安全多方计算技术综述

余欢

（华中科技大学 计算机科学与技术学院，武汉市 430074）

摘要 近年来，机器学习迅速地发展，给人们带来便利的同时，也带来极大的安全隐患。机器学习的安全与隐私问题已经成为其发展的绊脚石。机器学习模型的训练和预测均是基于大量的数据，而数据中可能包含敏感或隐私信息，随着数据安全与隐私泄露事件频发、泄露规模连年加剧，如何保证数据的安全与隐私引发科学界和工业界的广泛关注。因此，需要能够保证机器学习中数据和模型的隐私的安全框架。而安全多方计算是其中主流的解决方案。本文将重点介绍安全多方计算中的同态加密以及秘密共享的技术原理和目前的研究现状。最后，总结和展望安全多方计算在隐私保护机器学习上的未来发展方向。

关键词 机器学习；安全多方计算；同态加密；秘密共享；

An overview of secure multiparty computing technologies

Yu Huan

(Department of Huazhong University of Science and Technology,City WuHan)

Abstract In recent years, machine learning has developed rapidly, which not only brings convenience to people, but also brings great potential safety hazards. The security and privacy of machine learning has become a stumbling block to its development. The training and prediction of machine learning model are based on a large amount of data, and the data may contain sensitive or privacy information. With the frequent occurrence of data security and privacy leakage events and the aggravation of the leakage scale year after year, how to ensure the security and privacy of data has attracted extensive attention in the scientific and industrial circles. Therefore, a security framework that can ensure the privacy of data and models in machine learning is needed. Secure multi-party computing is the mainstream solution. This paper will focus on the technical principle and current research status of homomorphic encryption and secret sharing in secure multi-party computing. Finally, the future development direction of secure multi-party computing in privacy protection machine learning is summarized and prospected.

Key words machine learning；secure multi-party computation；homomorphic encryption；secret sharing；

1 引言

机器学习是一种实现人工智能的方式，是近些年主要研究的领域。目前机器学习方案在很多领域都有着成熟的应用，如天气预报、能源勘探、环境监测等，通过收集相关数据进行分析学习，可以提高这些工作的准确性；还有如在垃圾邮件检测、个性化广告推荐、信用卡欺诈检测、自动驾驶、人脸识别、自然语言处理、语音识别、搜索引擎的优化等各个领域，机器学习都扮演着重要的角色。然而，蓬勃发展的机器学习技术使数据安全与隐私面临更加严峻的挑战，因为机器学习的更精准模型需要大量的训练数据为支撑。

自 2013 年斯诺登的“棱镜”事件以来，全球信息泄露规模连年加剧，引起社会的广泛关注。2016 年 9 月 Yahoo 被曝出曾被黑客盗取了至少 5 亿个用户账号信息；2017 年微软 Skype 软件服务遭受 DDOS 攻击，导致用户无法通过平台进行通信；2018 年 3 月美国《纽约时报》和英国《卫报》均报道：剑桥分析（Cambridge Anaalytica）数据分析公司在未经用户许可的情况下，盗用了高达 5 千万个 Facebook 的用户个人资料。2019 年美国网络安全公司 UpGuard 发现上亿条保存在亚马逊 AWS 云计算服务器上的 Facebook 用户信息记录，可被任何人轻易地获取；IBM 在未经当事人许可的情况下，从网络图库 Flickr 上获得了接近 100 万张照片，借此训练人脸识别程序，并与外部研究人员分享。2020 年 4 月《华盛顿邮报》报道视频会议软件 Zoom 存在的重大安全漏洞：数以万计的私人 Zoom 视频被上传至公开网页，任何人都可在线围观，很多视频都包含个人可识别信息，甚至是在家里进行的私密谈话。信息泄露的途径主要分为内部人员或第三方合作伙伴泄露、信息系统无法杜绝的漏洞、机构本身的防护机制不健全、对数据的重要程度不敏感，以及对安全配置的疏忽大意等。可见，数据隐私的泄露已不单单是满足某些外部人员好奇心所驱使，而是已成为一种重要的商业获利而被广泛关注，其中不乏内外勾结、合谋获取用户的隐私等行为。由此可见，机器学习中的安全与隐私问题已经非常严重。

目前，研究人员提出了许多解决机器学习中的隐私问题的方法。而目前最受关注的处理方案是安全多方计算。安全多方计算(secure multiparty computation,SMC)起源于姚期智的百万富翁问题,主要用于解决一组互不信任的参与方之间保持隐私的协同计算问题。在 SMC 中，参与方将各自的秘密数据输入到一 个约定函数进行协同计算，即使在一方甚至多方被攻击的情况下，SMC 仍能保证参与方的原始秘密数据不被泄露,并且保证函数计算结果的正确性。自 SMC 理论创立以来，已经衍生 出多个技术分支，包括混淆电路、秘密分享、同态加密和不经意传输。

2 原理和优势

本节简单同态加密和基于秘密共享的安全多方计算的技术原理。

2.1 同态加密

同态加密（homomorphicencryption，HE）允许直接在密文上做运算，运算之后解密的结果与明文下做运算的结果一样。假设存在加密函数 f,使得明文 M 加密后变成密文 M’,明文 N 加密后变成密文 N’,即 $f(M) = M^*, f(N) = N^*$,存在 f 的解密函数 f^{-1}能够将 f 加密后的密文解密成加密前的明文。将 M^*与 N^*相加得到 P^*,如果解密函数 f^{-1}对 P^*解密后的结果等于 M 和 N 相加的结果,即 $f^{-1}(P^*)=f^{-1}(M^*+N^*)=M+N$,则 f 是可以进行同态加密的加密函数。

同态加密可以分为加法同态、乘法同态以及全同态。加法同态指的是加密算法满足 $f(M)+f(N)=f(M+N)$,乘法同态指的是加密算法满足 $f(M)*f(N)=f(M*N)$。而全同态加密指的是一个加密函数同时满足加法同态和乘法同态,全同态加密函数可以完成加减乘除、多项式求值、指数、对数、三角函数等运算。

同态加密是真正的端到端加密系统,有望从根本上解决当今数据模型的信任问题,它使用户能够更好地控制其数据,同时受益于远程服务器提供的计算服务。例如,在集中式机器学习中,用户将训练数据以密文形式上传至服务器,服务器进行模型训练但并不知道用户原始训练集,因而保护了用户数据隐私;在联邦学习中,各个参与方将模型参数或者梯度加密后上传至中央服务器,中央服务器在不知道每个参与方上传的原始模型参数或者梯度的同时完成了模型训练迭代,从而保护了模型和用户原始数据的隐私。

目前主流的技术实现方案是基于格上的困难问题构建的。例如 BFV 方案基于环上的带错误学习问题，但目前同态加密技术的计算复杂度过高，相比起明文的机器学习速度相差悬殊，学者们在如何缩短两者的差距上不断探索。

2.2 安全多方计算

安全多方计算形式化描述为:假定有 m 个参与方 $P_1, P_2, \cdots, P_m$,他们拥有各自的数据集 $d_1, d_2, \cdots, d_m$,在无可信第三方的情况下,如何安全地计算一个约定函数 $y=(d_1, d_2, \cdots, d_m)$,同时要求每个参与方除了计算结果外不能得到其他参与方任何输入信息。

SMC 具有输入独立性、计算正确性、去中心化等特征。SMC 基础密码协议包括 OT(obliviousstransferprotocol)协议、GC(garbledcircuits)协议、SS(secretsharing)协议、GMW(Goldreich-Micali-Wigderson)协议等。这些协议都是重要的密码学工具,可以看作特殊的安全多方计算问题。SMC 是多种密码学基础工具的综合应用,因此在实现安全多方计算时也广泛地应用了上述同态加密技术，本文将它们分开进行讨论。举一个简单的基于秘密共享协议实现的安全多方计算的例子，假如我们需要计算 300（用 x 表示）和 600（用 y 表示）的和，但是不能让任一方知道这俩个数据，我们可以简单的加性秘密共享给三方，例如可以将 x 随机切分成 x_1，x_2，x_3的和，对 y 也进行同样的处理，然后将 x_1，y_1 分发给参与计算方 P_1，x_2，y_2 分发给 P_2，x_3，y_3 分发给 P_3，这三方只需要将各自得到的部分相加得到 z_1，z_2，z_3，最后将结果

聚合就可以得到我们想要的数据同时参与计算的三方不会学习到任何关于 x 和 y 的信息，对于乘法计算也是类似的思想，只是实现上更为复杂，需要的计算和通信消耗更大。

相较于同态加密技术，基于秘密共享安全多方计算所需的计算复杂度更低，但因为需要联合多方进行计算，带来了更高的通信消耗和安全问题。其中一方腐败就有可能导致协议失效甚至泄露数据。另外，同态加密也可以用于安全多方计算当中。目前也有很多研究在一个框架中混合采用两种技术，且取得了不错的成果，故有时也把同态加密当作安全多方计算的子协议。

3 研究进展

本节主要介绍同态加密的主流方案以及三方、四方安全计算的代表方案。

3.1 同态加密研究进展

早在 1978 年 Rivest 等人就提出了隐私同态(privacy homomorphism)的概念，但这个概念一直被认为是一个开放性的问题，直到 2009 年 Gentry 提出首个全同态加密方案，证明了在加密数据上计算任何函数的可行性。

同态加密可以划分为部分同态加密(partial homomorphic encryption, PHE)、浅同态加密(somewhat homomorphic encryption, SHE)和全同态加密(fully homomorphic encryption, FHE)。部分同态加密仅支持单一类型的密文同态运算，主要包括加法同态(additive homomorphic encryption, AHE)和乘法同态(multiplicative homomorphic encryption, MHE)，代表方案分别为 Paillier[1]和 ElGamal[2]。浅同态加密支持低次多项式运算。全同态加密的构造均遵循 Gentry 的思想蓝图，利用自举(bootstrapping)技术将浅同态加密方案转换为全同态加密方案，即通过同态地执行解密电路以更新密文、约减噪音，从而支持进一步的同态运算。目前主流的 FHE 方案大多基于格上的困难问题(主要是 LWE,RLWE)构造，代表方案包括 BGV[3],BFV[4],GSW[5], CKKS[6]等。

BGV（Brakerski-Gentry-Vaikuntanathan）方案是目前主流的全同态加密算法中效率最高的方案。在 BGV 方案中，密文和密钥均以向量表示，而密文的乘积和对应的密钥乘积则为张量，因此密文乘法运算会造成密文维数的爆炸式增长，导致方案只能进行常数次的乘法运算。BGV 方案采用密钥交换技术控制密文向量的维数膨胀，在进行密文计算后通过密钥交换将膨胀的密文维数恢复为原密文的维数。同时，BGV 方案可采用模交换技术替代 Gentry 方案中的“Bootstrapping”过程，用于控制密文同态运算产生的噪声增长，而不需要通过复杂的解密电路实现。因此，在每次进行密文乘法运算后，首先需要通过密钥交换技术降低密文的维数，然后通过模交换技术降低密文的噪声，从而能够继续进行下一次计算。

BFV（Brakerski/Fan-Vercauteren）方案是与 BGV 方案类似的另一种第二代全同态加密方案，同样可基于 LWE 和 RLWE 构造。BFV 方案不需要通过模交换进行密文噪声控制，但同样需

要通过密钥交换解决密文乘法带来的密文维数膨胀问题。

GSW（Gentry-Sahai-Waters）方案是一种基于近似特征向量的全同态加密方案。该方案基于 LWE 并可推广至 RLWE，但其的性能不如 BGV 方案等其他基于 RLWE 的方案。GSW 方案的密文为矩阵的形式，而矩阵相乘并不会导致矩阵维数的改变，因此 GSW 方案解决了以往方案中密文向量相乘导致的密文维数膨胀问题，无需进行用于降低密文维数的密钥交换过程。

CKKS（Cheon-Kim-Kim-Song）方案是 2017 年提出的一种新方案，支持针对实数或复数的浮点数加法和乘法同态运算，得到的计算结果为近似值，适用于机器学习模型训练等不需要精确结果的场景。由于浮点数同态运算在特定场景的必要性，HElib 和 SEAL 两个全同态加密开源库均支持了 CKKS 方案。上述方案及其基本特性和应用情况总览如下表 3-1 所示。

表 3-1 各类同态加密算法

类型		算法	时间	说明	实际应用
半同态加密	乘法同态	ElGamal 算法	1985	随机化加密	DSS 数字签名标准基于 ElGamal 数字签名算法的变体
	加法同态	Pallier 算法	1999	应用最为成熟	联邦学习
全同态加密		Gentry 方案	2009	第一代全同态加密，性能较差	/
		BGV 方案	2012	第二代全同态加密方案，性能较好	IBM HElib 开源库
		BFV 方案	2012	第二代全同态加密方案，与 BGV 类似	微软 SEAL 开源库
		GSW 方案	2013	第三代全同态加密方案，基于特征向量	TFHE 开源库
		CKKS 方案	2017	可实现浮点数近似计算，适合机器学习建模场景	HElib 和 SEAL

尽管全同态技术方案一直在发展创新，但计算复杂度仍然是它们的主要瓶颈。近期的研究有在神经网络私有推理框架中运用上述同态加密方案，并进行了一定程度的优化。2018 年提出的 Gazelle[7]是一个可扩展的低延迟安全神经网络推理系统，它使用了同态加密 BFV 和传统的两方计算技术（在论文中具体指的混淆电路）。具体来说，这个计算框架由客户端和云端组成，其中客户端将数据加密后发送给云端，云端对加密后的数据执行如卷积一类的线性运算，而碰到非线性运算时再返回给客户端，客户端解密后执行如池化等非线性运算。如此循环直到计算结束。而客户端采用的技术是混淆电路，云端采用的技术则是 BFV 同态加密方案。

2020 年提出的 Cheetah[8]基于 Gazelle 进一步对 BFV 方案优化，可达到明文的推理速度。如下图 3-1 是 Cheetah 的框架和系统设计，其中红色数字是在 ResNet50 上实验得到的加速比。Cheetah 在 Gazelle 的基础上针对云端的同态加密计算部分提出了三点优化策略。

- HE-PTUNE 通过针对每个网络层选择合适的 BFV 参数，在性能和噪声中取得更好的平衡；
- Sched-PA 在卷积层和全连接层，通过合理的点积调度策略，降低了噪声增长，从而降低了每个 HE 操作的消耗；
- 提出了高度利用核内以及核间并行化的硬件加速框架如图 3-2。

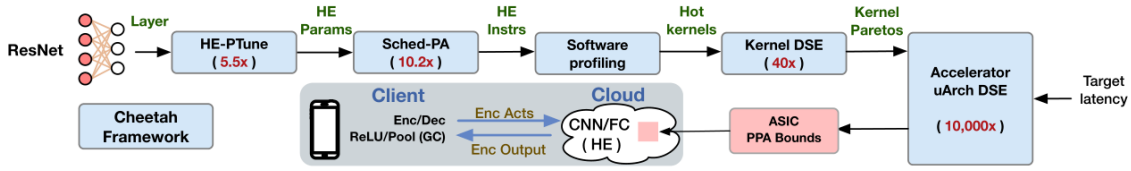

图 3-1 Cheetah 框架和系统设计

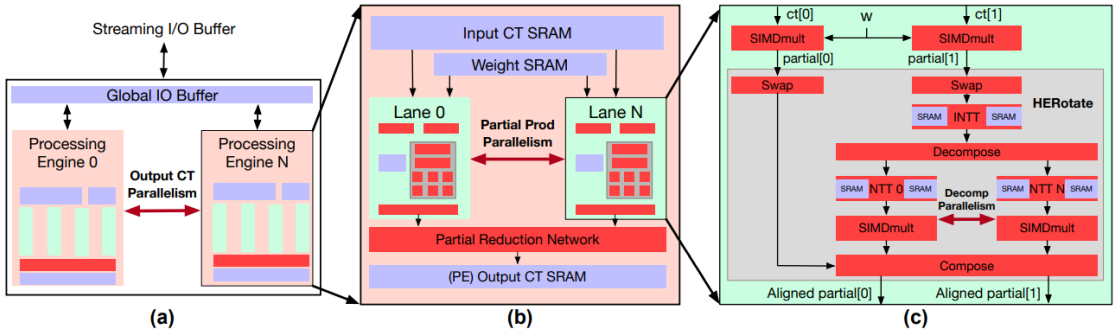

图 3-2 Cheetah 加速架构

3.2 安全多方计算

安全多方计算是通过一系列基础密码协议组合实现的隐私计算，其中包括 OT，GC，SS 等。

以前的研究通常采用 GC 技术进行非线性运算，而将 SS 技术用于线性运算，但由于 GC 的高通信量和计算复杂度，近期的研究都是采用的不同的 SS 协议计算线性和非线性运算，或和 HE 结合。所以，在此我们重点介绍基于秘密分享的安全多方计算(secret sharing-secure multiparty computation, SS-MPC)的研究发展现状。

SS-MPC 根据秘密的生成方式可以分为基于多项式插值的秘密分享和加性秘密分享。基于多项式插值的 SS-MPC 容易实现任意门限，但是需要模幂运算，导致其计算效率低。加性秘密分享根据秘密数据形式可以分为算术秘密分享和布尔秘密分享，在计算线性运算时通常需要使用算术秘密分享，在计算非线性运算时通常需要使用布尔秘密分享。加性秘密分享的优势是计算效率高，其门限方案的实现需要参与方持有冗余份额。目前 SS-MPC 在实际中应用最广泛，能够高效地实现线性运算和非线性运算。

目前主流的安全多方计算的参与方通常是三方或四方。其中三方的代表方案有 ABY3[9]和 SecureNN[10]，而四方的代表方案有 PrivPy[11]。

2018 年提出的 ABY^{3}受到设置冗余份额的启发，基于此设计了新的 2-out-of-3SS-MPC 方案，进一步通过份额校验实现了恶意安全方案。秘密分享的过程为：数据拥有方计算长度为 1 位的秘密数据 x,y 的加性份额 $x_i, y_i (i=1,2,3)$,并向 3 个计算方 P_i分发份额$(x_i, x_{i+1}), (y_i, y_{i+1})$。计算乘法的过程为：$P_i$ 在本地计算得到积的份额 $z_i=x_iy_i+x_iy_{i+1}+x_{i+1}y_i+\alpha_i$ 并将其发给 P_{i-1}，其中 $\alpha_1+\alpha_2+\alpha_3=0$。在秘密恢复阶段，任意 2 个计算方可以恢复出 xy。ABY^{3}通过使用 Beaver 三元组来完成份额校验，实现的恶意安全方案最多能够容忍一个恶意参与方。ABY^{3}实现的安全架构图如下 3-3 所示。

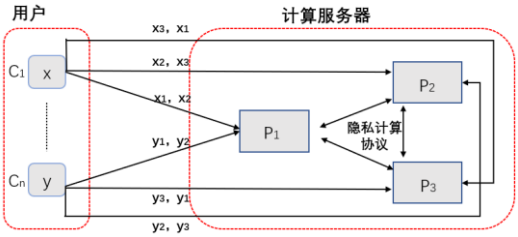

图 3-3 ABY^{3}计算架构

P_1、P_2和 P_3三方接收用户传来的数据，并根据隐私计算协议共同计算目标函数。其中的隐私计算协议包含基本的加法、乘法协议，另外由于算术秘密共享协议是针对整数的，所以对浮点型的数据进行处理时需要将其扩大一定的倍数转换成整数，所以在进行乘法后需要截断协议，ABY^{3}提供了两种截断协议。ABY^{3}采用比特秘密共享协议来实现 Relu 和池化等非线性运算，所以还提供了算术秘密共享与比特秘密共享相互转换的协议。

SecureNN 是 2019 年提出的半诚实安全三方方案。在并行和独立的工作中，与上述 ABY^{3}实现了类似的结果，但使用的技术与根本不同。ABY^{3}从理论上描述了如何转换协议以实现恶意安全，并提供了半诚实安全协议的实现和实验数字。相比之下，SecureNN 的性能数据既适用于半诚实的安全性，也适用于恶意的隐私。SecureNN 协议的实现非常简单，这使它们在现实世界的部署中具有优势(因为它们不需要在使用乱码电路时需要的大量优化)。数据拥有方生

成 m×n 维秘密矩阵 x,n×v 维秘密矩阵 y 的加性份额 $x_i, y_i (i=1,2)$,并分发给计算方 P_1, P_2,矩阵元素是长度为 1 位的比特串。三方秘密分享方案中计算乘法时采用 Beaver 三元组的思想，P_0 作为辅助节点来生成 Beaver 三元组以及其他随机数，P_1 和 P_2 是计算节点。四方秘密分享方案中 P_1, P_2 是计算节点，P_3, P_4 是辅助计算节点。计算乘法运算时首先 P_1, P_2 分别在本地计算 x_1y_1, x_2y_2;然后，P_1 分别把 x_1, y_1 发给 P_3, P_4,P_2 分别把 y_2, x_2 发给 P_3, P_4;收到份额后，P_3, P_4 分别计算交叉项 x_1y_2, x_2y_1 并发送给 P_1, P_2。在秘密恢复阶段，将 P_1 和 P_2 拥有的份额相加即可恢复出矩阵 xy。

PrivPy 是 2018 年李艺博士在他的博士论文中提出的。如图 3-4 是 PrivPy 的计算引擎架构。

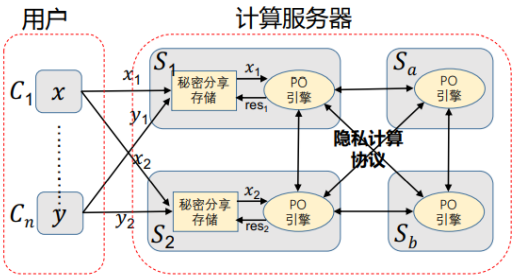

图 3-3PrivPy 计算引擎架构

引擎共包含 4 个半诚实的计算服务器，用户只需要将数据拆分后发给计算服务器，计算的中间过程不需要参与。这 4 个服务器当中，S_1 和 S_2 是主服务器，同时负责存储和计算。对每个数据 x 而言，S_1 只存储 x_1，而 S_2 只存储 x_2。S_a 和 S_b 是辅助服务器，不存储数据，只负责辅助计算。我们把负责存储的模块叫 SS Store，同时把负责计算的模块叫做 PO。

计算引擎的计算流程大致如下：

- 步骤 1：启动阶段。计算任务开始前，各参与者在线下就计算任务的内容达成一致，包括谁提供隐私输入，计算过程，以及最终结果给谁等。
- 步骤 2：程序准备。前端解析器解析程序，将任务代码解析为基本操作，并进行一定的优化，以提高运行效率。
- 步骤 3：拉取秘密分享中的编码数据。用户将其输入编码成秘密分享要求的格式，并将不同的数据发给相应的计算服务器。
- 步骤 4：在计算服务器上执行 PO 计算。计算服务器收到数据后，进行计算，过程中不需要用户的参与。
- 步骤 5：将结果发给结果方。当一个名为 reveal 的操作被触发的时候，程序会通知用户来取结果。有相应权限的用户能将将结果取回。

PrivPy 采用混淆电路技术来实现非线性计算，所以其隐私计算协议与 ABY[3] 略有不同，下图 3-5 给出 PrivPy 种的操作关系。

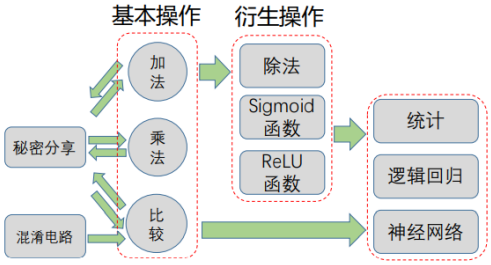

图 3-5 PrivPy 中操作间的关系

4 总结与展望

安全多方计算和全同态加密技术经过十几年的理论探索后在 PPML 领域目前都有了相应的

系统实现,但它们也都还存在很多问题，如何提高性能即如何减少通信和计算复杂度一直都是研究的重点。未来的工作可围绕以下 2 方面展开：

1）进一步提升准确率与系统性能。MPC 实现的 PPML 方案开销依然远大于明文模型的运算，准确率与明文模型相比还是会有一定的损失。PPML 方案非线性层的实现方式会为准确率带来很大的影响，相对而言 GC 精度较高但性能开销大，SS 性能较好但无法达到 GC 的精度，现有 PPML 方案都在效率和准确率之间进行折中。可通过硬件加速的方案来提升性能，如 CUHE 库是用 GPU 加速的 HE 库，这就要求我们寻找 GPU 友好的密码协议。此外，在 PPML 使用混合技术方案中降低不同技术的转换开销也将进一步提升 PPML 技术的可用性。

2）提升系统安全性。现有的很多方案都只满足半诚实安全模型，未来需要进一步加强对恶意敌手的防御。另一方面，早期的恶意安全方案实现了中止安全性，2019 年之后的方案都达到了更强的安全属性-公平性，提升系统的安全属性也是今后研究的一个重要方向，未来要实现的安全目标是能够保证结果输出(GOD),并且在性能开销也需要控制在可接受范围内。

参考文献

[1] Paillier P.Public-key cryptosystems based on composite degree residuosity classes[C]//Proc of Int Conf on the Theory and Applications Cryptographic Techniques.Berlin:Springer,1999:223-238.

[2] ElGamal T.A public key cryptosystem and a signature scheme based on discrete logarithms[J].IEEE Transactions on Information Theory,1985,31(4):469-472.

[3] Brakerski Z,Gentry C,Vaikuntanathan V.(Leveled)Fully homomorphic encryption without bootstrapping[J].ACMTransactions on Computation Theory,2014,6(3):1-36.

[4] Fan J,Vercauteren F.Somewhat practical fully homomorphic encryption[OL].2012 [2021-07-29].https: //eprint.iacr.org/2012/144.

[5] Gentry C,Sahai A,Waters B.Homomorphic encryption from learning with errors:Conceptually-simpler,asymptoticallyfaster,attribute-based[C]//Proc of Annual Cryptology Conf.Berlin:Springer,2013:75-92.

[6] Cheon J H,Kim A,Kim M,et al.Homomorphic encryption for arithmetic of approximate numbers[C]//Proc of Int Conf on the Theory and Application of Cryptology and Information Security.Berlin:Springer,2017:409-437.

[7] C. Juvekar, V . V aikuntanathan, and A. Chandrakasan, “Gazelle: A low latency framework for secure neural network inference,”arXiv preprint arXiv:1801.05507, 2018.

[8] B. Reagen et al., "Cheetah: Optimizing and Accelerating Homomorphic Encryption for Private Inference," 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2021, pp. 26-39, doi: 10.1109/HPCA51647.2021.00013.

[9] Mohassel P,Rindal P.ABY3:A mixed protocol framework for MAChine learning[C]//Proc of the 2018ACM SIGSACConf on Computer and Communications Security.New York:ACM,2018:35-52.

[10] Wagh S,Gupta D,Chandran N.SecureNN:3-party secure computation for neural network training[J].Proceedings on Privacy Enhancing Technologies,2019,2019(3):26-49.

[11] Li Y , Duan Y , Yu Y , et al. PrivPy: Enabling Scalable and General Privacy-Preserving Machine Learning[J]. 2018.