具有隐私保护的可验证计算研究进展

李世敏[1,2], 王 欣[3], 薛 锐[2,4]

[1] 中电科网络安全科技股份有限公司 摩石实验室 北京 中国 100043
[2] 中国科学院信息工程研究所 信息安全国家重点实验室 北京 中国 100085
[3] 蚂蚁云创数字科技(北京)有限公司 北京 中国 100020
[4] 中国科学院大学 网络空间安全学院 北京 中国 100049

摘要 随着信息产业的高速发展, 复杂的计算任务与用户有限的计算能力之间的矛盾愈加突出, 如何借助云平台提供的计算服务, 实现安全可靠的外包计算, 引起了人们的广泛关注。具有隐私保护的可验证计算为该问题提供了有效途径, 它能够解决外包计算中的两大安全问题——计算结果不可信和用户隐私数据泄露。根据客户端存储能力是否受限, 可验证计算可分为计算外包模式和数据外包模式, 本文分别对这两种模式下具有隐私保护的可验证计算进行梳理和总结。对于计算外包模式, 本文以方案涉及的服务器数量为分类依据, 分别梳理了单服务器情形和多服务器情形下的相关工作。其中, 对于单服务器情形下具有隐私保护的可验证计算, 提炼出了一般化的通用构造方法和针对具体函数的构造技术, 并对多服务器情形下的相关方案进行了分析对比。对于数据外包模式, 本文根据实现工具的不同, 分别梳理了基于同态认证加密的可验证计算和基于上下文隐藏的同态签名的可验证计算。具体地, 本文从函数类型、安全强度、困难假设、验证方式、证明规模等多个维度对现有的同态认证加密方案进行了分析对比; 此外, 本文还对同态签名不同的隐私性定义进行了总结对比, 包括单密钥情形下的弱上下文隐藏性、强上下文隐藏性、完全上下文隐藏性和基于模拟的上下文隐藏性, 以及多密钥情形下的内部上下文隐藏性和外部上下文隐藏性。最后, 通过分析现有方案在性能、功能和安全性三个方面的优势及不足, 对具有隐私保护的可验证计算未来的研究重点进行了讨论与展望。

关键词 云计算; 可验证计算; 数据隐私; 计算外包模式; 数据外包模式; 隐私保护的同态消息认证码; 上下文隐藏的同态签名
中图法分类号 TP309.2 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2024.07.12

A Survey on Verifiable Computation with Privacy Protection

LI Shimin[1,2], WANG Xin[3], XUE Rui[2,4]

[1] Westone Cryptologic Research Center, CETC Cyberspace Security Technology Co., Ltd., Beijing 100043, China
[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China
[3] Ant Group, Beijing 100020, China
[4] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract With the rapid development of the information industry, the contradiction between complex computing tasks and users' limited computing ability is becoming more and more prominent. How to achieve secure and reliable outsourcing computation by using computing services provided by cloud platforms has attracted widespread attention. Verifiable computation (VC) with privacy protection provides an effective approach to this problem, which can solve two major security problems in outsourcing computation: the unreliability of computing results and the leakage of user's privacy data. Verifiable computation can be divided into computation-outsourcing mode and data-outsourcing mode according to whether the storage capacity of the client is limited. This paper sorts out and summarizes the important research progress of privacy-preserving verifiable computation in the above two modes, respectively. For the computation-outsourcing mode, this paper sorts out the relevant work under the single-server situation and the multi-server situation based on the number of servers involved in the scheme. Among them, for VC with privacy protection in the case of single server, the generic construction methods and the construction techniques for specific functions are summarized, thereafter the relevant schemes in the case of multiple servers are analyzed and compared. For the data-outsourcing mode, according to different implementation tools this paper investigates VC based on homomorphic authenticated encryption and that based on context hiding homomorphic signature, respectively. Specifically, this paper analyzes and compares the existing homomorphic authenticated encryption schemes in several aspects such as function types, security levels, assumptions, verification mode and proof size. In addition, this paper also summarizes and compares different privacy

通讯作者: 薛锐, 博士, 研究员, Email: xuerui@iie.ac.cn。
本课题得到国家自然科学基金(No. 62172405), 信息安全国家重点实验室开放课题基金(No. 2021-MS-02)资助。
收稿日期: 2022-10-17; 修改日期: 2023-01-12; 定稿日期: 2024-04-08

definitions of homomorphic signature, including weakly context hiding, strong context hiding, completely context hiding and simulation-based context hiding in the case of single-key, and internally context hiding and externally context hiding in the case of multi-key. Finally, by analyzing the advantages and disadvantages of existing schemes in terms of performance, functionality and security, the future research emphasis of VC with privacy protection is discussed and prospected.

Key words cloud computing; verifiable computation; data privacy; computing-outsourcing mode; data-outsourcing mode; privacy-preserving homomorphic message authenticator; context hiding homomorphic signature

1 引言

随着社会信息化进程的不断加快, 越来越多的数据涌入到人们的日常生活中, 比如网络社交数据、电子医疗数据、物联网终端数据等。与此同时, 为了携带方便、操作简易, 终端设备(比如各种移动通讯设备、终端传感器、穿戴设备等)的设计越来越轻量化, 其存储资源和计算能力愈加有限, 无法胜任大量数据的存储和复杂任务的计算。面对这种数据处理需求与资源能力无法匹配的情形, 亟需探寻一种有效的手段, 使人们可以在能力范围内解决数据的存储和计算问题。

云计算的出现为解决上述问题提供了新思路[1-2]。在云计算模式下, 存储空间和计算资源作为一种服务而存在。用户可以根据自身的需求, 向云服务提供商(Cloud service provider, CSP)弹性地购买这些服务, 从而可以在不受时间和空间限制的情况下, 享受云端近乎无限的资源。计算能力较弱的用户将复杂的计算任务委托给 CSP 完成的过程, 称为外包计算。这种计算方式既能有效地整合分布式网络资源, 使其得到合理充分的利用, 还最大程度地降低了用户的成本花销, 因此特别适合个人用户和中小企业用户的发展需求。正是看中了云计算广阔的发展前景, 越来越多的信息技术企业开始向云计算转型, 市面上出现了各式各样的云平台, 例如国内的阿里云[3]、华为云[4]、百度智能云[5]、腾讯云[6]等, 以及国外的 Microsoft Azure[7]、Google Cloud Platform[8]、Amazon Web Services[9]、IBM Cloud[10]等。

然而, 任何新技术的应用都伴随着安全隐患, 云计算技术也不例外。在实际的外包计算场景中, CSP 可能会为了经济利益而“偷懒”, 最终返回一个未经计算的错误结果。或者出现一些客观因素, 比如软件故障、磁盘损坏、系统遭受恶意攻击等, 致使计算结果出错。然而, 无论出现哪种情况, 用户都将按照既定协议付费, 这极大地侵害了他们的权益。因此, 计算可靠性是外包计算安全研究的关键问题之一[11]。

作为一种能够对计算结果进行正确性验证的工具, 可验证计算(Verifiable computation, VC)就是一种能够保障外包用户正当权益的有效手段。在 VC 模型中, 客户端将复杂的计算任务委托给服务器执行, 服务器完成计算后, 返回计算结果及其相应的证明。验证者 (客户端自身或者第三方) 通过对该证明进行验证, 便能判断计算结果是否正确。一个可验证计算方案需要具备三个基本条件:

(1) 正确性: 服务器诚实执行方案得到的正确的计算结果一定能够通过验证;

(2) 可靠性: 客户端不会接受服务器返回的错误的计算结果;

(3) 高效性(或可外包性): 客户端的验证时间要远小于直接计算函数本身所需的时间, 否则计算便失去了外包的意义。

早在分布式计算的研究中, 针对各计算实体是否返回正确结果的研究就已开展, 这也引出了可验证计算。迄今为止, 可验证计算的发展已有三十多年的历程。根据技术手段以及功能侧重点的不同, 相关工作可划分为三大领域: 应用安全领域、计算机理论领域和密码学领域。

在应用安全领域, 通常采用重复计算[12-13]或者审计的方法[14-15]来保证实体返回正确的计算结果。重复计算是指将同一个任务交给多个实体计算, 通过对结果的比对保留正确的计算结果; 审计方法则要求客户采取小样本重算的方式进行结果验证。此外, 还有一种手段是使用安全硬件, 比如采用安全的协处理器[16-17], 提供一个安全的隔离环境, 从而避免敌手改变协处理器内部的工作状态, 最终保证数据的隐私性和完整性。

在计算机理论领域, 可验证计算大多都是利用交互式证明系统[18-19]实现, 比如“Muggle Proof”模型[20]及其改进版本[21]。此外, 还有基于概率可检验证明(Probabilistically checkable proofs, PCP)的可验证计算, 比如 PCP 模型[22-23], 以及基于二次张成程序的可验证计算, 比如简洁的非交互式论证系统(Succinct non-interactive argument, SNARG)[24]和实例化系统 Pinocchio[25]、SNARK-for-C[26]等。

Chaum 和 Pedersen[27] 于 1992 年提出“电子钱包”模型, 并利用密码学工具——盲签名构造了具体

协议, 这是密码学领域可验证计算研究的开端。本文仅仅关注密码学领域的可验证计算, 对于应用安全领域及计算机理论领域的相关成果, 不作具体介绍。关于计算机理论领域可验证计算的研究进展, 可参见综述文献[28]中的详细介绍。

在密码学领域, 根据客户端存储能力是否受限, 可验证计算可分为两种不同的模式: 计算外包模式(如图 1 所示)和数据外包模式(如图 2 所示)。前者描述的是客户端仅将计算委托给云服务器, 每次计算前客户端自己产生计算输入; 后者则描述客户端存储能力有限, 因而将大量的数据先外包存储在云端, 之后客户端可以委托服务器对这些外包数据进行相应的计算。具体地, 在计算外包模式中, 假设客户端想要委托服务器计算函数值 $F(x)$, 他首先将函数 $F(\cdot)$和输入 x 外包给服务器, 服务器随后返回计算结果 y 以及对应的证明 $proof_F^x(y)$, 客户端对该证明进行验证, 以判断 y 是否等于 $F(x)$。在外包计算模式中, 由于客户端存储能力有限, 他首先将大量的本地数据 $(x_1,\tau_1,\sigma_1),...,(x_N,\tau_N,\sigma_N)$外包存储服务器端, 其中$l_i$为标识数据$x_i$的标签, σ_i为x_i的认证值, $i \in [1,N]$; 之后, 客户端发送带标签的程序 $P:=(f,\tau_1,...,\tau_k)$, 请求服务器对标签为$\tau_1,...,\tau_k$的数据$x_1,...,x_k$执行函数$f(\cdot,...,\cdot)$运算; 服务器最终返回计算结果 y, 以及对应的认证值$\sigma_{y,f}$; 客户端对该认证值进行验证, 判断 y 是否等于 $f(x_1,...,x_k)$。需要注意的是, 外包计算模式的可验证计算通常由同态消息认证码(Homomorphic message authenticator, 同态 MAC)[84]和同态签名[128]两种密码学工具实现, 因此图示中的认证值通常对应标记(tag)或签名。

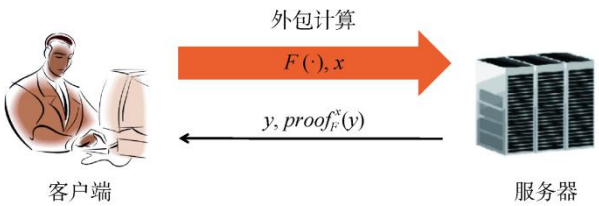

图 1 计算外包模式的可验证计算

Figure 1 Verifiable computation under computing-outsourcing mode

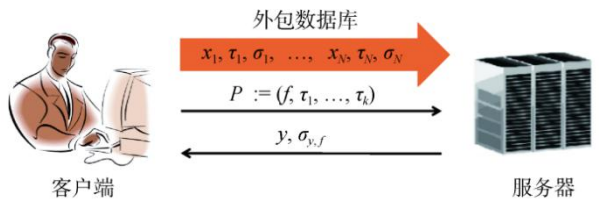

图 2 数据外包模式的可验证计算

Figure 2 Verifiable computation under data-outsourcing mode

无论采用哪种模式, 用户都需要将自身数据作为计算输入发送给云服务器, 尤其是对于数据外包模式的可验证计算, 由于存储能力有限, 用户将数据存储在云端后, 通常不会再在本地保留备份。这些数据可能涉及个人的隐私信息, 比如身份信息、健康数据、收支记录等, 也可能包含企业的技术信息、经营数据等商业机密。然而, 云平台始终存在内部或外部的安全隐患, 比如技术故障、安全漏洞、黑客攻击, 甚至 CSP 违背协议规定恶意处理、非法操作等[29-31]。这些安全隐患都有可能威胁到用户的数据安全, 造成数据丢失或隐私泄露等后果。因此, 这也引出了外包计算中另一个重要的安全研究课题——用户数据隐私保护[32]。

为了兼顾计算可靠性和数据隐私性的双重需求, 一些密码学者考虑对可验证计算增加隐私保护的功能。近年来, 涌现出了许多具有隐私保护的可验证计算方案, 实现了用户计算输入相对于服务器的隐私性、计算结果相对于服务器的隐私性, 以及用户数据相对于第三方验证者的隐私性等。这些研究成果丰富了可验证计算的功能, 拓宽了外包服务的模式, 为实现数据安全、服务可靠的外包计算提供了有效的解决方案。

本文将围绕外包计算中计算结果不可信和用户隐私数据泄露两大安全问题, 对具有隐私保护功能的可验证计算方案进行梳理和总结。根据外包模式的不同, 第 2 节梳理计算外包模式下具有隐私保护的可验证计算方案, 第 3 节梳理数据外包模式下具有隐私保护的可验证计算方案。对于前者的梳理, 依据方案涉及的服务器数量进行分类总结; 对于后者的梳理, 则根据实现工具的不同分别介绍。第 4 节对后续研究的重点问题进行讨论与展望, 最后一节对本文进行简单总结。本文的研究路线如图 3 所示。

2 计算外包模式具有隐私保护的可验证计算

由于场景的具体需求不同, 可验证计算方案涉及的客户端和服务器数量也有所不同。在实际的研究工作中, 以单服务器单客户端情形居多, 还有一些工作专门探讨多客户端或多服务器的情形。为简便起见, 本节将分别对单服务器单客户端情形、单服务器多客户端情形和多服务器情形下具有隐私保护的可验证计算进行介绍。在单服务器单客户端情形中, 根据可验证计算方案适用的函数类别不同, 又细分为针对一般函数和针对具体函数两条路线讨论。

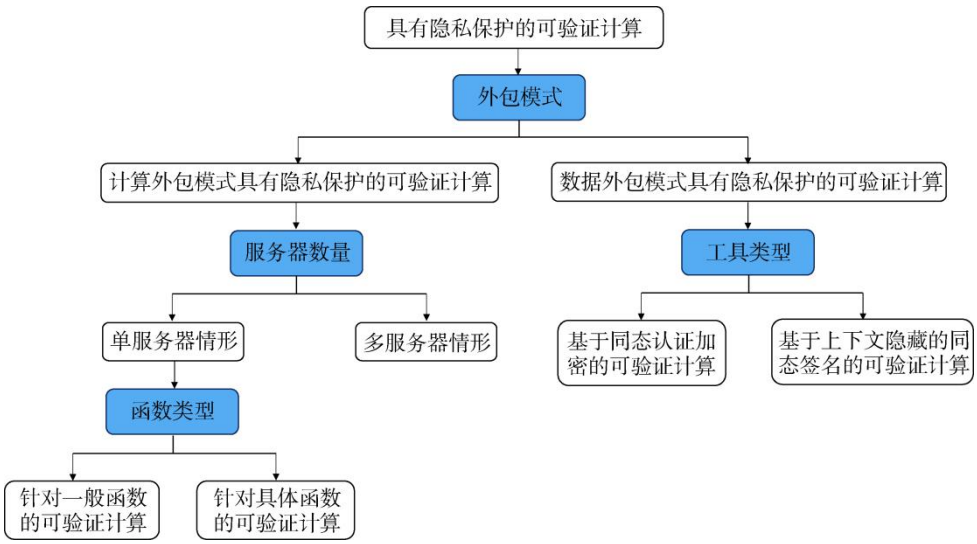

图 3　研究路线图

Figure 3　Overview roadmap

2.1　单服务器情形具有隐私保护的可验证计算

单服务器单客户端情形的可验证计算是指计算任务由单个服务器完成, 外包计算的输入数据也由单个客户端提供。

2.1.1　针对一般函数的可验证计算

一般函数是指布尔函数$f_{boolean}$:$\{0,1\}^* \to \{0,1\}$, 又称布尔电路。由于对于任意一个函数的计算都可以通过计算一系列具有逻辑结构的布尔电路实现, 因此一个针对布尔函数的可验证计算方案也适用于任意函数[33-34]。

2010 年, Gennaro 等人[35]首次形式化地定义了非交互式可验证计算,并且利用姚式混淆电路(Garbled circuits, GCs) 和全同态加密(Fully homomorphic encryption, FHE)构造了一个针对一般函数的可验证计算方案。需要说明的是, Gennaro 等人定义的是数据外包模式的可验证计算, 之后该模式下的可验证计算都是基于他们的模型和定义。根据文献[35]的定义, 一个可验证计算方案 $\mathcal{VC}$ 是一个多项式时间的双方协议, 由客户端和服务器共同执行, 以完成对某个函数值的计算。下面, 给出具体定义。

定义 1.可验证计算方案。一个可验证计算方案 **VC** 由以下四个概率多项式时间的算法组成:

- $KeyGen(F, \lambda) \to (PK, SK)$: 密钥生成算法, 基于安全参数$\lambda$, 对输入函数 F 进行编码, 得到的公钥 PK 用于服务器计算函数 F, 对应的私钥 SK 由客户端自己保存。该过程即为预处理操作。
- $ProbGen(SK, x) \to (\omega_x, \mu_x)$: 问题生成算法, 利用私钥 SK 对输入 x 进行编码, 得到的公开值ω_x和私有值μ_x, 前者发送给服务器, 后者客户端自己保存。
- $Compute(PK, \omega_x) \to \omega_y$: 计算算法, 服务器利用公钥 PK 和公开值ω_x, 计算得到函数的输出值 $y = F(x)$的编码ω_y。
- $Verify(SK, \mu_x, \omega_y) \to y \cup \perp$: 验证算法, 客户端利用私钥 SK 和私有值μ_x, 对ω_y进行解码, 输出计算结果 $y = F(x)$或者$\perp$, 其中$\perp$表示服务器计算出错, 客户端拒绝接受该结果。

一个可验证计算方案需同时满足正确性和安全性。正确性是指如果服务器诚实地执行了计算算法, 则输出结果一定能够通过验证。安全性则要求客户端接受一个错误的计算结果的概率可忽略。

此外, 文献[35]还形式化地定义了输入隐私性(Input privacy)和输出隐私性(Output privacy)。前者要求服务器不会获取到任何关于客户端输入数据的信息, 后者则要求真正的计算结果不会泄露给服务器。两个定义都是基于不可区分的论证方式给出的。下述实验刻画了输入隐私性, 输出隐私性的实验与其类似, 此处不作赘述。其中, 应答器 $PubProbGen(SK, x)$ 执行 $ProbGen(SK, x)$, 但仅回答公开值ω_x。

$\mathbf{Exp}_{\mathrm{A}}^{Priv}[\mathbf{VC}, F, \lambda]$:

$(PK, SK) \xleftarrow{R} KeyGen(F, \lambda)$;

$(x_0, x_1) \leftarrow \mathcal{A}^{PubProbGen(SK,\cdot)}(PK)$;

$(\omega_0, \mu_0) \leftarrow ProbGen(SK, x_0)$;

$(\omega_1, \mu_1) \leftarrow ProbGen(SK, x_1)$;

$b \xleftarrow{R} \{0, 1\}$;

$\hat{b} \leftarrow \mathcal{A}^{PubProbGen(SK,\cdot)}(PK, x_0, x_1, \omega_b)$;

若 $\hat{b}$ =b, 则输出 1; 否则, 输出 0。

定义 2 .输入隐私性。对于一个可验证计算方案

VC, 如果对任意多项式时间的敌手 $\mathcal{A}$, 下式成立

$$Adv_{\mathcal{A}}^{Priv}(\mathbf{VC}, F, \lambda)=|Prob[\mathbf{Exp}_{\mathcal{A}}^{Priv}[\mathbf{VC}, F, \lambda]=1] -1/2\leqslant negl(\lambda),$$

其中, $negl(\cdot)$是一个可忽略函数, 则称 **VC** 对于函数 F 满足输入隐私性。

在文献[35]中, Gennaro 等人首先利用姚式混淆电路, 构造了一个一次安全的 VC 方案。一次安全性要求对一个函数执行预处理操作后, 仅能执行一次关于该函数的计算。通过使用 FHE 对输入编码和输出编码加密, Gennaro 等人得到了一个多次安全的 VC 方案。需要说明的是, 两种安全性的方案都实现了输入隐私性和输出隐私性。前者的隐私性可以归约到混淆方案(Garbling scheme, GS)的可靠性(authenticity)①, 后者的隐私性则由 FHE 方案的安全性保证。

紧接着, Chung 等人[37]在不使用姚式混淆电路的情况下, 仅基于 FHE 设计了一个新的非交互式可验证计算方案, 其计算可靠性由通用论证系统(Universal argument)[38]保证。与文献[35]相比, 文献[37]极大地缩减了方案的公钥长度。对于 Gennaro 等人[35]的方案, 其"在线阶段"十分高效, 客户端仅花费 poly(logT)时间, 其中 T 为函数 F 的时间复杂度。但是, 该方案的预处理阶段需要花费客户端 poly(T)的时间对函数 F 进行编码, 最终得到长度为 poly(T)的公钥。对于 Chung 等人[37]的方案, 由于无需姚式混淆电路, 因此方案无公钥。此外, 在该方案中, 密钥生成算法被外包给服务器完成, 因此预处理操作仅需 poly(logT)时间。但是, 这是以增加客户端的外包成本为代价实现的。

从安全性角度分析, 文献[35]和文献[37]中的方案都面临"拒绝问题(Rejection problem)"。具体来说, 对于服务器返回的错误计算值, 客户端不能将相应的验证结果泄露给它, 否则方案将不再安全。因此, 方案安全性的一个隐含前提是敌手无法获知验证结果。通过提出代理同态加密(Delegatable homomorphic encryption, DHE)的密码学组件, Barbosa 和 Farshim[39]构造了第一个可以抵抗"拒绝问题"的可验证计算方案。DHE 可以看作 VC 公钥形式的一种推广, 其中客户端的功能被分解成了三部分: 可信的权威机构执行预处理操作, 发送方提供输入值, 接收方获取计算结果并进行验证。文献[39]利用函数加密、全同态加密和消息认证码三个组件, 构造了一个安全的 DHE 方案, 进而得到了一个针对一般函数的 VC 方案。但是, 该方案的实施需要一个额外的可信第三方以及安全的通信通道。

上述几个 VC 方案有一个共同点, 即都使用了 FHE 对输入值和输出结果进行加密保护, 因此方案自然地满足输入隐私性和输出隐私性。

2012 年, Parno 等人[40]首先考虑了可公开验证计算(Publicly verifiable computation, PVC), 并给出了形式化定义。根据文献[40]中的定义, 可公开验证计算要求满足公开代理性和公开验证性。前者表示函数 F 预处理后的信息可以公开发布, 任何想要外包计算该函数的客户端都可以使用这些信息; 后者则表示服务器返回的结果可以被所有人验证。

Parno 等人给出了一种将密钥策略(key-policy)的基于属性的加密(Attribute-based encryption, ABE)转换为 PVC 方案的通用方法。在密钥策略的 ABE 中, 只有当密钥中的访问结构 F(也即密钥对应的函数 F)与密文中的属性 x 相匹配时, 即 $F(x)=1$, 才能正确解密得到明文。ABE 的这种性质自然地提供了一种证明 $F(x)=1$ 方法, 这便建立起了 ABE 与 VC 之间的联系。具体地, 在 VC 方案中, 假设客户端想要委托服务器计算函数值 $F(x)$, 具体操作过程如下:

1)预处理阶段: 客户端首先生成 ABE 的主公私钥对(mpk,msk), 其中 mpk 即为 VC 方案的公钥, 然后利用 msk 生成函数 F 对应的密钥sk_F,作为 VC 方案的计算密钥;

2)问题生成阶段: 以 x 为属性, 对随机消息 m 进行加密, 然后将得到的密文和计算密钥sk_F发送给服务器;

3)计算阶段: 服务器执行 ABE 的解密算法, 将解密结果返回给客户端。

根据 ABE 的性质, 只有当 $F(x)=1$ 时, 服务器才能正确恢复出消息m; 如若$F(x)=0$, 则服务器正确解出消息 m 的概率可忽略。因此, 当服务器返回的值为 m 时, 可以确定 $F(x)=1$。但若服务器返回的值不等于 m, 还存在两种情况: 要么 $F(x)=0$, 服务器无法正确恢复m; 要么$F(x)=1$, 但服务器故意拒绝解密。为了杜绝后者的发生, 文献[40]考虑分别针对函数 F 和它的补函数$\overline{F}$, 重复运行上述协议两次。这样一来, 无论哪种结果, 总会伴随函数值为 0 的情形, 即 $F(x)=0$ 或者$\overline{F}(x)=0$, 以此便可约束服务器的欺骗行为。

ABE 的性质保证了方案的公开代理性和公开验证性, 但却限制其只能计算单比特输出的函数。如果想要实现多比特输出函数的外包计算, 则需要多次运行上述协议, 这将极大地增加委托成本。由于函数

① Bellare 等人[36]对姚式混淆电路进行概念抽象, 形式化地定义了混淆方案, 其中专门定义了可靠性。

的输入是作为属性嵌入密文, 因此若中间组件 ABE 满足属性隐藏(Attribute-hiding), 即密文不泄露属性的信息, 则 VC 方案自然满足输入隐私性; 输出隐私性则可以简单地通过随机置换密文和密钥对, 使得服务器无法判断正确解密对应的计算结果为 0 还是 1 来实现。

Parno 等人开启了一种以 ABE 作为中间组件构造 VC 方案的方法, 之后有很多工作基于该思想, 实现具有隐私保护的可验证计算[41-48]。

需要说明的是, 将 ABE 替换为更一般化的加密模式——函数加密(Functional encryption, FE)时, 文献[40]中的转换方法依然适用。利用 FE 构造得到的 VC 方案, 除了依然满足公开代理和公开验证性质外, 还天然地满足输入隐私性。这是由于 FE 的安全性保证了, 当服务器利用函数 F 对应的私钥sk_F对密文 c 进行计算时, 除了 $F(x)$外, 无法获取任何关于明文 x 的信息。因此, 遵循文献[40]的转化方法, 具有隐私保护的公开可验证计算成了 FE 的一个关键应用。在 FE 的研究工作中, 典型的针对一般函数的方案可见文献[49-55]。

上面介绍的 VC 方案的计算输入都来自一个客户端, 但在现实场景中, 有时需要多方各自提供部分输入, 共同完成同一计算。比如, 由分布式节点(传感器)组成的网络, 它们共同收集数据, 以作为某些计算的输入。为此, Choi 等人[56]将 Gennaro 等人[35]定义的单客户端(Single-client)模式扩展为多客户端(Multi-client)模式, 提出了多客户端非交互式的可验证计算(Multi-client verifiable computation, MVC)。在 MVC 模型中, 假设有 n 个计算能力较弱的客户端, 他们各自拥有数据x_i, 并且想要在无交互的情况下, 委托服务器完成函数 F 对于联合输入$(x_1, ..., x_n)$的计算。

为简便起见, 文献[56]假定在 MVC 模型中, 仅第一个客户端获得服务器返回的输出结果。具体地, 其他 $n-1$ 个客户端将输入$(x_2, ..., x_n)$发送给第一个客户端, 然后由第一个客户端负责运行一个单客户端情形的 VC 方案, 并将正确的计算结果返回给其他客户端。在这个过程中, 为了保证其他 $n-1$ 个客户端的输入不会泄露给第一个用户, 并且相互之间无交互, Choi 等人对双方的不经意传输协议进行了扩展, 引入适用于三方的非交互式代理不经意传输(Non-interactive proxy oblivious transfer, POT)协议。

POT 协议是对 Naor 等人[57]的交互式的代理不经意传输协议的变形和扩展。一个 POT 协议涉及三方: 发送者、选择者以及代理人。发送者输入消息(x_0,x_1), 选择方输入 b, 双方运行完 POT 协议后, 代理人仅得到x_b, 而发送者和选择者相互间不会获取对方的任何信息。一个 POT 协议需要满足两种隐私性: 发送方隐私性和选择方隐私性。前者表示代理方仅能得到对应于选择方输入比特的发送方的输入; 后者则表示代理方无法获知选择方的输入比特。在文献[56]中, Choi 等人利用非交互式的密钥交换协议设计了一个满足上述两种隐私性的 POT 协议, 然后将该协议与混淆方案[36]相结合得到了一个一次安全的 MVC 方案, 再通过 FHE 将其转换成多次安全的方案。

在 MVC 模型中, Choi 等人定义了两种隐私性要求: 针对第一个客户端的隐私性(Privacy against the first client)和针对服务器的隐私性(Privacy against the server)。前者要求第一个客户端除了函数的输出结果外, 无法获取任何关于其他客户端输入的信息; 后者则要求服务器不会获取到任何关于客户端输入的信息。对于文献[56]构造的 MVC 方案, FHE 的安全性和 POT 的隐私性保证了这两种隐私性。

Goldwasser 等人[58]对函数加密进行了扩展, 提出了多输入的函数加密(Multi-input functional encryption, MIFE)。利用该工具, 函数 F 对应的私钥sk_F能够对多个密文$(c_1, ..., c_n)$进行计算, 最终得到函数值 $F(x_1, ..., x_n)$, 其中, $(c_1, ..., c_n)$ 由不同的密钥$(EK_1, ..., EK_n)$分别对消息$(x_1, ..., x_n)$加密得到。由于(单输入的)函数加密可以用于实现(单客户端的)可验证计算, 因此多输入的函数加密的一个自然的应用就是实现多客户端的可验证计算, 并且其安全性保证了 MVC 方案的输入隐私性, 即服务器无法获得任何关于输入数据的信息。文献[58]基于不可区分的混淆(Indistinguishability obfuscation, iO)构造了相应的方案, 因此其安全性证明需要基于不可证伪的假设(Non-falsifiable assumptions)或者亚指数困难性假设。Boneh 等人[59]在不使用 iO 的情况下, 利用多线性映射提升了文献[58]中方案的效率。之后, 还有一些工作对多输入的函数加密进行了专门研究, 例如文献[60-67]。尽管技术方法、安全性以及适用的函数范围等各有所异, 但是相应的方案都可以用于实现保证输入隐私性的多客户端的可验证计算。

在文献[56]和文献[58]中, MVC 方案的安全性要求客户端诚实执行协议, 即不存在客户端与服务器或者客户端之间的合谋问题。之后, Gordon 等人[68]考虑客户端可能是恶意敌手的情形, 针对 MVC 提出了更强的安全性定义。他们以一种通用的可组合性框架的方式提出了基于模拟的安全性模型, 该模型能够对计算可靠性和隐私性进行统一定义。他们利

用姚式混淆电路、全同态加密和基于属性的加密构造了相应的方案, 与 Goldwasser 等人[58]的方案不同, 该方案仅依赖于可证伪的假设。对于方案的安全性, 文献[68]给出了正反两个分析结论: 当服务器与客户端之间无合谋的情况下, 该方案实现了基于模拟的安全性; 但是, 一旦服务器与客户端之间存在合谋, 即使仅合谋一个半诚实的客户端, 并且引入了可信假设, 比如公钥基础设施(Public key infrastructure, PKI)、公共参考串(Common reference string, CRS)等, 针对一般函数构造一个基于模拟安全的 MVC 方案都是不可行的。

Xu 和 Zhang[69-70]提出了一种利用同态代理重认证(Homomorphic proxy re-authenticator, HPRA)实现 MVC 的通用转换方式。HPRA 是 Deler 等人[71]提出的一种在多用户的数据聚合场景中提供安全性和可验证性保障的密码学原语。利用文献[71]的 HPRA 方案对上述通用转换进行实例化, Xu 和 Zhang 得到了第一个实用的 MVC 方案。文献[69]提供的实验数据表明, 与之前的 MVC 方案相比, 该方案在效率上占有很大的优势。此外, 该 MVC 方案还分别实现了针对第一个客户端的隐私性和针对服务器的隐私性, 充分保障了客户端的数据隐私。

综上所述, 针对一般函数想要实现可验证计算的隐私性, 通常需要基于较强的密码学组件, 比如全同态加密、混淆方案、(多输入的)函数加密, 因此方案的运行效率较低, 不具有实际的应用意义。

2.1.2 针对具体函数的可验证计算

2011 年, Benabbas 等人[72]开辟了一条针对具体函数实现高效的可验证计算的途径。他们考虑高阶多项式函数, 基于判定性 Diffie Hellman(Decisional Diffie Hellman, DDH)假设及其变体构造了相应的可验证计算方案。作为重要的应用, Benabbas 等人还介绍了如何利用多项式可验证计算解决关键词搜索[73-74]、可恢复性证明[75-76]等问题。

假设客户端想要计算多项式 $\mathrm{P}(x)=\sum_{i=0}^{d} a_i x^i \in \mathbb{Z}_p[x]$, 其中 $p>0$ 是一个大素数, 文献[72]的操作过程如下:

首先, 客户端选取随机值 $\varphi \in \mathbb{Z}_p$ 和 $\boldsymbol{r}=(r_0, ..., r_d) \in \mathbb{Z}_p^{n+1}$,针对每个系数计算$t_i = g^{\varphi a_i + r_i}$。然后将公钥 $pk=\{(a_i, t_i)\}_{i=0}^{d}$发送给服务器, 自己保存验证私钥 $sk=(\varphi, \boldsymbol{r})$。

对于客户端的输入 $x \in \mathbb{Z}_p$,服务器计算并返回相应的结果 $y=\mathrm{P}(x)=\sum_{i=0}^{d} a_i x^i$, 另附证明 $t=\prod_{i=0}^{d} t_i^{x^i}$。

客户端利用 sk 验证等式 $t = g^{\varphi y+\mathrm{R}(x)}$是否成立, 其中 $\mathrm{R}(x)=\sum_{i=0}^{d} r_i x^i$。若成立, 则接受 y; 否则, 拒绝该结果。

如果服务器返回一个伪造结果$(y' \neq y, t')$,满足验证要求 $t' = g^{\varphi y'+\mathrm{R}(x)}$, 则结合 $t = g^{\varphi y+\mathrm{R}(x)}$, 可成功解得$\varphi$。由于$\varphi$是客户端完全随机选择的, 因此该情况出现的概率可忽略, 这也说明了该构造实现了信息论意义下的安全性。

尽管上述构造提供了信息论安全的验证机制, 但是验证过程需要计算 $\mathrm{R}(x)$, 该计算量与 $\mathrm{P}(x)$的计算量相当, 这就失去了计算外包的意义了。为解决该问题, Benabbas 等人提出了一个新概念——封闭式高效伪随机函数(Pseudorandom function, PRF)。具体地, 假设$r_i = F_K(i)$, 其中 F 是一个满足封闭式高效性的 PRF, 则对于多项式 $\mathrm{R}(x)=\sum_{i=0}^{d} r_i x^i$, 如果知道 PRF 的私钥 K, 任何人都可以高效地计算 $\mathrm{R}(x)$, 其计算复杂度与 d 亚线性相关。这样一来, 上述可验证计算方案便可实现高效验证。在文献[72]中, Benabbas 等人基于 DDH 假设及其变体, 给出了封闭式高效 PRF 的具体构造。

上述方案有一个弊端, 即客户端必须以明文形式将多项式存储于云端。为了实现方案的隐私性, 文献[72]指出可以使用加法同态加密方案(比如 ElGamal 或者 Paillier 等)对多项式的系数及其标签加密, 然后将密文发送给服务器操作。但是, 该方法仅能隐藏多项式函数的信息以及输出结果, 无法保证输入数据的隐私性。

Zhang 和 Safavi-Naini[77-78]考虑在不使用 FHE 的前提下, 针对具体函数实现具有隐私保护的可验证计算。他们认为, 具有隐私保护的 VC 方案应能同时向服务器隐藏客户端的输入信息和计算函数信息。在文献[77]中, 利用多线性映射[79-80], Zhang 和 Safavi-Naini 针对单变量多项式函数和矩阵乘积, 构造出了同时具备上述两种隐私性的 VC 方案, 并且介绍了如何利用该方案实现外包的隐私信息检索(Private information retrieval, PIR)。其中, 隐私性是通过将 Boneh, Goh 和 Nissim 三人提出的适用于 2-DNF(Disjunctive normal form, 析取范式)的同态加密方案[81](简称 BGN_2 方案), 扩展为适用于 k-DNF 的同态加密方案(简称 BGN_k 方案), 然后利用 BGN_k 对客户端的输入以及计算函数进行加密实现的。在文献[78]中, Zhang 和 Safavi-Naini 首先针对矩阵-向量乘法给出了一个公开可验证计算的基础方案, 该方案的可靠性基于离散对数假设。在此基础上, 通过引入线性同态加密方案对向量或者矩阵加密, 实现输入隐私性或函数隐私性。但是, 这两种隐私性要求无法同时满足。此外, 通过对多项式计算进行分解, 使

得计算开销较大的步骤为矩阵-向量乘法, 然后利用上面的方案对高阶多项式计算进行外包, 文献[78]最终得到了一个针对高阶多项式且满足输入隐私性(或函数隐私性)的可公开验证方案。

Fiore 等人[82]专门针对密文数据的情形, 考虑如何实现高效的可验证计算。他们提出了一个新的密码学组件, 叫做同态 Hash 函数, 具体定义如下。

定义 3.同态 Hash 函数。同态 Hash 函数族 H: $\mathcal{D} \to R$ 由以下三个算法组成:

- $KGen$: 以安全参数λ为输入, 输出函数描述 H_K。
- H: 输入 $\varepsilon_1, ..., \varepsilon_t \in \mathcal{D}$, 输出对应的函数值 $H_K(\varepsilon_1), ..., H_K(\varepsilon_t)$。
- $Eval$: 输入 $H_K(\varepsilon_1), ..., H_K(\varepsilon_t)$ 和函数 f 的描述, 计算 $f(\varepsilon_1, ..., \varepsilon_t)$ 的哈希值, 即

$H_K(f(\varepsilon_1, ..., \varepsilon_t)) = Eval(f, (H_K(\varepsilon_1), ..., H(\varepsilon_t)))$。

文献[82]的基本思想如下: 利用同态加密方案对数据进行加密以保证隐私性, 然后对密文添加认证机制实现计算可靠性。其中, 加密采用 Brakerski 和 Vaikuntanathan[83]提出的同态加密方案(简称 BV 方案), 认证则采用 Gennaro 和 Wichs[84]提出的同态 MAC 方案(简称 GW 方案)。客户端首先利用 BV 方案对原始数据 x_i 进行加密, 然后将密文 $c = (c_0, ..., c_t)$ 发送给服务器存储。然后再利用 GW 方案, 对 c_i 生成对应的标记 σ_i。如此, 根据 MAC 的同态性质便可对密文的同态操作进行认证, 从而保证计算结果的正确性。然而, BV 方案的计算电路为 $\hat{f}: \mathbb{F}_q^{2nt} \to \mathbb{F}_q^{3n}$, 而文献[83]中的外包函数为 $f: \mathbb{F}_q^t \to \mathbb{F}_q$, 因此 BV 方案的密文空间与函数 f 的输入空间无法适配。这就是 Fiore 等人提出同态 Hash 函数的原因: 利用 H_K, 将 BV 方案 $\mathbb{F}_q^{3n}$ 空间的密文压缩为 $\mathbb{F}_q$ 中的元素, 同时保证同态操作性。

文献[82]考虑了几类多项式函数: 线性组合、多变量二次函数和高阶单变量多项式, 通过对同态 Hash 函数、BV 方案以及 GW 方案进行组合, 得到了高效的可验证计算方案, 这些方案都保证了输入隐私性和输出隐私性。此外, 它们还允许敌手进行验证询问, 从而能够抵抗"拒绝问题"攻击。在此之前, 满足隐私性的 VC 方案都不支持验证询问。

Fiore 等人[85]对文献[82]的结果进行了改进。具体地, 它们通过构造一种能够对多项式商环进行高效计算的简洁的非交互知识论证(Succinct non-interactive arguments of knowledge, SNARK)工具, 将其与基于环容错学习(Ring learning with errors, RLWE)假设的同态加密方案相结合, 设计了一个新的针对密文数据的高效 VC 协议。与文献[82]的工作相比, 该协议除了同样满足输入和输出隐私性外, 还具有几个方面的优势: 首先, 该协议支持阶大于 2 的算术电路操作; 其次, 它允许用户公开地对计算结果进行验证; 此外, 协议的验证过程无需原始计算数据。

为了构造出针对密文数据的高效可验证计算方案, 文献[82]和[85]都采用了同一种模式: 将认证机制与同态加密相结合, 利用前者对后者密文的正确性进行验证, 从而同时实现数据隐私性和计算可靠性。然而, 这两个方案都面临同样的弊端, 即需严格限制同态加密(Homomorphic encryption, HE)方案的参数选择, 要求 HE 的密文空间为 $\mathbb{Z}_q$, 其中 q 是大素数, 致使方案的灵活性较差。

针对上述弊端, Bois 等人[86]提出了一种新方法, 在保证 VC 和 HE 最佳运行效率的同时, 允许灵活选择 HE 的参数。作为关键技术, 文献[86]提出了一种更一般化的同态哈希函数, 该函数能将一个密文元素从多项式环 $\mathbb{Z}_q[X]/(h), h \in \mathbb{Z}_q[X]$ 压缩到一个更小的 Galois 环, 而文献[82]和[85]中的同态哈希函数仅能压缩到 $\mathbb{Z}_q$ 上, 并且要求 q 是一个很大的素数。Bois 等人首先给出了通用构造方法, 然后分别利用文献[83]中的 BV 方案和文献[20]中的"Muggle Proof"论证系统对通用构造进行实例化。文献[86]的 VC 方案除了针对服务器实现了输入和输出隐私性外, 还保证验证者也不会得到任何关于计算输入的信息。

对于支持具体函数操作的可验证计算, 除了上面介绍的基于 HE 实现隐私性的相关工作外, 还有一条路线是基于密码学工具——内积函数加密(Inner product functional encryption, IPFE)。由于 FE 的一个典型应用就是构造具有隐私保护的可验证计算, 因此 IPFE 可自然地构造针对内积函数的可验证计算, 同时保证输入数据的隐私性。内积计算作为多项式以及矩阵乘积等计算的基础, 不仅在机器学习的特征描述中可用于获取权重值, 还可以在生物认证系统中用于计算两个比特串之间的汉明距离, 因此 IPFE 近年来得到了广泛研究, 典型工作包括文献[87-93], 这些工作都为构造具有隐私保护的可验证内积计算提供了工具。

为了进一步提高方案的运行效率, 一些工作尝试使用非密码学工具实现数据的隐私性, 这类工作通常使用不同的转换技术将原问题转换为随机问题, 同时保护敏感的输入和输出信息。比如, 胡等人[11]使用伪装(盲化)技术, 针对矩阵乘积、矩阵行列式、

矩阵的逆等计算构造了高效的可验证安全外包计算协议, 这些协议既保证了矩阵计算问题中输入/输出的隐私性, 也保证了计算结果的高效验证性。Lei 等人[94]使用稀疏矩阵(Sparse matrix)对输入矩阵进行盲化, 构造了一个针对矩阵乘积计算的高效可验证计算方案。Chen 等人[95]利用同样的盲化方法, 首次构造了一个求解大型线性方程组的高效外包计算方案。Sheng 等人[96]采用矩阵转换技术, 将二维矩阵转换为一维向量, 实现了针对矩阵乘积计算的高效外包方案。Elkhiyaoui 等人[97]利用多项式分解原理, 给出了支持多项式函数的高效可验证计算方案。Zhang 等人[98]考虑批量矩阵乘法 $\boldsymbol{M}_{\text{pub}}\boldsymbol{M}_i$, 其中 $\boldsymbol{M}_i$ 为不同客户端提供的矩阵, $\boldsymbol{M}_{\text{pub}}$ 为数据中心提供的公开矩阵, 基于矩阵转换技术和矩阵消解(Matrix digest)技术, 构造了一个针对批量矩阵乘法的公开可验证计算方案, 该方案能够保证 $\boldsymbol{M}_i$ 的隐私性。

通过对单服务器情形下具有隐私保护的可验证计算相关工作进行梳理分析, 可得出以下结论: 1)针对一般函数的通用构造通常基于较强的密码学组件, 比如混淆方案、全同态加密或者(多输入的)函数加密; 2)针对具体函数的构造方法可大致分为三种, 第一种是基于同态加密, 通过对“认证机制+同态加密”的模式进行实例化, 利用前者对后者的密文正确性进行验证, 从而同时实现数据隐私性和计算可靠性; 第二种是利用内积函数加密, 对输入数据进行加密的同时, 实现密文数据的可验证内积计算; 第三种是采用非密码学手段, 利用转换技术(比如盲化技术、多项式分解技术等)将原问题转换为随机问题, 从而保护敏感的输入和输出信息。相关梳理情况如图 4 所示。

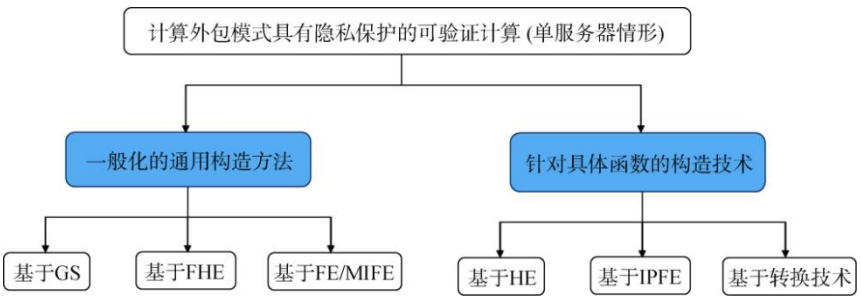

图 4 计算外包模式具有隐私保护的可验证计算构造方法

Figure 4 Methods for constructing verifiable computation with privacy protection under computing-outsourcing mode

2.2 多服务器情形具有隐私保护的可验证计算

多服务器的可验证计算是指外包计算任务由多个服务器共同完成。Canetti 等人[99-100]首先考虑了多服务器情形下的可验证计算, 但是他们的方案仅实现了计算可靠性, 即确保客户端接受的计算结果是正确的(只要至少存在一个服务器是诚实的), 并未考虑客户端输入数据的隐私性。

对于单服务器可验证计算的隐私性, 大多数方案都需要依赖全同态加密来实现。此外, 方案还要求客户端执行一个时间复杂度与计算函数本身相当的预处理过程, 因此外包计算的效率实际是建立在对同一个函数执行多次操作的均摊(Amortized)意义上的。这两个弊端严重削弱了可验证计算方案的实用性。对于多服务器的可验证计算, 是否能够避免上述两个问题? Ananth 等人[101]给出了肯定答案。在文献[101]中, 他们分别针对服务器个数 $n = 2$ 和 $n > 2$ 的两种情形, 构造出了无需预处理操作和 FHE 技术的可验证计算方案, 并且方案满足输入隐私性和输出隐私性。但是, 隐私性的实现依赖于复杂的混淆电路, 并且方案要求服务器之间进行连续交互: 客户端发送一个消息给第一个服务器, 自第二个服务器开始, 每一个服务器都从上一个服务器处得到一个消息, 最终由最后一个服务器将消息返回给客户端。

对于多服务器的可验证计算, 如果方案的运行要求服务器之间进行通信, 这不仅增加了服务器的通信和计算开销, 同时也增加了各服务器间合谋的风险, 可能导致客户端输入数据的恢复和泄露。因此, 无论是对于客户端还是服务器, 通信式的多服务器可验证计算方案都不是最佳选择。对此, 文献[101]提出了一个公开问题: 对于多服务器情形, 如何在无服务器通信的情况下, 实现具有隐私保护的可验证计算? 换言之, 对于多服务器的可验证计算, 如何同时保障客户端数据隐私和服务器之间无通信。

Zhang 等人[102]提出了多服务器的局部可验证计算(Verifiable local computation, VLC) 模型。利用 VLC 模型, 客户端首先将数量不受限的数据块

$x=(x_1, ..., x_n)$外包给多个服务器, 之后客户端可以委托服务器对外包数据块的任意部分进行计算, 使得 1)数据隐私性:恶意的服务器无法获得关于 x 的任何信息; 2)计算完整性: 恶意的服务器无法让客户端接受一个错误的计算结果。文献[102]分别基于伪随机函数和双线性映射, 构造了两个针对多项式函数操作的多服务器 VLC 方案Γ_1和Γ_2。这两个方案都实现了 1)ξ-privacy: 在合谋服务器数量不超过ξ的情况下, 任何服务器无法得到关于 x 的任何信息; 2)ξ-security: 在合谋服务器数量不超过ξ的情况下, 服务器无法使客户端接受一个错误的计算结果, 其中 $1\leqslant\xi<k$, k 为服务器个数。不同的是, Γ_1和Γ_2分别适用于阶数 $d<k/\xi$和 $d\leqslant 2<k/\xi$的多项式函数。

通过将单服务器的 Pinocchio 证明系统[103]运行于 3 个(及以上)服务器, Schoenmakers 等人[104]构造了一个适用于任意电路的可验证计算系统 Trinocchio。在该系统中, 每个服务器都执行和 Pinocchio 证明者一样的操作, 最终客户端得到一个正常的 Pinocchio 证明, 并执行同样高效的验证操作。这种将证明的生成分散于不同服务器的操作方式, 使得每个服务器都无法获知客户端的输入数据, 从而实现了信息论意义下的输入隐私性。但是, 在 Trinocchio 系统中, 各个服务器需要运行一个安全多方计算协议, 该协议的运行要求服务器相互之间进行通信, 并且安全性依赖于不可证伪的假设。

Zhang 等人[105-106]针对文献[101]提出的公开问题, 提出了多服务器可验证计算(Multi-server verifiable computation, MSVC)模型。该模型无需服务器之间进行通信, 同时保证客户端的输入相对于各服务器的隐私性。具体地, 客户端将输入 x 隐私地分发给非通信的多个服务器, 每个服务器在本地计算函数 F 得到部分结果, 客户端根据这些结果恢复出最终结果 $F(x)$。在文献[105]中, Zhang 等人提出了一个针对矩阵乘法 $\boldsymbol{Mv}$ 的 MSVC 通用构造, 其中 $\boldsymbol{M}$ 为矩阵, $\boldsymbol{v}$ 为向量。该构造实现了信息论意义下的安全性和隐私性, 即服务器无法让客户端接受一个错误的结果, 并且任何服务器都无法得到任何关于 $\boldsymbol{M}$ 和 $\boldsymbol{v}$ 的信息。通过对参数进行优化, 文献[105]还给出了服务器个数分别为 3 和 4 的两个实例化方案。其中, 3 服务器方案实现了服务器个数的最优; 4 服务器方案实现了计算开销的最优。在文献[106]中, Zhang 等人针对多项式$\mathrm{P}(q,l,d)$, 构造出了 5 个 MSVC 方案, 其中多项式有 l 个变量, 阶总和小于等于 d, 系数取自有限域 $\mathbb{F}_q$。这些方案满足 1)t-privacy: 任意 t 个服务器无法得到关于 x 的任何信息; 2)t-security: 任意 t 个服务器无法使得客户端输出错误的计算结果。上述两个性质要么在信息论意义下成立, 要么在计算意义下成立。

此外, 一些学者还专门针对双服务器模型下密文数据的可验证计算进行了研究。Li 等人[107]利用文献[56]构造 MVC 的思想, 基于混淆方案[36]、公钥加密(Public key encryption, PKE)方案和一次一密方案, 实现了双服务器模型下保证输入输出隐私性且可公开验证的 MVC 方案。该方案无需客户端之间进行交互, 但是两个服务器需要通信共同完成计算。

Chen 等人[108]提出了双服务器可验证同态秘密分享(Two-server verifiable homomorphic secret sharing, 2SVHSS)模型。在该模型中, 输入客户端利用公钥将自身的数据加密后发送给服务器; 每个服务器对收到的秘密数据进行计算, 得到部分结果; 输出客户端根据所有的部分结果, 恢复出最终的函数值。利用 Boyle 等人[109]提出的具有近似线性解密的公钥加密(PKE with nearly linear decryption, PKE-NLD), 他们构造了一个针对高阶多项式的 2SVHSS 方案, 该方案能够保证服务器不会得到客户端的原始数据和最终的计算结果, 并且服务器无法让客户端接受一个错误的计算结果。

Chen 等人[110]将标签同态加密[111]和双服务器的密文委托计算[112]相结合, 提出了基于标签的双服务器委托计算(Two-server delegation of computation on label-encrypted data, 2S-DCLED)模型。基于 PRF 存在性假设, 他们构造了一个针对二次函数的 2S-DCLED 方案, 该方案允许客户端在保证输入数据隐私的情况下, 将二次函数计算委托给两个无需通信的服务器。为了确保服务器返回的计算结果正确性, Chen 等人将 2S-DCLED 方案与同态 MAC 方案相结合, 得到了一个可以对结果进行验证的 2S-VDCLED 方案。此外, 他们还进一步对该方案进行扩展, 得到了一个在 d 服务器情形下针对 d 次多项式函数的 dS-VDCLED 方案。

上述具有隐私保护的多服务器可验证计算方案的对比情况如表 1 所示, 其中的缩写释义见表 2。在表 1 中, 本文分别梳理了各个方案支持的函数类型、服务器数量 k、安全性(包括隐私性和可靠性)基于的困难假设, 以及服务器之间是否需要通信、是否支持计算函数的公开代理、计算结果是否支持公开验证。

根据梳理情况可知, 不同的方案由于技术特点、功能侧重的不同, 在不同对比项上的表达不尽相同。总的来说, 现有方案能够支持多种计算类型, 包括

表 1　多服务器情形下具有隐私保护的可验证计算方案对比

Table 1　Comparison of multi-server verifiable computation schemes with privacy protection

方案	函数类型	服务器数量 k	隐私性基于的假设	可靠性基于的假设	非通信式	公开代理性	公开验证性
[101]	所有函数	$k=2$	OWF	OWF	×	×	×
	所有函数	$k>2$	DDH	DDH	×	×	×
[102]	二次函数	$k\geqslant 2$	PRF	DLIN	✓	×	×
	多项式函数	$k\geqslant 2$	PRF	PRF	✓	×	×
[104]	所有函数	$k\geqslant 3$	i.t.	PKE+q-PDH+q-SDH	×	×	✓
[105]	矩阵-向量乘法	$k\geqslant 3$	i.t.	i.t.	✓	✓	×
[106]	多项式函数 P(q,l,d)	$k=d(t+1)+1$	i.t.	i.t.	✓	✓	×
		$k=(d+1)t+1$	i.t.	i.t.	✓	✓	×
		$k=dt+1$	i.t.	i.t.	✓	✓	×
		$k=d(t+1)+1$	i.t.	DHI	✓	✓	✓
		$k=(d+1)t+1$	i.t.	DLog	✓	✓	✓
[107]	所有函数	$k=2$	PKE+One-time pad encryption	PKE+One-time pad encryption+GS	×	×	✓
[108]	多项式函数	$k=2$	PKE-NLD	i.t.	✓	✓	✓
[110]	二次函数	$k=2$	PRF	PRF	✓	✓	×
	d 次多项式	$k=d>2$	PRF	PRF	✓	✓	×

表 2　表 1 和表 3 中的缩写释义

Table 2　Abbreviations in Tables 1 and 3

缩写	英文全称	缩写	英文全称
OWF	One Way Function	GS	Garbled Scheme
DDH	Decisional Diffie-Hellman	PKE-NLD	PKE with Nearly Linear Decryption
PRF	Pseudorandom Function	k-APMDH	k-Augmented-Power Multilinear Diffie-Hellman
DLIN	Decisional Linear	EF-AGCD	Error-Free Approximate Greatest Common Divisor
q-PDH	q-power Diffie-Hellman	DCR	Decisional Composite Residuosity
q-SDH	q-strong Diffie-Hellman	IND-CPA	Indistinguishability under Chosen-Plaintext Attack
i.t.	information-theoretic	IND-CCA	Indistinguishability under Chosen-Ciphertext Attack
DHI	Diffie-Hellman Inversion	UF-CPA	Unforgeability under Chosen-Plaintext Attack
Dlog	Discrete Logarithm	UF-CCA	Unforgeability under Chosen-Ciphertext Attack

所有函数、二次函数、多项式函数、矩阵-向量乘法等, 涉及的服务器数量有两个或者两个以上的, 其中有的方案[101,107-108]固定了服务器数量, 有的方案服务器数量[106,110]与支持的多项式的阶数有关。对于方案的隐私性和可靠性, 大多数方案都需要基于相应的困难假设实现, 比如 PRF 存在性假设、DDH 假设等, 也有少数方案[105-106]实现了信息论意义下的安全性。对于如何在服务器间无通信的情况下, 实现具有隐私保护的可验证计算, Ananth 等人[101]曾将其作为公开问题提出。目前, 已有不少方案解决了这个公开问题, 比如 Zhang 等人[105-106]提出的多个 MSVC 方案, 以及 Chen 等人[108,110]提出的 2SVHSS 方案、2S-DCLED 方案、dS-VDCLED 方案。与此同时, 由于这些方案基于的模型支持计算函数的公开代理, 因此相应的方案自然满足公开代理性, 而其余的方案都不满足该性质。对于公开验证性, 单服务器情形已有多个 PVC 方案[40-55,78], 而多服务器情形目前仅有部分方案[104,106-108]满足该性质。

3　数据外包模式具有隐私保护的可验证计算

在不考虑隐私性时, 数据外包模式的可验证计算一般由两种密码学工具实现, 即同态 MAC 和同态签名。两者的功能类似: 允许服务器对认证数据(标记或签名)进行相关操作, 最终返回给客户端计算结果及相应的证明(短标记或短签名), 客户端通过对该证明进行验证, 以判断计算结果是否正确。唯一区别在于同态 MAC 是私密验证, 而同态签名则可以公开验证。

对于隐私性, 同态 MAC 考虑的是语义安全性, 即保证标记不泄露被认证数据的信息, 而同态签名

的隐私性[①]被定义为上下文隐藏性(Context hiding), 即要求同态操作得到的派生签名不泄露原始输入数据的信息。实现数据外包模式下具有隐私保护的可验证计算, 通常有两种方法: 第一, 将具有隐私性的同态 MAC 方案与同态加密方案相结合, 得到同态认证加密方案, 从而保证输入数据和输出结果对于服务器的隐私性; 第二, 直接利用上下文隐藏的同态签名构造 VC 方案, 实现原始数据对于验证者的隐私保护。两种方法得到的 VC 方案都能够允许客户端先将大量的数据存储在云端, 然后再委托服务器对其进行相应的计算, 外包数据与计算函数无绑定关系。

3.1 基于同态认证加密的可验证计算

Joo 和 Yun[113]首先提出了同态认证加密(Homomorphic authenticated encryption, HAE)的概念。一个 HAE 方案可看作对一个同态 MAC 方案增加额外的解密算法, 使得拥有私钥的用户可以从标记中将被认证的消息解密出来。对于 HAE 的安全性, 文献[113]分别定义了隐私性(privacy)和可靠性(authenticity), 前者要求密文不泄露明文的信息, 后者要求密文不能够被伪造。根据敌手是否能够进行加密或解密询问, 隐私性分为选择明文攻击下的不可区分性(Indistinguishability under chosen-plaintext attack, IND-CPA)和选择密文攻击下的不可区分性(Indistinguishability under chosen-ciphertext attack, IND-CCA), 可靠性分为选择明文攻击下的不可伪造性(Unforgeability under chosen-plaintext attack, UF-CPA)和选择密文攻击下的不可伪造性(Unforgeability under chosen-ciphertext attack, UF-CCA)。因此, 一个安全的 HAE 方案能够实现数据外包模式下针对密文数据的可验证计算。对于 HAE, 一种典型的构造方法是采用“Encrypt-and-MAC”组合模式, 即首先构造具有隐私性的同态 MAC, 然后将其与同态加密相结合。下面, 首先回顾已有的具有隐私性的同态 MAC 方案。

作为实现数据外包的可验证计算的一种有效工具, 同态 MAC 允许服务器对认证数据进行相关操作, 最终得到一个短标记以对计算结果进行认证。2013 年, Gennaro 等人[84]首次形式化地定义了同态 MAC, 并且基于 FHE 给出了一个适用于任意函数的全同态 MAC 方案。为了更加清楚方便地标识数据, 文献[84]引入带标签程序(Labeled-program)对同态 MAC 进行定义。

一个带标签程序 $\mathcal{P}=(f,\tau_1,...,\tau_k)$包含一个函数 $f:\{0,1\}^k\to\{0,1\}$, 以及每个输入位置 $i\in[k]$独有的标签$\tau_i\in\{0,1\}^*$。给定带标签程序$\mathcal{P}_1,...,\mathcal{P}_t$以及函数 $g:\{0,1\}^t\to\{0,1\}$, 还可以定义组合程序 $\mathcal{P}^*=g(\mathcal{P}_1,...,\mathcal{P}_t)$, 表示函数 g 对程序$\mathcal{P}_1,...,\mathcal{P}_t$的输出的计算结果。假设$g_{id}$是规范的恒等函数, $\tau\in\{0,1\}^*$是标签, 则$I_\tau:=(g_{id},\tau)$表示关于标签τ的恒等程序。其中, 任意的程序 $\mathcal{P}=(f,\tau_1,...,\tau_k)$都可表示为恒等标签的组合形式, 即 $\mathcal{P}=f(I_{\tau_1},...,I_{\tau k})$。

定义 4. 同态消息认证码方案。一个同态消息认证码方案 **HMAC** 由以下四个概率多项式时间的算法组成:

- $Setup(1^\lambda)\to(pk,sk)$: 密钥生成算法, 输入安全参数$\lambda$, 输出私钥 sk, 公钥 pk。其中, 公钥 pk 定义了消息空间 $\mathcal{M}$, 以及由所有容许函数 $f:\mathcal{M}^k\to\mathcal{M}$ 构成的集合 $\mathcal{F}$。
- $Auth(sk,m,\tau)\to\sigma$: 认证算法, 利用私钥认证消息 $m\in\mathcal{M}$ 及其对应的标签 $\tau\in\{0,1\}^*$, 输出标记σ。
- $Eval(pk,f,\boldsymbol{\sigma})\to\psi$: 确定性的计算算法, 以电路 $f\in\mathcal{F}$ 和标记$\boldsymbol{\sigma}=(\sigma_1,...,\sigma_k)$为输入, 利用密钥 evk 计算输出新的标记ψ。
- $Ver(sk,m,\mathcal{P},\sigma)\to 0/1$: 验证算法, 利用$\sigma$验证 $m\in\mathcal{M}$ 是否为程序 $\mathcal{P}$ 关于已认证数据的输出, 根据验证结果输出 1(接受)或 0(拒绝)。

关于正确性, **HMAC** 方案要求对于所有的密钥对$(pk,sk)\leftarrow Setup(1^\lambda)$, 下面两个性质成立:

1) 认证正确性: 对于任意的消息 $m\in\mathcal{M}$ 和任意的标签 $\tau\in\{0,1\}^*$, 如果 $\sigma\leftarrow Auth(sk,m,\tau)$, 则 $Ver(sk,m,I_\tau,\sigma)=1$。

2) 组合正确性: 给定容许函数 $g:\mathcal{M}^k\to\mathcal{M}$ 以及任意的消息/程序/标记集合$\{(m_i,\mathcal{P}_i,\sigma_i)\}_{i=1}^k$满足

$$Ver(sk,m_i,\mathcal{P}_i,\sigma_i)=1。$$

如果 $m^*:=g(m_1,...,m_k)$, $\mathcal{P}^*:=g(\mathcal{P}_1,...,\mathcal{P}_k)$, $\sigma^*:=Eval(pk,g,(\sigma_1,...,\sigma_k))$, 则

$$Ver(sk,m^*,\mathcal{P}^*,\sigma^*)=1。$$

一个安全的同态 MAC 方案要求满足不可伪造性: 任意概率多项式时间的敌手都无法伪造一个有效的标记。

由挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 交互进行的下述实验刻画了不可伪造性:

初始化阶段 $\mathcal{C}$ 执行$(pk,sk)\leftarrow Setup(1^\lambda)$, 将公钥 pk 给 $\mathcal{A}$, 私钥sk 自己保存, 并且初始化列表$\mathrm{T}=\emptyset$。

询问阶段 敌手 $\mathcal{A}$ 适应性地进行如下询问:

① 由于同态签名是公开验证, 因此无法像同态 MAC 一样针对签名定义语义安全性。

1) 认证询问: $\mathcal{A}$ 提交消息 $m \in \mathcal{M}$, 如果在列表 T 中存在元组(τ, m), 则 $\mathcal{C}$ 计算$\sigma \leftarrow Auth(sk, m, \tau)$; 否则 $\mathcal{C}$ 选择一个新标签$\tau \in \{0,1\}^*$, 更新列表 $\mathrm{T} = \mathrm{T} \cup (\tau, m)$, 并计算$\sigma \leftarrow Auth(sk, m, \tau)$。$\mathcal{C}$ 将标记σ返回给 $\mathcal{A}$。

2) 验证询问: $\mathcal{A}$ 提交元组$(m, \mathcal{P}, \sigma)$, 挑战者回答$Ver(sk, m, \mathcal{P}, \sigma)$的输出结果。

输出阶段 敌手 $\mathcal{A}$ 输出消息m^*, 带标签程序$\mathcal{P} = (f^*, \tau_1^*, ..., \tau_k^*)$以及标记$\sigma^*$。

敌手 $\mathcal{A}$ 成功当且仅当 $Ver(sk, m^*, \mathcal{P}^*, \sigma^*) = 1$, 并且下面两个条件之一成立:

1) 伪造类型 I: 存在 $i \in \{1, ..., k\}$使得$(\tau_i^* \notin \mathrm{T})$;

2) 伪造类型 II: 对于消息$m_1,...,m_k$, 列表 T 中包含元组$(\tau_1^*, m_1), ..., (\tau_k^*, m_k)$, 并且满足$m^* = f^*(m_1, ..., m_k)$。

定义 5.不可伪造性。对于一个同态消息认证码方案 **HMAC**, 如果对任意概率多项式时间的敌手 $\mathcal{A}$, 下式成立

$$Adv_{\mathcal{A}}^{Unforg}(\mathbf{HMAC}, \lambda) = \left|\Pr[\mathcal{A}\text{成功}]\right| \leqslant negl(\lambda),$$

其中, $negl(\cdot)$是一个可忽略函数, 则称 **HMAC** 方案是(强)不可伪造的。

如果在上述实验中, 敌手 $\mathcal{A}$ 不允许进行验证询问, 则称方案 **HMAC** 是弱不可伪造的。

之后, Lai 等人[114]首次考虑同态 MAC 的标记的隐私性, 提出了一个新的密码学原语——同态加密认证(Homomorphic encrypted authenticator, HEA)。大致来说, HEA 可看作特殊的同态 MAC, 除了不可伪造的安全性外, 它还额外满足语义安全性, 即要求消息 m(标签为τ)的标记σ不泄露任何关于 m 的信息。根据文献[114]的定义, 由挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 交互进行的下述实验刻画了语义安全性:

初始化阶段 $\mathcal{C}$ 执行$(pk, sk) \leftarrow Setup(1^\lambda)$, 将公钥 pk 给 $\mathcal{A}$, 私钥 sk 自己保存, 并且初始化列表 $\mathrm{T} = \emptyset$。

认证询问阶段 $\mathcal{A}$ 向 $\mathcal{C}$ 适应性地询问消息的标记。$\mathcal{A}$ 提交消息 $m \in \mathcal{M}$, 如果在列表 T 中存在元组(τ, m), 则 $\mathcal{C}$ 计算$\sigma \leftarrow Auth(sk, m, \tau)$; 否则 $\mathcal{C}$ 选择一个新标签$\tau \in \{0,1\}^*$, 更新列表 $\mathrm{T} = \mathrm{T} \cup (\tau, m)$, 并计算$\sigma \leftarrow Auth(sk, m, \tau)$。$\mathcal{C}$ 将标记σ返回给 $\mathcal{A}$。

挑战阶段 敌手 $\mathcal{A}$ 提交一个标签$\tau^* \in \{0,1\}^*$和两个消息$m_0, m_1 \in \mathcal{M}$。挑战者 $\mathcal{C}$ 选取一个随机比特 $\beta \in \{0,1\}$, 计算$\sigma^* \leftarrow Auth(sk, \tau^*, m_\beta)$, 然后将$\sigma^*$发送给 $\mathcal{A}$。

猜测阶段 敌手 $\mathcal{A}$ 输出比特$\beta' \in \{0,1\}$作为对β的猜测值。如果$\beta = \beta'$, 则 $\mathcal{A}$ 成功。

定义 6.语义安全性。对于一个同态加密认证方案 **HEA**, 如果对任意多项式时间的敌手 $\mathcal{A}$, 下式成立

$$Adv_{\mathcal{A}}^{Seman}(\mathbf{HEA}, \lambda) = \left|\Pr[\beta = \beta'] - \frac{1}{2}\right| \leqslant negl(\lambda),$$

其中, $negl(\cdot)$是一个可忽略函数, 则称 **HEA** 方案是语义安全的。

尽管文献[84]未特意考虑标记的隐私性, 但由于其标记是 FHE 的密文集合, 因此该方案自然地满足语义安全性。鉴于此, 在文献[114]中, Lai 等人将文献[84]的全同态 MAC 方案转换为了一个支持任意函数操作的全同态加密认证方案。但是, 该方案仅实现了弱不可伪造性, 原因在于敌手可以通过恶意的验证询问破坏方案的安全性。在实际应用中, 这意味着用户一旦得到一个验证失败的结果标记, 则需要立马终止方案。此外, 该方案的验证过程需要执行 FHE 的同态计算操作, 其时间复杂度与计算函数本身相当。为了提高验证效率, Gennaro 等人提出可以采用预处理方式, 将昂贵的同态计算过程外包给服务器提前计算, 而剩余的计算仅为简单的等值比较。通过将同态计算开销均摊在之后对于同一函数的多次验证中, 客户端便实现了均摊意义下的高效验证性。

Lai 等人构造了一个能够支持敌手进行任意验证询问的 HEA 方案。由于同态签名的验证过程可以公开进行, 因此敌手无法通过验证询问获取任何额外的信息。Lai 等人观察到这个事实可以支撑 HEA 方案实现强不可伪造性。但是, 签名不一定保证被签名消息的隐私性, 因此一个同态签名方案并非一定是一个 HEA 方案, 还需要采取额外的技术将前者转换为后者。在文献[114]中, Lai 等人考虑 Freeman[115]的线性同态签名方案, 通过判定线性(Decision linear, DLIN)假设技术为其增加语义安全性, 从而构造了一个针对线性函数的满足强不可伪造性的 HEA 方案。与 Gennaro 等人的方案相比, 该方案的验证无需均摊操作, 其主要开销仅为 4 次双线性映射。

Li 等人[116-117]继续对同态 MAC 的隐私性进行研究, 提出了一个新的密码学原语: 隐私保护的同态 MAC(Privacy-preserving homomorphic MACs, PHMAC)。与一般的同态 MAC 相比, PHMAC 要求额外满足消息隐私性和高效验证性。前者的定义与文献[114]中的语义安全性类似, 即要求标记不泄露被认证消息的信息; 后者则表示验证计算结果的开销要远小于计算函数本身所需的开销。Li 等人基于 Catalano 等人[118]的同态签名方案构造 PHMAC 方案。在文献[118]中, Catalano 等人利用多线性群上的 k-增

广幂多线性(k-Augmented-power multilinear)Diffie-Hellman 假设, 构造了一个针对多项式函数的满足(均摊意义下)高效验证性的同态签名方案。Li 等人首先引入了一个 PRF, 将验证密钥作为 PRF 的输出, 从而将文献[118]的同态签名方案转换成一个私密验证的同态 MAC 方案。对于变换得到的同态 MAC 方案, 将原来的签名由两个群元素适应性地扩展为三个群元素。该扩展操作在保证不影响方案正确性和安全性证明的前提下, 将方案的隐私性归约到 DDH 假设。最终, 得到了一个支持固定阶的多项式函数的 PHMAC 方案, 并且方案安全性允许敌手进行任意的验证询问。对于验证操作, Li 等人的方案除了执行 4 次多线性映射操作外, 还需再次执行多项式函数计算。因此, 该方案也仅实现了均摊意义下的高效验证性。

上述几个工作都实现了同态 MAC 方案的隐私性, 其中文献[84]的全同态 MAC 方案支持任意函数操作, 但是其构造依赖于全同态加密, 方案十分低效。此外, 该方案的安全性仅达到弱不可伪造性。文献[114]给出了两个 HEA 方案, 其中一个实现了强不可伪造但仅支持线性函数操作, 另一个适用于任意函数但仅满足弱不可伪造性。文献[116-117]中的 PHMAC 方案允许敌手进行任意的验证询问, 同时支持多项式函数操作。但是, 方案安全性基于的多线性映射其存在性尚未得到理论证实。

为了进一步实现数据外包模式下具有隐私保护的可验证计算, Lai 等人[114]提出了可验证认证加密(Verifiable homomorphic encryption, VHE)的概念, 即能够对密文数据进行可验证计算的对称模式的同态加密。需要说明的是, VHE 的定义与文献[113]的 HAE 几乎相同, 两者可以相互转换。通过将 HEA 方案与同态加密相结合, Lai 等人得到了一个针对线性函数的 VHE 方案, 该方案实现了强可靠性(Strong authenticity), 即敌手能够进行任意的解密询问。采用同样的转换方式, 文献[84]可相应地得到一个针对任意函数但仅满足弱可靠性(Weak authenticity)的 VHE 方案, 文献[116-117]可得到一个针对多项式函数且实现强可靠性的 VHE 方案。

对于 HAE(或 VHE)的构造方式, 除了将同态加密和具有隐私性的同态 MAC 按照“Encrypt-and-MAC”模式组合外, 还有一些其他的方法。比如, 在文献[113]中, Joo 和 Yun 直接基于(无噪声)近似最大公因子假设, 构造了一个针对低次多项式 $f:\mathbb{Z}^l \to \mathbb{Z}$ 的 HAE 方案。该方案引入了一个 PRF 函数, 构造较为简洁, 但是验证操作除了执行 l 次 PRF 求值外, 还要求客户端再次执行多项式 f。因此, 该方案也仅实现了均摊意义下的高效验证性。

Catalano 等人[119]提出了可公开验证的线性同态认证加密(Linearly homomorphic authenticated encryption with public verifiability, LAEPuV), 并且基于 Paillier 密码系统和文献[120]的线性同态签名方案, 实现了一个针对密文数据的简单高效的可公开验证计算机制。该工作的关键技术是对 Paillier 的密文进行盲化, 使得同态签名的对象是盲化后的明文, 从而保证真正的明文不会在签名过程中泄露。Struck 等人[121]指出文献[119]的 LAEPuV 方案存在缺陷, 其验证算法会给出错误的否定结果。因此, 他们构造了一个满足可证明正确性(Provable correctness)的可公开验证的线性同态认证加密方案, 简称 LEPCoV 方案。该方案在弥补 LAEPuV 缺陷的同时, 实现了更优的性能。对于验证过程, LAEPuV 方案和 LEPCoV 方案都仅需 k 次模幂运算, 其中 k 为线性函数的输入长度。

Tran 等人[122]分别针对线性函数和多变量二次函数, 构造了两个 VHE 方案。线性版本 VHE 通过将文献[123]的同态 MAC 方案与对称模式的同态加密相结合得到, 其安全性仅基于伪随机函数; 二次版本 VHE 则基于 Paillier 密码系统和 Backes 等人[124]的同态 MAC 技术实现。与文献[124]中的方案相比, 该方案无需双线性映射, 验证操作仅需 3 次模幂运算, 因此更加高效。

最近, Kim 等人[125]采用“all in one”的方式对 HAE 的安全性进行了重新定义。他们将 HAE 的隐私性和可靠性进行合并, 给出了一个更加简洁的统一的安全性定义。该定义能够推导出 IND-CCA 安全性和 UF-CCA 安全性。通过将 FHE、全同态 MAC 和 OR-同态 MAC 三种组件相结合, Kim 等人构造了一个满足新安全性的全同态认证加密方案。与文献[84]的全同态 MAC 一样, 该方案的验证操作也需预处理操作, 因此也仅实现了均摊意义下的高效验证性。

上述同态认证加密方案的对比见表 3, 其中的缩写释义见表 2。在表 3 中, 本文分别梳理了各同态认证加密方案支持的函数类型、隐私性/可靠性的类型、隐私性/可靠性基于的困难假设、计算结果是否可以公开验证、高效性的实现是否需要均摊操作, 以及服务器返回的证明规模。需要说明的是, 高效性是可验证计算的一个基本属性, 即要求客户端的验证时间要远小于直接计算函数本身所需的时间, 否则计算便失去外包的意义。因此, 作为数据外包模式的可验证计算的实现工具, 同态认证加密方案都需满足这个基本要求。但是, 一些方案的验证过程要求客户端执行一个时间复杂度与计算函数本身相当的预处理过程, 因此其高效验证性是建立在均摊意义上的。严谨地说, 这些方案仅实现了“均摊意义高效验证”。为了尽可能缩减可验证计算中客户端的开销, 除了实

表 3 同态认证加密方案对比

Table 3 Comparison of homomorphic authenticated encryption schemes

方案	函数类型	隐私性	隐私性基于的假设	可靠性	可靠性基于的假设	公开验证性	均摊意义高效验证	证明规模
[84]	所有函数	IND-CPA	FHE	UF-CPA	FHE+PRF	×	Yes	$O(\lambda)$
[114]	线性函数	IND-CPA	DLIN	UF-CCA	SDH	×	No	$O(1)$
	所有函数	IND-CPA	FHE	UF-CPA	FHE+PRF	×	Yes	$O(\lambda)$
[116]	k 阶多项式函数	IND-CPA	DDH	UF-CCA	k-APMDH	×	Yes	$O(1)$
[117]	k 阶多项式函数	IND-CPA	DDH	UF-CCA	k-APMDH	×	Yes	$O(1)$
[113]	多项式函数	IND-CCA	EF-AGCD	UF-CCA	EF-AGCD	×	Yes	$O(1)$
[119]	线性函数	IND-CCA	DCR+Strong-RSA	UF-CCA	Strong-RSA	✓	No	$O(1)$
[121]	线性函数	IND-CCA	DCR+Strong-RSA	UF-CCA	Strong-RSA	✓	No	$O(1)$
[122]	线性函数	IND-CPA	PRF	UF-CCA	PRF	×	No	$O(1)$
	二次函数	IND-CPA	PRF+DCR+DLog	UF-CCA	PRF+DLog	×	No	$O(1)$
[125]	所有函数	“all in one” security	FHE+同态 MAC	“all in one” security	FHE+同态 MAC	×	Yes	$O(1)$

(注: $O(\lambda)$表示证明规模与安全参数λ线性相关; $O(1)$表示证明规模为常数级别)

现高效的验证操作外, 还要求服务器端返回的证明尽量简短, 以减少客户端的通信量和计算量。因此, 表 3 专门对比了各个方案的证明规模。其中, “$O(\lambda)$”表示服务器返回的证明大小与安全参数相关, “$O(1)$”表示证明大小为固定的常数值, 不随输入长度、函数的阶数等参数而变化。

根据梳理情况可知, 现有的同态认证加密方案支持多种函数类型, 包括线性函数、二次函数、多项式函数和通用的所有函数。其中, 支持所有函数操作的方案都是基于 FHE 实现的, 并且高效验证性都是建立在均摊意义上, 而其余方案基于的安全性假设各有不同, 比如 DDH 假设、DLIN 假设、PRF 存在性假设等。对于隐私性而言, 大部分方案都只实现了针对选择明文攻击敌手的 IND-CPA 安全性, 与此不同的是, 大多数方案的可靠性都达到了抵抗选择密文攻击的 UF-CCA 安全性。作为一项特殊工作, 文献[125]中的方案实现了“all in one”安全性, 该安全性能够同时推导出 IND-CCA 和 UF-CCA 安全性。对于公开验证性, 除了文献[119]的 LAEPuV 方案和文献[121]的 LEPCoV 方案外, 绝大多数同态认证加密方案都未实现解密和验证操作的分离, 因此它们都不支持计算结果的公开验证。对于均摊意义的高效验证性, 除了针对所有函数的同态认证加密方案[84,114,125]外, 文献[116-117,113]中针对多项式函数的方案也需要均摊操作。可见, 计算函数越复杂, 验证的实际开销也随之增加。对于证明规模, 除了文献[84, 114]中基于 FHE 的全同态认证加密方案外, 绝大多数方案都实现了常数级别的证明规模。

3.2 基于上下文隐藏的同态签名的可验证计算

1993 年, Desmedt[126]第一次提出同态签名的概念, 之后 Johnson 等人[127]给出了它的形式化定义。利用同态签名, 客户端首先对数据进行签名, 然后将原始数据和对应的签名外包给服务器存储。与使用同态 MAC 一样, 客户端之后可以请求服务器对这些数据进行计算, 服务器完成计算后返回结果及对应的签名, 该签名可以被公开验证以判断结果是否正确。

在文献[128]中, Boneh 和 Freeman 基于小整数解(Small integer solution, SIS)假设, 构造了第一个能够支持二元域$\mathbb{F}_2$上线性操作的同态签名方案。此外, 该工作还首次考虑了同态签名的隐私性, 形式化地定义了弱上下文隐藏性(Weakly context hiding), 它要求同态操作得到的派生签名不泄露原始输入数据的信息。

为进一步介绍上下文隐藏性, 首先给出文献[128]中线性同态签名方案的定义。

定义 7.线性同态签名方案。假设 R 是一个主理想域。一个定义在 R 上的线性同态签名方案 **LHS** 由以下四个概率多项式时间的算法组成:

- $Setup(n, params)$: 输入安全参数 n 和公共参数 $params$, 其中 $params$ 包含空间维数

N, 以及被签名子空间的维数 k。输出公钥 pk 和私钥 sk。

- $Sign(sk, id, \boldsymbol{v})$: 输入私钥 sk, 向量 $\boldsymbol{v} \in \mathrm{R}^N$, 输出签名$\sigma$。
- $Combine\left(pk, id, \{(\alpha_i, \sigma_i)\}_{i=1}^{l}\right)$: 输入公钥$pk$, 文件标识符 id(用于标识同属一个文件或数据集的向量), 以及元组集合$\{(\alpha_i, \sigma_i)\}_{i=1}^{l}$, 其中 $\alpha_i \in \mathrm{R}$, 输出派生签名σ(σ即为$\sum_{i=1}^{l}\alpha_i \boldsymbol{v}_i$对应的签名)。
- $Verify(pk, id, \boldsymbol{y}, \sigma)$: 输入公钥 pk, 标识符 $id \in \{0,1\}^n$, 向量 $\boldsymbol{y} \in \mathrm{R}^N$, 以及签名$\sigma$, 输出 0(拒绝)或者 1(接受)。

对于每一对密钥$(pk, sk) \leftarrow Setup(n, params)$, **LHS** 要求满足:

1) 对于所有标识符 $id \in \{0,1\}^n$以及向量 $\boldsymbol{y} \in \mathrm{R}^N$, 如果$\sigma \leftarrow Sign(sk, id, \boldsymbol{y})$, 则

$$Verify(pk, id, \boldsymbol{y}, \sigma) = 1;$$

2) 对于所有标识符 $id \in \{0,1\}^n$以及三元组集合$\{(\alpha_i, \sigma_i, \boldsymbol{v}_i)\}_{i=1}^{l}$, 如果对于所有的下标 i, $Verify(pk, id, v_i, \sigma_i)=1$ 成立, 则

$$Verify\left(pk, id, \sum_{i=1}^{l}\alpha_i \boldsymbol{v}_i, Combine\left(pk, id, \{(\alpha_i, \sigma_i)\}_{i=1}^{l}\right)\right) = 1。$$

一个线性同态签名方案需要满足最基本的安全性要求, 即不可伪造性: 任意概率多项式时间的敌手都不能伪造一个有效签名。在安全性模型中, 敌手能够对自己选择的文件向量进行签名询问。因此, 根据敌手最终给出的签名对$(\boldsymbol{y}^*, \sigma^*)$是否被询问过, 伪造可分为两种类型:

1) 伪造类型 I: $(\boldsymbol{y}^*, \sigma^*)$能够正确验证未向签名者查询过的文件;

2) 伪造类型 II: $(\boldsymbol{y}^*, \sigma^*)$能够正确验证向签名者询问过的文件, 但是$\boldsymbol{y}^*$并不是被询问文件向量的线性组合的结果。

给定R^N上向量集$\boldsymbol{v}_1, ..., \boldsymbol{v}_k$的签名, 线性同态签名的弱上下文隐藏性要求同态操作得到的关于向量 $\boldsymbol{v}$ 的派生签名不会泄露任何关于$\boldsymbol{v}_1, ..., \boldsymbol{v}_k$的信息。由挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 交互进行的下述实验刻画了弱上下文隐藏性:

初始化阶段 $\mathcal{C}$ 执行$(pk, sk) \leftarrow Setup(n, params)$, 将得到公钥 pk 给 $\mathcal{A}$, 私钥 sk 自己保存。

挑战阶段 敌手 $\mathcal{A}$ 输出$(V_0, V_1, f_1, ..., f_s)$, 其中$V_0$和$V_1$作为$\mathrm{R}^N$上的线性空间, 分别代表 k 元组向量$(\boldsymbol{v}_1^{(b)}, ..., \boldsymbol{v}_k^{(b)}), b = 0, 1$。函数$f_1, ..., f_s$是定义在$(\mathrm{R}^N)^k$上的 R-线性函数, 对于所有的 $i = 1, ..., s$, 满足

$$f_i\left(\boldsymbol{v}_1^{(0)}, ..., \boldsymbol{v}_k^{(0)}\right) = f_i\left(\boldsymbol{v}_1^{(1)}, ..., \boldsymbol{v}_k^{(1)}\right)。$$

作为回复, 挑战者 $\mathcal{C}$ 生成一个随机比特 $b \in \{0,1\}$, 以及一个随机标签$\tau \in \{0,1\}^n$, 作为向量空间V_b的签名。接下来, 对于 $i = 1, ..., s$, $\mathcal{C}$ 利用 $Combine$ 算法对$f_i\left(\boldsymbol{v}_1^{(b)}, ..., \boldsymbol{v}_k^{(b)}\right)$进行计算, 得到衍生签名$\sigma_i$, 然后将$\sigma_1$,...,$\sigma_s$发送给 $\mathcal{A}$。在上述过程中, 函数$f_1, ..., f_s$可以在V_0,V_1输出后再适应性地输出。

输出阶段 敌手 $\mathcal{A}$ 输出比特 b'。

当$b' = b$ 时, 敌手 $\mathcal{A}$ 赢得实验。$\mathcal{A}$ 成功的优势即为它赢得实验的概率。

定义 8.弱上下文隐藏性。对于一个定义在 R 上的线性同态签名方案 **LHS**, 如果任意概率多项式时间的敌手在上述实验中成功的优势都可忽略, 则该方案满足弱上下文隐藏性。

需要说明的是, 文献[128]定义的上下文隐藏性之所以是“弱”版本, 是由于挑战者并未向敌手公开原始签名。之后, 还有一些工作针对同态签名, 提出了更强的隐私性定义, 后面再作具体介绍。

对于文献[128]构造的线性同态签名方案, 其隐私性依赖于离散高斯分布的一项研究结论: 对离散高斯样本求和后的分布与离散高斯分布统计接近, 该分布只依赖于求和结果, 与独立的样本无关。根据上述结论可知, 利用文献[128]的方案计算得到的 k 个签名的线性组合σ, 本身即为一个分布中取样的短向量, 而该分布仅与线性函数及函数输出 $\boldsymbol{v}$ 有关, 与函数的输入$\boldsymbol{v}_1, ..., \boldsymbol{v}_k$无关, 从而证明方案满足隐私性要求。

第一个支持多项式函数操作的同态签名方案由 Boneh 和 Freeman[129]于 2011 年提出, 该方案基于理想格实现。利用同样的构造思想, 他们还基于 SIS 假设, 给出了一个支持$\mathbb{F}_2$上线性操作的同态签名方案。与文献[128]中的线性同态签名方案相比, 该方案能够支持多项式规模的数据集操作, 而前者仅适用于常数规模的数据集。对于隐私性, 文献[119]中线性版本的同态签名方案满足弱上下文隐藏性, 而多项式版本的方案则无法保证隐私性。Boneh 和 Freeman 将构造具有隐私性的多项式同态签名方案作为一个公开问题提出。

Ahn 等人[130]考虑派生签名和原始签名之间的不可链接性(Unlinkability), 针对同态签名定义了更强的隐私性——强(Strong)上下文隐藏性。在文献[130]中, Ahn 等人基于 P-同态签名, 提出了一种计算认证数据的通用框架, 该框架覆盖了包括同态签名、可引用签名(Quotable signature)、可修订签名(Redactable signature)等在内的多个概念。利用 P-同态签名, 在满

足 $\mathrm{P}(m, m') = 1$ 的情况下, 任何人都可以根据 m 的签名派生得到 m'的签名, 其中谓词 P 刻画了 m 和 m'之间的“可认证关系(Authenticatable relationship)”。

对于 P-同态签名的隐私性, Ahn 等人定义了强上下文隐藏性: 利用诚实生成的原始签名得到的 m'的派生签名, 与 m'的新鲜签名统计不可区分。这就意味着, m'的派生签名与一个独立于 m 的签名统计不可区分。因此, 除了 m'外, 派生签名不会泄露任何关于原始数据 m 的信息。此外, 该性质要求即使公开 m 的原始签名, 上述结论依然成立。可见, 该定义强于文献中[129]弱上下文隐藏的定义。在文献[130]中, Ahn 等人针对任意的单变量封闭的谓词, 提出了计算同态签名的通用方法, 但是该方法并不高效; 此外, 他们还针对子串引用、子集谓词、以及加权平均值三类计算问题, 分别给出了高效的构造。无论是通用构造还是具体构造, 都满足强上下文隐藏性。

之后, Attrapadung 等人[131]对文献[130]的强上下文隐藏性定义进行进一步分析后指出, 该定义成立的隐含前提条件是原始签名是利用签名算法诚实生成的。但是, 对于一些签名方案(比如文献[132-134]), 存在签名不是诚实生成的, 但是却能通过验证的情况。因此, 当原始签名由敌手选择时(即重随机化后的原始签名), 文献[130]中的强上下文隐藏定义将无法保证派生签名与原始签名之间的不可链接性。Attrapadung 等人将该特殊情况纳入考虑, 针对同态签名提出了更强的隐私性定义——完全(completely)上下文隐藏性。该定义描述了在公开签名私钥 sk 的情况下, 利用 m 的签名σ_m派生得到的 m'的签名, 与 m'的新鲜签名统计不可区分, 其中σ_m仅要求能够通过验证即可。文献[131]利用标准模型下的双系统技术[135], 构造了一个新的线性同态签名方案, 该方案满足完全上下文隐藏性。

Catalano 等人[136]利用非对称可编程哈希函数(Asymmetric programmable hash functions, APHFs), 在标准模型下构造了第一个短公钥的线性同态签名方案, 该方案同样满足隐私性要求。在这之前的同态签名方案, 无论适用于线性函数[133,137]、多项式函数[118,129]还是多项式深度的任意电路[138], 其公钥规模与数据集的大小至少成线性关系。但是, 文献[136]中线性同态签名方案的公钥规模仅与数据集的大小 N 和被签名向量的维度 T 成亚线性关系, 即 $O(\sqrt{\mathrm{N}} + \sqrt{\mathrm{T}})$。对于方案的隐私性, Catalano 等人针对多标签程序(Multi-labeled program)给出了一个基于模拟的(Simulation-based)上下文隐藏性定义。该定义要求, 在只给定多标签程序$\mathcal{P}_{\Delta}$以及它的输出 m 和签名私钥 sk, 但是不公开$\mathcal{P}_{\Delta}$的输入的条件下, 可以模拟生成 m 的签名σ。对于任意的区分算法 $\mathcal{D}$, 在给定公私钥对(sk, vk)、多标签程序$\mathcal{P}_{\Delta}$以及原始签名的情况下, 该模拟签名与同态操作得到的签名统计不可区分。该定义强于 Boneh 和 Freeman[129]的弱上下文隐藏性, 但是弱于 Ahn 等人[130]的强上下文隐藏性。

2015 年, Gorbunov 等人[138]基于标准格中的 SIS 假设, 创新性地构造出了第一个支持多项式深度的任意电路操作的层次型全同态签名方案。为了实现方案的隐私性, 文献[138]首先构造了一个满足上下文隐藏性的同态陷门函数(Homomorphic trapdoor function, HTDF), 然后对该组件进行转换, 得到一个满足上下文隐藏性的层次型全同态签名方案。但是, 为了实现上下文隐藏性, 该方案需限制消息空间为$\{0,1\}$。

上述方案都是针对单用户签名情形的, Schabhüser 等人[139]考虑签名数据来源于多个用户, 定义了多密钥情形下的上下文隐藏性。根据敌手是否为参与计算的用户, 可分为内部敌手和外部敌手, 分别对应两种隐私性定义: 内部(internally)上下文隐藏性和外部(externally)上下文隐藏性, 前者的安全性强于后者。在文献[139]中, Schabhüser 等人构造了第一个满足(内部)上下文隐藏性的多密钥线性同态签名方案。

在表 4 中, 本文对同态签名不同隐私性的定义进行了总结对比, 包括定义名称、定义描述、定义特征以及相关工作。对于单用户(或单密钥)的同态签名, 其隐私性定义包括弱上下文隐藏性、强上下文隐藏性、完全上下文隐藏性和基于模拟的上下文隐藏性。与最后一种定义不同, 前三种隐私性都是基于不可区分性(Indistinguishability-based)定义的。这几种隐私性的强度排序: 弱上下文隐藏性<基于模拟的上下文隐藏性<强上下文隐藏性<完全上下文隐藏性。需要注意的是, 基于模拟的上下文隐藏性的实现需要额外的隐藏算法。多用户(或多密钥)同态签名的隐私性分为内部上下文隐藏性和外部上下文隐藏性, 分别对应于掌握部分用户私钥和输入的内部敌手, 以及不掌握任何用户私钥和输入的外部敌手。根据敌手能力可知, 内部上下文隐藏性的定义更强。

4 讨论与展望

本文梳理的具有隐私保护的可验证计算方案满足两个共性: 首先, 计算结果可验证, 保证其正确性; 其次, 计算过程不泄露客户端的敏感数据, 实现输入数据隐私性和(或)计算结果保密性。除了这两个基础共性外, 由于侧重点和技术手段的不同, 不同方案呈现出了不同的特点及优劣势。为了全面准确地把握现有研究的不足和欠缺, 以更好地对未来工作进行展望, 需要综合考虑以下三个方面的相关因素。

表 4 同态签名的隐私性定义对比

Table 4 Comparison of privacy definitions of homomorphic signature

用户数量	同态签名的隐私性定义	定义描述	定义特征	相关工作
单个	弱上下文隐藏性	利用诚实生成的原始签名σ_m得到的派生签名$\sigma_{m'}$, 不泄露原始输入 m 的信息。	原始签名σ_m不能向敌手公开。	Boneh et al.[128] Boneh et al.[129]
	强上下文隐藏性	利用诚实生成的原始签名σ_m得到的派生签名$\sigma_{m'}$, 与 m'的新鲜签名统计不可区分。	原始签名σ_m可以向敌手公开, 但是必须由签名算法诚实生成。	Ahn et al.[130]
	完全上下文隐藏性	在公开签名私钥 sk 的情况下, 利用 m 的签名σ_m派生得到的 m'的签名, 与 m'的新鲜签名统计不可区分, 其中σ_m仅要求能够通过验证即可。	原始签名σ_m仅要求能够通过验证即可。该定义能够保证派生签名与原始签名之间的不可链接性。	Attrapadung et al.[131]
	基于模拟的上下文隐藏性	在只给定多标签程序$\mathcal{P}_\Delta$以及它的输出 m'和签名私钥 sk, 但是不公开$\mathcal{P}_\Delta$输入的条件下, 可以模拟生成 m'的签名$\sigma_{m'}$, 该模拟签名与同态操作得到的派生签名统计不可区分。	上下文隐藏性的实现需要额外的隐藏算法。	Catalano et al.[136] Gorbunov et al.[138]
多个	内部上下文隐藏性	对于知道部分用户私钥和输入的内部敌手, 派生签名不会泄露其余用户的输入信息。	内部上下文隐藏性强于外部上下文隐藏性。	Schabhüser et al.[139]
	外部上下文隐藏性	对于不知道任何用户私钥和输入的外部敌手, 派生签名不会泄露任何用户的输入信息。		

(注: m 表示原始输入数据; σ_m表示 m 对应的签名; m'表示对 m 进行计算后的输出结果; $\sigma_{m'}$表示 m'对应的签名, 是通过对σ_m进行同态操作所得到的派生签名)

在性能方面, 主要考虑方案是否需要预处理操作、验证效率是否建立在均摊意义上, 以及验证开销、通信开销、服务器端的计算量、服务器返回的证明大小等。比如, 基于全同态加密实现隐私性的 VC 方案要求客户端执行一个时间复杂度与计算函数本身相当的预处理过程, 因此外包计算的效率实际是建立在对同一个函数执行多次操作的均摊意义上的, 这个弊端严重削弱了方案的实用性。又比如, 由于同态 MAC/签名要求同态操作后的标记/签名满足紧致性(succinctness), 因此对于采用这两种工具实现的方案, 服务器返回的证明一般都较小。

在功能方面, 需要考虑方案是否支持公开代理或公开验证, 是否要求服务器之间进行通信或者允许服务器之间合谋、验证过程是否需要原始数据参与, 以及方案支持的函数类型等。公开代理性能够提升多用户摊销模型下协议执行的效率, 公开验证性不仅可以有效预防各参与方单方面的恶意抵赖, 还允许用户将验证过程委托给第三方, 进一步节约开销。对于多服务器情形, 如果方案的运行要求服务器之间进行通信, 这不仅增加了服务器的通信和计算开销, 同时也增加了各服务器间合谋的风险, 可能导致客户端输入数据的恢复和泄露。此外, 方案支持的函数类型也是一个关键因素, 其直接决定了方案的应用范围。

在安全性方面, 需要考虑方案是否能够抵抗“拒绝问题”、是否允许敌手进行验证询问或解密询问、敌手是半诚实型还是恶意型, 安全性的实现是基于标准模型还是随机预言机模型, 以及相应的困难假设是标准的还是非标准的。对于不能抵抗“拒绝问题”的方案, 云服务器一旦得知用户是否接受了计算结果, 用户则只能重新运行昂贵的预处理过程, 生成新的密钥来保证方案的安全。此外, 敌手能力、方案的安全模型、困难问题的难易程度等因素都与方案的安全强度紧密相关, 决定了方案在实际应用时的健壮性和可靠性。

通过衡量现有工作在上述三个方面的优势及不足, 本文提出几个有待进一步研究的方向:

1)具有结果保密性的公开可验证外包计算

从密码安全理论的角度看, 公开可验证计算的验证算法仅用于判断计算结果正确性, 而无需输出结果本身; 从实际应用的角度看, 计算结果可能包含用户的私密信息, 为保护用户隐私, 应避免将其公开。因此, 公开可验证的安全性应该保证结果不被泄露。然而, 现有了大多数方案都无法同时满足公开验证性和结果保密性。Parno 等人[40]首先提出了可公开验证计算, 并且开启了以 ABE 作为中间组件构造 VC 方案的思路。但是, 无论是 ABE, 还是更一般化形式的 FE, 都不能直接保证计算结果的隐私性。对

于利用 FHE 或 HE 实现的可验证计算方案(比如文献[35,37,39,56,82,85,86]等), 由于计算结果是密文形式, 因此能够保证真实结果的隐私性。然而, 验证过程需要进行解密操作, 故无法公开验证。目前, 仅有 Catalano 等人[120]提出的可公开验证的线性同态认证加密能够对密文结果进行公开验证。但是, 相应的构造方案也仅有文献[119]的 LAEPuV 方案和文献[121]的 LEPCoV 方案, 并且两个方案都只适用于线性函数, 前者还被证明验证算法会给出错误的否定结果。因此, 对于可验证计算而言, 结果保密性和公开验证性之间的共存问题是一个研究基础薄弱, 但值得深入探讨的重要问题。

2)兼顾安全强度(比如强不可伪造性)和函数范围(多项式函数及以上)要求的具有隐私保护的同态 MAC

对于数据外包模式的可验证计算, 如果方案仅支持明文操作, 意味着客户端只能以明文形式将大量数据外包存储在云端, 在缺乏监督管理的情形下, 这种操作隐藏了很大的风险。因此, 外包密文数据将是实际应用中的主流选择。同态认证加密(HAE)是实现密文外包的可验证计算的有效工具, 该工具的一种典型构造方法是采用“Encrypt-and-MAC”组合模式 , 将同态加密与具有隐私保护的同态 MAC 相结合。具有隐私保护的同态 MAC 要求标记不仅能对消息提供完整性保护, 同时还不能泄露被认证消息的信息。目前, 仅有几个方案同时满足上述两个要求, 并且还存在待改进之处。Gennaro 等人[84]构造的全同态 MAC 方案需要基于 FHE 实现, 实用性受限, 并且方案不支持敌手进行验证询问, 仅实现了弱不可伪造性。Lai 等人[114]构造了一个强不可伪造性的 HEA 方案, 但是方案仅支持线性函数操作。Li 等人[116-117]基于多线性映射, 构造了一个针对多项式函数且满足强不可伪造性的 PHMAC 方案。相比前两个工作, 该工作在函数范围和安全强度的兼具性方面表现更好, 但是多线性映射的存在性尚未得到理论证实。因此, 为了较好地实现针对密文外包数据的可验证计算, 有必要对具有隐私保护的同态 MAC 这个基础工具进行深入研究, 使其在实现隐私性的同时, 能够兼顾现实场景对方案的安全强度、函数范围等的要求。

3)支持更广函数类型(比如二次函数、多项式函数、矩阵乘积等)的上下文隐藏的同态签名

作为一种实现数据外包模式的可验证计算的有效工具, 同态签名允许外包数据与计算函数相互分离, 并且支持任何人对计算结果进行验证。为了保证同态计算后的签名不泄露原始输入信息, 从而实现公开验证过程对于客户端原始数据的隐藏, 一些工作专门考虑了同态签名的隐私性, 提出了不同强度的隐私性定义, 比如单客户端情形下的弱上下文隐藏性[128-129]、强上下文隐藏性[130]、完全上下文隐藏性[131]、基于模拟的上下文隐藏性[136,138], 以及多客户端情形的内部上下文隐藏性和外包上下文隐藏性[139]等。从理论研究的角度看, 这些定义充分挖掘了同态签名的隐私性的内涵, 丰富了同态签名的安全层级。但是, 就方案构造而言, 目前实现隐私性的同态签名方案大多仅支持线性函数操作, 比如文献[128,131, 136, 139]。作为例外, Gorbunov 等人[138]构造的层次型全同态签名方案能够支持任意函数操作, 但是为了实现隐私性, 该方案需限定消息空间为比特。由此可见, 现有的上下文隐藏的同态签名方案还不够健壮, 实际适用范围还较窄。因此, 考虑针对更多更广的函数类型(比如二次函数、单变量多项式、多变量多项式、矩阵乘积等), 实现具有上下文隐私性的同态签名方案, 是未来需要深入探索的另一重要方向。

5 总结

具有隐私保护的可验证计算能够解决外包场景中的两大安全问题——计算结果不可信和用户隐私数据泄露, 是实现安全可靠的外包计算的有效手段。本文对现有相关工作进行分析和梳理, 分别讨论了计算外包模式和数据外包模式下具有隐私保护的可验证计算。考虑到云计算场景对于数据隐私保护的迫切需求, 本文基于对现有工作的总结和思考, 提出了具有隐私保护的可验证计算未来的几个重要研究方向, 希望能够提供一定的参考。

参考文献

[1] Feng D G, Zhang M, Zhang Y, et al. Study on Cloud Computing Security[J]. *Journal of Software*, 2011, 22(1): 71-83.
(冯登国, 张敏, 张妍, 等. 云计算安全研究[J]. *软件学报*, 2011, 22(1): 71-83.)

[2] Chen Q, Deng Q N. Cloud Computing and Its Key Techniques[J]. *Journal of Computer Applications*, 2009, 29(9): 2562-2567.
(陈全, 邓倩妮. 云计算及其关键技术[J]. *计算机应用*, 2009, 29(9): 2562-2567.)

[3] Alibaba Cloud. https://www.aliyun.com. 2009.
(阿里云. https://www.aliyun.com. 2009.)

[4] Huawei Cloud. https://www.huaweicloud.com. 2005.
(华为云. https://www.huaweicloud.com. 2005.)

[5] Baidu AI Cloud. https://cloud.baidu.com/cloudai.html. 2015.
(百度智能云. https://cloud.baidu.com/cloudai.html. 2015.)

[6] Tencent Cloud. https://cloud.tencent.com. 2010.
(腾讯云. https://cloud.tencent.com/. 2010.)

[7] Microsoft Azure (in Chinese). http://www.windowsazure.cn/zh-cn. 2012.

[8] Google Cloud (in Chinese). https://search.njau.cf/extdomains/cloud.google.com/?hl=zh-cn&. 2012.

[9] Amazon Web Services (in Chinese). https://docs.amazonaws.cn. 2014.

[10] IBM Cloud (in Chinese). https://www.ibm.com/cn-zh/cloud. 2014.

[11] Hu X, Pei D Y, Tang C M, et al. Verifiable and Secure Outsourcing of Matrix Calculation and Its Application[J]. *Scientia Sinica* (*Informationis*), 2013, 43(7): 842-852.
(胡杏, 裴定一, 唐春明, 等. 可验证安全外包矩阵计算及其应用[J]. *中国科学: 信息科学*, 2013, 43(7): 842-852.)

[12] Castro M, Liskov B. Practical Byzantine Fault Tolerance and Proactive Recovery[J]. *ACM Transactions on Computer Systems*, 2002, 20(4): 398-461.

[13] Parno B, McCune J M, Perrig A. Bootstrapping Trust in Modern Computers[M]. New York, NY: Springer New York, 2011.

[14] Monrose F, Wyckoff P, Rubin A D. Distributed execution with remote audit[C]. *Network and Distributed System Security Symposium*, 1999: 103-113.

[15] Belenkiy M, Chase M, Erway C C, et al. Incentivizing Outsourced Computation[C]. *The 3rd international workshop on Economics of networked systems*, 2008: 85-90.

[16] Smith S W, Weingart S. Building a High-Performance, Programmable Secure Coprocessor[J]. *Computer Networks*, 1999, 31(8): 831-860.

[17] Yee B, Tygar J D. Secure Coprocessors in Electronic Commerce Applications[C]. *The 1st conference on USENIX Workshop on Electronic Commerce - Volume 1*, 1995: 14.

[18] Goldwasser S, Micali S, Rackoff C. The Knowledge Complexity of Interactive Proof-Systems[C]. *The seventeenth annual ACM symposium on Theory of computing*, 1985: 291-304.

[19] Babai L. Trading group theory for randomness[C]. *STOC*, 1985: 421-429.

[20] Goldwasser S, Kalai Y T, Rothblum G N. Delegating Computation: Interactive Proofs for Muggles[J]. *Journal of the ACM*, 62(4): 27.

[21] Cormode G, Mitzenmacher M, Thaler J. Practical Verified Computation with Streaming Interactive Proofs[J]. *ArXiv e-Prints*, 2011: arXiv: 1105. 2003.

[22] Fortnow L, Rompel J, Sipser M. On the Power of Multi-Power Interactive Protocols[C]. *[1988] Proceedings Structure in Complexity Theory Third Annual Conference*, 1988: 156-161.

[23] Arora S, Safra S. Probabilistic checkable proof: A new characterization of NP[J]. *Journal of ACM*, 1998, 45:70-122.

[24] Gennaro R, Gentry C, Parno B, et al. Quadratic Span Programs and Succinct NIZKs without PCPS[M]. Advances in Cryptology - EUROCRYPT 2013. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 626-645.

[25] Parno B, Howell J, Gentry C, et al. Pinocchio: Nearly Practical Verifiable Computation[C]. *2013 IEEE Symposium on Security and Privacy*, 2013: 238-252.

[26] Ben-Sasson E, Chiesa A, Genkin D, et al. SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge[C]. *Annual Cryptology Conference*, 2013: 90-108.

[27] Chaum D, Pedersen T P. Wallet Databases with Observers[M]. Advances in Cryptology — CRYPTO' 92. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 89-105.

[28] Xue R, Wu Y, Liu M H, et al. Progress in Verifiable Computation[J]. *Scientia Sinica* (*Informationis*), 2015, 45(11): 1370-1388.
(薛锐, 吴迎, 刘牧华, 等. 可验证计算研究进展[J]. *中国科学(信息科学)*, 2015, 45(11): 1370-1388.)

[29] Chow R, Golle P, Jakobsson M, et al. Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control[C]. *The 2009 ACM workshop on Cloud computing security*, 2009: 85-90.

[30] Ren K, Wang C, Wang Q. Security Challenges for the Public Cloud[J]. *IEEE Internet Computing*, 2012, 16(1): 69-73.

[31] Cloud Security Alliance. The notorious nine: Could computing top threats in 2013[R]. CSA, 2013.

[32] Bösch C, Hartel P, Jonker W, et al. A Survey of Provably Secure Searchable Encryption[J]. *ACM Computing Surveys*, 47(2): 18.

[33] Ostrovsky R, Sahai A, Waters B. Attribute-Based Encryption with Non-Monotonic Access Structures[C]. *The 14th ACM conference on Computer and communications security*, 2007: 195-203.

[34] Lewko A, Okamoto T, Sahai A, et al. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption[C]. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2010: 62-91.

[35] Gennaro R, Gentry C, Parno B. Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers[C]. *Annual Cryptology Conference*, 2010: 465-482.

[36] Bellare M, Hoang V T, Rogaway P. Foundations of Garbled Circuits[C]. *The 2012 ACM conference on Computer and communications security*, 2012: 784-796.

[37] Chung K M, Kalai Y, Vadhan S. Improved Delegation of Computation Using Fully Homomorphic Encryption[C]. *The 30th annual conference on Advances in cryptology*, 2010: 483-501.

[38] Barak B, Goldreich O. Universal Arguments and Their Applications[C]. *Proceedings 17th IEEE Annual Conference on Computational Complexity*, 2002: 194-203.

[39] Barbosa M, Farshim P. Delegatable Homomorphic Encryption with Applications to Secure Outsourcing of Computation[C]. *Cryptographers' Track at the RSA Conference*, 2012: 296-312.

[40] Parno B, Raykova M, Vaikuntanathan V. How to Delegate and Verify in Public: Verifiable Computation from Attribute-Based Encryption[C]. *Theory of Cryptography Conference*, 2012: 422-439.

[41] Sun J M, Zhu B R, Qin J, et al. Confidentiality-Preserving Publicly Verifiable Computation Schemes for Polynomial Evaluation and Matrix-Vector Multiplication[J]. *Security and Communication Networks*, 2018, 2018: 5275132.

[42] Wu Y, Liu M H, Xue R, et al. Attribute-Based Multi-Function Verifiable Computation[J]. *Future Generation Computer Systems*, 2018, 78(P3): 995-1004.

[43] Alderman J, Janson C, Cid C, et al. Access Control in Publicly Verifiable Outsourced Computation[C]. *The 10th ACM Symposium on Information, Computer and Communications Security*, 2015: 657-662.

[44] Yang H N, Sun J M, Qin J, et al. An Improved Scheme for Outsourced Computation with Attribute-Based Encryption[J]. *Concurrency and Computation: Practice and Experience*, 2019, 31(21): e4833.

[45] Alderman J, Janson C, Cid C, et al. Hybrid Publicly Verifiable Computation[C]. *Cryptographers' Track at the RSA Conference*, 2016: 147-163.

[46] Dagher G G, Fung B C M, Mohammed N, et al. SecDM:

Privacy-Preserving Data Outsourcing Framework with Differential Privacy[J]. *Knowledge and Information Systems*, 2020, 62(5): 1923-1960.

[47] Meng F, Cheng L X, Wang M Q. Ciphertext-Policy Attribute-Based Encryption with Hidden Sensitive Policy from Keyword Search Techniques in Smart City[J]. *EURASIP Journal on Wireless Communications and Networking*, 2021, 2021(1): 20.

[48] Luo R, Yao Y Z, Li W H, et al. PTAC: Privacy-Preserving Time and Attribute Factors Combined Cloud Data Access Control with Computation Outsourcing[C]. *International Conference on Artificial Intelligence and Security*, 2022: 540-555.

[49] Goldwasser S, Kalai Y, Popa R A, et al. Reusable Garbled Circuits and Succinct Functional Encryption[C]. *The forty-fifth annual ACM symposium on Theory of Computing*, 2013: 555-564.

[50] Garg S, Gentry C, Halevi S, et al. Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits[C]. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 2013: 40-49.

[51] Waters B. A Punctured Programming Approach to Adaptively Secure Functional Encryption[C]. *Advances in Cryptology -- CRYPTO 2015*: 678-697.

[52] Gorbunov S, Vaikuntanathan V, Wee H. Predicate Encryption for Circuits from LWE[C]. *Advances in Cryptology -- CRYPTO 2015*: 503-523.

[53] Brakerski Z, Segev G. Function-Private Functional Encryption in the Private-Key Setting[J]. *Journal of Cryptology*, 2018, 31(1): 202-225.

[54] Kitagawa F, Nishimaki R, Tanaka K. Simple and Generic Constructions of Succinct Functional Encryption[J]. *Journal of Cryptology*, 2021, 34(3): 25.

[55] Garg R, Goyal R, Lu G, et al. Dynamic Collusion Bounded Functional Encryption from Identity-Based Encryption[C]. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2022: 736-763.

[56] Choi S G, Katz J, Kumaresan R, et al. Multi-Client Non-Interactive Verifiable Computation[C]. *Theory of Cryptography Conference*, 2013: 499-518.

[57] Naor M, Pinkas B, Sumner R. Privacy Preserving Auctions and Mechanism Design[C]. *The 1st ACM conference on Electronic commerce*, 1999: 129-139.

[58] Goldwasser S, Gordon S D, Goyal V, et al. Multi-Input Functional Encryption[M]. Advances in Cryptology - EUROCRYPT 2014 Berlin Heidelberg, 2014: 578-602.

[59] Boneh D, Lewi K, Raykova M, et al. Semantically Secure Order-Revealing Encryption: Multi-Input Functional Encryption without Obfuscation[C]. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2015: 563-594.

[60] Badrinarayanan S, Gupta D, Jain A, et al. Multi-Input Functional Encryption for Unbounded Arity Functions[C]. *Proceedings, Part I, of the 21st International Conference on Advances in Cryptology -- ASIACRYPT 2015 - Volume 9452*, 2015: 27-51.

[61] Li P L, Xu H X, Ji Y Y. Multi-Input Functional Encryption and Its Application in Outsourcing Computation[C]. *International Conference on Information and Communications Security*, 2016: 220-235.

[62] Abdalla M, Gay R, Raykova M, et al. Multi-Input Inner-Product Functional Encryption from Pairings[C]. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2017: 601-626.

[63] Attrapadung N, Hanaoka G, Hirano T, et al. Token-Based Multi-Input Functional Encryption[C]. *International Conference on Provable Security*, 2018: 147-164.

[64] Tomida J. Tightly Secure Inner Product Functional Encryption: Multi-input and Function-Hiding Constructions[C]. *ASIACRYPT,* 2019: 459-488.

[65] Abdalla M, Bourse F, Marival H, et al. Multi-Client Inner-Product Functional Encryption in the Random-Oracle Model[C]. *Security and Cryptography for Networks*: *12th International Conference*, 2020: 525-545.

[66] Ciampi M, Siniscalchi L, Waldner H. Multi-Client Functional Encryption for Separable Functions[C]. *IACR International Conference on Public-Key Cryptography*, 2021: 724-753.

[67] Lee K, Seo M. Functional Encryption for Set Intersection in the Multi-Client Setting[J]. *Designs*, *Codes*, *and Cryptography*, 2022, 90(1): 17-47.

[68] Gordon S D, Katz J, Liu F H, et al. Multi-Client Verifiable Computation with Stronger Security Guarantees[C]. *Theory of Cryptography Conference*, 2015: 144-168.

[69] Xu S. An Efficient HPRA-Based Multiclient Verifiable Computation: Transform and Instantiation[J]. *Security and Communication Networks*, 2021: 6612614.

[70] Xu S, Zhang L. A homomorphic proxy re-authenticators based efficient multi-client non-interactive verifiable computation scheme[C]. *International Conference on Information Systems Security and Privacy*, 2020: 195-206.

[71] Derler D, Ramacher S, Slamanig D. Homomorphic Proxy re-Authenticators and Applications to Verifiable Multi-User Data Aggregation[C]. *International Conference on Financial Cryptography and Data Security*, 2017: 124-142.

[72] Benabbas S, Gennaro R, Vahlis Y. Verifiable Delegation of Computation over Large Datasets[C]. *Annual Cryptology Conference*, 2011: 111-131.

[73] He D B, Ma M M, Zeadally S, et al. Certificateless Public Key Authenticated Encryption with Keyword Search for Industrial Internet of Things[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(8): 3618-3627.

[74] Cai C J, Weng J, Yuan X L, et al. Enabling Reliable Keyword Search in Encrypted Decentralized Storage with Fairness[J]. *IEEE Transactions on Dependable and Secure Computing*, 2021, 18(1): 131-144.

[75] Naor M, Rothblum G N. The Complexity of Online Memory Checking[C]. *46th Annual IEEE Symposium on Foundations of Computer Science*, 2005: 573-582.

[76] Juels A, Kaliski B S Jr. Pors: Proofs of Retrievability for Large Files[C]. *The 14th ACM conference on Computer and communications security*, 2007: 584-597.

[77] Zhang L F, Safavi-Naini R. Private Outsourcing of Polynomial Evaluation and Matrix Multiplication Using Multilinear Maps[C]. *International Conference on Cryptology and Network Security*, 2013: 329-348.

[78] Zhang L F, Safavi-Naini R. Protecting Data Privacy in Publicly

Verifiable Delegation of Matrix and Polynomial Functions[J]. *Designs, Codes and Cryptography*, 2020, 88(4): 677-709.

[79] Garg S, Gentry C, Halevi S. Candidate Multilinear Maps from Ideal Lattices[M]. Advances in Cryptology - EUROCRYPT 2013 Berlin Heidelberg, 2013: 1-17.

[80] Garg S, Gentry C, Halevi S, et al. Attribute-Based Encryption for Circuits from Multilinear Maps[C]. *Annual Cryptology Conference*, 2013: 479-499.

[81] Boneh D, Goh E J, Nissim K. Evaluating 2-DNF Formulas on Ciphertexts[C]. *Theory of Cryptography Conference*, 2005: 325-341.

[82] Fiore D, Gennaro R, Pastro V. Efficiently Verifiable Computation on Encrypted Data[C]. *The 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014: 844-855.

[83] Brakerski Z, Vaikuntanathan V. Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages[C]. *The 31st annual conference on Advances in cryptology*, 2011: 505-524.

[84] Gennaro R, Wichs D. Fully Homomorphic Message Authenticators[M]. Advances in Cryptology - ASIACRYPT 2013 Berlin Heidelberg, 2013: 301-320.

[85] Fiore D, Nitulescu A, Pointcheval D. Boosting Verifiable Computation on Encrypted Data[C]. *IACR International Conference on Public-Key Cryptography*, 2020: 124-154.

[86] Bois A, Cascudo I, Fiore D, et al. Flexible and Efficient Verifiable Computation on Encrypted Data[C]. *IACR International Conference on Public-Key Cryptography*, 2021: 528-558.

[87] Abdalla M, Bourse F, De Caro A, et al. Simple Functional Encryption Schemes for Inner Products[C]. *IACR International Workshop on Public Key Cryptography*, 2015: 733-751.

[88] Datta P, Dutta R, Mukhopadhyay S. Functional Encryption for Inner Product with Full Function Privacy[C]. *Public-Key Cryptography - PKC 2016*, 2016: 164-195.

[89] Benhamouda F, Bourse F, Lipmaa H. CCA-secure inner-product functional encryption from projective hash functions[C]. *Public-Key Cryptography*, 2017: 36-66.

[90] Castagnos G, Laguillaumie F, Tucker I. Practical Fully Secure Unrestricted Inner Product Functional Encryption Modulo P[C]. *International Conference on the Theory and Application of Cryptology and Information Security*, 2018: 733-764.

[91] Barbosa M, Catalano D, Soleimanian A, et al. Efficient Function-Hiding Functional Encryption: From Inner-Products to Orthogonality[C]. *Cryptographers' Track at the RSA Conference*, 2019: 127-148.

[92] Abdalla M, Catalano D, Gay R, et al. Inner-Product Functional Encryption with Fine-Grained Access Control[C]. *International Conference on the Theory and Application of Cryptology and Information Security*, 2020: 467-497.

[93] Agrawal S, Libert B, Maitra M, et al. Adaptive Simulation Security for Inner Product Functional Encryption[C]. *IACR International Conference on Public-Key Cryptography*, 2020: 34-64.

[94] Lei X Y, Liao X F, Huang T W, et al. Achieving Security, Robust Cheating Resistance, and High-Efficiency for Outsourcing Large Matrix Multiplication Computation to a Malicious Cloud[J]. *Information Sciences*, 2014, 280: 205-217.

[95] Chen X F, Huang X Y, Li J, et al. New Algorithms for Secure Outsourcing of Large-Scale Systems of Linear Equations[J]. *IEEE Transactions on Information Forensics and Security*, 2015, 10(1): 69-78.

[96] Sheng G, Tang C M, Gao W, et al. MD-VC Matrix: An efficient scheme for publicly verifiable computation of outsourced matrix multiplication[C]. *International Conference on Network and System Security*, 2016: 349-362.

[97] Elkhiyaoui K, Önen M, Azraoui M, et al. Efficient Techniques for Publicly Verifiable Delegation of Computation[C]. *The 11th ACM on Asia Conference on Computer and Communications Security*, 2016: 119-128.

[98] Zhang X Y, Jiang T, Li K C, et al. New Publicly Verifiable Computation for Batch Matrix Multiplication[J]. *Information Sciences*, 2019, 479: 664-678.

[99] Canetti R, Riva B, Rothblum G N. Practical Delegation of Computation Using Multiple Servers[C]. *The 18th ACM conference on Computer and communications security*, 2011: 445-454.

[100] Canetti R, Riva B, Rothblum, G.N. Two protocols for delegation of Computation[C]. *ICITS*, 2012: 37-61.

[101] Ananth P, Chandran N, Goyal V, et al. Achieving Privacy in Verifiable Computation with Multiple Servers — without FHE and without Pre-Processing[C]. *The 17th International Conference on Public-Key Cryptography — PKC 2014 - Volume 8383*, 2014: 149-166.

[102] Zhang L F, Safavi-Naini R, Liu X W. Verifiable Local Computation on Distributed Data[C]. *The 2nd international workshop on Security in cloud computing*, 2014: 3-10.

[103] Parno B, Howell J, Gentry C, et al. Pinocchio: Nearly Practical Verifiable Computation[C]. *2013 IEEE Symposium on Security and Privacy*, 2013: 238-252.

[104] Schoenmakers B, Veeningen M, de Vreede N. Trinocchio: Privacy-Preserving Outsourcing by Distributed Verifiable Computation[C]. *International Conference on Applied Cryptography and Network Security*, 2016: 346-366.

[105] Zhang L F. Multi-Server Verifiable Delegation of Computations: Unconditional Security and Practical Efficiency[J]. *Information and Computation*, 2021, 281: 104740.

[106] Zhang L F, Wang H X. Multi-Server Verifiable Computation of Low-Degree Polynomials[C]. *2022 IEEE Symposium on Security and Privacy*, 2022: 596-613.

[107] Li P L, Xu H X, Ji Y Y. Multi-Client Outsourced Computation[C]. *International Conference on Information Security and Cryptology*, 2016: 397-409.

[108] Chen X, Zhang L F. Two-Server Verifiable Homomorphic Secret Sharing for High-Degree Polynomials[C]. *International Conference on Information Security*, 2020: 75-91.

[109] Boyle E, Kohl L, Scholl P. Homomorphic Secret Sharing from Lattices without FHE[M]. Advances in Cryptology - EUROCRYPT 2019 International Publishing, 2019: 3-33.

[110] Chen X, Zhang L F. Two-Server Delegation of Computation on Label-Encrypted Data[J]. *IEEE Transactions on Cloud Computing*, 2021, 9(4): 1645-1656.

[111] Barbosa M, Catalano D, Fiore D. Labeled Homomorphic Encryption[C]. *European Symposium on Research in Computer Security*, 2017: 146-166.

[112] Catalano D, Fiore D. Boosting Linearly-Homomorphic Encryption to Evaluate Degree-2 Functions on Encrypted Data[J]. *IACR Cryptology EPrint Archive*, 2014: 813.

[113] Joo C, Yun A. Homomorphic Authenticated Encryption Secure Against Chosen-Ciphertext Attack[M]. Lecture Notes in Computer Science Berlin Heidelberg, 2014: 173-192.

[114] Lai J Z, Deng R H, Pang H, et al. Verifiable Computation on Outsourced Encrypted Data[C]. *Computer Security - ESORICS 2014*: 273-291.

[115] Freeman D M. Improved Security for Linearly Homomorphic Signatures: A Generic Framework[C]. *International Workshop on Public Key Cryptography*, 2012: 697-714.

[116] Li S M, Wang X, Zhang R. Privacy-Preserving Homomorphic MACs with Efficient Verification[C]. *International Conference on Web Services*, 2018: 100-115.

[117] Li S M, Wang X, Xue R. Toward both Privacy and Efficiency of Homomorphic MACs for Polynomial Functions and Its Applications[J]. *The Computer Journal*, 2022, 65(4): 1020-1028.

[118] Catalano D, Fiore D, Warinschi B. Homomorphic Signatures with Efficient Verification for Polynomial Functions[C]. *Annual Cryptology Conference*, 2014: 371-389.

[119] Catalano D, Marcedone A, Puglisi O. Authenticating Computation on Groups: New Homomorphic Primitives and Applications[C]. *International Conference on the Theory and Application of Cryptology and Information Security*, 2014: 193-212.

[120] Catalano D, Fiore D, Warinschi B. Efficient Network Coding Signatures in the Standard Model[C]. *International Workshop on Public Key Cryptography*, 2012: 680-696.

[121] Struck P, Schabhüser L, Demirel D, et al. Linearly Homomorphic Authenticated Encryption with Provable Correctness and Public Verifiability[C]. *International Conference on Codes, Cryptology, and Information Security*, 2017: 142-160.

[122] Tran N H, Pang H, Deng R H. Efficient Verifiable Computation of Linear and Quadratic Functions over Encrypted Data[C]. *The 11th ACM on Asia Conference on Computer and Communications Security*, 2016: 605-616.

[123] Catalano D, Fiore D. Practical Homomorphic MACs for Arithmetic Circuits[M]. Advances in Cryptology - EUROCRYPT 2013 Berlin Heidelberg, 2013: 336-352.

[124] Backes M, Fiore D, Reischuk R M. Verifiable Delegation of Computation on Outsourced Data[C]. *The 2013 ACM SIGSAC conference on Computer & communications security*, 2013: 863-874.

[125] Kim J, Yun A. Secure Fully Homomorphic Authenticated Encryption[J]. *IEEE Access*, 2021, 9: 107279-107297.

[126] Desmedt Y. Computer Security by Redefining what a Computer is[C]. *NSPW '92-93: Proceedings on the 1992-1993 workshop on New security paradigms*, 1993: 160-166.

[127] Johnson R, Molnar D, Song D, et al. Homomorphic Signature Schemes[M]. Topics in Cryptology — CT-RSA 2002 Berlin Heidelberg, 2002: 244-262.

[128] Boneh D, Freeman D M. Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures[C]. International *Workshop on Public Key Cryptography*, 2011: 1-16.

[129] Boneh D, Freeman D M. Homomorphic Signatures for Polynomial Functions[M]. Advances in Cryptology - EUROCRYPT 2011 Berlin Heidelberg, 2011: 149-168.

[130] Ahn J H, Boneh D, Camenisch J, et al. Computing on Authenticated Data[J]. *Journal of Cryptology*, 2015, 28(2): 351-395.

[131] Attrapadung N, Libert B, Peters T. Computing on Authenticated Data: New Privacy Definitions and Constructions[C]. *International Conference on the Theory and Application of Cryptology and Information Security*, 2012: 367-385.

[132] Lewko A, Waters B. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts[C]. *Theory of Cryptography Conference*, 2010: 455-479.

[133] Attrapadung N, Libert B. Homomorphic Network Coding Signatures in the Standard Model[C]. *International Workshop on Public Key Cryptography*, 2011: 17-34.

[134] Gerbush M, Lewko A, O'Neill A, et al. Dual Form Signatures: An Approach for Proving Security from Static Assumptions[C]. *International Conference on the Theory and Application of Cryptology and Information Security*, 2012: 25-42.

[135] Waters B. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions[C]. *Annual International Cryptology Conference*, 2009: 619-636.

[136] Catalano D, Fiore D, Nizzardo L. Programmable Hash Functions Go Private: Constructions and Applications to (Homomorphic) Signatures with Shorter Public Keys[C]. *Advances in Cryptology -- CRYPTO 2015*: 254-274.

[137] Boneh D, Freeman D, Katz J, et al. Signing a Linear Subspace: Signature Schemes for Network Coding[C]. *International Workshop on Public Key Cryptography*, 2009: 68-87.

[138] Gorbunov S, Vaikuntanathan V, Wichs D. Leveled Fully Homomorphic Signatures from Standard Lattices[C]. *The forty-seventh annual ACM symposium on Theory of Computing*, 2015: 469-477.

[139] Schabhüser L, Butin D, Buchmann J. Context Hiding Multi-Key Linearly Homomorphic Authenticators[C]. *Cryptographers' Track at the RSA Conference*, 2019: 493-513.

李世敏 于 2020 年在中国科学院大学网络空间安全专业获得博士学位。现任中电科网络安全科技股份有限公司高级工程师。研究领域为应用密码学。研究兴趣包括可验证计算、安全协议等。Email: li.shimin@cetccst.com.cn

王欣 于 2020 年在中国科学院大学网络空间安全专业获得博士学位。现任北京蚂蚁云金融信息服务有限公司高级算法工程师。研究领域为公钥密码学。研究兴趣包括可搜索加密、可证明安全理论等。Email: wangxinsy1991@foxmail.com

薛锐 于 1999 年在北京师范大学数学系获得博士学位。现任中国科学院信息工程研究所研究员、博士生导师。研究领域为公钥密码学。研究兴趣包括安全协议、计算复杂性理论、密码协议的形式化方法等。Email: xuerui@iie.ac.cn