同态加密在隐私计算中的应用综述

邵航　高思琪　钟离　傅致晖　孟丹　李晓林

（同盾科技人工智能研究院，杭州 311121）

摘要：同态加密技术是一种基于数学难题的计算复杂性理论的密码学技术，支持数据以密态方式进行计算，计算结果解密后与明文计算的结果一致，在多样化复杂应用场景中具有很好的普适性，是目前隐私计算领域的一个热点研究方向。通过对同态加密技术的发展历程以及相关的技术路线进行梳理，解析了同态加密在安全求交、隐匿查询、多方联合计算、多方联合建模等典型隐私计算应用场景的技术融合应用，并对同态加密目前广泛落地应用过程中碰到的关键问题进行分析，最后对同态加密的研究发展方向进行探讨。

关键词：同态加密；多方安全计算；联邦学习；隐私集合求交；隐私信息检索；隐私计算

中图分类号：TP309.2　　**文献标志码**：A

引用格式：邵航，高思琪，钟离，等．同态加密在隐私计算中的应用综述[J]．信息通信技术与政策，2022，48(8)：75-88.

DOI：10.12267/j.issn.2096-5931.2022.08.012

0　引言

大数据和人工智能快速发展，目前已经形成产业规模，并上升到国家战略层面，海量数据的交叉协同计算和人工智能技术的落地应用为各行各业提供了更好的支持和用户体验；但与此同时，数据安全与隐私保护问题日益凸显，国际与国内各项隐私保护的政策法规相继制定，安全、可信、合规使用大数据成为数据要素流通与人工智能进一步发展的一个必须要解决的难题。近年来，兴起的隐私计算技术被认为是解决数据隐私保护与数据安全流通、数据智能化应用两难问题的重要技术手段。目前，隐私计算主要分为多方安全计算（Secure Multi-Party Computation，MPC）、联邦学习（Federated Learning，FL）、可信执行环境（Trusted Execution Environment，TEE）这 3 个主要技术路线，而同态加密技术（Homomorphic Encryption，HE）作为一个重要的基础性密码算法协议，被广泛应用于不同技术路线的隐私计算解决方案中，例如基于多方安全计算的联合统计、基于联邦学习的联合建模、基于隐私信息检索（Private Information Retrieval，PIR）的联合查询等。本文首先从技术角度来介绍同态加密技术的发展历程和一些常见的实现机制，然后结合典型的隐私计算应用场景介绍基于同态加密的技术解决方案，最后分析当前同态加密技术面临的挑战和未来可能的发展方向。

1　同态加密技术综述

1.1　同态加密简介

同态加密是基于数学难题的计算复杂性理论的密码学技术。主要思想是：对经过同态加密的数据进行某种方法计算得到一个输出，将这一输出进行解密，其结果与用同种方法计算未加密的原始数据得到的输出结果一致。

一般来说，同态加密具有加法同态性和乘法同态

性,可利用加法和乘法构造任意的计算方法对密文进行运算。根据同态性质可分为:部分同态加密(Partially Homomorphic Encryption,PHE)、有限层次全同态加密(Leveled Fully Homomorphic Encryption,LFHE)、类同态加密(SomeWhat Homomorphic Encryption,SWHE)和全同态加密(Fully Homomorphic Encryption,FHE)方案。部分同态加密方案是指具有单一的加法同态性或乘法同态性,例如 RSA 算法、Elgamal 算法和 Paillier 算法等。有限层次全同态加密方案是指支持对密文进行有限次数的同态加法和同态乘法,例如 BGV12 方案、GSW13 方案和 CKKS17 方案等。全同态加密方案支持对密文进行无限次数的加法同态和乘法同态,即任何类型的计算,而自举(Bootstrapping)技术[1]是目前能实现全同态加密方案的唯一方法。

1.2 同态加密发展历程

1977 年,Rivest、Shamir 和 Adleman[2]提出 RSA 密码算法,该算法基于大整数难分解问题,起初是用作加解密和消息签名,但因其具有乘法同态性而成为部分同态加密方案的一员。而在之后的 1985 年,Taher ElGamal[3]基于循环群上的离散对数问题提出了具有乘法同态性的 ElGamal 算法。Pascal Paillier[4]于 1999 年基于符合剩余类困难问题(Composite Degree Residuosity Classes)提出具有加法同态性的 Paillier 算法。这三个较为典型的部分同态加密方案,为全同态的发展奠定了坚实的基础。

真正意义上的全同态是在 2009 年,Gentry[5]基于理想格(Ideal lattice)构造出了第一个可行的全同态加密方案 Gentry。在该方案中,先构造了一个有限层次全同态加密方案,然后通过自举实现全同态加密,该构造全同态加密的框架成为后续大多数方案的重要思路,从而全同态加密被冠以“密码学的圣杯”。Gentry[6]于 2010 年提出基于整数的全同态加密方案 DGHV,仅在整数上进行简单计算,其安全性可以归约为近似最大公约数问题,然后使用自举实现全同态加密。因此,DGHV 方案被认为是第一代全同态加密方案的代表[6]。

第二代全同态加密方案以基于带错误学习(Learning With Error,LWE)问题为特点。Brakerki 和 Vaikuntanathan[7]在 2011 年提出 BV 方案,是第二代全同态加密方案的代表之一,BV 方案使用重线性化技术(Relinearization)使得在不增加密文尺寸(维数)的情况下完成一次乘法同态。由于和 LWE 问题相比,环上带错误学习(Ring Learning With Error,RLWE)问题中的多项式计算可使用快速傅立叶变换(Fast Fourier Transform,FFT)进行加速计算。2012 年,Brakerki、Gentry 和 Vaikuntanathan[8]构造了基于 RLWE 问题的有限层次全同态加密方案 BGV。BGV 方案是第二代全同态加密方案的另一个典型代表方案,使用了模切换(Modulus Switching)降低密文噪音,并采用密钥切换(Key Switching)控制密文尺寸,还支持单指令多数据流(Single Instruction Multiple Data,SIMD)编码,能对多比特明文编码进行打包处理,可明显提升计算性能。2012 年,Fan 等[9]提出 BFV 方案,较 BGV 更加轻量。与绝大数基于整数上计算的方案不同,Cheon 等[10]于 2017 年提出一个基于 RLWE 问题的有限层次全同态加密方案 CKKS,能够近似地执行同态加法和同态乘法运算,并支持浮点数运算,非常适合统计和机器学习应用。

第三代全同态加密方案以 Gentry 等[11]在 2013 年设计的使用近似特征向量技术构造出一种无需计算密钥的有限层次全同态加密方案 GSW 为代表。GSW 方案中密文是矩阵形式,因此密文的加法和乘法相当于对矩阵做加法和乘法。在进行同态计算时,密文尺寸不会变大,也无需引入计算密钥。另外,在同态计算过程中,密文噪音的增长是非对称的(即 C_1 和 C_2 是密文,$C_1 \otimes C_2$ 和 $C_2 \otimes C_1$ 产生的噪音不同,$\otimes$表示加法或乘法运算),且密文的噪音呈线形增长,这使得之前用的降噪技术(如模切换等)不再适用,GSW 提出比特分解(BitDecomp)和展开(Flatten)技术实现降噪。

基于 GSW,Alperin-Sheriff 等[12]加入自举技术,进而实现真正的全同态加密方案,之后的 FEHW[13]和 TFHE[14]方案更是优化了自举,并设计了对应的开源库,其中 TFHE 自举一个比特仅需 13 ms,这是目前自举效率最高的方案之一。2018 年,Cheon 等[14]基于 CKKS,加入自举技术提出了全同态加密方案[15],接着在文献[16]中优化了自举,使得自举时间降低了两个数量级。同态加密的主要技术发展历程如图 1 所示。

纵观同态加密方案近十多年的发展,虽然存在效率低、密文膨胀等瓶颈,但各种优化技术是加速同态加

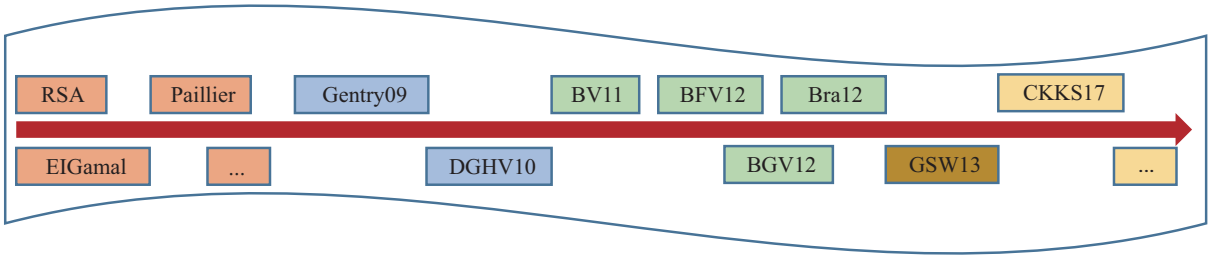

图 1　同态加密技术的发展历程

密技术在隐私计算领域中的规模化使用及工业化落地的催化剂。各大公司和组织也相继开源了同态加密库，表 1 列举了目前常用的同态开源库。

表 1　同态加密开源库列举

开源库	贡献者	HE 方案
Cryptofun[17]	Arnaucube	RSA/ElGamal/Paillier/etc
HELIB[18]	IBM	BGV/CKKS
SEAL[19]	微软	BGV/BFV/CKKS
FEAAN[20]	首尔大学	CKKS
FHEW[21]	Ducas、Micciancio	FHEW
TFHE[22]	Chillotti 等	TFHE 和 FHEW
PALISADE[23]	MIT、UCSD	BFV/BGV/FHEW/TFHE/CKKS

2　同态加密应用解决方案

当前，“数据就是资源”逐渐成为现实，顺应大数据云计算的潮流，大量数据上云，为了数据安全，必须加密存储。但为了能更好地释放数据价值，数据的开放共享、交流互换、联合计算也是大势所趋。既想要数据，又想要安全，如何兼顾风险和效率，在保证数据安全的前提下，发挥更大的数据价值是当前重要的研究话题。隐私计算应运而生，通过联邦学习、多方安全计算、可信执行环境等技术，在解决实际业务问题的同时兼顾数据安全，进而实现“数据可用不可见，知识共创可共享”的目标。

同态加密技术因其具有允许用户直接在密文上进行运算，运算结果解密后和对明文运算的结果一致的性质，是实现数据“可用不可见”的关键技术之一。在安全求交、隐匿查询、多方联合计算、多方联合建模等隐私计算的应用场景中具有举足轻重的地位。

2.1　同态加密在安全求交中的应用

安全求交，即隐私集合求交（Private Set Intersection，PSI），能够使得多个参与方在不公开各自数据的前提下，共同找出交集数据，且不能泄露交集数据以外的信息。

PSI 通常是隐私计算中重要的前置步骤，可在安全计算或建模之前找出多方共有的样本，并且保证不泄露各方独有的样本。作为一项成熟的“小技术”，PSI 技术目前已在隐私保护的实名认证、联合风控、数据发现、数据对齐等多个场景得到“大应用”。以联合风控为例，银行在客户信息核查阶段可以运用 PSI 技术，探查同业金融机构的风险名单或同时多次借贷的客户信息，同时保证优质客户信息不被泄露。另外，在数据发现场景下，社交软件基于用户授权，可运用 PSI 技术与用户通信录中的手机号码进行隐私安全求交，进而根据此信息提供好友账号推荐。PSI 还可应用在纵向联邦学习场景中。在纵向联邦学习的建模场景中，PSI 也称样本对齐（Sample Alignment），即各参与方首先计算出训练样本的交集，再进行联邦模型训练。

PSI 根据参与者的数量可分为两方和多方，下面均以两方为例。假设参与 PSI 的两方为发送方 S（Sender）和接受方 R（Receiver），分别持有数据集 X 和 Y。根据两方数据集大小的不同，分为平衡场景（Balanced PSI）和非平衡场景（Unbalanced PSI）。在平衡场景下，双方样本数量相差不大，适用于双方客群有较多重叠的场景，如集团子部门之间的安全求交场景；在非平衡场景下，双方样本数量相差非常大，如营销场景中筛选本机构种子用户与外部数据方海量用户群中的共有用户群，实现目标客群筛选。

在平衡场景中，PSI 的实现方式目前较为成熟的有基于 RSA、基于 Diffie-Hellman（DH）和基于不经意传输（Oblivious Transfer，OT）等。综合安全性、效率和通信量考虑，基于 OT 以及 OT Extension[24] 系列的对齐方案被广泛使用。

在非平衡场景中，可在基于 OT 的 PSI 方案基础上，将同态加密技术结合特定的优化方法，实现高效的 PSI。下面介绍两种高效的基于同态加密的非平衡 PSI 方案。

2017 年，Chen 等[25] 在 CCS2017 中将同态加密应用到非平衡 PSI 中，综合使用了 Cuckoo hash、Partition、Window 和 Modulus Switch 技术，具体流程如图 2 所示。试验表明，在数据量为 $N_X = 5\ 000$，$N_Y = 1\ 600$ 万时，CCS2017[25] 方案的通信量为 12.5 MB，求交时间为 36 s。

2018 年，Chen 等[25] 改进了 CCS2017，提出了一个能抵抗恶意攻击的非平衡 PSI 方案（CCS2018[26]）。与 CCS2017[25] 相比，该方案支持更高位（512 位和 1 024

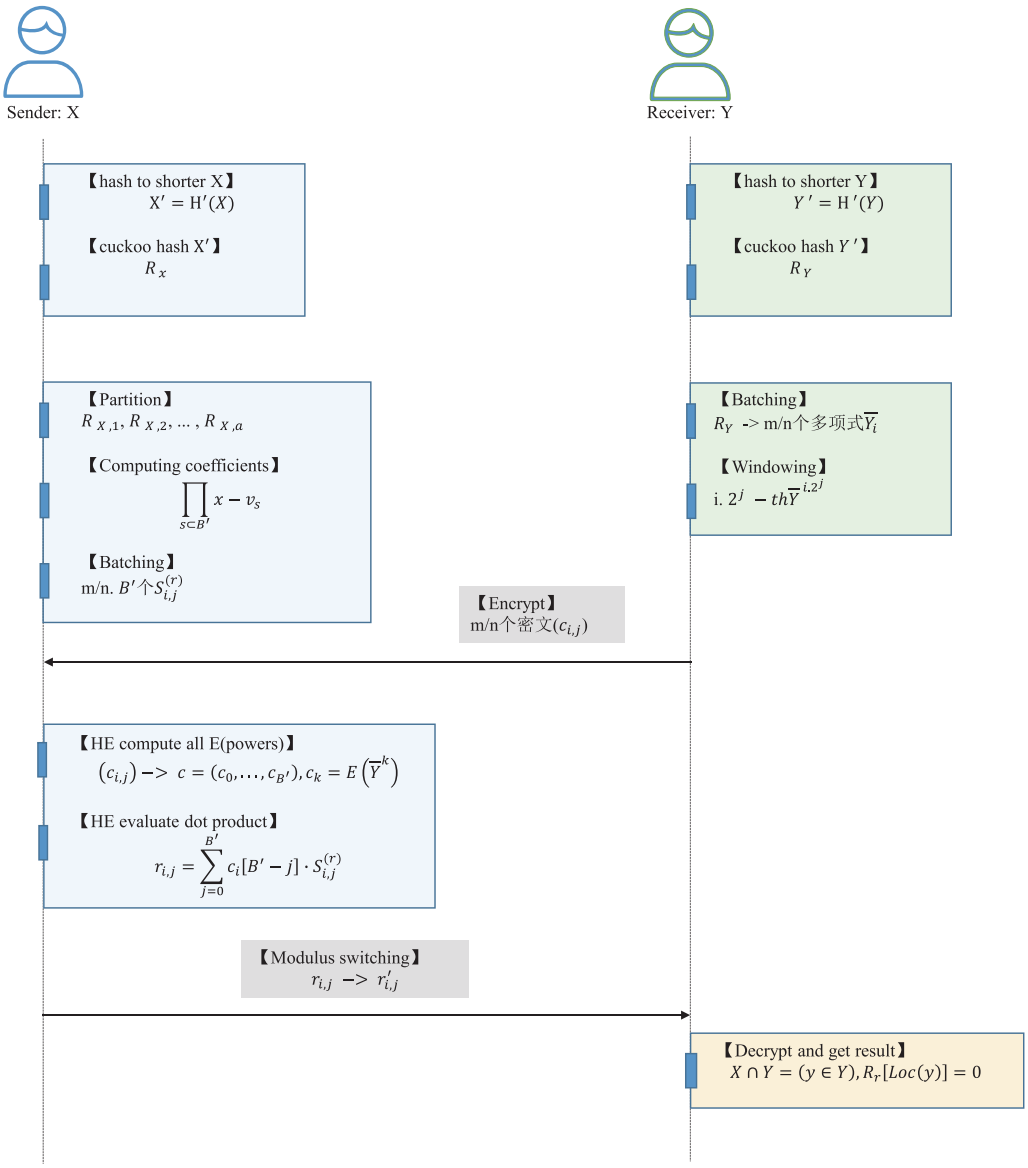

图 2　基于同态加密的非平衡 PSI 方案（CCS2017[25]）

位)的 Item(CCS2017 支持 32 位),改进了 SIMD 编码,在不增加加密参数的前提下,提升了效率和安全性,具体协议如图 3 所示。该方案较 CCS2017 在性能上做出很大改进(见表 2),在数据量为 $N_X = 2^{24}$ 和 $N_Y = 5\,535$ 时,CCS2017 需要 20 MB 的通信量和 40 s 的在线计算时间,CCS2018 通信量为 16 MB,在线计算时间为 22 s(单线程),运行时间几乎缩短 2 倍和通信量节约 27%。此外,当接受方 R 的数据集更小时,CCS2018 的同态加密的参数会更小,能进行更少的密文计算,当接受方 R 的数据量为 512 或 1 024 时,该方案运行时间分别只需 9.1 s 和 17.7 s,以及 8.2 MB 的通信量,较 CCS2017 快 2~4 倍,发送的数据量降低一半,同时支持任意长度的 Item。

2.2 同态加密在隐匿查询中的应用

早期的信息检索是用户根据自己的需求生成查询请求并发送给存有数据库的服务器,然后服务器返回一个或多个结果给用户。但用户查询请求信息(如查询 ID)是“暴露”的,针对这一需求,隐私信息检索

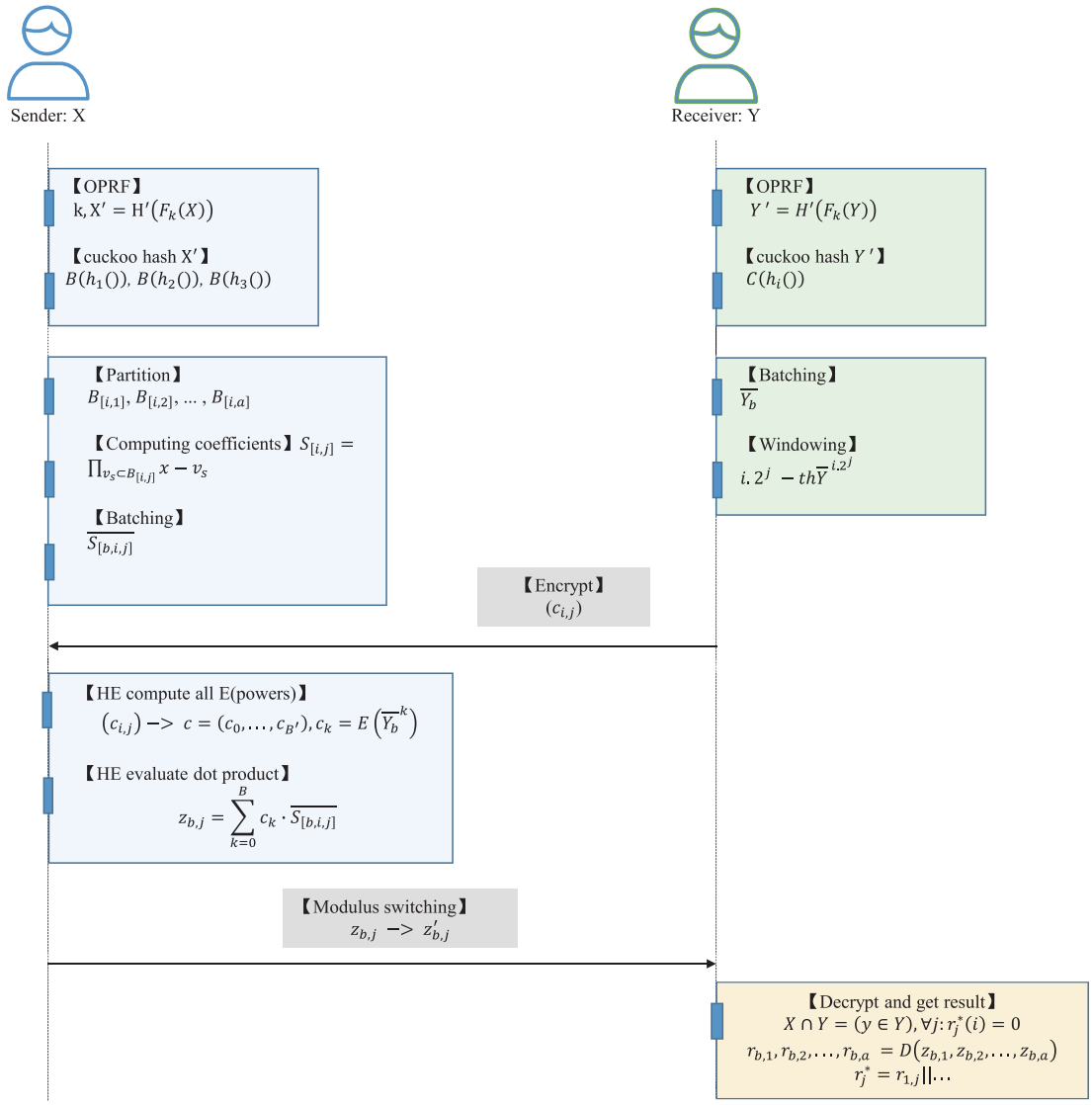

图 3 基于同态加密的非平衡 PSI 方案(CCS2018[26])

表 2　方案对比

N_X	N_Y	方案	离线时间/s	在线时间/s	在线通信量/MB
2^{24}	11041	CCS2018[26]	656.00	20.10	41.48
		CCS2017[25]	71.00	44.70	23.20
	5535	CCS2018[26]	806.00	22.01	16.39
		CCS2017[25]	64.00	40.10	20.10
2^{20}	11041	CCS2018[26]	43.00	4.49	14.34
		CCS2017[25]	6.40	6.40	11.50
	5535	CCS2018[26]	43.00	4.23	11.50
		CCS2017[25]	4.30	4.30	5.60

(Private Information Retrieval,PIR)就应运而生,它允许用户从数据库检索时,能同时隐藏检索内容,比如保险机构(查询方)对投保人作信用评估时,需要向大数据机构中心查询所需用户信息(年龄、心率、BMI 等),大数据机构(被查询方)在不能获取保险机构查询对象的情况下,提供匹配的查询结果,且保险机构也无法获取除查询结果以外的信息。比较“直接简单”的方法就是下载整个数据库,服务器不得知用户的检索内容,这样明显是不切实际的。

自 1995 年,Chor 等[27]提出了开创性的 PIR 方案后,为隐私计算的发展打下了坚实的理论基础,能应用在隐匿查询场景中,但也一直存在着诸多问题,比如通信复杂度和计算复杂度过高、服务器共谋等。

而将同态加密应用在 PIR 中是一大突破,早在 2011 年,BV11[7]中就提出了一个基于同态加密的单服务器(Single Server)PIR 方案,且该方案使用了混合加密来降低同态加密的密文长度。近年来,随着同态加密技术的发展,基于同态加密的 PIR 方案在性能上逐渐高效,更具备使用价值。微软的 Angel 等[28]在 2018 年提出了一个基于全同态加密的隐私信息检索方案 SealPIR,适用于算力强大的服务器,较低配置的用户客户端的场景,该方案对应的开源库为 SEALPIR[29]。表 3 给出了协议的伪代码,查询方执行 Query 和 Extract 操作,被查询方运行 Setup 和 Answer 操作,其中 DB 是密态数据库(即通过同态加密后的数据库),n 是数据库大小,FHE 是同态加密算法,idx 是要查询的序号。

表 3　SealPIR 协议框架

FHE = {FHE.Keygen, FHE.Enc, FHE.Const_Add, FHE.Add, FHE.Const_Mult, FHE.Mult, FHE.Dec}

1. function Setup(DB):
return (pk,sk)=FHE.Keygen() //生成公、私秘钥对

2. function Query(pk,idx,n):
for i=0 to n−1 do
ci=Enc(pk,i==idx? 1:0)
return q=$\{c_0,c_1,\cdots,c_{n-1}\}$

3. function Answer(q=$\{c_0,c_1,\cdots,c_{n-1}\}$,DB):
for i=0 to n−1 do
a_i=FHE.Const_Mult(DB_i,c_i)　//明文*密文
return a=FHE.Add($a_0,a_1,\cdots,a_{n-1}$)　//密文+密文

4. function Extract(sk,a):
Return FHE.Dec(sk,a)

2018 年,谷歌提出了基于同态加密的 PSIR[30]协议,引入了具有“状态”信息的查询方,且该信息查询方可以自行更新,在无法恢复的情况下,由被查询方更新。PSIR 协议原理如表 4 所示,首先查询方和被查询方分别初始化,查询方输入安全参数 λ 和可以存储记录数量 c,输出初始状态 st,被查询方输入安全参数 λ 和包含 n 条数据的密态数据库 $D=(B_1,\cdots,B_n)$,不输出。然后,查询方输入查询索引 $q\in n$ 和当前状态 st,基于加密执行 PSIR.Query,将输出 Query 发送给被查询方,更新状态为 st。被查询方将接收到的 Query 作

为输入，和密态数据库 D 进行同态计算，并将结果返回给查询方；最后查询方进行 Extract 操作，对结果解密。查询方和被查询方执行 PSIR. UpdateState 更新各自的状态。

表 4　PSIR 协议框架

查询方：R；被查询方：S
R：st = PSIR. Init(1^{λ}，1^{c}) S：⊥ = *PSIR. Init* (1^{λ}，D)
R：(*st′*，*Query*) = *PSIR. Query*(*q*，*st*)
S：*Reply* = *PSIR. Reply*(*Query*，*D*)
R：*B* = *PSIR. Extract*(*Reply*，*st′*)
R：*st″* = *PSIR. UpdateState*((*st′*)，(*D*)) *S*：⊥ = *PSIR. UpdateState*((*st′*)，(*D*))

试验结果表明，PSIR[30] 具有较大的效率优势：数据库大小在 100 k ~ 1.288 M 范围内，总体性能比 SEALPIR[28] 高；与 SEALPIR 相比，使用 Seal 库加密的 PSIR（SealPSIR），被查询方速度至少提升 4.5 倍，使用 Paillier 加密的 PSIR（PaillierPSIR），在线带宽至少降低 5~10 倍，总带宽降低 1.3~4 倍。

2.3　同态加密在多方计算中的应用

为了解决多方计算场景中云端数据的安全性，数据提供方通常是先将数据加密后上传到中心代理服务器上，基于密文进行数据的密态计算。

图 4 给出一个基于同态加密的金融机构数据共享方案[31]。首先，金融认证中心生成身份凭证码和数字签名密钥以及金融秘钥管理中心生成同态密钥分配给各个机构，然后各个机构加密自己的数据信息提交给信贷统计中心，之后信贷统计中心对数据验签后进行密态计算。具体而言，是将所有 ID 相同的记录（即相同用户）求和，然后将计算结果返回给金融机构，最后各个机构对结果验签，解密得到所需的共享数据。

此外，金融业常见的多头共债问题也可通过基于同态加密的联合计算来解决。多头共债指借款人在多个平台上同时存在债务，给平台造成了过度授信的风险。假设有由授信评估方发起方提供要查询的用户 ID 和其偿债能力 r，共有 K 个服务方（金融机构、平台）参与风险评估，则每个服务方 k 提供该用户的贷款

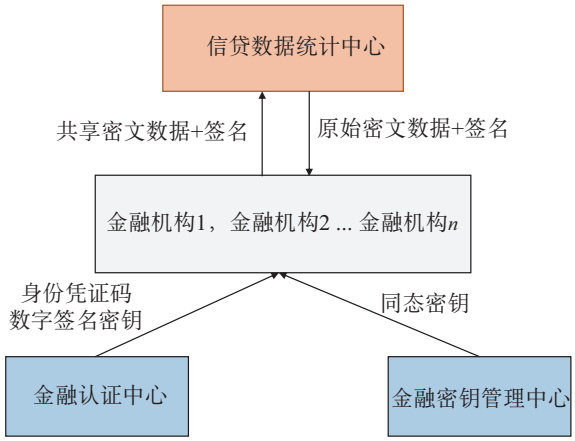

图 4　基于同态加密的金融机构数据共享方案

数据 $l^{(k)}$，这些数据经同态加密后发送给第三方进行风险计算 $[T] = \sum_{k=1}^{K} [l^{(k)}] - [r]$ 将密文结果发送给发起方，发起方解密后根据 T 是否大于 0 来判断是否存在信贷风险。

这种通过将多方数据汇总到中心代理计算服务器上然后利用同态加密进行密态计算的方法，既能保证数据安全性又实现了各方所需计算。除了上述应用于数据共享（统计）场景外，可以应用于多方的信息检索、联合计算、电子投票等诸多场景，具有较高的实用性。

2.4　同态加密在多方联合建模中的应用

多方联合建模的主要目标是在不泄露任何隐私的前提下，结合多方数据以提高模型效果。例如，医院之间可以联合诊断疾病，银行和保险公司可以联合进行反欺诈，电商之间可以联合从用户购买行为中学习模型从而改进推荐。将不同机构的数据在合规的前提下进行联合建模，将会获得更大收益。

同态加密应用在分布式联合建模场景时，对经过本地建模得到需要交互的中间态数据进行加密、传输，在密文上进行统计计算（见图 5）。该架构能够同时保证输入数据的隐私性和计算结果的准确性，从而广泛应用于联邦学习的建模场景。

同态加密在联邦学习中的应用，多数采用安全聚合的方式。例如，在横向联邦学习建模过程中，利用加法同态加密（Additive Homomorphic Encryption，AHE）实现密文下的梯度聚合[32]，该方法适用于 LWE、

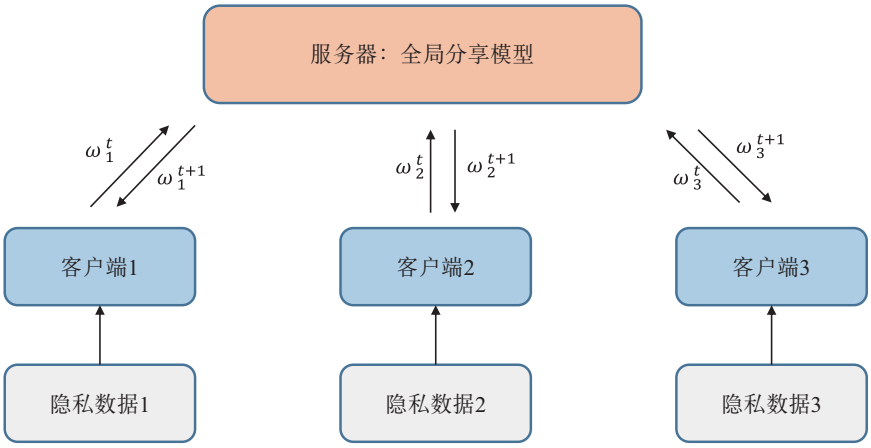

图 5　基于同态加密的参数聚合基本架构

Paillier 等同态加密机制，可视为一种参数聚合的基本方案。Zhang 等[33]在此基础上，提出了一种梯度量化的方式，并对量化梯度进行批量编码再进行同态加密的操作，减少了加密次数和密文数量，从而提升了效率。谷歌的 McMahan 等[34]于 2017 年提出的联邦平均算法（Federated Averaging，Fed AVG）适用于所有有限加和形式的损失函数，其重点在于将各用户自己训练的权重整合起来进行平均，在不同参与方的数据集非独立同分布的情况下，可以用这一形式来融合模型。将 AHE 加入到 Fed AVG 算法中，可以为中间梯度信息提供安全保护。Liu 等[35]应用 AHE 提出了一种安全的、基于特征的联邦迁移学习框架，可用于训练神经网络。

下面以实际业务应用中常见的纵向联邦逻辑回归（Logistic Regression，LR）算法和纵向极端梯度提升（eXtreme Gradient Boosting，XGBoost）算法为例，分别介绍基于同态加密以及同态加密与秘密共享（Secret Sharing，SS）融合的联合建模。

2.4.1　基于同态加密的联合建模

假设某地的 A 银行想要建立个人信用风险评估模型，为了改善模型表现与当地的一家 B 电商合作进行模型训练。两家的用户群体重叠较大而特征交集较小。设 A 银行持有特征 x_A 和标签 y，B 电商持有特征 x_B，两方的样本已进行安全对齐。

可以采用逻辑回归建立风险评估建模，则基于 Paillier 同态的两方纵向 LR 建模流程如图 6 所示[36]。主要的实现机制是对模型建模过程的中间计算结果（如 Loss、梯度等），经过同态加密后在参与方之间交互和密态计算来完成建模任务。进一步地，可以将 Paillier 同态加密替换为满足加性同态加密的其他加密方案，如 CKKS、OU 等。也可对流程做适当推演，扩展到多方纵向 LR 建模。

还可以采用 XGBoost 进行模型训练，Secure Boost 是 2019 年由 Cheng 等[37]提出的联邦学习典型算法，使用同态加密的方案进行纵向 XGBoost 的联邦建模。拥有标签的一方为发起方（A 银行），以及只拥有特征数据的服务方（B 电商）。

基于同态加密的纵向 XGBoost 建模的核心在于改造节点分裂的过程。如图 7 所示，整个流程由发起方主导，大致包括：由发起方计算每个样本的一阶和二阶导数，并将其同态加密为 $[g_i]$ 和 $[h_i]$；将当前分裂结点的实例空间 $I=\{id_1,\cdots,id_n\}$ 和对应的密文导数 $[g_i]$，$[h_i]$ 发送给服务方；服务方在本地枚举持有的特征 f_m 及该特征的分桶信息 $idset_{mn}$，其中 m 和 n 分别表示特征编号和分桶编号（图 8 中以 $m=2$，$n=k$ 为示例），使用同态加法计算分桶内的导数累加，并将所有潜在分裂点的导数累加得到的 $\sum[g_i]$，$\sum[h_i]$ 密文发给发起方；发起方解密得到明文的导数累加，采取 XGBoost 方法中的方式计算增益 $max(gain)$，得到最优分裂结点信息 $argmax(gain)$，包括特征分裂方、特征编号和分桶编号；发起方同步当前结点的最优分裂方，特征编号和分桶编号，最优分裂方记录上述信息，

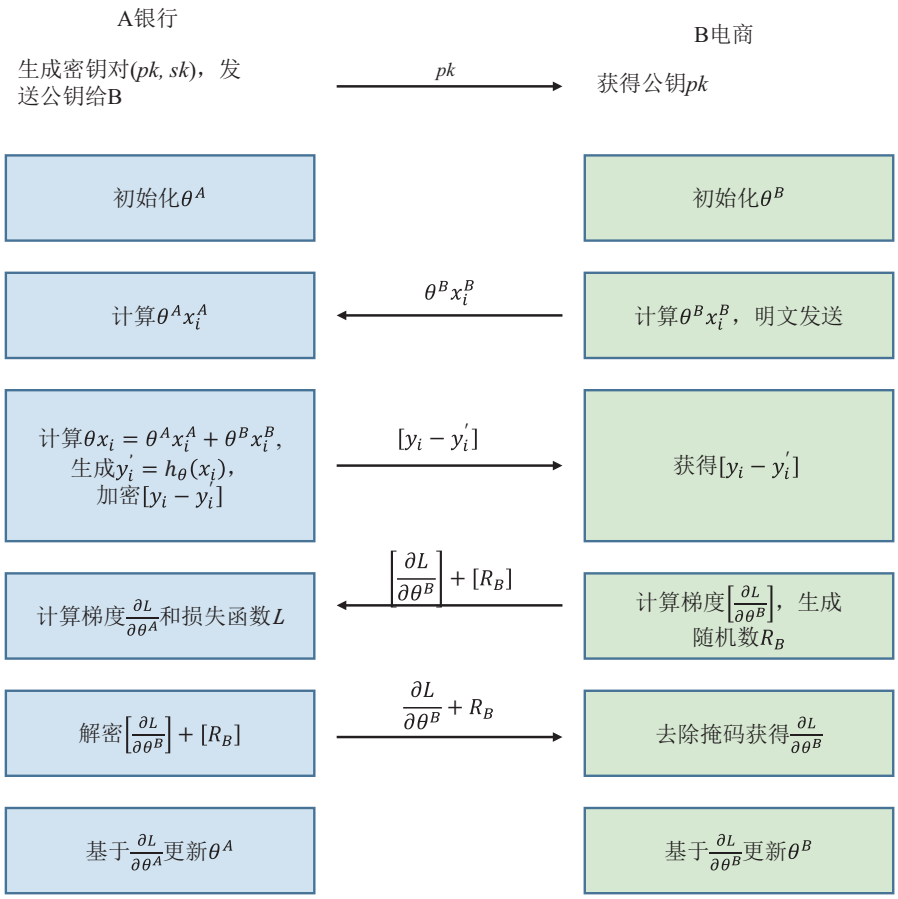

图 6　基于 Paillier 同态加密的两方纵向 LR 建模

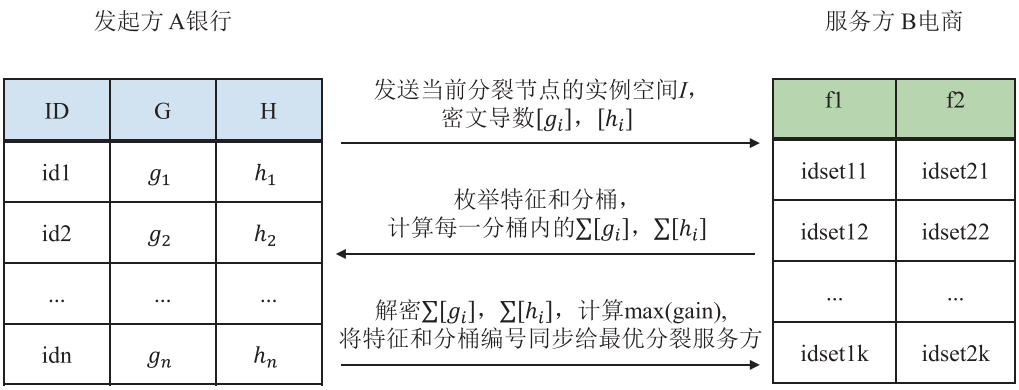

图 7　纵向 XGBoost 算法中的节点分裂过程

并将分裂记录 ID 和样本划分信息返回给发起方；发起方在对应结点记录分裂方和记录 ID 以便后续预测，并生成左右子结点；对子结点递归执行以上步骤 2～6 直至达到停止条件，计算叶结点权重。

2.4.2　基于同态加密与秘密共享融合的模型训练

在基于同态加密的纵向 LR 模型中，每次迭代都会有部分明文的中间信息暴露给其他方，即使在参与者都是半诚实的假设下，模型也存在着一定风

险[38]。而基于秘密共享的纵向 LR 模型难以处理业务场景中存在的高维稀疏数据。为了解决上述问题，Chen 等[39]2021 年提出了一种基于同态加密和秘密共享融合的纵向 LR 方案，其中如何计算模型预测值 $y'=\theta x$ 和逻辑函数是模型更新迭代的关键。算法的基本思想是，A 和 B 在彼此间秘密共享模型，在训练过程中均为密文状态，直到训练结束才会恢复明文。同时，A 和 B 各自保护自己的特征和标签隐私，训练时使用安全矩阵乘法协议来计算 θx，并且用多项式来近似逻辑函数。然后，B 方计算密文的预测值并使用同态加密域的秘密共享协议来进行共享，A 和 B 计算误差份额和梯度份额，其中也应用到了两个基础协议。最后，基于梯度下降来更新各自的模型份额。在以上训练过程中，中间结果都是以同态加密或秘密共享的形式交互的，直至迭代结束后，才会恢复明文的模型。仍然以 A 银行与 B 电商为例，基于同态加密与秘密共享的两方纵向 LR 建模流程如图 8 所示。

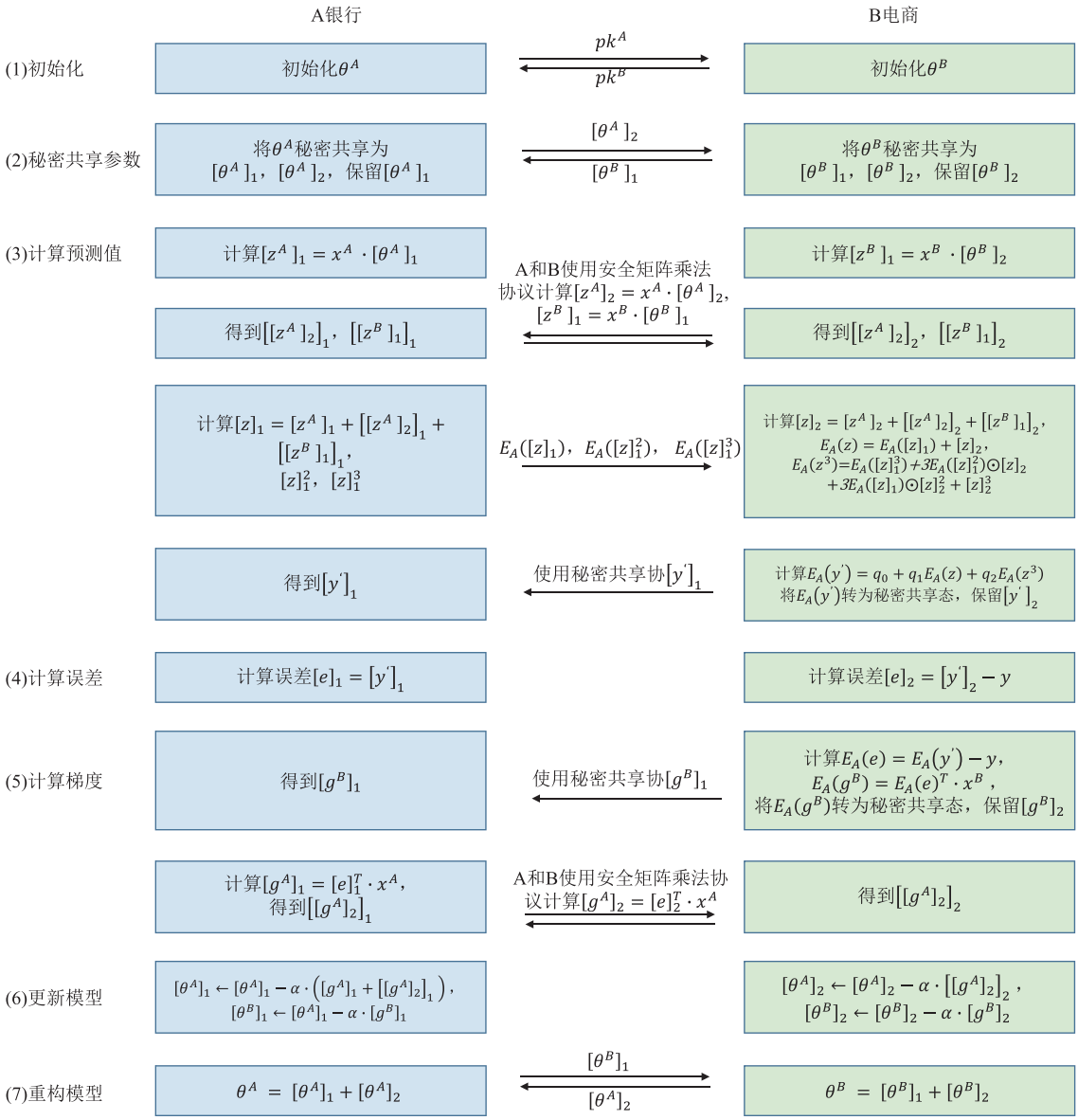

图 8　基于同态加密与秘密共享的两方纵向 LR 建模

同样，同态加密和秘密共享融合的框架也可实现纵向联邦 XGBoost 的建模[40]，在基于秘密共享实现的安全 XGBoost 算法的基础上，将同态加密应用于分桶累积这一步骤，以优化矩阵乘法带来的巨大通信量。

总的来说，同态加密计算和存储的开销较大，而通信量较少，仅应用同态加密的方案适用于算力充足的场景，能实现高效但有一定数据安全风险的模型训练。秘密分享的计算性能和安全性相对较好，但会存在大量的数据传输，在带宽巨大的步骤中引入同态加密来优化，以计算来换通信，即是二者结合的方案，平衡了效率和安全性。在实际应用中，为同时实现效率、安全、精确的目标，密码学技术通常不会单独使用，而是将几种技术结合起来以设计更具实用性的混合协议。

3 同态加密的发展与挑战

同态加密技术经过近几十年的发展，从最开始的半同态方案到全同态方案陆续提出，更好地满足不同应用场景的复杂计算需求，但由于计算资源开销太大，计算效率问题仍然是目前影响同态加密技术在实际应用场景中被采用的一个重要因素。半同态加密能够高效计算，并能支持无限次加法或乘法，目前在隐私计算技术方案中应用较为广泛，成为隐私计算的一个重要基础组件，可辅助完成多种隐私计算功能，例如隐私保护的数据聚合、秘密共享中乘法三元组生成、构造不经意传输协议等特定的隐私保护协议、门限签名、隐私集合求交等。但同时半同态加密因为只支持加法或乘法单一计算，较难实现复杂算法，应用面受限。某些场景虽然通过泰勒展开等方式逼近某个函数来实现复杂计算，但会造成计算精度的损失和计算效率的降低。而全同态加密能够同时支持无限次的加法和乘法，因而能够支持任意的函数更好满足多样性应用场景需求，但全同态加密目前的实现路径主要基于自举，普遍存在效率低的问题，使得全同态加密几乎无法在实际生产环境应用。近年来，学术界致力于研究从密码学层面提高同态加密的效率，例如针对全同态的自举效率问题，AlperinSheriff 和 Peikert[41] 提出的利用对称群和置换矩阵构造快速自举一个比特的同态加密方案，Ducas 和 Micciancio[42] 将该技术扩展到环上实现性能的进一步提升等。虽然目前通过算法优化，同态加密的性能有了很大的优化，但同态加密的计算复杂度仍然很高，与明文的操作仍有很大的性能差异，性能优化仍有很大的提升空间。

与此同时，在产业界除了通过秘密学技术的优化来提升同态加密的性能，也在尝试通过与硬件结合的方式进行加速，以尽快提升同态加密在不同应用场景的可用性。同态加密通过密码学技术以密文的方式进行计算，不可避免地引入大量的计算开销，面对大数据量的应用场景，传统的 CPU 的计算能力难以满足实际应用的高性能要求，而 FPGA 是可以根据需求对底层电路结构进行设计更新的芯片，通过使用 FPGA 内部逻辑资源构建计算电路，例化大量计算引擎，可以提高计算并发度，实现指定算法的加速计算。目前，产业界已经研究基于 FPGA 的同态加密加速方案来大幅度提高计算效率，并已取得显著成果，能够提升 5 倍以上的同态加密计算性能。

4 结束语

随着近年来国内外对数据安全和隐私保护的需求越来越高，同态加密技术不断优化和突破，开始走向商用阶段，在云计算和隐私计算等场景中被逐步应用。同态加密实现密文间的多种计算功能，即先计算后解密得到的结果可以等价于先解密后计算结果，这个特性对于保护敏感数据跨域计算过程的安全具有重要意义。同态加密可适用于金融、医疗、政务等跨机构间的联合查询、联合统计、联合建模、联合预测等多种落地场景。同态加密应用过程可单独使用进行集中式代理计算、分布式的多方联合统计等，也可应用于联邦学习、隐匿查询、安全求交等技术方案作为一个底层支撑密码学技术组件，满足更广泛的多样性应用需求，扩展到更大、更复杂的实际业务场景范围。

目前，同态加密技术的研究重点主要在于性能提升方面：通过密码算法的优化提升同态加密的性能，打破同态加密技术的性能瓶颈；研究软硬结合的性能优化方案，以 GPU、FPGA、ASIC 等硬件技术来推动同态加密性能的提升，快速提高同态加密的实际业务可用性；通过对隐私计算方案的流程优化，对复杂计算进行拆分，融合使用半同态和全同态技术，提升整体运行性能。

同态加密技术可以实现对密文的计算，在隐私计算流程可以减少通信代价，也可以密态转移计算任务，

可通过代理计算等方式平衡各方的计算代价;同态加密技术只有私钥持有方能获知最后的结果,可以提高结果的安全性。由于同态加密在计算复杂性、通信复杂性与安全性上的优势,越来越多的学术界和产业界的力量都投入到其理论和应用的探索中,随着性能的进一步优化提升,必将有着更广阔的应用前景。

参考文献

[1] 刘钦菊, 路献辉, 李杰, 等. 全同态加密自举技术的研究现状及发展趋势[J]. 密码学报, 2021,8(5):795-807. DOI:10.13868/j.cnki.jcr.000477.

[2] RIVEST R L, SHAMIR A, ADLEMAN L M. Cryptographic communications system and method: US Patent 4405829[P], 1983.

[3] GAMAL T E. A public key cryptosystem and a signature scheme based on discrete logarithms [J]. IEEE Transactions on Information Theory, 1984(31):469-472.

[4] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes [J]. EUROCRYPT'99, Czech Republic, May, 1999.

[5] GENTRY C. Fully homomorphic encryption using ideal lattices[C]. The Forty-First Annual ACM Symposium on Theory of Computing, 2009:169-178.

[6] DIJK M V, GENTRY C, HALEVI S, et al. Fully homomorphic encryption over the integers[J]. Springer, Berlin, Heidelberg, 2010.

[7] BRAKERSKI Z, VAIKUNTANATHAN V. Efficient fully homomorphic encryption from (standard) LWE[J]. SIAM Journal on Computing, 2014,43(2):831-871.

[8] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. (Leveled) fully homomorphic encryption without bootstrapping[J]. ACM Transactions on Computation Theory (TOCT), 2014,6(3):1-36.

[9] FAN J, VERCAUTEREN F. Somewhat practical fully homomorphic encryption [J]. Cryptology Eprint Archive, 2012.

[10] CHEON J H, KIM A, KIM M, et al. Homomorphic encryption for arithmetic of approximate numbers[J]. International Conference on the Theory and Application of Cryptology and Information Security, 2017:409-437.

[11] GENTRY C, SAHAI A, WATERS B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based [J]. Annual Cryptology Conference, 2013:75-92.

[12] ALPERIN-SHERIFF J, PEIKERT C. Faster bootstrapping with polynomial error [J]. Annual Cryptology Conference, 2014: 297-314.

[13] DUCAS L, MICCIANCIO D. FHEW: bootstrapping homomorphic encryption in less than a second [J]. Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015: 617-640.

[14] CHILLOTTI I, GAMA N, GEORGIEVA M, et al. TFHE: fast fully homomorphic encryption over the torus [J]. Journal of Cryptology, 2020, 33(1): 34-91.

[15] CHEON J H, HAN K, KIM A, et al. Bootstrapping for approximate homomorphic encryption [J]. Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2018:360-384.

[16] CHEN H, CHILLOTTI I, SONG Y. Improved bootstrapping for approximate homomorphic encryption [J]. Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019:34-54.

[17] Cryptofun[EB/OL]. [2022-03-01]. https://github.com/arnaucube/cryptofun.

[18] IBM. HELIB [EB/OL]. [2022-03-01]. https://github.com/homenc/Helib.

[19] Microsoft. SEAL[EB/OL]. [2022-03-01]. https://github.com/Microsoft/SEAL.

[20] HEAAN[EB/OL]. [2022-03-01]. https://github.com/snucrypto/HEAAN.

[21] FHEW[EB/OL]. [2022-03-01]. https://github.com/lducas/FHEW.

[22] TFHE[EB/OL]. [2022-03-01]. https://github.com/tfhe/tfhe.

[23] PALISADE [EB/OL]. [2022-03-01]. https://gitlab.com/palisade/palisade-development.

[24] KOLESNIKOV V, KUMARESAN R, ROSULEK M, et al. Efficient batched oblivious PRF with applications to private set intersection [C]. 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016: 818-829.

[25] CHEN H, LAINE K, RINDAL P. Fast private set intersection from homomorphic encryption [C]. 2017

ACM SIGSAC Conference on Computer and Communications Security. 2017:1243-1255.

[26] CHEN H, HUANG Z, LAINE K, et al. Labeled PSI from fully homomorphic encryption with malicious security [C]. 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018: 1223-1237.

[27] CHOR B, GOLDREICH O, KUSHILEVITZ E, et al. Private information retrieval [C]. IEEE 36th Annual Foundations of Computer Science, 1995: 41-50.

[28] ANGEL S, CHEN H, LAINE K, et al. PIR with compressed queries and amortized query processing [C]//2018 IEEE symposium on security and privacy (SP), 2018: 962-979.

[29] Microsoft. SEALPIR [EB/OL]. [2022-03-01]. https://github.com/microsoft/SealPIR.

[30] PATEL S, PERSIANO G, YEO K. Private stateful information retrieval [C]. 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018: 1002-1019.

[31] 王婧琳. 基于同态加密的金融数据安全共享方案研究及实现[D]. 哈尔滨工业大学, 2020.

[32] AONO Y, HAYASHI T, WANG L, et al. Privacy-preserving deep learning via additively homomorphic encryption [J]. IEEE Transactions on Information Forensics and Security, 2017, 13(5): 1333-1345.

[33] ZHANG C, LI S, XIA J, et al. BatchCrypt: efficient homomorphic encryption for cross-silo federated learning [C]//2020 USENIX Annual Technical Conference, 2020: 493-506.

[34] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]//Artificial intelligence and statistics. PMLR, 2017:1273-1282.

[35] LIU Y, KANG Y, XING C, et al. A secure federated transfer learning framework [J]. IEEE Intelligent Systems, 2020,35(4):70-82.

[36] YANG S, REN B, ZHOU X, et al. Parallel distributed logistic regression for vertical federated learning without third-party coordinator [J]. arXiv preprint arXiv: 1911.09824, 2019.

[37] CHENG K, FAN T, JIN Y, et al. Secureboost: A lossless federated learning framework [J]. IEEE Intelligent Systems, 2021,36(6): 87-98.

[38] LI Z, HUAGN Z, CHEN C, et al. Quantification of the leakage in federated learning[J]. arXiv preprint arXiv: 1910.05467, 2019.

[39] CHEN C, ZHOU J, WANG L, et al. When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control [C]//27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021: 2652-2662.

[40] FANG W, ZHAO D, TAN J, et al. Large-scale secure XGB for vertical federated learning [C]//30th ACM International Conference on Information & Knowledge Management, 2021: 443-452.

[41] ALPERIN-SHERIFF J, PEIKERT C. Faster bootstrapping with polynomial error [C]//Annual Cryptology Conference, 2014:297-314.

[42] DUCAS L, MICCIANCIO D. FHEW: bootstrapping homomorphic encryption in less than a second [C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015: 617-640.

作者简介：

邵航 同盾科技有限公司人工智能研究院算法工程师，主要研究领域为同态加密、应用密码学、隐私计算等

高思琪 同盾科技有限公司人工智能研究院算法工程师，研究领域包括统计理论方法、机器学习、联邦学习、多方安全计算等

钟离 同盾科技有限公司人工智能研究院算法专家，目前负责同盾科技多方安全计算平台相关开发工作，研究方向为机器学习、联邦学习、分布式计算及密码学等

傅致晖 同盾科技有限公司人工智能研究院联邦学习算法专家，作为核心研发人员参与知识联邦参考实现智邦平台的开发，并撰写了多篇联邦学习相关专利及论文

孟丹 同盾科技有限公司人工智能研究院联邦学习

部负责人，主要研究领域为隐私计算、联邦学习、多方安全计算等

李晓林 同盾科技有限公司合伙人，副总裁，人工智能研究院院长，中科院医学所智慧医疗首席科学家，知识联邦产学研联盟理事长；首创“知识联邦”理论体系，主要研究领域为隐私计算、机器学习、深度学习、分布式系统、智慧医疗等

Homomorphic encryption protocols and applications in privacy preserving computation

SHAO Hang，GAO Siqi，ZHONG Li，FU Zhihui，MENG Dan，LI Xiaolin

（AI Institute，Tongdun Technology，Hangzhou 311121，China）

Abstract：Homomorphic encryption is a cryptography technology based on the computational complexity theory of mathematical problems. It supports the calculation of data under the encrypted state，and the decrypted calculation result is consistent with the plaintext calculation result. As a hot research topic in privacy preserving computation，homomorphic encryption can be used in diverse application scenarios. We first introduce the development process of homomorphic encryption and relevant technologies. We then elaborate on the use cases of homomorphic encryption in typical application scenarios，such as secure intersection，secure querying，multi-party joint computing，and multi-party joint modeling. Finally，this paper analyzes and discusses the technical challenges and some research directions of current homomorphic encryption technologies.

Keywords：homomorphic encryption；private set intersection；private information retrieval；secure multi-party computation；federated learning；privacy preserving computating

（收稿日期：2022-03-10）