之间的传输开销.

4.4.3 具有隐私保护的空间查询验证方案

然而, 就在基于位置的服务给人们带来了巨大便利的同时, 位置数据等隐私信息的泄露面临着严峻挑战. 因此, 如何保证在空间数据验证的同时保证高质量的位置服务而不泄露用户隐私受到了高度关注. 为实现隐私保护, Hu 等 [162] 结合 R-tree 和网格文件索引 (grid-file index) 提出了分别适用于大型查询、小型查询和静态数据集查询的 3 种空间数据范围查询验证方案. 他们利用线性排序以及将每个节点映射到最小边界框 (minimum bounding boxes, MBB), 实现了对数据的有效修剪, 重新获得了多维数据的线性形式, 并利用保序加密 (order-preserving encryption) 对数据进行加密来保护用户数据的隐私. 但是, 由于网格文件索引使用签名链进行查询验证, 查询用户必须花费更多的计算资源进行签名验证. Ku 等 [163] 提出了一种能够确保外包空间数据隐私性的完整性验证方案. 他们通过随机选择一部分的外包数据进行概率复制并使用不同的加密密钥进行加密, 采用一种基于空间填充曲线 Hilbert 的单向函数来对空间数据进行加密来保证数据的隐私, 然后将加密后的空间数据外包给 LBS, 根据服务器返回的证明数据用户能够验证空间数据查询 (包括范围查询和 k-NN 查询) 结果的完整性, 同时还支持数据的更新、插入和删除操作, 但该方案是一个概率性的验证方案, 而且还引入了第三方可信安全网关, 有一定的实现困难. 为了实现具有隐私保护的查询验证, Jang 等 [164] 设计了一种基于周期函数的数据分组方案, 秘密地将空间数据库划分为若干小组, 并为每个组生成用于检查查询结果完整性的签名, 保证了用户数据的机密性和查询结果的完整性. 但这些方案都只能处理特定的范围查询, Wang 等 [165] 基于处理向量内积的对称可搜索加密设计了一种具有隐私保护的空间查询验证方案. 他们基于 BF 的概率性加密为同一消息生成不同的密文实现数据隐私保护, 借助 BF 集合交运算转化为向量内积计算, 实现在密文状态下检验两个 BF 的集合交运算, 进而完成验证方案的判定. 但在该方案中, 每次访问时用户都需要与 DO 交互, 以获取方案的访问令牌, 若查询用户过多, DO 会成为方案的瓶颈.

4.4.4 方案对比分析

综上所述, 本小节从查询者的状态、安全性评价 (包括确定性、最小化)、实现功能 (包括支持查询类型、公开可验证、隐私保护)、构造 ADS 时使用的技术等方面对现有典型空间数据查询验证方案进行对比分析, 如表 9 [147,149,150,152~155,158,160~163,165] 所示. 我们发现, 为了减少查询响应时间、存储开销和索引结构构建成本, 大量基于 KD-tree 和 R-tree 改进的 ADS 研究方案被提出. 这类树形的可验证数据结构在空间查询验证中应用的也很广泛, 它可以结合其他验证数据结构的优点, 来支持空间数据库中多属性的位置数据验证, 同时还能提高验证效率. 此外, 为了加快验证过程的查询效率, 空间划分技术 Hilbert 曲线和 Voronoi 图也在空间数据查询验证中得到了广泛的应用.

4.5 可搜索加密验证技术

在现实应用要求以及相关法律法规要求下, 为了防止外包数据的隐私信息泄露给第三方服务器, 外包数据往往是以密态的形式存在, 如果每次验证都需要先解密显然是不现实的, 因此如何无需解密直接对密态数据查询结果进行可信性验证是值得深入研究的问题. 可搜索加密 (searchable encryption,

表 9 空间数据查询验证方案比较

Table 9 Comparison of spatial data query verification schemes

Scheme	Querier status	ADS	Supported query types	Determinacy	Minimality	Public verification	Data privacy
Cheng et al. [147]	Static	KD-tree; VR-tree	Range queries; K-NN	✓	✓	✓	×
Yang et al. [149]	Static	MR-tree	Range queries; K-NN	✓	×	✓	×
Yang et al. [150]	Static	MR*-tree	Range queries; K-NN	✓	×	✓	×
Mouratidis et al. [152]	Static	PMKD-tree	Range queries; K-NN	✓	×	✓	×
Sharifzadeh et al. [153]	Static	VoR-tree	Range queries; K-NN; R-NN; K-ANN	✓	✓	✓	×
VN-auth [154]	Static	Sig-chain	Range queries; K-ANN; K-NNR-NN; Skyline	✓	×	✓	×
Zhang et al. [155]	Static	Distributed MR-tree	Range queries; K-NN	✓	✓	✓	×
Song et al. [158]	Mobile	R-tree	Range queries; K-NN	✓	×	×	×
Yung et al. [160]	Mobile	MR-tree	Range queries; K-NN	✓	×	✓	×
Papadopoulos et al. [161]	Mobile	DPM-tree	Range queries; K-NN	✓	✓	✓	×
Hu et al. [162]	Static	R-tree; Grid-file index	Range queries; K-NN	✓	✓	✓	✓
Ku et al. [163]	Static	Double space encryption	Range queries; K-NN	✓	✓	×	✓
Wang et al. [165]	Static	Bloom filter	Range queries; K-NN	✓	✓	✓	✓

SE) [166] 技术的出现为外包加密数据的搜索查询提供一种有效解决方法. 云环境下可搜索加密模型的基本工作原理是用户将本地数据加密后上传至云服务器, 随后数据检索方通过某些关键词生成陷门, 云服务器利用陷门对密文进行搜索操作, 得到包含目标关键词的密文数据并返回给数据检索方. 虽然传统的可搜索加密技术可以提供隐私保护, 但没有考虑搜索结果的可验证性, 云服务器可能会为了节省计算资源而自私, 从而只返回一部分结果来回复搜索查询. 因此, 对可搜索加密查询结果进行验证是非常值得研究的问题.

根据数据搜索方的身份可搜索加密协议可以分为两类: 当数据检索方为数据拥有者, 即用户本身时, 称为对称可搜索加密 (symmetric searchable encryption, SSE) [167]; 当数据检索方为非用户方的第三方时, 称为非对称可搜索加密, 或者公钥可搜索加密 (public key encryption with keyword search, PEKS) [168]. 基于此, 可搜索加密验证技术也可分为两类, 对称搜索加密可验证技术和公钥搜索加密可验证技术, 下面分别进行详细论述.

4.5.1 对称搜索加密可验证技术

2012 年, Chai 等 [169] 提出了第 1 个可验证搜索加密的概念, 给出了可验证搜索加密 (verifiable searchable encryption, VSE) 协议的形式化定义, 并且构造了一个具体的支持单关键词精确搜索的对称可搜索加密 (verifiable symmetric searchable encryption, VSSE) 方案. 在执行搜索后, 服务器会给出搜索结果和根据搜索结果的标识符串联生成的证据, 用户验证接收到的证据是否与预处理生成的查找树节点的标识符匹配, 通过概率性验证来检测服务器伪造行为. 随后, Wang 等 [170] 和 Sun 等 [171] 分别提出了支持模糊关键词搜索和多关键词搜索的 VSSE 方案. Wang 等 [170] 提出了一种基于索引树构造验证结构的可验证模糊关键词搜索的可验证方案, 索引树中的每个节点的路径都是唯一的, 拥有私钥的授权用户可以通过重构证据来验证搜索结果的正确性, 但由于验证时使用了私钥, 不能从真正意义上实现公开验证. 为了支持多关键字搜索和搜索结果排序, Sun 等 [171] 提出了基于词频的搜索索引和具有余弦相似性度量的向量空间模型, 以获得更高的搜索结果精度, 还提出了基于索引树的搜索结构, 使得实际搜索效率远高于线性搜索.

然而, 以上这几种 VSSE 方案只适用于静态数据, 且仅支持单个用户, 这意味着只有 DO 才能进行搜索, 而实际上人们通常使用云存储与其他用户共享文件. 于是, Zhu 等 [172] 提出了单所有者 – 多用户模型中第 1 个通用的可验证 SSE 方案. 为了更普遍地支持结果验证, 首先将证明索引解耦, 然后利用 Merkle 前缀树 (Merkle patricia tree, MPT) 和增量哈希来构建支持数据更新的证明索引, 还新增了一个时间戳链, 用于维护跨多个用户的数据新鲜度, 能够支持数据的有效更新. 2020 年, Liu 等 [173] 也提出了一个多用户可验证搜索对称加密方案 MVSSE, 实现了支持动态数据操作的具有隐私保护的可验证 SSE, 并允许多个用户执行搜索. 该方案分别在数据和加密的索引项两处部署了用于验证的累加器, 数据用户可以通过服务器返回的证明来检验搜索结果的正确性, 还利用双关键字索引和逻辑操作组合减少搜索时间. 此外, 通过结合委托多项式函数和可验证的前向安全 SSE, Miao 等 [174] 提出了一种具体的基于关键字的 VDB 方案, 不仅可以在 VDB 上实现前向安全关键字搜索, 而且支持所有更新操作.

4.5.2 公钥搜索加密可验证技术

由于攻击者可以通过离线关键字猜测攻击推断出有问题的关键字, 基于对称加密算法实现的安全可验证关键字搜索在公钥可搜索加密中可能是不安全的. 于是在可验证公钥加密可搜索领域, 2014 年, Zheng 等 [175] 第 1 次提出了可验证的基于属性的关键词搜索 (verifiable attribute-based keyword search, VABKS) 协议, 采用基于 BF 和数字签名的验证方案实现了数据用户可以有效地验证云是否诚实地对外包加密数据执行了用户的搜索操作, 同时还实现了访问控制. 此外, 由于关键字密文、搜索令牌和算法容易被外部对手捕获, 对手可以通过匹配碰撞来破坏关键字密文的不可区分性, 无法抵抗离线攻击. 为了抵抗离线猜测攻击, Liu 等 [176] 在 VABKS 的基础上改进, 只有 SP 可以使用公钥重新加密关键字密文和搜索令牌, 匹配关键字密文和搜索令牌, 有效地防止离线猜测攻击, 还降低了验证的计算复杂度, 更具有实用性.

在文献 [175,176] 中只能由发出查询的实体验证结果, 而在实际中公开验证搜索结果应用更广泛. 因此, Alderman 等 [177] 构造了一个基于密文策略属性加密的公开可验证搜索加密协议. 该协议不仅支持更细粒度的关键词表达 (包括: 与、或、非、等逻辑询问和关键词的简单计算), 还支持阈值运算、多项式计算以及模糊搜索等, 同时实现了公开验证搜索结果的功能. 但该方案只支持静态数据, 而且还需要提前规定数据使用者具有的搜索权限, 数据使用者和持有者需进行多次交互. 2017 年, Zhu 等 [178] 基于双线性对提出了较为完整的公开可验证公钥搜索加密模型, 可以实现密文和陷门的不可区分性, 同时抵抗关键词猜测攻击. 为了丰富公钥搜索加密可验证技术的应用场景, Liu 等 [179] 提出了一个支持多个 DO 和多个数据检索者场景下的公钥搜索加密可验证方案. 他们设计了一种可以有选择地共享搜索和验证权限的聚合密钥, 减少分享密钥的计算、通信开销, DO 只需向其他用户分发一个聚合密钥, 用户不仅可以使用它生成的陷门作为关键字搜索查询, 还可以验证服务器是否进行了完整的搜索.

为了提高搜索精度, Liang 等 [180] 实现了加密文件的精确多关键字细粒度搜索方案 VPAMS, 方案中的数据用户不仅可以检查返回的文件是否包含多个查询关键字, 还可以通过基于属性的加密实现对搜索文件的授权解密. 随着量子处理器的进步, 包括云计算在内的各个技术领域的加密结构的安全性受到了严重威胁. Yousefipoor 等 [181] 提出了第 1 个实用的、语义安全的、基于秩度量码的后量子 PEKS 方案, 对关键字猜测攻击、反应攻击和密钥暴露都是安全的, 并且具有关键词连读搜索、使用 Bloom 过滤器和混淆 Bloom 过滤器实现了搜索结果可验证.

表 10 可搜索加密验证方案比较

Table 10 Comparison of verification schemes for searchable encryption

Scheme	Type of SE	Key technology	Supported search types	Applicable scene	Public verification	Public delegatability
VSSE [169]	SSE	Index tree	Single-keyword	Single-owner & single-user	×	×
Wang et al. [170]	SSE	Index tree	Fuzzy-keyword	Single-owner & single-user	×	×
Sun et al. [171]	SSE	Index tree	Multi-keyword	Single-owner & single-user	×	×
GSSE [172]	SSE	MPT; Incremental Hash	Single-keyword	Single-owner & multi-user	×	✓
MVSSE [173]	SSE	Accumulator	Single-keyword	Single-owner & multi-user	×	✓
VABKS [175]	PEKS	ABE; BF	Single-keyword	Single-owner & single-user	×	×
Liu et al. [176]	PEKS	KP-ABE	Single-keyword	Single-owner & single-user	×	×
Alderman et al. [177]	PEKS	CP-ABE	Fine-grained query	Single-owner & single-user	✓	×
Zhu et al. [178]	PEKS	Bilinear pairing	Single-keyword	Single-owner & single-user	✓	×
Liu et al. [179]	PEKS	Aggregate key	Single-keyword	Multi-owner & multi-user	✓	×

4.5.3 方案对比分析

如表 10 [169~173, 175~179] 所示, 我们从 SE 类型、关键技术、支持搜索类型以及适用场景对可搜索加密验证技术进行归纳, 并分析了这些方案是否实现了支持动态数据和公开可验证等功能. 其中, 适用场景 (applicable scene) 中的 owner 指的就是数据本身的拥有者 (data owner), user 指的是使用数据的用户, 数据拥有者可以是使用数据的用户, 但使用数据的用户不一定是数据拥有者. 从中我们发现, 基于索引树的搜索结构使得搜索效率更高, 利用 MPT 用于支持数据更新, 时间戳链能够用于维护数据新鲜度, 支持数据的有效更新, 基于属性的关键词搜索协议还能够实现访问控制. 现有的可搜索加密验证方案大都围绕关键词检索以及范围检索, 支持的搜索方式不够丰富, 对动态数据的支持也不够理想, 因此未来针对查询验证的隐私保护和丰富查询模式的研究是值得探索的.

4.6 基于可信硬件的数据完整性验证技术

以上可验证方案大多数是基于软件定义的 ADS (如, 哈希表、向量承诺、MHT) 来实现数据完整性验证, 对方案的运行环境没有特定要求. 而随着可信硬件技术的发展, 通过在不受信任服务器上的部署防篡改、受信任的可信硬件 (trusted hardware) 来实现可验证技术也逐渐引起了人们的关注. 可信硬件可以独立地完成预先写入的函数或密码算法而不受外界干扰, 同时能够很大程度地确保其内部的代码或状态信息不被外界获取或修改, 能保证计算和存储受硬件保护. 目前, 适用于大规模数据处理的可信硬件技术主要有 3 种, 安全协处理器 (security coprocessors, SCPU)、现场可编程阵列 (field-programmable gate arrays, FPGA) 和软件保护扩展 (software guard extension, SGX). 可信硬件技术中的验证工作由 SCPU 完成, 通用的 SCPU 可以被编程来执行任意的查询验证, 消除了对查询的表达限

制, 并且可以用来验证范围、连接和聚合查询, 以及其他更加复杂的查询验证. 此外, SCPU 通过 PCI 总线与主机服务器进行本地通信, 还能降低验证时用户与 SP 的通信开销.

Bajaj 等 [182] 于 2013 年的 SIGMOD (Special Interest Group on Management of Data) 会议上首次提出基于可信硬件安全数据库查询验证方案 Corre-ctDB, 它们利用 IBM-4764 SCPU, 在不受信任的云环境中提供了一个安全的存储与计算环境来实现数据的查询验证. 他们的方案依然需要借助 ADS, 主要思想是在构建 MHT 后, 将根哈希值存储在 SCPU 中, 由于 SCPU 可以保证根哈希值的安全, 这样就可以无需再对生成的根哈希进行签名, 从而在数据更新与验证都不需要再进行开销昂贵的签名操作. 因此, 使用基于 SPCU 的设计可以避免 MHT 在更新时面临的同步问题. DO 可以直接向 SCPU 发出查询请求, 所有的更新和查询结果验证都由 SCPU 执行, 这样不仅不会产生额外的数据传输开销, 还避免了用户重新更新根节点信息. Arasu 等 [183] 则在 2017 年的 SIGMOD 会议, 完全抛弃了以往基于软件定义的 ADS 的方案, 基于 FPGA 和 SGX 可信硬件实现了支持并行更新存储的完整性验证方案 Concerto. Concerto 设计的核心理念是将 key-value 存储数据的完整性验证问题简化为内存完整性验证, 通过使用效率高、操作开销恒定的离线内存检查算法 (offline memory checking) 验证内存完整性, 还采用最先进的并行优化索引技术将数据完整性验证与查询索引分离, 较大地提升方案处理效率, 但 Concerto 只适用于 key-value 存储下的环境处理简单的键值查询. Bajaj 等 [184] 基于同样的设计理念在 2019 年的 SIGMOD 会议提出了针对关系数据库上的高并发更新环境下 SQL 查询验证方案 ConcurDB. 与 Concerto 类似, ConcurDB 将查询验证转化为离线的内存验证, 能处理多种 SQL 查询验证 (如, 范围查询、连接查询、映射查询) 并支持数据插入、更新、删除的高效离线查询验证, 还可以基于可信硬件 SCPU 的强制访问控制机制过滤恶意更新操作.

虽然可信硬件有其优点, 但是并不能完全取代基于非硬件的查询验证技术, 两者是互补的. 从物理角度看, 可信硬件处于不可信的服务器上, 作为服务器的组件, 它与服务器直接相连, 通信延迟低、带宽高, 而且可以被多个租户共享使用, 摊销成本. 从安全角度看, 可信硬件的安全保护比查询验证更全面, 不只可以保护数据库中的数据, 数据库执行状态的机密性和完整性也可以得到保障. 但问题是可信硬件中的存储资源和计算能力都相对有限, 如果将所有查询处理都放在可信硬件中, 而不充分利用不可信硬件, 那么系统整体性能表现难以达到最优. 而且随着查询验证技术由当前简单的数据库 SQL 查询已逐渐向越来越复杂的机器学习运算任务的发展, 可信硬件在这种更加复杂的外包运算任务中表现出越来越严重的不足. 出于可信硬件与不可信硬件之间负载平衡的考虑, 基于可信硬件的安全云数据库系统上也需要利用非硬件的查询验证技术把部分工作负载放在不可信硬件上完成. 在可部署性方面, 可信硬件的部署往往需要 DO 与服务商共同协商, 而这一问题则在技术之外. 虽然当前都认为可信硬件安全强度高, 但可信硬件的部署无疑将攻击焦点都集中在了可信硬件上, 一旦被攻破, 则会造成大规模的安全泄露问题, 因此这种技术虽然非常有前景, 但是依然不可过分依赖.

5 可验证技术的新应用

随着大数据、边缘计算、区块链、机器学习、联邦学习等新技术的快速发展, 数据的产生、存储、计算以及验证方式都发生了巨大的变化. 与此同时, 人们对数据隐私、服务体验的关心重视也给数据验证带来了诸多挑战, 本节结合新技术的发展, 对可验证技术在新技术中的应用现状进行论述并对其下一步发展趋势进行展望.

5.1 面向大数据的可验证技术

大数据的容量 (volume)、多样性 (variety) 和速度 (velocity) 3V 属性对可验证技术提出了不同的挑战, 大数据容量要求可验证方式必须具有可扩展性, 在初始化和验证时不会增加验证本身的大量开销. 速度属性要求验证方法必须有效地支持频繁的更新. 多样性要求能够支持各种数据类型的验证, 简单的方法是为每种数据类型构造一个 ADS, 但这可能会带来巨大的验证开销. MapReduce 已经成为大数据中重要的计算模型, 其基础设施主要由两个原语组成, 一个是将输入分配给多个映射节点用来并行生成中间值的 map 函数, 一个是允许多个还原节点合并关联中间值来生成最终输出的 reduce 函数. 然而, 所有映射节点和还原节点都可能被破坏, 返回错误的中间结果或最终结果. 虽然文献 [185~187] 对 MapReduce 模型的计算结果的可验证进行了研究, 构造 ADS 以支持对映射节点得到的中间结果和还原节点得到的最终结果的有效验证. 但当前研究成果还很难应用于 MapReduce 等分布式计算框架的远程计算结果验证, 一方面是由于采用复杂性理论和密码学理论构造的可验证计算协议性能开销过大, 另一方面是由于大部分可验证计算协议不支持远程输入, 即验证者必须处理所有的输入和输出, 无法表示 MapReduce 程序. 综上, 未来在面向大数据的可验证技术研究中一方面针对大数据的容量、多样性和速度等特性, 研究通用的可适应多种数据类型的、可扩展的高效可验证数据结构, 于此同时还需考虑大数据局部细粒度验证与整体高效验证之间融合问题, 以及频繁且微小更新与持续增长并行对大数据可验证的影响; 一方面针对大数据中的计算模型, 重点研究由大多数工作者构成的映射节点上, 将计算分解为更小的部分 (即中间结果生成), 以解决分布式查询结果的验证开销问题, 以及构建模块化的结构, 来验证计算生成的中间和最终结果的完整性.

5.2 面向边缘计算的可验证技术

随着边缘计算技术的快速发展, 许多应用程序提供商将他们的应用程序和相关数据部署到分布式边缘服务器上, 为附近的用户提供高效的低延迟服务. 然而, 在高度分布式、动态、资源受限、安全手段缺失的边缘计算环境中, 这样的缓存数据容易受到有意或无意的破坏. 因此, Tong 等 [188] 第 1 个提出了两种用于移动边缘计算中的可验证方案, 一种是基于可信第三方且具有隐私保护的数据完整性验证协议 ICE-basic, 一种是基于聚合验证思想的数据完整性验证协议 ICE-batch, 分别用于用户希望在单个边缘云节点或多个边缘云节点上检查数据完整性的情况. 为解决边缘云数据完整性问题的多副本验证, Li 等 [189] 提出了一种基于采样的轻量级的概率检测方法 EDL-V, 用于在计算资源有限的边缘服务器上验证副本完整性, 边缘云只需生成特定抽选的子树进行数据完整性验证. 然而, 实际情况边缘数据副本的总数很大, 在该方案中使用一棵哈希树来管理所有副本并不可行. 因此, Li 等 [190] 又提出了一个基于聚合验证思想的 EDI-S 方法, 利用椭圆曲线签名为每个副本生成证明, 通过聚合验证减少复杂乘积计算次数, 可以任意选择聚合子集进行并行计算, 还可预先计算验证时需要用到的一部分参数, 从而实现大量数据的高效完整性验证. 但是, 该方案为了传递验证消息, 通信开销较大, 且不能在数据块级别定位或修复损坏的边缘数据副本. 为了更高效地解决边缘云数据完整性问题, 文献 [191] 又基于分布式共识机制设计了一种数据完整性验证方案 CooperEDI, 形成了去中心化的自管理边缘缓存系统, 边缘服务器协同保证缓存副本的完整性和修复损坏副本, 无需应用程序供应商的远程控制就可以自主交互. 由此可见, 边缘计算完整性验证与云计算完整性验证有不同之处, 需要对多个副本数据验证, 而且还需找出被损坏的副本. 此外, 边缘计算环境中往往需要接入海量的数据, 边缘服务器和应用程序提供商都需要更高效且轻量级的可验证方案. 根据已有研究, 得知采用概率性方案不能完全保证所有副本的完整性, 而采用聚合验证虽然可以验证所有副本但是通信开销较大. 因此, 在边缘计

算环境中实现高效的数据验证以及精确定位损坏副本位置, 在未来仍具有研究价值.

5.3 面向区块链的可验证技术

去中心化的数据外包是一个新的发展趋势, 最近, 包括 IBM[4], Oracle[5]和 SAP[6]在内的主要数据库供应商都将区块链与他们的数据库管理系统集成在一起, 并且它们允许用户通过数据库前端在区块链上执行查询. 区块链是一种仅支持追加操作的数据结构, 分布式存储在网络中. 随着以太坊代表的第二代区块链的出现, 该技术被用作一般数据 (如, 文本、图像等) 的可信存储解决方案. 区块链从两个方面确保了数据完整性, 首先, 借助哈希链技术确保存储在区块链上的数据是不可篡改的; 其次, 区块链共识协议可确保所有对等方都维护相同的数据副本. 但是, 区块链数据库也处于一个公开的环境, 这样的受信方并不总是存在, 使得查询结果的完整性并不能得到保证. Xu 等 [192] 于 2019 年的 SIMGOD 会议上首次研究了区块链数据库的可验证查询处理问题, 提出了一种新的可验证处理框架 vChain. 该框架将存储和查询服务委托给一个功能强大的全节点, 而查询用户仅作为一个轻节点接收结果, 来解决区块链数据库的外包数据完整性验证问题. Wang 等 [193] 又在 vChain 的基础上改进, 提出了一种新的可验证范围查询区块链系统 vChain+. 他们设计了一种滑动窗口累加器 (sliding window accumulator, SWA) 用在每个块中构建 ADS, 实现高效的查询处理, 方案中还设计了一个对象注册索引 (object registration index), 用一个小整数 ID 注册和索引每个数据对象, 从而限制了公钥大小, 用户使用对应于 ID 的具有完整性保证的索引检索最终查询结果. 由于区块链的不可变性、全节点不可信性以及数据量的持续增长性等问题, 区块链中的可验证问题仍然面临着巨大挑战. 如何构建新颖的 ADS 使其适应数据量持续增长且趋于无限的区块链数据库, 如何基于可验证技术验证用户上链数据与链下数据的一致性, 如何满足资源有限的验证用户, 特别是移动用户的需求, 这些都是值得思考和研究的问题. 此外, 还需要进一步研究在有效工作量证明共识机制中使用可验证计算来验证工作, 而不依赖于可信的执行环境. 因此, 研究区块链下的外包数据可验证技术是一个极具吸引力的研究课题.

5.4 面向 MLaaS 的可验证技术

机器学习即服务 (machine learning as a service, MLaaS) 已经成为一个很有前景的服务范式, 它使弱用户能够在强大的云基础设施中训练机器学习模型或者进行预测运算. 然而这些外包计算也带来了安全问题, 客户如何确保不可信的云是否进行了正确的深度神经网络训练, 相信返回的结果确实是通过完整的原始数据如实训练且没有遭受外部攻击, 这就提出了如何验证这些外包机器学习计算结果可信性的问题. 然而, 现有的 ADS (如 MB-tree, VB-tree, MHT, IHT) 不能适用于外包机器学习和数据挖掘计算的验证解决方案. 为了确保深度学习中外包神经网络训练返回结果的正确性, Ghodsi 等 [194] 基于和校验协议提出了 SafetyNets 模型. Zhao 等 [195] 提出了一种外包 MLaaS 的概率性验证可信性框架 VeriML, 使训练过程具有可恢复性, 验证者可以对任意特定迭代的输入和输出进行重构, 同时可以在不获取任何训练模型信息的情况下验证结果正确性, 避免用户在验证阶段获取到任何关于训练完毕的模型信息. 大数据分析、机器学习过程中, 数据共享越来越重要, 为了解决敏感数据的共享和分析, Froelicher 等 [196] 提出了一个 Drynx 系统用于对分布式数据集进行具有隐私的统计查询以及机器学习回归模型的训练和评估. Drynx 依靠一组计算节点来实现标准差或极值等统计数据的计算, 以及对敏感和分布式数据的机器学习模型的训练和评估. 为了确保数据的保密性和数据提供者的隐私, Drynx

4) Enterprise blockchain solutions and services. 2018. https://www.ibm.com/blockchain.
5) Transforming the enterprise with oracle blockchain platform. 2018. https://www.oracle.com/cloud/blockchain/.
6) Blockchain applications and services. 2018. https://www.sap.com/products/leonardo/blockchain.html.

结合了互动协议、同态加密、零知识正确性证明和差分隐私, 能够对输入数据和所有系统的计算进行有效和分散的验证, 从而在一个强对抗性的模型中提供可审计性, 在这个模型中没有实体需要被单独信任.

最近, 联邦学习以其无需收集用户原始数据的训练网络能力逐渐受到学术界和业界的关注, 所有用户和云服务器只通过共享本地的梯度和全局参数协同工作. 此时, 用户隐私依然存在被泄露的风险, 一旦对手获得了用户本地梯度就能够操纵返回给用户的聚合结果 (全局模型), 同时用户很容易被服务器返回的伪造结果所欺骗, 导致训练失败. Xu 等 [197] 提出了第 1 个可验证的具有隐私保护的联邦学习框架 VerifyNet, 在深度神经网络训练过程中, 基于结合伪随机技术的同态哈希函数验证聚合结果的可信性, 同时使用双掩码协议来保证用户数据隐私, 确保训练过程中存在退出或掉线用户时依然能够实现验证, 但该方案的验证开销与模型中参数的数量呈线性关系. 为了提高验证效率和降低通信开销, Guo 等 [198] 将线性同态哈希与承诺方案结合提出了验证聚合结果完整性方案 VeriFL, 并采用批量验证的方法, 平摊昂贵的哈希计算开销, 但随着用户退出次数的增加, 为了恢复出用户的相关参数, 通信开销还会增加, 系统性能也会下降. 为了更适用于真实场景, Han 等 [199] 在 VerifyNet 基础上利用无证书签名的密钥交换协议, 提出了一种无需可信第三方的可验证联邦学习方案. 此外, Fu 等 [200] 结合单掩码和拉格朗日 (Lagrange) 插值技术实现了用户梯度和全局模型的隐私保护, 但处理用户退出问题并不完善. Xu 等 [201] 提出一种基于双服务器的非交互式可验证隐私保护联邦学习体系结构 NIVP-DS, 利用随机矩阵编码 (random matrix coding) 结合交叉验证实现了局部梯度的隐私保护和聚合梯度的验证, 无需用于处理用户退出或掉线的额外开销. 通过研究分析, 现有方案主要基于可验证技术判定联邦学习中 SP 计算聚合结果是否可信的问题, 但可验证技术的引入还需同时兼顾用户梯度和全局模型的隐私保护问题, 支持用户在训练过程中掉线或退出, 以及提高聚合效率、降低通信开销等问题还需要进一步深入研究.

随着外包环境下可验证计算任务趋于复杂化, 概率验证方案在计算通用性方面的优势能得到更大的发挥, 现有涉及各种类型的数据挖掘计算的验证方案大多都采用的了概率验证的方法. 但是, 概率验证方案容易受到伪造数据点的恶意用户攻击, 攻击者可以区分 “哨兵” 数据点与真实数据点, 而确定性的验证方法会带来昂贵的开销, 也不适用于高度复杂的机器学习计算. 因此, 未来针对这种复杂的外包 MLaaS 计算机制, 研究安全高效的可验证方案仍是一个值得深入探索的方向.

6 总结与展望

当今时代, 数据量剧增, 算法、算力要求越来越高, 通过外包服务来实现大量数据存储和计算是必然的发展趋势, 但这些外包数据或外包计算结果是否正确、是否可信等问题也日益突出, 可验证技术被视为解决该问题的一种有效且具有广泛应用前景的技术. 目前, 由于验证需求、验证场景、验证功能、验证类型等不同, 已形成了许多不同的面向外包服务的可验证方案和方法, 我们通过总结归纳现有可验证技术方案, 梳理给出了一种面向外包服务的可验证技术框架, 基于该框架从关键技术、服务类型、实现功能以及在新技术中的应用等 4 个方面对现有可验证技术进行分类论述, 并从安全性、实现功能、性能和应用场景等方面对典型方案进行比较分析, 指出了目前该类研究存在的问题和未来的研究热点. 最后, 在新应用背景下对可验证技术未来的研究方向进行了展望, 并提出了一些有待解决的问题.

虽然面向外包服务的可验证技术已取得了很多重要成果, 但随着新技术、新应用的不断涌现以及实用性要求的不断提高, 还需要进行大量的深入研究和创新性研究. 未来的研究热点及面临的挑战本

文前面已进行分类论述, 总体来说主要可以归纳为以下 4 个方面.

(1) 可重构、可组合的智能化可验证通用模型研究未来是一个重要研究方向. 随着数字中国智慧应用新场景的不断落地应用, 数字中国建设进入创新发展新时代, 未来数字化、智能化必然不断深入到人们生活的方方面面, 在此背景下, 未来应用场景多变, 数据体量会越来越大、动态更新越来越频繁, 计算复杂度越来越高, 可重构、可组合的智能化可验证通用模型研究必然是一个重要研究方向. 然而现有可验证方案大多采用复杂性理论或密码学理论进行构造, 可验证计算性能开销过大, 很难适应于未来大数据规模、复杂计算的结果验证, 此外传统的可验证模型验证证据的计算和生成都是完全集中的, 存在单点故障和隐私泄露的风险, 如何创新性构建新颖的自主可重构、可组合的智能化可验证通用模型, 将可验证计算、验证模型的构建、验证证据的生成与验证都进一步分解化、模块化、组件化, 如何根据应用场景需求、数据体量变化、算力要求、效能评估进行智能化重构和组合, 降低单个节点的计算开销, 防止单点破坏、局部失控和隐私泄露, 提高可验证计算模型自身的容错率、鲁棒性和抗攻击能力是需要深入探索的问题.

(2) 面向外包服务的可验证技术基础理论研究. 计算复杂性和密码学是构建可验证方案的重要理论基础, SNARKs 理论、可随机检查证明定理 (PCP theorem)、可靠性证明 (CS proof)、密码算法、同态加密、零知识证明等理论被广泛应用于构建满足不同场景、不同应用需求的可验证计算方案. 随着新的证明系统 (如 STARKs, SNARKs, Bulletproofs)、多变量公钥密码学 (multivariate public key cryptography)、量子计算、格密码等新理论的发展与应用必然会成为可验证技术发展的重要支撑, 为可验证技术的理论发展带来颠覆性影响, 因此基于计算复杂性新理论和密码学新理论开展面向外包服务可验证技术基础理论研究是非常重要的研究方向. 如, 大规模量子计算机的实现将破解现有广泛使用的公钥加密算法, 甚至会威胁某些后量子候选算法, 而目前绝大部分的可验证方案是无法抵抗量子攻击的, 因此研究抗量子攻击的可验证方案也必然会成为一个新方向.

(3) 新颖的具有隐私保护的轻量级动态实时可验证技术研究. 大数据分析与挖掘、统计查询、机器学习及 AI 协同等领域往往都需要大量的数据作为支撑. 尽管多年来数据存储和计算成本已经下降, 特别是由于低成本和强大的云计算的出现, 但这些数据的共享仍然很困难. 因为随着物联网、智慧医疗、无人驾驶等应用的发展, 每天产生的大量数据中可能包括个人的行动、兴趣、爱好、健康、购物习惯、政治和宗教观点等信息, 这些信息涉及隐私或及其敏感, 欧洲数据保护法规《通用数据保护条例》(GDPR) 的出台, 我国《数据安全法》《个人信息保护法》的出台, 数据的存储、使用、出境等都提出了具体的新要求, 如, 数据共享之前必须进行匿名化或加密. 隐私问题和道德/法律约束使得人们或企业更加不愿或不能共享数据. 因此, 数据的隐私问题不仅从技术上而且从法律法规上都将进一步制约和限制外包数据存储、分析、计算和管理等外包服务的应用与发展, 处理好数据隐私问题未来将成为评价可验证技术的重要指标. 虽然目前已存在具有隐私保护功能的可验证方案, 但仍然存在强诚实假设、至少需要一个可信实体、计算开销大等不足, 而且随着时代的发展和需求的变化, 隐私保护的内涵和技术较传统隐私保护都已有所扩展, 如, 在隐私计算中, 目前基于差分隐私来实现数据隐私保护是一个热点研究问题, 通过向原始数据随机加入噪声来保护数据隐私, 可验证技术在验证过程中如何识别随机噪声, 随机噪声与概率性可验证方案中的随机 “检查点” 能否协同工作来提高效率等问题都需要进一步深入研究, 此外, 目前大多数终端设备虽然能够源源不断的产生大量数据, 但自身普遍是资源受限的设备, 这样就要求验证方案既要能够对终端源源不断的数据进行动态实时验证又要满足轻量级要求. 因此, 如何基于先进的隐私计算技术, 研究具有隐私保护的轻量级动态实时可验证方案, 如何在隐私保护安全和效率之间找到最优的平衡点等必将是未来面临的主要挑战和研究热点, 具有隐私保护的轻量级高效可验证技术未来可广泛应用于工业控制和自动化、跨域医疗、智慧交通等领域, 具有

重要的研究意义.

(4) 可验证技术与新兴技术相互促进发展. 可验证技术与边缘计算、区块链、联邦学习等新兴技术相互促进发展. 一方面, 在边缘计算、区块链、联邦学习等新兴技术架构下, 可验证技术依然具有广泛应用前景但研究还处于起步阶段, 仍然面临着许多待解决的难点问题, 需要我们进一步深入研究和攻克. 如, 在边缘计算环境下, 资源受限的用户面对海量数据, 需要更高效且轻量级的数据验证的方案. 现有的采用概率性方案不能完全保证所有副本的完整性, 而采用聚合验证虽然可以验证所有副本但是通信开销较大. 因此, 在边缘计算环境中实现高效数据验证的同时还能精确定位损坏副本位置, 仍值得进一步研究. 在区块链环境下, 数据采用分布式副本存储, 每一个节点都保存有一份数据副本, 且区块链的链式哈希指针结构可确保其上数据无法被任意删改, 可以提供有效的数据完整性保证. 但有众多节点参与的大型数据区块链存储及验证成本较为昂贵, 结合具体的应用场景找到合适的激励机制和共识算法仍是一个研究热点. 又如, 联邦学习为当今数据安全共享提供了一种全新的非聚合式数据共享模式, 由于联邦学习环境下的聚合计算的验证数据不是元数据而是模型, 可验证技术如何实现分布式机器学习环境下模型高效验证的同时防止模型攻击、中毒攻击、隐私泄露等问题还尚未解决, 需要进一步深入研究. 反之, 新兴技术的发展也会进一步促进可验证技术的发展, 如, 基于区块链技术研究可验证技术的去中心化和可追溯问题, 基于隐私计算技术研究可验证技术的隐私保护问题等研究都需要我们进一步深入挖掘和探索.

参考文献

1 Hacigumus H, Iyer B, Mehrotra S. Providing database as a service. In: Proceedings of the 18th IEEE International Conference on Data Engineering, San Jose, 2002. 29–38

2 Cloud Security Alliance. The Notorious Nine Cloud Computing Top Threats in 2013. Technical Report, 2013

3 Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, 2007. 598–609

4 Juels A, Kaliski B S. PORs: proofs of retrievability for large files. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, 2007. 584–597

5 Zhu Y, Ahn G J, Hu H X, et al. Dynamic audit services for outsourced storages in clouds. IEEE Trans Serv Comput, 2013, 6: 227–238

6 Ateniese G, Pietro R D, Mancini L V, et al. Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, Istanbul, 2008. 1–10

7 Erway C C, Küpçü A, Papamanthou C, et al. Dynamic provable data possession. ACM Trans Inf Syst Secur, 2015, 17: 1–29

8 Sookhak M, Yu F R, Zomaya A Y. Auditing big data storage in cloud computing using divide and conquer tables. IEEE Trans Parallel Distrib Syst, 2018, 29: 999–1012

9 Shen J, Liu D, Bhuiyan M Z A, et al. Secure verifiable database supporting efficient dynamic operations in cloud computing. IEEE Trans Emerg Top Comput, 2017, 8: 280–290

10 Chen X, Li J, Huang X, et al. New publicly verifiable databases with efficient updates. IEEE Trans Dependable Secure Comput, 2015, 12: 546–556

11 Zhang Z, Chen X, Li J, et al. HVDB: a hierarchical verifiable database scheme with scalable updates. J Ambient Intell Hum Comput, 2019, 10: 3045–3057

12 Miao M, Wang J, Ma J, et al. Publicly verifiable databases with efficient insertion/deletion operations. J Comput Syst Sci, 2017, 86: 49–58

13 Chen X, Li H, Li J, et al. Publicly verifiable databases with all efficient updating operations. IEEE Trans Knowl Data Eng, 2021, 33: 3729–3740

14 Chen X, Li J, Weng J, et al. Verifiable computation over large database with incremental updates. IEEE Trans Comput, 2016, 65: 3184–3195

15 Wang Q, Wang C, Ren K, et al. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans Parallel Distrib Syst, 2011, 22: 847–859
16 Li L. Research on some issues of data security in cloud computing services. Dissertation for Ph.D. Degree. Hefei: University of Science and Technology of China, 2013 [李凌. 云计算服务中数据安全的若干问题研究. 合肥: 中国科学技术大学, 2013]
17 Liu C, Chen J, Yang L T, et al. Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates. IEEE Trans Parallel Distrib Syst, 2014, 25: 2234–2244
18 Chen X, Tao S, Kim I, et al. A remote data integrity checking scheme for big data storage. In: Proceeding of the 2nd International Conference on Data Science in Cyberspace, Shenzhen, 2017. 53–59
19 Yang K, Jia X. An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans Parallel Distrib Syst, 2012, 24: 1717–1726
20 Wang C, Chow S S M, Wang Q, et al. Privacy-preserving public auditing for secure cloud storage. IEEE Trans Comput, 2013, 62: 362–375
21 Shah M A, Baker M, Mogul J C, et al. Auditing to keep online storage services honest. In: Proceedings of the 11th USENIX Workshop on Hot Topics in Operating Systems, 2007. 1–6
22 Shah M A, Swaminathan R, Baker M. Privacy-preserving audit and extraction of digital contents. 2008. https://eprint.iacr.org/2008/186.pdf
23 Zhu Y, Wang H X, Hu Z X, et al. Zero-knowledge proofs of retrievability. Sci China Inf Sci, 2011, 54: 1608–1617
24 Wei L, Zhu H, Cao Z, et al. Security and privacy for storage and computation in cloud computing. Inf Sci, 2014, 258: 371–386
25 Du W, Jia J, Mangal M, et al. Uncheatable grid computing. In: Proceedings of the International Conference on Distributed Computing Systems, Tokyo, 2004. 4–11
26 Saeednia S, Kremer S, Markowitch O. An efficient strong designated verifier signature scheme. In: Proceedings of International Conference on Information Security and Cryptology, 2004. 40–54
27 Wang B Y, Li B C, Li H. Oruta: privacy-preserving public auditing for shared data in the cloud. IEEE Trans Cloud Comput, 2014, 2: 43–56
28 Wang B, Li H, Li M. Privacy-preserving public auditing for shared cloud data supporting group dynamics. In: Proceedings of the IEEE International Conference on Communications, Budapest, 2013. 1946–1950
29 Wang B, Li B, Li H. Public auditing for shared data with efficient user revocation in the cloud. In: Proceedings of the IEEE International Conference on Computer Communications, Turin, 2013. 2904–2912
30 Yuan J, Yu S. Efficient public integrity checking for cloud data sharing with multi-user modification. In: Proceeding of the IEEE International Conference on Computer Communications, Toronto, 2014. 2121–2129
31 Wang C, Wang Q, Ren K, et al. Ensuring data storage security in cloud computing. In: Proceedings of the International Conference on Parallel Distributed & Grid Computing, Solan, 2010. 217–222
32 Shacham H, Waters B. Compact proofs of retrievability. J Cryptol, 2013, 26: 442–483
33 Boneh D, Lynn B, Shacham H. Short signatures from the weil pairing. J Cryptology, 2004, 17: 297–319
34 Benabbas S, Gennaro R, Vahlis Y. Verifiable delegation of computation over large datasets. In: Proceedings of Annual Cryptology Conference, 2011. 111–131
35 Catalano D, Fiore D. Vector commitments and their applications. In: Proceedings of International Workshop on Public Key Cryptography, 2013. 55–72
36 Miao M, Ma J, Huang X, et al. Efficient verifiable databases with insertion/deletion operations from delegating polynomial functions. IEEE Trans Inform Forensic Secur, 2018, 13: 511–520
37 Wang Q, Zhou F, Xu J, et al. Efficient verifiable databases with additional insertion and deletion operations in cloud computing. Future Generation Comput Syst, 2021, 115: 553–567
38 Anthoine G, Dumas J-G, Jonghe M, et al. Dynamic proofs of retrievability with low server storage. In: Proceedings of the 30th USENIX Security Symposium, 2021. 537–554
39 Curtmola R, Khan O, Burns R, et al. MR-PDP: multiple-replica provable data possession. In: Proceedings of the 28th International Conference on Distributed Computing Systems, Beijing, 2008. 411–420
40 Hao Z, Yu N. A multiple-replica remote data possession checking protocol with public verifiability. In: Proceedings of the 2nd International Symposium on Data, Privacy, and E-Commerce, Buffalo, 2010. 84–89

41 Barsoum A F, Hasan M A. Integrity verification of multiple data copies over untrusted cloud servers. In: Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Ottawa, 2012. 829–834
42 Yi M, Wei J, Song L. Efficient integrity verification of replicated data in cloud computing system. Comput Secur, 2017, 65: 202–212
43 Li L M, Yang Y H, Wu Z H. FMR-PDP: flexible multiple-replica provable data possession in cloud storage. In: Proceedings of the IEEE Information Security and Countermeasures Contest, Heraklion, 2017. 1115–1121
44 Barsoum A F, Hasan M A. On verifying dynamic multiple data copies over cloud servers. 2011. https://eprint.iacr.org/2011/447.pdf
45 Liu C, Ranjan R, Yang C, et al. MuR-DPA: top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud. IEEE Trans Comput, 2015, 64: 2609–2622
46 Zhang Y, Ni J, Tao X, et al. Provable multiple replication data possession with full dynamics for secure cloud storage. Concurr Comput-Pract Exper, 2016, 28: 1161–1173
47 Guo W, Qin S, Gao F, et al. Dynamic proof of data possession and replication with tree sharing and batch verification in the cloud. IEEE Trans Serv Comput, 2022, 15: 1813–1824
48 Yu H, Yang Z, Waqas M, et al. Efficient dynamic multi-replica auditing for the cloud with geographic location. Future Gener Comput Syst, 2021, 125: 285–298
49 Zhu Y, Hu H, Ahn G J, et al. Cooperative provable data possession for integrity verification in multicloud storage. IEEE Trans Parallel Distrib Syst, 2012, 23: 2231–2244
50 Barsoum A F, Hasan M A. Provable multicopy dynamic data possession in cloud computing systems. IEEE Trans Inform Forensic Secur, 2015, 10: 485–497
51 Wang H. Identity-based distributed provable data possession in multicloud storage. IEEE Trans Serv Comput, 2015, 8: 328–340
52 Peng S, Zhou F, Li J, et al. Efficient, dynamic and identity-based remote data integrity checking for multiple replicas. J Netw Comput Appl, 2019, 134: 72–88
53 Li J, Yan H, Zhang Y. Efficient identity-based provable multi-copy data possession in multi-cloud storage. IEEE Trans Cloud Comput, 2022, 10: 356–365
54 Chang J, Shao B, Ji Y, et al. Efficient identity-based provable multi-copy data possession in multi-cloud storage, revisited. IEEE Commun Lett, 2020, 24: 2723–2727
55 Papamanthou C, Shi E, Tamassia R, et al. Streaming authenticated data structures. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2013. 9–22
56 Yu C M. POSTER: lightweight streaming authenticated data structures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, 2015. 1693–1695
57 Xu J, Wei L, Wu W, et al. Privacy-preserving data integrity verification by using lightweight streaming authenticated data structures for healthcare cyber-physical system. Future Gener Comput Syst, 2020, 108: 1287–1296
58 Xu J, Wei L, Zhang Y, et al. Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures. J Netw Comput Appl, 2018, 107: 113–124
59 Schroeder D, Schroeder H. Verifiable data streaming. In: Proceedings of the ACM Conference on Computer and Communications Security, Raleigh, 2012. 953–964
60 Backes M, Gerling R W, Gerling S, et al. WebTrust — a comprehensive authenticity and integrity framework for HTTP. In: Proceedings of International Conference on Applied Cryptography and Network Security, 2014. 401–418
61 Schoder D, Simkin M. VeriStream-a framework for verifiable data streaming. In: Financial Cryptography and Data Security. Berlin: Springer-Verlag, 2015. 548–566
62 Krupp J, Schroder D, Simkin M, et al. Nearly optimal verifiable data streaming. In: Public-Key Cryptography. Berlin: Springer-Verlag, 2016. 417–445
63 Wu Q, Zhou F, Xu J, et al. Secure data stream outsourcing with publicly verifiable integrity in cloud storage. J Inf Security Appl, 2019, 49: 102392
64 Li R, Liu A X, Wang A L, et al. Fast range query processing with strong privacy protection for cloud computing. Proc VLDB Endow, 2014, 7: 1953–1964
65 Tsai I C, Yu C M, Yokota H, et al. VENUS: verifiable range query in data streaming. In: Proceedings of IEEE Conference on Computer Communications Workshops, Honolulu, 2018. 160–165

66 Sun Y, Chen X Y, Du X H, et al. Dynamic authenticated method for outsourcing data stream with access control in cloud. Chinese J Comput, 2017, 40: 337–350 [孙奕, 陈性元, 杜学绘, 等. 一种具有访问控制的云平台下外包数据流动态可验证方法. 计算机学报, 2017, 40: 337–350]
67 Sun Y, Liu Q, Chen X, et al. An adaptive authenticated data structure with privacy-preserving for big data stream in cloud. IEEE Trans Inform Forensic Secur, 2020, 15: 3295–3310
68 Xu J, Meng Q, Wu J, et al. Efficient and lightweight data streaming authentication in industrial control and automation systems. IEEE Trans Ind Inf, 2021, 17: 4279–4287
69 Miao M, Wei J, Wu J, et al. Verifiable data streaming with efficient update for intelligent automation systems. Int J Intell Sys, 2022, 37: 1322–1338
70 Miao M, Li J, Wang Y, et al. Verifiable data streaming protocol supporting update history queries. Int J Intell Sys, 2022, 37: 11342–11361
71 Zhang Z, Chen X, Ma J, et al. New efficient constructions of verifiable data streaming with accountability. Ann Telecommun, 2019, 74: 483–499
72 Wei J H, Tian G H, Shen J, et al. Optimal verifiable data streaming protocol with data auditing. In: Proceedings of the 26th European Symposium on Research in Computer Security, 2021. 296–312
73 Gennaro R, Gentry C, Parno B. Non-interactive verifiable computing: outsourcing computation to untrusted workers. In: Proceedings of Annual Cryptology Conference, 2010. 465–482
74 Chen X F. Introduction to Secure Outsourcing Computation. Berlin: Springer, 2016
75 Yao A C. Protocols for secure computations. In: Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science, 1982. 160–164
76 Yao A C. How to generate and exchange secrets. In: Proceedings of the 27th Annual Symposium on Foundations of Computer ScienceInt, Toronto, 1986. 162–167
77 Parno B, Raykova M, Vaikuntanathan V. How to delegate and verify in public: verifiable computation from attribute-based encryption. In: Proceedings of Theory of Cryptography Conference, 2012. 422–439
78 Choi S G, Katz J, Kumaresan R, et al. Multi-client non-interactive verifiable computation. In: Proceedings of Theory of Cryptography Conference, 2013. 499–518
79 Gordon S D, Katz J, Liu F-H, et al. Multi-client verifiable computation with stronger security guarantees. In: Proceedings of Theory of Cryptography Conference, 2015. 144–168
80 Li P, Xu H, Ji Y. Multi-client outsourced computation. In: Proceedings of the 10th International Conference on Information Security and Cryptology, 2016. 397–409
81 Yang H, Sun J, Qin J, et al. An improved scheme for outsourced computation with attribute-based encryption. Concurr Comput Pract Exper, 2019, 31: 4833
82 Sun J, Zhu B, Qin J, et al. Confidentiality-preserving publicly verifiable computation. Int J Found Comput Sci, 2017, 28: 799–818
83 Yang H, Su Y, Qin J, et al. Verifiable inner product computation on outsourced database for authenticated multi-user data sharing. Inf Sci, 2020, 539: 295–311
84 Blanton M, Atallah M J, Frikken K B, et al. Secure and efficient outsourcing of sequence comparisons. In: Proceedings of European Symposium on Research in Computer Security, 2012. 505–522
85 Atallah M J, Li J. Secure outsourcing of sequence comparisons. Int J Inf Secur, 2005, 4: 277–287
86 Backes M, Fiore D, Reischuk R M. Verifiable delegation of computation on outsourced data. In: Proceedings of the ACM SIGSAC Conference on Computer & Communications Security, 2013. 863–874
87 Fiore D, Gennaro R. Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: Proceedings of the ACM Conference on Computer and Communications Security, Raleigh, 2012. 501–512
88 Sun Y, Yu Y, Li X, et al. Batch verifiable computation with public verifiability for outsourcing polynomials and matrix computations. In: Proceedings of the 21st Australasian Conference on Information Security and Privacy, 2016. 293–309
89 Yu X, Yan Z, Zhang R. Verifiable outsourced computation over encrypted data. Inf Sci, 2019, 479: 372–385
90 Wang C, Ren K, Wang J, et al. Harnessing the cloud for securely outsourcing large-scale systems of linear equations. IEEE Trans Parallel Distrib Syst, 2013, 24: 1172–1181
91 Chen X F, Huang X Y, Li J, et al. New algorithms for secure outsourcing of large-scale systems of linear equations.

IEEE Trans Inform Forensic Secur, 2015, 10: 69–78
92 Elkhiyaoui K, Onen M, Azraoui M, et al. Efficient techniques for publicly verifiable delegation of computation. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Xi'an, 2016. 119–128
93 Zhang X, Jiang T, Li K C, et al. New publicly verifiable computation for batch matrix multiplication. Inf Sci, 2019, 479: 664–678
94 Sun J, Zhu B, Qin J, et al. Confidentiality-preserving publicly verifiable computation schemes for polynomial evaluation and matrix-vector multiplication. Secur Commun Netw, 2018, 2018: 1–15
95 Lei X Y, Liao X F, Huang T W, et al. Outsourcing large matrix inversion computation to a public cloud. IEEE Trans Cloud Comput, 2013, 1: 78–87
96 Hu C, Alhothaily A, Alrawais A, et al. A secure and verifiable outsourcing scheme for matrix inverse computation. In: Proceedings of the 3rd IEEE International Conference on Computer Communications, Atlanta, 2017. 1–9
97 Pan S R, Wang Q X, Zheng F Y, et al. Secure and efficient outsourcing of large-scale matrix inverse computation. In: Proceedings of International Conference on Wireless Algorithms, Systems, and Applications, 2018. 374–386
98 Chen Z, Fu A, Xiao K, et al. Secure and verifiable outsourcing of large-scale matrix inversion without precondition in cloud computing. In: Proceedings of the IEEE International Conference on Communications, Kansas City, 2018. 1–6
99 Hohenberger S, Lysyanskaya A. How to securely outsource cryptographic computations. In: Proceedings of Theory of Cryptography Conference, 2005. 264–282
100 Wang Y J, Wu Q H, Wong D S, et al. Securely outsourcing exponentiations with single untrusted program for cloud storage. In: Proceedings of European Symposium on Research in Computer Security, 2014. 326–343
101 Chen X, Li J, Ma J, et al. New algorithms for secure outsourcing of modular exponentiations. In: Proceedings of IEEE Transactions on Parallel and Distributed Systems, 2014. 2386–2396
102 Ding Y, Xu Z, Ye J, et al. Secure outsourcing of modular exponentiations under single untrusted programme model. J Comput Syst Sci, 2017, 90: 1–13
103 Su Q, Zhang R, Xue R. Secure outsourcing algorithms for composite modular exponentiation based on single untrusted cloud. Comput J, 2020, 63: 1271
104 Ren Y, Dong M, Qian Z, et al. Efficient algorithm for secure outsourcing of modular exponentiation with single server. IEEE Trans Cloud Comput, 2021, 9: 145–154
105 Qi X, Zheng Y, Tian C. A note on two outsourcing algorithms of modular exponentiations. Int J Netw Secur, 2022, 24: 797–801
106 Chevallier-Mames B, Coron J-S, McCullagh N, et al. Secure delegation of elliptic-curve pairing. In: Proceedings of International Conference on Smart Card Research and Advanced Applications, 2010. 24–35
107 Chen X, Susilo W, Li J, et al. Efficient algorithms for secure outsourcing of bilinear pairings. Theor Comput Sci, 2015, 562: 112–121
108 Dong M, Ren Y. Efficient and secure outsourcing of bilinear pairings with single server. Sci China Inf Sci, 2018, 61: 039104
109 Tong L, Yu J, Zhang H. Secure outsourcing algorithm for bilinear pairings without pre-computation. In: Proceedings of IEEE Conference on Dependable and Secure Computing, Hangzhou, 2019. 1–7
110 Xue R, Wu Y, Liu M H, et al. Progress in verifiable computation. Sci Sin Inform, 2015, 45: 1370–1388 [薛锐, 吴迎, 刘牧华, 等. 可验证计算研究进展. 中国科学: 信息科学, 2015, 45: 1370–1388]
111 Devanbu P, Gertz M, Martel C, et al. Authentic data publication over the Internet. J Comput Secur, 2003, 11: 291–314
112 Pang H H, Tan K-L. Authenticating query results in edge computing. In: Proceedings of the 20th International Conference on Data Engineering, Boston, 2004. 560–571
113 Nuckolls G. Verified query results from hybrid authentication trees. In: Proceedings of IFIP Annual Conference on Data and Applications Security and Privacy, 2005. 84–98
114 Benaloh J, Mare M D. One-way accumulators: a decentralized alternative to digital signatures (extended abstract). In: Proceedings of Workshop on the Theory and Application of of Cryptographic Techniques, 1994. 274–285
115 Li F, Hadjieleftheriou M, Kollios G, et al. Dynamic authenticated index structures for outsourced databases.

In: Proceedings of the ACM SIGMOD International Conference on Management of data, Chicago, 2006. 121–132
116 Yang Y, Papadias D, Papadopoulos S, et al. Authenticated join processing in outsourced databases. In: Proceedings of the ACM SIGMOD International Conference on Management of data, Providence, 2009. 5–18
117 Li F, Hadjieleftheriou M, Kollios G, et al. Authenticated index structures for aggregation queries. ACM Trans Inf Syst Secur, 2010, 13: 1–35
118 Mykletun E, Narasimha M, Tsudik G. Authentication and integrity in outsourced databases. ACM Trans Storage, 2006, 2: 107–138
119 Pang H H, Zhang J, Mouratidis K. Scalable verification for outsourced dynamic databases. Proc VLDB Endow, 2009, 2: 802–813
120 Wang J, Chen X, Huang X, et al. Verifiable auditing for outsourced database in cloud computing. IEEE Trans Comput, 2015, 64: 3293–3303
121 Wang J, Chen X, Li J, et al. Towards achieving flexible and verifiable search for outsourced database in cloud computing. Future Gener Comput Syst, 2016, 67: 266–275
122 Papamanthou C, Tamassia R, Triandopoulos N. Optimal verification of operations on dynamic sets. In: Proceedings of Annual Cryptology Conference, 2011. 91–110
123 Canetti R, Paneth O, Papadopoulos D, et al. Verifiable set operations over outsourced databases. In: Proceedings of International Workshop on Public Key Cryptography, 2014. 113–130
124 Papadopoulos D, Papadopoulos S, Triandopoulos N. Taking authenticated range queries to arbitrary dimensions. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, 2014. 819–830
125 Zhang Y, Katz J, Papamanthou C. IntegriDB: verifiable SQL for outsourced databases. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, 2015. 1480–1491
126 Wu S, Li Q, Li G, et al. ServeDB: secure, verifiable, and efficient range queries on outsourced database. In: Proceedings of the 35th IEEE International Conference on Data Engineering, Macao, 2019. 626–637
127 Xie M, Wang H, Yin J, et al. Integrity auditing of outsourced data. In: Proceedings of the 33rd International Conference on Very Large Data bases, Washington, 2007. 782–793
128 Bitansky N, Canetti R, Chiesa A, et al. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, 2012. 326–349
129 Gennaro R, Gentry C, Parno B, et al. Quadratic span programs and succinct NIZKs without PCPs. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2013. 626–645
130 Ben-Sasson E, Chiesa A, Genkin D, et al. SNARKs for C: verifying program executions succinctly and in zero knowledge. In: Proceedings of Annual Cryptology Conference, 2013. 90–108
131 Costello C, Fournet C, Howell J, et al. Geppetto: versatile verifiable computation. In: Proceedings of the IEEE Symposium on Security and Privacy, San Jose, 2015. 253–270
132 Parno B, Howell J, Gentry C, et al. Pinocchio: nearly practical verifiable computation. Commun ACM, 2016, 59: 103–112
133 Setty S, Braun B, Vu V, et al. Resolving the conflict between generality and plausibility in verified computation. In: Proceedings of the 8th ACM European Conference on Computer Systems, Prague, 2013. 71–84
134 Wahby R S, Setty S, Ren Z, et al. Efficient RAM and control flow in verifiable outsourced computation. In: Proceedings of Network and Distributed System Security Symposium, San Diego, 2015. 8–25
135 Ben-Sasson E, Chiesa A, Tromer E, et al. Succinct non-interactive zero knowledge for a von Neumann architecture. In: Proceedings of the 23rd USENIX Security Symposium, San Diego, 2014. 781–796
136 Zhang Y, Genkin D, Katz J, et al. vSQL: verifying arbitrary SQL queries over dynamic outsourced databases. In: Proceedings of the IEEE Symposium on Security and Privacy, San Jose, 2017. 863–880
137 Cormode G, Mitzenmacher M, Thaler J. Practical verified computation with streaming interactive proofs. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, 2012. 90–112
138 Li F, Yi K, Hadjieleftheriou M, et al. Proof-infused streams: enabling authentication of sliding window queries on streams. In: Proceedings of the 33rd international conference on Very large data bases, 2007. 147–158
139 Papadopoulos S, Yang Y, Papadias D. CADS: continuous authentication on data streams. In: Proceedings of the

33rd International Conference on Very Large Data Bases, 2007. 23–35
140 Yi K, Li F, Cormode G, et al. Small synopses for group-by query verification on outsourced data streams. ACM Trans Database Syst, 2009, 34: 1–42
141 Nath S, Venkatesan R. Publicly verifiable grouped aggregation queries on outsourced data streams. In: Proceedings of the 29th IEEE International Conference on Data Engineering, Brisbane, 2013. 517–528
142 Papadopoulos S, Cormode G, Deligiannakis A, et al. Lightweight authentication of linear algebraic queries on data streams. In: Proceedings of the Special Interest Group on Management of Data, New York, 2013. 881–892
143 Chor B, Kushilevitz E, Goldreich O, et al. Private information retrieval. J ACM, 1998, 45: 965–981
144 Zhao L, Wang X, Huang X. Verifiable single-server private information retrieval from LWE with binary errors. Inf Sci, 2021, 546: 897–923
145 Zhang L F, Safavi-Naini R. Verifiable multi-server private information retrieval. In: Proceedings of International Conference on Applied Cryptography and Network Security, 2014. 62–79
146 Ke P, Zhang L F. Two-server private information retrieval with result verification. In: Proceedings of the International Symposium on Information Theory, Espoo, 2022. 408–413
147 Cheng W W, Pang H H, Tan K L. Authenticating multi-dimensional query results in data publishing. In: Proceedings of IFIP Annual Conference on Data and Applications Security and Privacy, 2006. 60–73
148 Berg M, Cheong O, Kreveld M, et al. Computational Geometry: Algorithms and Applications. Berlin: Springer, 2000
149 Yang Y, Papadopoulos S, Papadias D, et al. Spatial outsourcing for location-based services. In: Proceedings of the IEEE International Conference on Data Engineering, 2008. 1082–1091
150 Yang Y, Papadopoulos S, Papadias D, et al. Authenticated indexing for outsourced spatial databases. VLDB J, 2009, 18: 631–648
151 Agarwal P K, Berg D, Gudmundsson J, et al. Box-trees and R-trees with near-optimal query time. Discrete Comput Geom, 2002, 28: 291–312
152 Mouratidis K, Sacharidis D, Pang H H. Partially materialized digest scheme: an efficient verification method for outsourced databases. VLDB J, 2009, 18: 363–381
153 Sharifzadeh M, Shahabi C. VoR-tree: R-trees with Voronoi diagrams for efficient processing of spatial nearest neighbor queries. Proc VLDB Endow, 2010, 3: 1231–1242
154 Hu L, Ku W S, Bakiras S, et al. Spatial query integrity with voronoi neighbors. IEEE Trans Knowl Data Eng, 2013, 25: 863–876
155 Zhang C, Xu C, Xu J, et al. Distributed KNN query authentication. In: Proceedings of IEEE International Conference on Mobile Data Management'19, Aalborg, 2018. 167–176
156 Tao Y, Papadias D, Shen Q. Continuous nearest neighbor search. In: Proceedings of the 28th International Conference Very Large DataBase, Elsevier, 2002. 287–298
157 Iwerks G S, Samet H, Smith K P. Maintenance of k-NN and spatial join queries on continuously moving points. Javnost-Public, 2006, 13: 485–536
158 Song Z, Roussopoulos N. K-nearest neighbor search for moving query point. In: Proceedings of International Symposium on Spatial and Temporal Databases, 2001. 12–15
159 Nutanong S, Zhang R, Tanin E, et al. The V*Diagram: a query-dependent approach to moving KNN queries. Proc VLDB Endow, 2008, 1: 1095–1106
160 Yung D, Lo E, Yiu M L. Authentication of moving range queries. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, 2012. 1372–1381
161 Papadopoulos S, Yang Y, Bakiras S, et al. Continuous spatial authentication. In: Proceedings of International Symposium on Spatial and Temporal Databases, 2009. 62–79
162 Hu H, Xu J, Chen Q, et al. Authenticating location-based services without compromising location privacy. In: Proceedings of the Special Interest Group on Management of Data, Scottsdale, 2012. 301–312
163 Ku W S, Hu L, Shahabi C, et al. A query integrity assurance scheme for accessing outsourced spatial databases. GeoInformatica, 2013, 17: 97–124
164 Jang M, Yoon M, Chang J-W. A privacy-aware query authentication index for database outsourcing. In: Proceedings of the IEEE International Conference on Big Data and Smart Computing, Bangkok, 2014. 72–76

165 Wang B, Li M, Wang H. Geometric range search on encrypted spatial data. IEEE Trans Inform Forensic Secur, 2015, 11: 704–719

166 Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In: Proceedings of IEEE Symposium on Security and Privacy, Berkeley, 2000. 44–55

167 Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: improved definitions and efficient constructions. J Comput Secur, 2006, 19: 895–934

168 Boneh D, Crescenzo G D, Ostrovsky R, et al. Public key encryption with keyword Search. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, 2004. 506–522

169 Chai Q, Gong G. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. In: Proceedings of IEEE International Conference on Communications, Ottawa, 2012. 917–922

170 Wang J, Ma H, Tang Q, et al. Efficient verifiable fuzzy keyword search over encrypted data in cloud computing. ComSIS, 2013, 10: 667–684

171 Sun W, Wang B, Cao N, et al. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In: Proceedings of the 8th ACM ASIA Conference on Computer and Communications Security, Hangzhou, 2013. 71–82

172 Zhu J, Li Q, Wang C, et al. Enabling generic, verifiable, and secure data search in cloud services. IEEE Trans Parallel Distrib Syst, 2018, 29: 1721–1735

173 Liu X, Yang G, Mu Y, et al. Multi-user verifiable searchable symmetric encryption for cloud storage. IEEE Trans Dependable Secure Comput, 2020, 17: 1322–1332

174 Miao M, Wang Y, Wang J, et al. Verifiable database supporting keyword searches with forward security. Comput Standards Interfaces, 2021, 77: 103491

175 Zheng Q, Xu S, Ateniese G. VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In: Proceedings of the IEEE International Conference on Computer Communications, Toronto, 2014. 522–530

176 Liu P, Wang J, Ma H, et al. Efficient verifiable public key encryption with keyword search based on KP-ABE. In: Proceedings of the 9th International Conference on Broadband and Wireless Computing, Communication and Applications, 2014. 584–589

177 Alderman J, Janson C, Martin K M, et al. Extended functionality in verifiable searchable encryption. In: Proceedings of International Conference on Cryptography and Information Security in the Balkans, 2016. 187–205

178 Zhu B, Sun J, Qin J, et al. The public verifiability of public key encryption with keyword search. In: Proceedings of 9th International Conference on Mobile Networks and Management, 2018. 299–312

179 Liu Z, Li T, Li P, et al. Verifiable searchable encryption with aggregate keys for data sharing system. Future Gener Comput Syst, 2018, 78: 778–788

180 Liang Y, Li Y, Cao Q, et al. VPAMS: verifiable and practical attribute-based multi-keyword search over encrypted cloud data. J Syst Archit, 2020, 108: 101741

181 Yousefipoor V, Eghlidos T. An efficient, secure and verifiable conjunctive keyword search scheme based on rank metric codes over encrypted outsourced cloud data. Comput Electrical Eng, 2023, 105: 108523

182 Bajaj S, Sion R. CorrectDB: SQL engine with practical query authentication. Proc VLDB Endow, 2013, 6: 529–540

183 Arasu A, Eguro K, Kaushik R, et al. Concerto: a high concurrency key-value store with integrity. In: Proceedings of the 17th Special Interest Group on Management of Data, Chicago, 2017. 251–266

184 Bajaj S, Chakraborti A, Sion R. ConcurDB: concurrent query authentication for outsourced databases. IEEE Trans Knowl Data Eng, 2021, 33: 1401–1412

185 Wang Y, Wei J. VIAF: verification-based integrity assurance framework for mapReduce. In: Proceedings of the 33rd IEEE International Conference on Cloud Computing, Washington DC, 2011. 300–307

186 Ulusoy H, Kantarcioglu M, Pattuk E. TrustMR: computation integrity assurance system for MapReduce. In: Proceedings of IEEE International Conference on BigData, Santa Clara, 2015. 441–450

187 Ghosh E, Ohrimenko O, Tamassia R. Efficient verifiable range and closest point queries in zero-knowledge. In: Proceedings on Privacy Enhancing Technologies, 2016. 373–388

188 Tong W, Jiang B, Xu F, et al. Privacy-preserving data integrity verification in mobile edge computing. In: Proceedings of the 39th IEEE International Conference on Distributed Computing Systems, Dallas, 2019. 1007–1018

189 Li B, He Q, Chen F, et al. Auditing cache data integrity in the edge computing environment. IEEE Trans Parallel Distrib Syst, 2021, 32: 1210–1223
190 Li B, He Q, Chen F, et al. Inspecting edge data integrity with aggregate signature in distributed edge computing environment. IEEE Trans Cloud Comput, 2021, 10: 2691–2703
191 Li B, He Q, Chen F, et al. Cooperative assurance of cache data integrity for mobile edge computing. IEEE Trans Inform Forensic Secur, 2021, 16: 4648–4662
192 Xu C, Zhang C, Xu J. vChain: enabling verifiable Boolean range queries over blockchain databases. In: Proceedings of Special Interest Group on Management of Data, Amsterdam, 2019. 141–158
193 Wang H, Xu C, Zhang C, et al. vChain+: optimizing verifiable blockchain Boolean range queries. In: Proceedings of the 38th IEEE International Conference on Data Engineering, Kuala Lumpur, 2022. 1927–1940
194 Ghodsi Z, Gu T, Garg S. SafetyNets: verifiable execution of deep neural networks on an untrusted cloud. In: Proceedings of Conference and Workshop on Neural Information Processing Systems, 2017. 4672–4681
195 Zhao L, Wang Q, Wang C, et al. VeriML: enabling integrity assurances and fair payments for machine learning as a service. IEEE Trans Parallel Distrib Syst, 2021, 32: 2524–2540
196 Froelicher D, Troncoso-Pastoriza J R, Sousa J S, et al. Drynx: decentralized, secure, verifiable system for statistical queries and machine learning on distributed datasets. IEEE Trans Inform Forensic Secur, 2020, 15: 3035–3050
197 Xu G, Li H, Liu S, et al. VerifyNet: secure and verifiable federated learning. IEEE Trans Inform Forensic Secur, 2020, 15: 911–926
198 Guo X, Liu Z, Li J, et al. VeriFL: communication-efficient and fast verifiable aggregation for federated learning. IEEE Trans Inform Forensic Secur, 2021, 16: 1736–1751
199 Han G, Zhang T, Zhang Y, et al. Verifiable and privacy preserving federated learning without fully trusted centers. J Ambient Intell Hum Comput, 2022, 13: 1431–1441
200 Fu A, Zhang X, Xiong N, et al. VFL: a verifiable federated learning with privacy-preserving for big data in industrial IoT. IEEE Trans Ind Inf, 2022, 18: 3316–3326
201 Xu Y, Peng C, Tan W, et al. Non-interactive verifiable privacy-preserving federated learning. Future Generation Comput Syst, 2022, 128: 365–380

Research progress of verifiable technologies for outsourcing services

Yi SUN[1], Fan YANG[1], Xingyuan CHEN[1,2*], Xuehui DU[1] & Wei LIN[1]

1. *School of Cryptographic Engineering, Information Engineering University, Zhengzhou* 450001*, China*;
2. *State Key Laboratory of Cryptology, Beijing* 100094*, China*
* Corresponding author. E-mail: chxy302@vip.sina.com

Abstract With the advent of the digital economy, data has become an important factor in production. There is an ever-growing trend of outsourcing data to third parties for storage, management, analysis and computation, which causes increasingly prominent trust issues with outsourcing services such as outsourced data and computing. As an effective and promising technology for determining the trustworthiness of outsourcing services, verifiable technology is rapidly developed, rich in research, diverse and varied in technology. With the emergence of new outsourcing service requirements such as blockchain, artificial intelligence, and privacy computing, there come new challenges and opportunities. In this paper, the progress of verifiable technologies for outsourcing services is reviewed and summarized. First, we summarize the frameworks, classification methods, security objectives and evaluation system of verifiable technology for outsourcing services. Then we formally define the verifiable technology and discuss the typical verifiable schemes from the perspectives of outsourcing data and outsourcing computing, respectively. From the former perspective, this paper, based on the verifiable data types, together with the implemented functions and methods, discusses the outsourced data holdings and integrity verification technologies under different data types, scenarios and implementation methods; from the latter perspective, we summarize the characteristics and progress of five typical outsourced computing verifiable techniques, and conduct a detailed comparative analysis of typical solutions, based on the proposed evaluation system. Finally, in the context of emerging technologies and application hot issues, the developing trend and application prospect of verifiable technologies and challenges are viewed.

Keywords outsourcing services, outsourcing data verification, verifiable computation, authenticated data structures, dynamic verifiable structures