# 综述:基于密码技术的人工智能隐私保护计算模型

田海博,梁岫琪
(中山大学计算机学院,广东广州 510275)

**摘 要:** 人工智能隐私保护的应用场景多种多样. 在不同的场景中,完成隐私保护计算的实体可信程度和数量不尽相同. 这些实体的可信程度和数量对隐私保护计算方法能否实际应用具有重要影响. 本文从实体的可信程度和数量出发,将基于密码技术的人工智能隐私保护计算方法归类为4种计算模型,分别是多中心模型、双中心模型、单中心模型和现实模型. 除现实模型外,其它计算模型都存在可信实体. 对每一种计算模型,本文给出当前基于密码学工具给出的人工智能隐私保护方法涉及的典型计算和采取的典型算法,并指出提升算法的效率和安全性是对每种计算模型都适用的研究方向.

**关键词:** 人工智能;隐私保护;计算模型;算法;协议;密码技术

**基金项目:** 广东省基础与应用基础研究重大项目(No.2019B030302008);广东省重点领域研发计划项目(No.2020B010166005);华为技术有限公司(No.TC20210407007, No.YBN2019105017)

**中图分类号:** TP309.2 **文献标识码:** A **文章编号:** 0372-2112(2023)08-2260-17

**电子学报 URL:**http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210702

# A Survey: Computing Models of Artificial Intelligence Privacy Protection Based on Cryptographic Techniques

TIAN Hai-bo, LIANG Xiu-qi
(*School of Computer Science and Engineering*, *Sun Yat-Sen University*, *Guangzhou*, *Guangdong* 510275, *China*)

**Abstract:** The application scenarios of artificial intelligence privacy protection are diverse. In different scenarios, the trustness and number of entities fulfilling privacy protection computation are different. The trustness and number of these entities have an important impact on the technical choices of privacy protection computation. Starting from the trustness and number of entities, this paper classifies the computation methods of artificial intelligence privacy protection, which are based on cryptographic techniques into four types of computation models: multiple centers model, double centers model, single center model and real model. Except for the real model, there are trusted entities in all other computation models. For each kind of computation model, this paper presents the typical computations and algorithms, which are involved in the current artificial intelligence privacy protection methods based on cryptography tools. And this paper also points out that improving the efficiency and security of algorithms is an applicable research direction for each model.

**Key words:** artificial intelligence; privacy protection; computation models; algorithms; protocols; cryptographic techniques

**Foundation Item(s):** Guangdong Major Project of Basic and Applied Basic Research (No.2019B030302008); Key-Area Research and Development Program of Guangdong Province (No.2020B010166005); Huawei Technologies Co., Ltd. (No.TC20210407007, No.YBN2019105017)

## 1 引言

人工智能隐私保护具有现实需求. 人工智能计算的应用场景非常丰富. 例如银行可以通过用户的数据建立模型,然后用该模型对用户的信用进行评估. 银行和电信公司可以通过两方的数据建立更好的模型,从而对用户信用进行更为准确的评估. 多家医院可以通过各自用户的数据建立关于疾病的模型,以更好地完成诊疗服务. 在不同的应用场景中,对模型、数据隐私保护的需求有所不同. 模型隐私保护的需求往往来自其商业价值,因为一个预测准确率高的模型往往是经过大量的资金和人力投入获得的,并且可以作为一种服务[1]提供给消费者或者合作伙伴. 数据隐私保护的

收稿日期:2021-06-04;修回日期:2022-07-25;责任编辑:王天慧

需求实际来自各种法律法规. 我国在《中华人民共和国民法典》和《中华人民共和国网络安全法》中明确了个人隐私数据和网络运营者使用这些数据时应当遵循的法律条文. 这迫使数据提供方对数据附有各种使用策略[2],以满足法律法规的要求.

人工智能隐私保护的计算主要指在保证模型、数据隐私的条件下,完成模型的训练或者在已有模型上完成预测. 完成隐私保护计算的实体数量和可信程度是不同的. 例如银行和电信公司是两个参与方,为了进行隐私保护的计算,一种方法是引入一个隐私计算服务提供方和一个密钥分配服务提供方. 假设密钥分配和隐私计算两个服务提供方不勾结,可以实现基于同态加密等技术的双参与方、双服务提供方的隐私保护计算. 又例如多家医院形成多个参与方,为了进行隐私保护的计算,一种方法是引入一个联邦平均算法服务提供方. 假设该服务提供方是半诚实的,可以实现多参与方、单服务提供方且能保护用户数据隐私的横向联邦学习.

多个参与方完成的计算形成了多方计算的场景. 密码学领域中的安全多方计算[3]是一种自然的技术选择. 安全多方计算使$n$个互不信任的参与方$[P_1,P_2,\cdots,P_n]$能够完成某个函数$F$的计算任务$(y_1,y_2,\cdots,y_n)=F(x_1,x_2,\cdots,x_n)$,其中$x_i$是参与方$P_i$的隐私数据,$1\leqslant i\leqslant n$,$F$代表任意可以在图灵机模型下计算的函数,$(y_1,y_2,\cdots,y_n)$是该函数的输出结果[4]. 在计算任务完成后,参与方$P_i$能获得输出$y_i$,同时不泄露各自的隐私数据$x_i$. 多个参与方通过安全多方计算可以在保护数据、模型隐私的条件下,完成函数$F$为训练或者预测的计算.

安全多方计算定义了理想模型以明确安全的内涵[4]. 在理想模型中,有一个可信第三方TTP(Trusted Third Party),每个参与方把隐私数据安全的传输给该TTP,由TTP运行函数$F$,得到输出$(y_1,y_2,\cdots,y_n)$,然后把输出$y_i$安全地发给参与者$P_i$,$1\leqslant i\leqslant n$. 现实中很难找到这样的TTP,因此安全多方计算的主要任务是提供模拟TTP功能的方法. 安全多方计算从姚先生的百万富翁问题开始,经过40年的发展,已经形成了以混淆电路、不经意传输、GMW(Goldreich,Micali,Wigderson)协议、BGW(Ben-Or,Goldwasser,Wigderson)协议、选择-分割协议、零知识证明等为基础的完整技术路线[5]. 这些技术在半诚实攻击者或者恶意攻击者条件下可以让各个参与方直接模拟TTP的功能,得到相应的输出.

在人工智能隐私保护方面,安全多方计算理论与实际需求之间还存在较大的差距. 人工智能隐私保护中,需要保护的数据规模往往过万,需要保护的模型参数规模经常超过百万. 当我们将小规模数据上运行良好的安全多方计算协议放到人工智能隐私保护的实际环境中时,这些协议的通信量或者计算量过大,导致协议的性能往往不能满足实际需求. 我们注意到当前的人工智能隐私保护计算在事实上采用了折衷的思路,即引入服务提供方,并对服务提供方进行一定的置信假设,以得到满足实际需求的解决方案. 在这些解决方案中,服务提供方的数量和可信程度有所不同. 然而,其任务却是相似的. 借用安全多方计算的概念,可以说服务提供方模拟了TTP的功能,为参与方提供隐私保护的计算. 例如,一个双参与方、双服务提供方的隐私保护计算,可以看成两个互不信任的参与方,通过两个服务提供方模拟的TTP完成函数$F$为某种人工智能算法的计算过程.

基于上述事实,我们认为对当前的人工智能隐私保护方案,根据服务提供方的数量和可信程度,可以划分为若干计算模型,进而梳理不同计算模型下进行的典型计算和完成计算任务时基于密码学工具的典型算法,可以让读者从实际需求出发,快速了解某类计算模型的当前进展和主要采用的密码技术,为进一步研究或者应用做好准备.

具体地,我们将当前人工智能隐私保护的计算归类为多中心、双中心、单中心模型和现实模型4类. 现实模型中不存在服务提供方及其置信假设. 其他各类计算模型的非正式描述如下.

(1)多中心模型:存在多个服务提供方$[S_1,S_2,\cdots,S_m]$,$m>2$,合作完成$F$函数的计算.

(2)双中心模型:存在两个服务提供方$S_1$和$S_2$,合作完成$F$函数的计算.

(3)单中心模型:存在一个服务提供方$S$,完成函数$F$的计算.

这些服务提供方的置信假设包括不勾结、可信、半诚实等. 例如密钥分配中心一般是可信的,不会与其他服务提供方勾结;多个服务提供方中有若干是可信的,不会泄漏用户数据等. 在具体的应用场景中,有些假设是可以满足的. 例如多个有竞争关系的公司联合为用户提供某种服务,根据用户自身对公司的信赖程度,可以实现多个服务提供方中有若干是可信的这一假设. 有些假设则不容易满足,例如可信的密钥分配中心. 因此,隐私保护方案所属的计算模型可以反映其在某些场景中是否能实际落地,以及是否能完成隐私保护的任务.

## 2　问题与挑战

目前,人工智能的隐私保护计算已有较多的高质量综述. 这些综述大多以隐私保护方案采取的不同技术和策略或应用场景中的计算任务来分类,并对每一类的当前进展给出了详细、全面的描述.

文献[6]在“模型隐私风险与保护”一节,把隐私保护方法分为基于差分隐私和基于密码学2类,其中密码学技术主要涉及同态加密和安全多方计算. 文献[7]按照差分隐私、同态加密和安全多方计算技术详细地介绍了当前的隐私保护方案. 在“机器学习隐私保护方案的分类”一节,该文按照模型训练方式,把隐私保护方案分为集中式、分布式和联邦学习3类. 文献[8]在“机器学习中的隐私防御方案”一节,按照基于扰动、近似、泛化、对抗和本地5种策略总结了隐私保护防御方案,其涉及的技术包括差分隐私、同态加密、L2正则化、对抗网络、安全协议等. 文献[9]把隐私保护机器学习方案分为使用密码学方法的方案和使用数据扰动方法的方案2类:对于采用密码学方法的解决方案,进一步按照同态加密、Garbled电路、秘密分享和安全处理器4种技术对现有典型方案进行了梳理;对于采用数据扰动方法的解决方案,进一步按照差分隐私、本地扰动、降维3种技术对现有典型方案进行了梳理. 其中差分隐私主要是加噪声,本地扰动主要是反馈随机响应,降维主要是对数据源变换以降低数据的维度.

对于深度学习计算任务,文献[10]把隐私保护计算涉及的技术分为同态加密、安全多方计算和差分隐私3类,并对每一类中的隐私保护深度学习方案进行了描述. 对于推荐系统计算任务,文献[11]把隐私保护计算涉及的技术分为同态加密、安全多方计算、秘密分享和零知识证明4类,其中零知识证明主要是在基于位置的推荐系统中,用于匿名的证明用户的身份. 对于基因检测计算任务,文献[12]把隐私保护计算涉及的技术分为同态加密、Garbled电路、不经意传输、隐私信息提取和加密的有限自动机几类,并总结了该类计算任务涉及的具体计算函数,包括编辑距离、疾病易感性、身份测试、族系测试、亲子鉴定、个人医疗和基因匹配等.

当前综述主要着眼于人工智能隐私保护方案和人工智能的计算任务,从技术、策略、应用的角度进行了分类总结. 然而计算模型对实际的生产活动有直接的影响. 计算模型下的假设与实际的人工智能隐私保护方案的应用场景是否匹配,决定了该类计算模型下的方案能否实际部署. 因此,本文力图从计算模型的角度,对不同隐私保护方案和计算任务进行整理,给出当前各种计算模型下隐私保护方案完成的主要计算任务和采取的主要算法,并对一些采取密码学工具的算法进行原理性分析和性能分析,以展示当前的技术进展. 在对各种计算模型下的隐私保护方案进行梳理之后,本文指出提升隐私保护算法的效率和安全性是对各类计算模型都有益的研究方向.

## 3 研究现状分析

本节按照4类计算模型,梳理当前的人工智能隐私保护方案涉及的计算任务和典型的基于密码学工具的算法. 表1总结了本节用到的各种符号及其含义.

**表1 本文所用符号及其含义**

| 符号 | 含义 |
|---|---|
| $[P_1,P_2,\cdots,P_n]$ | 表示含有$n$个参与者的列表 |
| $[S_1,S_2,\cdots,S_m]$ | 表示含有$m$个服务提供方的列表 |
| $x_i$ | 参与者$P_i$的隐私数据 |
| $x_{ij}$ | $x_i$的秘密份额,给$S_j$或$P_j$ |
| $\mathrm{sx}_{ij}$ | $x_i$的加性秘密分享份额,给$S_j$或$P_j$ |
| $\mathrm{sp}_{ij}$ | $x_i\cdot x_i'$的加性秘密分享份额,给$S_j$或$P_j$ |
| PRF | 伪随机数生成函数 |
| OT | 不经意传输 |
| $k$ | 数据的比特长度 |
| $z\gg k$ | 整数$z$右移$k$位 |
| $x_{ij}^{(t)}$ | $x_i$的秘密份额,给$S_j$或$P_j$,门限为t |
| $r_j^{(l)}$ | 随机数$r$的第$l$比特份额,给服务提供方$S_j$ |
| $x_{>j}$ | 服务提供方$S_j$关于$x_i>x_i'$的秘密份额 |
| $x_{-j}$ | 服务提供方$S_j$关于$x_i'-x_i$差的秘密份额 |
| $x_{\pm j}$ | 服务提供方$S_j$关于$x$的符号的秘密份额 |
| $k_w^y$ | Garbled电路中比特y对应的线密钥 |
| $\mathrm{LSB}(k_w^y)$ | 线密钥$k_w^y$的第0比特 |
| $\{x_i\}$ | 数据$x_i$的密文 |
| $\pi$ | 随机置换函数 |
| $G,\ g,\ q$ | $G=<g>$为循环群,$q$为群的阶 |
| $\mathrm{sk}_{i,u}$ | 参与者$P_i$和$P_u$的共享密钥 |

### 3.1 多中心模型

多中心模型的主要内容是用超过两个服务提供方模拟安全多方计算TTP的功能,以实现隐私保护计算的目的. 目前的隐私保护方案最多使用三个服务提供方,因此下面给出三中心模型的定义,一般性定义可以类似地扩展.

**定义1** (三中心模型)三中心模型的参与者记为$P_1,P_2,\cdots,P_n$,服务提供方记为$S_0,S_1,S_2$. 对于$1\leqslant i\leqslant n$和$0\leqslant j\leqslant 2$,参与者$P_i$向服务提供方$S_j$上传秘密份额$x_{ij}$. 在参与者的秘密份额之上,3个服务提供方合作完成某个计算函数$(y_1,y_2,\cdots,y_n)=F(x_1,x_2,\cdots,x_n)$,并将输出$y_i$返回给$P_i$.

三中心模型中,参与者通过秘密分享技术把自己的数据以秘密份额的形式上传到3个服务提供方,然后3个服务提供方在秘密份额的基础上完成隐私保护的计算. 3个服务提供方显然不能都由攻击者控制. 这意味着在实际的应用中,参与者需要相信$t$个服务提供方是不会泄露隐私数据的,是可信的,其中t是参与者采

用的秘密分享技术的门限值.

一般意义上,多中心模型中服务提供方的计算依赖秘密分享方案. 例如文献[13]给出了ABY$^3$模型,参与者以任意接入结构[14]的秘密分享方案,把自己的隐私数据进行分享. 设参与者$P_i$的隐私数据$x_i \bmod 2^k$为$k$比特的整数,算法1给出了分享的过程,算法1的输出$x_{ij}$上传给$S_j$,$0 \leqslant j \leqslant 2$.

**算法1　参与者分享隐私数据算法**

**输入**:参与者$P_i$的$k$比特隐私数据$x_i \bmod 2^k$

**输出**:给服务提供方$S_j$的秘密份额$x_{ij}$,$0 \leqslant j \leqslant 2$

1. $P_i$随机选取$\mathrm{sx}_{i0}, \mathrm{sx}_{i1} \bmod 2^k$,计算$\mathrm{sx}_{i2} = x_i - \mathrm{sx}_{i0} - \mathrm{sx}_{i1} \bmod 2^k$.

2. $P_i$生成秘密份额$x_{i0} = (\mathrm{sx}_{i0}, \mathrm{sx}_{i1}), x_{i1} = (\mathrm{sx}_{i1}, \mathrm{sx}_{i2}), x_{i2} = (\mathrm{sx}_{i2}, \mathrm{sx}_{i0})$.

3. 输出$x_{ij}$,$0 \leqslant j \leqslant 2$.

根据算法1的分享过程,显然任意两个服务提供方都可以恢复秘密,所以是一种(2,3)门限的接入结构. 文献[15]给出了Sharemind框架,参与者按照加性秘密分享把自己的秘密分为3个份额,分别给3个服务提供方,形成(3,3)门限的接入结构.

服务提供方在参与者的份额上,可以完成加法和乘法计算. 其中2个秘密值的和的份额、公开常数与秘密值的积的份额等运算都可以在本地完成,无须通信. 乘法的运算则与秘密分享方案相关. 当秘密分享方案为任意接入结构[14]时,假设需要计算$x_i \cdot x_i'$的积的秘密份额,可以列出式(3-1),其中第$j$行的计算可以由服务提供方$S_j$单独完成无须通信. 因此,三方事实上就各自得到了一个(3,3)的关于$x_i \cdot x_i'$的积的份额$\mathrm{sp}_{ij}$,$0 \leqslant j \leqslant 2$,满足$x_i \cdot x_i' = \mathrm{sp}_{i0} + \mathrm{sp}_{i1} + \mathrm{sp}_{i2}$. 这3个份额经过0的(3,3)份额的扰动之后,由服务提供方$S_j$发送给$S_{j'}$,$j' = j + 1 \bmod 3$,就完成了(2,3)门限的$x_i \cdot x_i'$的积的份额的计算.

$$\begin{aligned} x_i x_i' &= \left(\mathrm{sx}_{i0} + \mathrm{sx}_{i1} + \mathrm{sx}_{i2}\right)\left(\mathrm{sx}_{i0}' + \mathrm{sx}_{i1}' + \mathrm{sx}_{i2}'\right) \\ &= \mathrm{sx}_{i0}\,\mathrm{sx}_{i1}' + \mathrm{sx}_{i1}\mathrm{sx}_{i0}' + \mathrm{sx}_{i0}\mathrm{sx}_{i0}' \\ &\quad + \mathrm{sx}_{i1}\,\mathrm{sx}_{i2}' + \mathrm{sx}_{i2}\,\mathrm{sx}_{i1}' + \mathrm{sx}_{i1}\mathrm{sx}_{i1}' \\ &\quad + \mathrm{sx}_{i2}\,\mathrm{sx}_{i0}' + \mathrm{sx}_{i0}\mathrm{sx}_{i2}' + \mathrm{sx}_{i2}\mathrm{sx}_{i2}' \end{aligned} \tag{1}$$

当把$k$比特整数看成定点数时,设包含$d$比特的小数部分,则乘法操作还需要除以$2^d$,以保持定点数的性质. 假设两个随机数$r = \mathrm{sr}_0 + \mathrm{sr}_1 + \mathrm{sr}_2$和$r/2^d = \mathrm{srd}_0 + \mathrm{srd}_1 + \mathrm{srd}_2$. 服务提供方$S_j$有这2个随机数的秘密份额$(\mathrm{sr}_j, \mathrm{srd}_j, \mathrm{sr}_{j'}, \mathrm{srd}_{j'})$,$j' = j + 1 \bmod 3$. 设0的(3,3)份额在$S_j$处为$\mathrm{sz}_j$. 那么在得到$x_i \cdot x_i'$直接乘积未经缩放的秘密份额$\mathrm{sp}_{ij}$后,还需要根据算法2进行缩放,以满足定点数的要求. 当2个包含$k$比特定点数的向量做内积时,也可以按照式(1)的方式先在本地计算每一个分量的(3,3)份额,做本地加和之后得到最终内积的(3,3)份额,然后再按照上述过程除以$2^d$,得到内积的最终份额.

当秘密分享方案为加性秘密分享时,文献[15]采取了先把加性分享的份额转化为任意接入结构的份额,然后按照式(1)进行计算的方法. 文献[16]则采用了Beaver预计算的方法. 假设整数$c = ab$,且服务提供方$S_j$已经有了$a, b, c$的份额$a_j, b_j, c_j$. 另外服务提供方$S_j$有整数$x_i$的份额$\mathrm{sx}_{ij}$和整数$x_i'$的份额$\mathrm{sx}_{ij}'$. 为计算$x_i \cdot x_i'$,服务提供方$S_j$只需要计算$\mathrm{sx}_{ij} - a_j$和$\mathrm{sx}_{ij}' - b_j$,然后服务提供方恢复$x_i - a$和$x_i' - b$的值. 通过式(2)可知,此时服务提供方$S_j$可以本地计算$x_i x_i'$的份额$c_j + a_j(x_i' - b) + b_j(x_i - a) + (x_i - a)(x_i' - b)$.

$$\begin{aligned} x_i x_i' &= \left(x_i + a - a\right)\left(x_i' + b - b\right) \\ &= ab + a\left(x_i' - b\right) \\ &\quad + b\left(x_i - a\right) + \left(x_i - a\right)\left(x_i' - b\right) \end{aligned} \tag{2}$$

从上面的算法可以看到,秘密份额上的乘法运算需要0的秘密份额或者乘法预计算的份额. 对于任意接入结构的秘密分享方案,假设服务提供方$S_0, S_1, S_2$分别拥有了份额$x_{i0}, x_{i1}, x_{i2}$,文献[17]给出了伪随机数秘密份额的生成算法,如算法3所示. 该算法中$x_{ij}'$是一个新的随机数$x_i' = \mathrm{sx}_{i0}' + \mathrm{sx}_{i1}' + \mathrm{sx}_{i2}'$在服务提供方$S_j$处的秘密份额.

**算法2　秘密份额定点数乘法中积的缩放算法**

**输入**:服务提供方$S_j$的秘密份额$(\mathrm{sp}_{ij}, \mathrm{sz}_j, \mathrm{sr}_j, \mathrm{srd}_j, \mathrm{sr}_{j'}, \mathrm{srd}_{j'})$,$j' = j + 1 \bmod 3$,$0 \leqslant j \leqslant 2$

**输出**:$x_i \cdot x_i'$定点数积的秘密份额

1. 服务提供方$S_j$公开$\mathrm{sp}_{ij} + \mathrm{sz}_j - \mathrm{sr}_j$.

2. 所有服务提供方一起恢复$\mathrm{tmp} = x_i \cdot x_i' - r$.

3. 服务提供方$S_j$计算$\mathrm{tmp}/2^d + \mathrm{srd}_j$和$\mathrm{tmp}/2^d + \mathrm{srd}_{j'}$.

4. 输出$(\mathrm{tmp}/2^d + \mathrm{srd}_j, \mathrm{tmp}/2^d + \mathrm{srd}_{j'})$.

**算法3　伪随机数秘密份额的生成算法**

**输入**:服务提供方$S_j$拥有$x_{ij} = (\mathrm{sx}_{ij}, \mathrm{sx}_{ij'})$,$j' = j + 1 \bmod 3$,执行轮数$z$

**输出**:$x_{ij}' = (\mathrm{sx}_{ij}', \mathrm{sx}_{ij'}')$

1. 服务提供方$S_j$通过一个伪随机数生成函数PRF计算$\mathrm{sx}_{ij}' = \mathrm{PRF}_{\mathrm{sx}_{ij}}(z)$和$\mathrm{sx}_{ij'}' = \mathrm{PRF}_{\mathrm{sx}_{ij'}}(z)$,$j' = j + 1 \bmod 3$.

2. 输出$x_{ij}' = (\mathrm{sx}_{ij}', \mathrm{sx}_{ij'}')$.

算法3的原理在于,对任意接入结构的秘密分享,所有人的秘密份额数量的和一定是最大不合格接入结构集合规模的倍数,从而所有秘密份额的异或最终会导致0的出现. 例如,$\mathrm{sx}_{i0}' \oplus \mathrm{sx}_{i1}'$,$\mathrm{sx}_{i1}' \oplus \mathrm{sx}_{i2}'$和$\mathrm{sx}_{i2}' \oplus \mathrm{sx}_{i0}'$是0的3个秘密份额. 文献[13]采用了这样的方法. 乘法预计算的份额在文献[15]中采用了算法4来完成.

特别需要注意的是,式(1)和式(2)所展示的算法适用于多中心模型,并不限于3个服务提供方. 例如采用(3,5)门限的一般接入结构秘密分享方案,式(1)的计算方法依旧成立. 事实上,当服务提供方的数量较多时,有一种更为高效的计算乘法的方法. 当秘密分享方案为Shamir秘密分享时,文献[18]给出了采用两种门

**算法4 乘法预计算算法**

**输入:**分布式全同态加密公钥

**输出:**乘法预计算的秘密份额

1. 服务提供方$S_j$生成随机数$a_j, b_j, r_j$,使用公钥进行全同态加密后形成密文$\{a_j\}, \{b_j\}, \{r_j\}$,并公开.

2. 服务提供方$S_j$分别聚合密文形成$\{a\}, \{b\}$和$\{r\}$,计算$\{a\}, \{b\}$的乘积得到$\{ab\}$,计算$\{ab-r\}$.

3. 服务提供方对密文$\{ab-r\}$联合解密并公开结果.

4. 如果$j=0$,服务提供方$S_j$设置$c_j=ab-r+r_j$,否则设置$c_j=r_j$.

5. 输出$c_j$.

限值来计算乘法的方法. 假设较小的门限为$t$,服务提供方$S_j$有$x_i$的$t$门限份额$x_{ij}^{(t)}$和$x_i'$的t门限份额$x_{ij}'^{(t)}$,另有同一个随机数$r$的$t$门限份额$r_j^{(t)}$和$2t$门限份额$r_j^{(2t)}$. 此时乘法份额的计算如算法5所示. 算法5在三中心模型中不能使用,因为门限太小,所以秘密份额就是秘密值本身. 然而,对于更为一般的多中心模型,并且容忍比较小的门限时,该方法只需要一次秘密恢复的操作,消耗一对随机份额,就可以完成一次乘法.

**算法5 基于门限值的乘法算法**

**输入:**服务提供方$S_j$的秘密份额$(x_{ij}^{(t)}, x_{ij}'^{(t)}, r_j^{(t)}, r_j^{(2t)})$

**输出:**服务提供方$S_j$关于$x_ix_i'$的t门限秘密份额

1. 服务提供方$S_j$计算$x_{ij}^{(t)}x_{ij}'^{(t)}+r_j^{(2t)}$.

2. 不少于门限$2t$的服务提供方恢复$\mathrm{tmp}=x_ix_i'+r$.

3. 输出$\mathrm{tmp}-r_j^{(t)}$.

在秘密份额的基础上完成一些复杂运算需要的通信量较高. 以比较运算为例,假设服务提供方$S_j$拥有某种秘密分享方案下$k$比特$x_i$的份额$x_{ij}$和同样长度的$x_i'$的份额$x_{ij}'$,现在服务提供方希望得到$x_i>x_i'$的份额$x_{>j}$. 文献[19]给出了计算方法:首先$S_j$计算$x_{ij}'-x_{ij}$得到关于$x_i'-x_i$的份额$x_{-j}$,之后根据算法6得到该差的符号的份额$x_{\pm j}$,进而得到$x_i>x_i'$的份额$1-x_{\pm j}$. 为了计算$x_{\pm j}$,服务提供方需要预计算$k$个二进制随机数的份额,一个随机数$\alpha$的份额和随机数$\beta\in\{+1,-1\}$的份额. 设服务提供方$S_j$参与预计算后获得$r_j^{(1)}, r_j^{(2)}, \cdots, r_j^{(k)}, \alpha_j$和$\beta_j$.

不考虑预计算的代价,算法6需要2次秘密恢复、一次连乘和$k$次乘法. 因为一次乘法需要至少一次通信,考虑并发操作,上述比较协议的通信轮数是$O(\log k)$级别,通信量是$O(k)$级别的. 文献[20]通过采用Legendre符号,得到了通信次数常数轮、通信量为$O\left(\sqrt{\frac{k}{\log k}}\right)$级别的协议,该协议更为高效.

如果可以限制多中心模型中服务提供方的数量,可以用通信次数更少的Garbled电路技术. 文献[13]中限制服务提供方的数量为三,约定采用任意接入结构的秘密分享方案,半诚实模型下,可以在秘密份额的基础上进行Garbled电路的计算. 设秘密为一个比特$b_i=\mathrm{sb}_{i1}\oplus\mathrm{sb}_{i2}\oplus\mathrm{sb}_{i3}$,服务提供方$S_j$拥有份额$b_{ij}=(\mathrm{sb}_{ij}, \mathrm{sb}_{ij'})$,$j'=j+1 \bmod 3$. 设服务提供方$S_0$为计算者,计算$f(b_i)$,服务提供方$S_1$为混淆者,生成$f$的Garbled电路. 文献[13]中给出的算法如算法7所示.

**算法6 基于秘密份额的符号提取算法**

**输入:**服务提供方$S_j$的秘密份额$(x_{-j}; r_j^{(1)}, r_j^{(2)}, \cdots, r_j^{(k)}; \alpha_j; \beta_j)$

**输出:**服务提供方$S_j$的秘密份额$x_{\pm j}$

1. 服务提供方$S_j$本地计算二进制份额$r_j^{(1)}, r_j^{(2)}, \cdots, r_j^{(k)}$对应的$k$比特随机数$r$的份额$r_j$,计算份额$x_{-j}+r_j+2^k+2^k\alpha_j$.

2. 服务提供方一起恢复$c_1=(x_i'-x_i)+r+2^k+\alpha 2^k$,并计算$c_2=\left(x_i'-x_i\right)+r \bmod 2^k$.

3. 服务提供方$S_j$把$c_2$看成$k$个二进制数$c^{(1)}, c^{(2)}, \cdots, c^{(k)}$,按照式(3-3)计算$k+1$个份额$e_j^{(i)}, 1\leqslant i\leqslant k+1$,其中求和符号当下标大于上标时自然认定为0.

$$e_j^{(i)}=\beta_j+r_j^{(i)}-c^{(i)}+3\left(\sum_{l=i+1}^{k}\left(r_j^{(l)}\oplus c^{(l)}\right)\right)$$

$$e_j^{(k+1)}=\beta_j-1+3\left(\sum_{l=1}^{k}\left(r_j^{(l)}\oplus c^{(l)}\right)\right) \tag{3-3}$$

分析式(3-3)可知,如果$c_2\geqslant r$,那么$\beta_j$对应秘密为+1时,$e_j$份额对应的秘密序列$e^{(i)}$中会出现$0, 1\leqslant i\leqslant k$;如果$c_2<r$且$\beta_j$对应的秘密为−1时,该秘密序列$e^{(i)}$中会出现0.

4. 服务提供方通过各自的份额计算并恢复连乘积$\mathrm{tmp}=\prod_{i=1}^{k+1}e^{(i)}$.

5. 如果$\mathrm{tmp}=0$,服务提供方$S_j$设定$h_j=3-\beta_j$,否则设定为$h_j=3+\beta_j$. 可以看到,当$c_2\geqslant r$时,$h_j$份额对应的秘密为2,反之为4.

6. 服务提供方$S_j$计算$x_{\pm j}=(c_2-(x_{-j}+r_j+2^k)+2^{k-1}h_j)/2^k$. 注意到如果$x_{\pm j}$对应秘密为0,则$c_2=x_{-j}+r_j$或者$c_2=x_{-j}+r_j-2^k$,分别有$c_2\geqslant r$和$c_2<r$,从而使得$c_2-(x_{-j}+r_j+2^k)+2^{k-1}h_j$对应的秘密为0,反之亦然.

7. 输出$x_{\pm j}$.

**算法7 基于秘密份额的Garbled电路**

**输入:**服务提供方$S_j$的秘密份额$b_{ij}=(\mathrm{sb}_{ij}, \mathrm{sb}_{ij'})$,$j'=j+1, j=0$或1

**输出:**服务提供方$S_0$输出$f(b_i)$的线密钥

1. 服务提供方$S_1$计算$\mathrm{st}=\mathrm{sb}_{i1}\oplus\mathrm{sb}_{i2}$,然后生成st和$\mathrm{sb}_{i1}$关于$f$的Garbled电路,线密钥分别为$k_w^{\mathrm{st}}, k_w^{\mathrm{sb0}}$和$k_w^{\mathrm{sb1}}$.

2. 服务提供方$S_1$把Garbled电路,线密钥$k_w^{\mathrm{st}}$发送给$S_0$,并由不经意传输协议传输$k_w^{\mathrm{sb}_{i0}}\in\{k_w^{\mathrm{sb0}}, k_w^{\mathrm{sb1}}\}$给服务提供方$S_0$.

3. 服务提供方$S_0$把$k_w^{\mathrm{st}}$和$k_w^{\mathrm{sb}_{i0}}$作为Garbled电路的输入,计算该电路的输出$k_w^y$.

4. 服务提供方$S_0$输出$k_w^y$.

为了把算法7的输出再次转化为秘密份额,可以由服务提供方$S_1$把线密钥$k_w^{y0}$给服务提供方$S_2$,并由服务提供方$S_0$和$S_1$共同生成一个随机比特$r$,然后服务提供方$S_0$计算$\mathrm{LSB}(k_w^y)\oplus r$并将该值发送给服务提供方$S_2$,其中$\mathrm{LSB}(k_w^y)$表示$k_w^y$的第0比特. 此时,3个服务提供方得到了输出比特$y$的秘密份额$y=$

$(\mathrm{LSB}(k_w^y)\oplus r)\oplus(r)\oplus\mathrm{LSB}(k_w^{y0})$,各个服务提供方秘密份额的分配如表2所示.

**表2　Garbled电路输出转为秘密份额的示例**

| $y$ | 秘密份额1 | 秘密份额2 |
|---|---|---|
| 服务提供方$S_0$ | $\mathrm{LSB}(k_w^y)\oplus r$ | $r$ |
| 服务提供方$S_1$ | $r$ | $\mathrm{LSB}(k_w^{y0})$ |
| 服务提供方$S_2$ | $\mathrm{LSB}(k_w^{y0})$ | $\mathrm{LSB}(k_w^y)\oplus r$ |

文献[13]中还给出了一种采用三方不经意传输协议完成$k$比特秘密和二进制秘密相乘的计算方法,其中秘密分享采用的是任意接入结构的秘密分享方案,安全模型为半诚实模型. 该算法采用了一个三方的不经意传输协议. 设定发送方为$S_2$,发送的消息为$m_0$和$m_1$,接收方为$S_1$,辅助方为$S_0$,接收方和辅助方都知道选择的比特$c$,发送方和辅助方有共同的随机比特串$r_0$和$r_1$. 在此设定下,不经意传输如算法8所示.

**算法8　三方不经意传输算法**

**输入:**发送方$S_2$拥有消息$m_0$和$m_1$,接收方$S_1$和辅助方$S_0$共享比特$c$,发送方$S_2$和辅助方$S_0$共享随机比特串$r_0$和$r_1$

**输出:**接收方输出$m_c$

1. 发送方发送$m_0\oplus r_0, m_1\oplus r_1$给接收方.
2. 辅助方直接发送$r_c$给接收方.
3. 假设消息$m_0, m_1$与随机数可以区分,那么接收方可以得到$m_c$.
4. 接收方输出$m_c$.

现在设$a$为公开的$k$比特整数,$b=b_0\oplus b_1\oplus b_2$为二进制秘密,$S_j$持有秘密份额$(b_j, b_{j'})$,$j'=j+1 \bmod 3$,设0的(3,3)份额在$S_j$处为$\mathrm{sz}_j$,则可以按照算法9计算$a$和$b$乘积的秘密份额.

当$a$是$k$比特秘密时,可以通过执行2次三方不经意传输,然后对份额再做加和,以完成$k$比特秘密和二进制秘密相乘. 需要注意的是,该方法假设了接收方可以区分三方不经意传输的真实消息和随机消息,因此

**算法9　基于三方OT的乘法算法**

**输入:**服务提供方$S_j$的秘密份额$(b_j, b_{j'})$,$j'=j+1 \bmod 3$,公开的$k$比特整数$a$

**输出:**乘积$ab$的秘密份额

1. 服务提供方$S_2$选择$k$比特随机数$r$,设置$m_i=(i\oplus b_2\oplus b_0)a-r$,$i\in\{0,1\}$.
2. 服务提供方$S_1$以$b_1$作为选择,经过三方OT后,$S_1$得到$m_c=ba-r$.
3. 服务提供方$S_1$计算$\mathrm{tmp}_1=m_c+\mathrm{sz}_1$,服务提供方$S_2$计算$\mathrm{tmp}_2=r+\mathrm{sz}_2$,服务提供方$S_0$设置$\mathrm{tmp}_0=\mathrm{sz}_0$.
4. 服务提供方$S_j$传输$\mathrm{tmp}_j$给$S_{j'}$.
5. 服务提供方$S_j$输出$(\mathrm{tmp}_j, \mathrm{tmp}_{j'})$,$j'=j+1 \bmod 3$.

在实现时,$r_0$和$r_1$的比特长度应该大于$k$.

表3总结了前述多中心模型下、基于秘密份额可以执行的计算. 可以看到,基于秘密分享方案进行计算的最大优势在于加法可以本地完成,乘法则根据不同的秘密分享方案有不同的折衷,比较运算在基于秘密份额计算时需要较多的通信次数. 考虑到其他非线性运算大多采用多项式近似的方式完成,因而秘密分享方案主要的代价在于乘法所需要的通信次数和通信量. 当服务提供方的数量较少时,对于一些非线性运算,有可能采用Garbled电路来完成,从而减少通信的次数,然而Garbled电路构造时需要用线密钥替换分享的比特份额,在通信量上付出了代价,且需要完成$k$比特份额和其二进制份额的转化. OT也是一种计算乘法的可选方式,通过把一个乘数分解为比特,就可以通过上述OT乘来完成计算. OT乘尽管通信量大,但是效率较高,根据文献[21],使用扩展OT技术[22]后,每秒大约可以进行$10^6$次OT. 需要指出的是,在秘密分享方案的基础上,还可以完成模、幂[23]、除法、移位[15]等运算,其主要的代价也是通信次数和通信量. 另外,恶意模型下基于秘密分享方案的计算需要额外增加一些校验的内容,参见文献[24].

**表3　多中心模型下的若干计算**

| 服务提供方数量 | $n$ | $n$ | $n$ | 3 |
|---|---|---|---|---|
| 秘密分享方案 | Shamir秘密分享 | 加性秘密分享 | 任意接入结构 | |
| 加法 | 免费 | 免费 | 免费 | |
| 乘法 | 恢复1个秘密,消耗2个随机数 | 恢复2个秘密,消耗3个随机数 | 传输1个份额,消耗1个随机数 | |
| 比较 | 恢复2个秘密,计算$k$比特连乘,消耗$k$+2个随机数 | | | |
| Garbled | - | - | - | 1次传输 |
| OT乘 | - | - | - | 2次传输 |

### 3.2　双中心模型

双中心模型中有2个服务提供方. 这2个服务提供方依旧可以采用秘密分享的方案,基于份额进行计算,其支持的计算和付出的代价与多中心模型类似,如文献[21]给出的计算方法. 本节重点阐述2个服务提供方有不同功能的计算方法,其中一个服务提供方$S_1$通常称为服务提供者(Service Provider,SP),另外一个服务提供方$S_2$通常称为密码服务提供者(Cryptographic Service Pro-

vider,CSP). 我们称此类模型为非对称双中心模型.

**定义 2** (非对称双中心模型)非对称双中心模型中的参与方记为$P_1,P_2,\cdots,P_n$,服务提供方记为$S_1$和$S_2$. 对于$1\leqslant i\leqslant n$,参与方$P_i$向服务提供方$S_1$提交加密数据$\{x_i\}$,并使服务提供方$S_2$具有解密能力. 服务提供方$S_1$和$S_2$合作完成某个计算函数$(y_1,y_2,\cdots,y_n)=F(x_1,x_2,\cdots,x_n)$,并将输出$y_i$返回给$P_i$.

双中心模型中,需要假定CSP和SP是不会勾结的且互相不信任,所以参与方把密文发送给SP和CSP不会直接获得参与方的数据,同时SP还会通过加扰等方法来防止CSP获得计算过程中的敏感数据. 在实际的应用中,CSP和SP往往需要是不同的服务提供方,具有不同的核心利益诉求.

双中心模型下的计算一般依赖某种加密方案. 例如文献[25]采用了部分同态加密方案. CSP提供部分同态加密的密钥,SP接收参与方同态加密的密文. 如果部分同态加密算法支持加法同态,SP可以在密文上直接计算对应明文的加法,可以计算明文与密文的标量乘等运算,无须通信,且本地的计算时间一般也很快. 乘法的计算则需要通过协议完成. 设SP有用户提交的密文$\{x_i\}$和$\{x_i'\}$,为了计算$\{x_ix_i'\}$,并且不泄露明文给CSP,文献[25]给出了算法10.

**算法10 基于加法同态加密的乘法运算**

**输入:**服务提供方SP输入$\{x_i\}$和$\{x_i'\}$,服务提供方CSP输入解密密钥

**输出:**服务提供方SP获得$\{x_ix_i'\}$

1. SP选择随机数$r_i$和$r_i'$,计算$c_1=\{x_i-r_i\}$和$c_2=\{x_i'-r_i'\}$,并发送$c_1$和$c_2$给CSP.

2. CSP解密并计算$(x_i-r_i)(x_i'-r_i')$,重新加密后返回SP密文$c_3=\{(x_i-r_i)(x_i'-r_i')\}$.

3. SP计算$\{x_ix_i'\}=c_3+\{r_i'x_i\}+\{r_ix_i'\}-\{r_ir_i'\}$.

4. SP输出$\{x_ix_i'\}$.

当在双中心模型下可以计算加法和乘法时,与基于秘密分享的多中心模型一样,理论上可以完成任意的计算. 例如上节多中心模型中的比较算法,在半诚实模型下,可以直接应用到双中心模型中. 然而,得益于双中心模型的特殊设定,比较运算可以更为简单地完成. 依旧设SP有用户提交的密文$\{x_i\}$和$\{x_i'\}$,为了获得$\{x_i>x_i'\}$,文献[25]总结了一种比较算法:

注意到算法11中$z \bmod 2^k$和$r \bmod 2^k$分别在CSP和SP处以明文的形式出现,因此双方可以按照算法12进行比较. 设$a=r \bmod 2^k=a_{\{k-1\}},a_{\{k-2\}},\cdots,a_0$;$b=z \bmod 2^k=b_{\{k-1\}},b_{\{k-2\}},\cdots,b_0$. 算法12给出了在加法同态加密算法支持下如何完成比较.

算法12需要$k$+1轮通信,通信量是$O(k)$级别,无须

**算法11 基于加法同态加密的比较运算**

**输入:**服务提供方SP输入$\{x_i\}$和$\{x_i'\}$,服务提供方CSP输入解密密钥,共同输入包括数据的比特长度$k$

**输出:**服务提供方SP获得密文$\{x_i>x_i'\}$

1. SP选择二进制比特长度远大于$k$的一个随机数$r$,计算$\{z\}=\{2^k+(x_i'-x_i)+r\}$,并发送给CSP.

2. CSP解密后获得$z$,计算$z\gg k$,然后重新加密移位后的结果给SP.

3. SP计算$\{x_i>x_i'\}=\{(z\gg k)-(r\gg k)-t\}$,其中t的取值取决于$z \bmod 2^k$和$r \bmod 2^k$. 当$r \bmod 2^k>z \bmod 2^k$时,表明$z$的第$k$比特产生了进位,因此$t=1$,以消除进位的影响,否则$t=0$. 算法12给出该比较算法.

4. SP输出$\{x_i>x_i'\}$.

**算法12 基于加法同态加密的$k$比特明文比较协议**

**输入:**服务提供方SP输入$a_{\{k-1\}},a_{\{k-2\}},\cdots,a_0$和同态加密密钥,服务提供方CSP输入$b_{\{k-1\}},b_{\{k-2\}},\cdots,b_0$和解密密钥

**输出:**服务提供方SP获得$\{a>b\}$

1. CSP发送$\{b_0\}$给SP;

2. SP计算$\{t_1\}=\{a_0(1-b_0)\}$. 若$a_0=0$或$a_0=b_0=1$,则$a_0\leqslant b_0$,设置$t_1=0$;否则$a_0=1$且$b_0=0$,则$a_0>b_0$,设置$t_1=1$.

3. SP和CSP继续交互$k-1$次,设次数为$i=1,2,\cdots,k-1$.

3.1 SP选择随机比特$r$,计算$\{\tau\}=\{r\oplus t_i\}$,发送$\{\tau\}$给CSP.

3.2 CSP解密得到$\tau$,计算$b_i\tau$,加密获得$\{b_i\tau\}$和$\{b_i\}$,并发送给SP.

3.3 SP计算$\{b_it_i\}=(1-r)\{b_i\tau\}+r(\{b_i\}-\{b_i\tau\})$,即若$r=1$,有$b_it_i=b_i-b_i(1-t_i)=b_i-b_i\tau$.

3.4 SP计算$\{t_{i+1}\}=(1-a_i)\{t_i\}-\{b_it_i\}+a_i(1-\{b_i\})$.

4. SP输出$\{t_k\}$.

预计算的随机数. 根据文献[26]的测试结果,采用格上的同态加密算法,得益于SIMD操作[27],计算速度优于Paillier加密. 文献[28]也采取了类似的比较操作,并指出为了减小通信次数,可以使用Garbled电路完成已知明文的比较. 文献[29]给出了两比特输入的Garbled电路. 作者对该电路进行了并联和级联处理,测试了2个8比特明文的比较电路,Garbled电路规模约16.6 KB;当扩展到16比特明文时,Garbled电路规模约36.0 KB. 在通信轮数上,因为Garbled电路需要评估电路的一方通过OT的方式获得电路的输入,因此其通信次数至少是$k$+1次. 相比文献[25]中给出的算法,使用Garbled电路和扩展OT可以避免基于公钥的加密、解密、同态等运算,因此计算速度较快,特别是在大量数据需要比较时,速度优势会比较明显.

有了比较操作之后,在非对称双中心模型下还可以求一组密文的最大值. 假设SP有$l$个密文$\{x_1\},\{x_2\},\cdots,\{x_l\}$,可以通过二叉树的方式成对比较2个密文中的较大者,最后得到最大的密文. 例如设

$\{x_{12}\}$为$\{x_1\}$,$\{x_2\}$中较大者,则$x_{12}=(x_1>x_2)x_1+\left(1-\left(x_1>x_2\right)\right)x_2$,即需要计算2次乘法完成一次密文较大值的计算.为了得到最大的密文在原密文序列中的索引,还需要类似的设置密文索引,以$\{\text{index}_{12}\}$表示$\{x_1\}$,$\{x_2\}$中较大值的索引,则$\text{index}_{12}=(x_1>x_2)\times 1+\left(1-\left(x_1>x_2\right)\right)\times 2$,即需要在SP得到密文的比较值以后,本地计算$\{\text{index}_{12}\}$. 文献[28]给出了算法13让SP计算一组密文中的最大值$\{x_{\max}\}$在原序列中的位置.

**算法13　基于加法同态加密的找最大值位置的算法**

**输入:**服务提供方SP输入$l$个密文$\{x_1\},\{x_2\},\cdots,\{x_l\}$和同态加密密钥,服务提供方CSP输入解密密钥

**输出:**服务提供方SP获得$\{x_{\max}\}$的位置

1. SP对密文序列进行$\pi$随机置换得到$\{x_{(1)}\},\{x_{(2)}\},\cdots,\{x_{(l)}\}$,并设置$\{x_{\max}\}=\{x_{(1)}\}$.

2. CSP设置$m=1$.

3. SP与CSP交互$l-1$次,设置$i=2$到$l$,表示这$l-1$次交互.

　3.1　SP和CSP运行算法11比较$\{x_{\max}\}$和$\{x_{(i)}\}$,比较结果交给CSP,CSP解密得到$b_i=1-\left(x_{\max}>x_{(i)}\right)$.

　3.2　SP选择远大于$k$比特的两个随机数$r_i$和$s_i$,计算$\{m_i'\}=\{x_{\max}+r_i\}$和$\{x_{(i)}^{'}\}=\{x_{(i)}+s_i\}$,结果发送给CSP.

　3.3　如果$b_i=1$,CSP设置$m=i$且$\{v_i\}=\{x_{(i)}^{'}+0\}$,否则CSP设置$\{v_i\}=\{m_i'+0\}$,然后CSP加密$b_i$,返回密文$\{v_i\}$和$\{b_i\}$给SP;这里加0的含义是需要解密后重新加密.

　3.4　SP计算$\{x_{\max}\}=\{v_i\}+\left(\{b_i\}-1\right)r_i-\{b_i\}s_i$.

4. CSP发送$m$给SP.

5. SP输出$\pi^{-1}(m)$.

在两方非对称模型下,使用Garbled电路除了进行比较外,还可以进行其他的任意计算.如文献[26]给出了一种使用Garbled电路提取某个密文的最高比特位的方法.设SP拥有一个$k$比特明文$x_i$的密文$\{x_i\}$,CSP希望获得最高比特位$x_i^{(k-1)}$. SP可以执行算法14,这仅会泄露最高比特位给CSP.

文献[30]使用Garbled电路给出了线性回归问题的解,具体的运算涉及Garbled电路上的加减乘除和平方根运算,不在赘述.

前述非对称双中心模型中CSP能够解密的原因是用户直接使用CSP的公钥加密,这一约束条件是可以进行适当弱化的.文献[31]采用了文献[32]的同态加密算法,该算法结合了Paillier加密和ElGamal加密,使得每个用户有独立的公私钥对,同时CSP能够解密所有密文.这样前述所有的计算依旧可以完成,用户也不用CSP的公钥加密.文献[33]提出了一个基于代理重加密的DeepPAR协议,每个用户也有自己的公私钥.每个用户通过与CSP和SP交互形成代理重加密密钥,并将该密钥存储在SP处.每个用户用自己的公钥加密数据并上传给SP. SP再使用代理重加密密钥将用户的密文统一转化为使用CSP公钥加密的密文,如果转化后的密文具有同态性,就再次形成了前述的非对称双中心模型,可以进行各种计算.

**算法14　基于Garbled电路的同态加密密文最高比特位提取算法**

**输入:**服务提供方SP输入密文$\{x_i\}$和同态加密密钥,服务提供方CSP输入解密密钥,数据比特长度$k$为公开输入

**输出:**服务提供方CSP获得$x_i^{(k-1)}$

1. SP选择$k$比特随机数$r$,计算$\{x_i\oplus r\}$并发送给CSP.

2. CSP解密后获得$x_i\oplus r$,并将其作为Garbled的一个输入,另外一个输入为SP选择的$r$,之后CSP构造Garbled电路计算两个输入的异或,并输出最高比特位,然后将Garbled电路和关于$x_i\oplus r$的线密钥发送给SP.

3. SP通过OT协议获得关于$r$的线密钥,对Garbled电路进行评估,输出线密钥返回给CSP.

4. CSP根据输出线密钥确定最高比特位的值$x_i^{(k-1)}$.

5. CSP输出$x_i^{(k-1)}$.

当双中心模型下采用的加密方案是层级全同态加密方案时,CSP的主要作用在于减少SP乘法计算的次数,以满足加密方案的要求.例如文献[34]采用全同态加密实现了隐私保护的平凡贝叶斯分类器,其中CSP用于辅助SP完成批量的比较运算. SP首先随机旋转待比较的2个对象,然后CSP解密旋转后的对象,在明文下生成比较的结果,之后任何人得到比较的结果和随机旋转的指数,都可以恢复所关心数据的正确比较结果.需要注意的是全同态加密本身是可以完成比较运算的,文献[35]给出了详细的比较运算的算法.

当双中心模型下采用的加密方案是功能加密时,CSP主要的工作是为SP和参与者分配密钥,不再参与到具体计算函数中完成解密操作.文献[36]给出了一个用功能加密做内积的例子.参与者使用CSP的公钥加密数据并提交给SP,SP从CSP获取内积函数的密钥,之后SP使用内积函数的密钥对用户的加密数据解密,获得参与者与SP各自输入的内积.例如SP的输入为全1向量,则获得的是参与者输入的和.文献[37]中给出了一个用功能加密做二次多项式的例子.二次多项式可以等价地看成一个两层的神经网络,完成一些人工智能的计算任务.需要注意的是,当使用功能加密做预测时,SP只需要从CSP获取一次关于模型参数的功能函数密钥,就可以完成多次预测.

表4总结了前述非对称双中心模型下、通过加密方案可以执行的部分计算.可以看到,基于加法同态加密方案进行计算,加法是直接在密文上进行同态运算的,

乘法则需要SP和CSP通过协议完成. 加法同态运算主要优势在于可用方案较多,且效率较高. 例如Paillier加密方案和各种基于格的全同态加密方案. 注意到全同态加密方案当然可以仅使用其加法同态功能,作为加法同态方案来使用,同时其参数规模可以缩小. 采用层级全同态加密方案的主要优势在于可以减少通信轮数,其代价则在于较大的参数规模和较长的本地运算时间. 功能加密限于具体的实现方案,目前仅支持内积、二次型多项式等计算,同时其功能密钥生成的方式对实际的应用场景也是有影响的. 采用Garbled电路在双中心模型下是非常自然的选择,主要优势在于非线性运算的效率. 双中心模型的共同缺点在于CSP服务提供方往往很难在工程实践中落地,因此其应用场景受到限制.

**表4 非对称双中心模型下的若干计算**

| 加密方案 | 加法同态 | 层级全同态 | 功能加密 |
|---|---|---|---|
| 加法 | 同态 | 同态 | 特定内积函数 |
| 乘法 | 一轮通信 | 有限次乘法同态 | 特定内积函数 |
| 比较 | $k$比特输入,$k$+1轮通信 | 批量比较,一轮通信 | - |
| 求最大值及顺序 | $l$个密文,$l$次比较,$l$轮通信 | - | - |
| Garbled | 支持加、减、乘、除、平方根等计算,$k$比特输入,$k$+1轮通信 | | |

### 3.3 单中心模型

单中心模型中只有一个服务提供方$S$,该服务提供方完成对某个函数$F$的计算. 根据安全假设和安全方案的不同,参与者向服务提供方输入明文或者密文,服务提供方完成计算后,返回参与者计算结果. 模型描述如下.

**定义3** (单中心模型)单中心模型中的参与者记为$P_1,P_2,\cdots,P_n$,服务提供方记为$S$. 对于$1\leqslant i\leqslant n$,参与者$P_i$向服务提供方$S$提交数据$x_i$或$\{x_i\}$,服务提供方$S$完成某个计算函数$(y_1,y_2,\cdots,y_n)=F(x_1,x_2,\cdots,x_n)$,并将输出$y_i$返回给$P_i$.

单中心模型中对服务提供方$S$有不同的置信假设. 如果设服务提供方$S$具有硬件中实现的软件防护扩展SGX(Software Guard Extensions),又假设参与者愿意信任SGX硬件,那么参与者可以直接把SGX作为可信第三方,提交明文,获取计算结果. 如果假设服务提供方是半诚实的,那么参与者向服务提供方提交的数据要么是加密的,要么是非敏感的,以防止服务提供方获得用户的原始数据. 进一步,如果假设服务提供方是恶意的,那么还需要参与者检查服务提供方工作是否正常,在检查通过后才可以提交数据.

在基于SGX的方案中,参与者与服务提供方$S$的隔离区建立安全通道,向隔离区存入解密密钥,然后把人工智能算法所需要的数据加密后发送给服务提供方$S$,由服务提供方$S$送入隔离区进行解密,然后在隔离区运行函数$F$得到结果,并在需要时返回给用户. 由于隔离区是明文运算,所以各种计算机上可执行的运算都可以执行. 文献[38]考虑了对SGX的侧信道攻击问题,提出SGX中运行的算法在访问SGX之外的内存、硬盘、网络时,其访问模式不应泄露SGX之内的敏感数据信息.

基于数据扰动技术的方案希望向服务提供方$S$提交非敏感的数据. 文献[9]中主要把差分隐私、本地扰动和降维3种技术归类到数据扰动中. 差分隐私的主要问题是如何加噪声,使单个用户的隐私可以被保护且人工智能算法的准确率受影响较小. 本地扰动的主要问题是如何给出随机响应,使单个用户的原始数据可以被保护,并且返回的响应包含用户原始数据尽可能多的特征. 降维主要是对用户的高维数据进行降维,使低维数据不泄露用户的隐私,并且含有足够的信息值得被学习. 文献[39]引入了JOHNSON-LINDENSTRAUSS引理. 该引理说明高维数据可以通过随机投影矩阵映射为随机的低维数据,并且高维空间中两个点的欧氏距离与映射后的相应点的距离相差不大. 基于这个引理,高维空间数据和低维空间数据在内积计算等方面具有相似的统计特性,服务提供方$S$可以基于低维数据进行正常的数据分析.

基于加密技术的方案所采取的加密技术也包括了功能加密、半同态加密和全同态加密. 其中功能加密方案是通过参与者模拟CSP的功能,去掉在非对称双中心模型下CSP之后形成的,其中SP的输入向量是公开的. 文献[40]给出了一个协议实现如算法15所示的功能.

**算法15 分布式功能函数密钥生成算法**

**输入:**公开值$y=y_1,y_2,\cdots,y_n$, 安全哈希函数$H$,同态加函数$f$

**输出:**服务提供方$S$公开功能函数的公钥,参与者获得部分私钥

1. $n$个参与者协商关于零的加性秘密分享份额,协商完成后参与者$P_i$持有秘密份额$\mathrm{sz}_i$.

2. 参与者$P_i$生成部分私钥$s_i$,计算功能函数的部分密钥$f(s_iy_i+\mathrm{sz}_iH(y))$并发送给服务提供方$S$.

3. 服务提供方$S$合成功能函数公钥$f\left(\sum_{i=1}^{n}s_iy_i+\mathrm{sz}_iH(y)\right)=f\left(\sum_{i=1}^{n}s_iy_i\right)$.

4. 服务提供方$S$输出合成的功能函数公钥,参与者$P_i$输出部分私钥$s_i$.

在单中心模型下,采用半同态加密的方案可以支持仅需要单一运算的应用场景. 例如,文献[41]仅涉及了明文乘和密文加这样的线性运算,使用支持加法同态的半同态加密就可以完成,其实现了基于线性函数的预测过程和基于半同态加密的隐私数据提取过程.

文献[42]仅涉及了密文幂次和密文乘2种运算,使用支持乘法同态的半同态加密就可以完成,实现了推荐系统的预测过程. 联邦学习[43]仅需要半同态加密方案就可以完成计算,因为服务提供方的计算仅仅是求和. 基于算法15,一个简单的联邦学习方案可以描述如下:

(1) $n$个参与者协商关于零的加性秘密分享份额,协商完成后参与者$P_i$持有秘密份额$\mathrm{sz}_i$;

(2) 参与者$P_i$计算$c_i=\mathrm{sz}_i+x_i \bmod p$,发送$c_i$给服务提供方$S$;

(3) 服务提供方$S$合成$\sum_{i=1}^{n}x_i=\sum_{i=1}^{n}c_i$.

上述方案中模数p属于全局参数,大于$\sum_{i=1}^{n}x_i$且小于任意用户的秘密份额$\mathrm{sz}_i$. 该方案的主要问题在于任意用户掉线都会导致服务提供方合成不了想要的结果. 文献[43]解决了这一问题,给出了一个精巧的四轮协议. 首先设参与者$P$和服务提供方$S$都具有门限值为$t$的秘密分享方案,有Diffie-Hellman密钥协商参数$(G, g, q)$,其中$g$为循环群$G$的生成元、$q$为群的阶,有伪随机数生成函数PRF,输入种子、输出$d$维的伪随机数,参与者$P$输入数据的维度为$d$. 设每个参与者与服务提供方$S$都有安全通道. 算法16描述了该协议.

算法16在半诚实模型下可以达到保护单个用户数据的目的,使服务提供方$S$只能得到聚合的结果. 该协议不需要参与者之间有任何预先分配的秘密信息,并且可以容忍用户掉线,适合动态临时群组的应用场景. 如果群组成员相对固定,只是为了防止协议执行过程中某个参与者掉线,那么掉线的用户重新上线后只需要重新生成一对公私钥,并分享该私钥,从而可以去掉第1轮的交互,得到一个更为简化的协议. 事实上,在群组成员相对固定时,甚至可以假设参与者共享一对同态加密的公私钥,那么上述过程可以直接简化第一轮,只需要传输同态加密的密文和聚合后的密文,如文献[44~46]. 在群组成员有缓慢的变化时,可以采用群组密钥协商协议来维护一对动态的同态加密的密钥对,如文献[47]. 文献[43]也给出了服务提供方$S$主动攻击的情况,通过增加一轮确认过程和数字签名来防止主动攻击者. 必须指出的是,文献[48]在服务提供方$S$输出的聚合结果的基础上,能够有效地恢复参与者的原始数据,这在一定程度上限制了上述算法在联邦学习中的应用场景.

在单中心模型下,采用全同态加密的方案是很自然地选择. 参与者上传全同态加密的密文给服务提供方$S$,服务提供方$S$通过密文上的加运算和乘运算完成定义3.3中函数$F$的计算,然后返回密文结果给参与者. 文献[35]给出了全同态密文上的比较、二进制加法、查表等计算方法,扩展了全同态加密密文上可以执行的计算.

**算法16　安全的联邦学习算法**

**输入:**参与者$P_i$的$d$维数据$x_i$, $(G, g, q)$, 门限$t$的秘密分享方案, 对称加密算法, PRF

**输出:** $\sum_{u\in U_3}x_u$,其中$U_3$集合含有不少于$t$个参与者

1. 参与者$P_i$生成两对密钥$\left(g^{c_i}, c_i\right)$和$\left(g^{s_i}, s_i\right)$, 发送$\mathrm{msg}_{1_i}=\left(i, g^{c_i}, g^{s_i}\right)$给服务提供方$S$, 进入下一轮. 服务提供方$S$收集至少$t$个消息, 否则超时退出. 服务提供方$S$构造列表$U_1$包含所接收消息的发送者身份, 设定$|U_1|=n_1$, 构造消息$\mathrm{res}_1=\left[\left(P_u, g^{c_u}, g^{s_u}\right)\right]$, $u\in U_1$,向$U_1$列表中的参与者发送消息$\mathrm{res}_1$.

2. 参与者$P_i$确认收到的消息$\mathrm{res}_1$中有至少$t$对不同的公钥, 否则退出. $P_i$构造列表$U_1'$包含$\mathrm{res}_1$中的参与者身份,设定$\left|U_1'\right|=n_1'$. $P_i$选择一个随机数$b_i$,使用秘密分享体制得到$n_1'$份该随机数的秘密份额$b_{iu}$, 再次使用秘密分享体制得到私钥$s_i$的$\mathrm{n}_1'$份秘密份额$s_{iu}$, $u\in U_1'$. 然后$P_i$计算与参与者$P_u$的对称加密密钥$\left(g^{c_u}\right)^{c_i}$, 加密$\left(b_{iu}, s_{iu}\right)$得到密文$\left\{\left(b_{iu}, s_{iu}\right)\right\}, u\neq i, u\in U_1'$. 最后$P_i$构造消息$\mathrm{msg}_{2_i}=\left(P_i, \left[\left(P_u, \left\{\left(b_{iu}, s_{iu}\right)\right\}\right)\right], u\in U_1', u\neq i\right)$, 发送消息$\mathrm{msg}_{2_i}$给服务提供方$S$, 进入下一轮. 服务提供方$S$接收至少$t$个消息,否则超时退出. 服务提供方$S$构造列表$U_2$包含所接收消息的发送者身份, 设定$|U_2|=n_2$, 构造消息$\mathrm{res}_{2_u}=\left[\left(P_i, \left\{\left(b_{iu}, s_{iu}\right)\right\}\right)\right], i\in U_2, i\neq u$, 然后服务提供方$S$把消息$\mathrm{res}_{2_u}$单独发送给$P_u, u\in U_2$.

3. 参与者$P_i$确认收到的消息至少包含$t-1$个不同参与者的密文, 否则退出; 参与者$P_i$存储消息$\mathrm{res}_{2_i}$, 以备在第4轮使用, 构造列表$U_2'$包含$\mathrm{res}_{2_i}$中的参与者身份, 设定$\left|U_2'\right|=n_2'$. 参与者$P_i$计算与$P_u$的共享密钥$\mathrm{sk}_{i,u}=\left(g^{s_u}\right)^{s_i}, u\in U_2'$. 计算$d$维向量$\mathrm{zs}_{\mathrm{i,u}}=\delta_{i,u}\,\mathrm{PRF}(\mathrm{sk}_{\mathrm{i,u}})$, 当$i>u$时,$\delta_{i,u}=1$,否则$\delta_{i,u}=-1$. 参与者$P_i$计算$sl_i=\mathrm{PRF}(b_i)$, 其中$b_i$是步骤2中该参与者所分享的随机数. 参与者$P_i$计算$c_i=sl_i+x_i+\sum_{u\in U_2'}\mathrm{zs}_{i,u}$, 其中$x_i$为参与者$P_i$需要加密的数据, 其维度为$d$. 参与者$P_i$发送$\mathrm{msg}_{3_i}=(P_i, c_i)$给服务提供方$S$,进入下一轮. 服务提供方$S$接收至少$t$个消息,否则超时退出. 服务提供方$S$构造列表$U_3$包含所接收消息的发送者身份,设定$|U_3|=n_3$, 构造消息$\mathrm{res}_3=U_3$, 发送给$U_3$列表中的参与者.

4. 参与者$P_i$确认收到的$U_3'$列表规模至少为$t$, 否则退出. 参与者$P_i$解密消息$\mathrm{res}_{2_i}$中的密文, 恢复$U_2'$中的参与者给$P_i$的秘密份额. 参与者$P_i$构造消息$\mathrm{msg}_{4_i}=[(P_u, \mathrm{ss}_u)]$, $P_u\in U_2'$, 其中, 如果$P_u\notin U_3'$, 则$\mathrm{ss}_u=s_{ui}$, 否则$\mathrm{ss}_u=b_{ui}$. 参与者$P_i$发送消息$\mathrm{msg}_{4_i}$, 之后退出协议. 服务提供方$S$接收至少$t$个参与者发送的$\mathrm{msg}_{4_i}$, 否则超时退出. 对于$P_u\in U_2$且$P_u\notin U_3$的参与者, 服务提供方$S$使用秘密恢复体制恢复其私钥$s_u$, 从而能够计算参与者$P_u$在步骤3所用的共享密钥$\mathrm{sk}_{u,i}=\left(g^{s_i}\right)^{s_u}, i\in U_2$. 对于$P_u\in \mathrm{U}_3$的参与者, 服务提供方$S$使用秘密恢复体制恢复随机数$b_u$, 从而能够通过计算$\sum_{u\in U_3}(c_u-\mathrm{sl}_u)+\sum_{u\in U_2\text{且}u\notin U_3}(\sum_{i\in U_3}\mathrm{zs}_{u,i})$, 输出聚合结果$\sum_{u\in U_3}x_u$.

设 $k$ 比特整数 $a=a_{k-1},a_{k-2},\cdots,a_0$ 和 $b=b_{k-1},b_{k-2},\cdots,b_0$，逐个比特加密后得到 $\{a\}=\{a_{k-1}\},\{a_{k-2}\},\cdots,\{a_0\}$ 和 $\{b\}=\{b_{k-1}\},\{b_{k-2}\},\cdots,\{b_0\}$，文献[35]给出了一个算法可以同时得到密文的比较结果、较大值和较小值：

**算法17 全同态加密密文上的比较算法**

**输入**:$\{a\}=\{a_{k-1}\},\{a_{k-2}\},\cdots,\{a_0\}$和$\{b\}=\{b_{k-1}\},\{b_{k-2}\},\cdots,\{b_0\}$

**输出**:$\{a>b\},\{\max(a,b)\},\{\min(a,b)\}$

1. 对于$0\leqslant i<k$,计算$\{e_i\}=\{a_i\}\oplus\{b_i\}\oplus 1$和$\{g_i\}=\{a_i\}\{a_i\}\{b_i\}$.

2. 计算$\{e_i^*\}=\prod_{\{j\geqslant i\}}\{e_j\}$和$\{g_i^*\}=\{g_i\}\prod_{\{j>i\}}\{e_j\}$.

3. 计算$\{\tilde{g}_i\}=\sum_{\{j\geqslant i\}}\{g_1^*\}$.

4. 计算$\{x_i\}=\left(\{a_i\}\oplus\{b_i\}\right)\{\tilde{g}_i\}\oplus\{b_i\}$,计算$\{y_i\}=\{a_i\}\oplus\{b_i\}\oplus\{x_i\}$.

5. 输出$\{\tilde{g}_0\}$为$\{a>b\}$,$\{x\}=\{x_{k-1}\},\{x_{k-2}\},\cdots,\{x_0\}$为$\{\max(a,b)\}$,$\{y\}=\{y_{k-1}\},\{y_{k-2}\},\cdots,\{y_0\}$为$\{\min(a,b)\}$.

算法17的第1步，可以得到密文状态下的$\{e\}$序列和$\{g\}$序列，其中$e$序列表明对应位置的2个比特是否相等，相等为1，否则为0，而$g$序列表明对应比特位置是否有$a_i>b_i$，是则为1，否则为0. 算法17的第2步，可以得到密文状态下的$\{e^*\}$序列和$\{g^*\}$序列，其中$e^*$序列的第$i$比特$e_i^*$表明从第$k-1$比特到第$i$比特，是否全部相等，而$g^*$序列的第$i$比特$g_i^*$表明$a$和$b$是否是从第$i$比特开始不同的. 假设$a$和$b$是从第$i$比特开始不同的，那么显然第$k-1$比特到第$i+1$比特是相同的，从而$g$序列中第$k-1$比特到第$i+1$比特全为0，进而$g^*$序列这些比特全部为0；进一步，第$i-1$比特到第0比特，因为第$i$比特不同，所以$\prod_{\{j>i\}}e_j$为0，进而$g^*$序列这些比特全部为0；因此，只有$g_i^*=g_i$保留下来，表明$a$和$b$比较的结果. 算法17的第3步，得到$\tilde{g}$序列，如果依旧假设$a$和$b$是从第$i$比特开始不同的，那么$\tilde{g}$序列从第$i$比特开始，一直等于$g_i^*$，而大于$i$的位置全部为0，由此显然得到$\{\tilde{g}_0\}$为比较结果. 算法17的第4步，如果假设$a$和$b$是从第$i$比特开始不同的，则对于大于$i$的那些比特位置，$a$和$b$在对应位置相同，$\tilde{g}$序列对应位置为0；从第$i$比特开始，输出$x_i=a_i$. 上述计算的乘法次数为$O(k)$.

当在二进制密文上进行计算时，全同态加密所支持的是二进制的异或运算及与运算. 为了计算加法，需要采用全加器电路. 文献[35]给出了一种计算过程. 依旧设$k$比特整数$a=a_{k-1},a_{k-2},\cdots,a_0$和$b=b_{k-1},b_{k-2},\cdots,b_0$，逐个比特加密后得到$\{a\}=\{a_{k-1}\},\cdots,\{a_0\}$和$\{b\}=\{b_{k-1}\},\{b_{k-2}\},\cdots,\{b_0\}$.

算法18的第1步，从输入分别计算2个序列，其中$e$序列表明两个输入比特是否相等，$g$序列则将两比特输入下需要进位的情况单独标识出来. 第2步中考虑第$i$比特到第$j$比特的片段. 其中$g_i$的值对于第$j$比特的进位有这样的影响：如果$g_i$为1，并且从第$i$+1比特到$j$比特的片段每一比特的加和结果都是1，则对第$j$比特进位$c_j$的影响为加1. 这其实就是$g_{[i,j]}$计算式的含义. 第3步中实际计算第j比特的进位$c_j$. 从第0比特的位置到第$j$比特的位置，每一个$g_{[i,j]}$都可能影响$c_j$的值. 但是，如果是其中的第$i$比特位置的影响为加1，即$g_i=1$，则$e_i=0$，那么对于$i'<i$，有$g_{[i',j]}$为0；对于$i''>i$，有$e_{i''}=1$，$g_{i''}=0$，从而$g_{[i'',j]}$=0. 所以第3步的求和中最多只有一个1，其他均为0，即影响$c_j$值的有效比特位置只有一个. 第4步中，考虑第$i$比特的输出，如果$g_i=1$，则$e_i=0$，输出为$c_{i-1}$；如果$g_i=0$，则因为进位在第3步已经充分考虑，这里只需要直接给出关于第$i$比特$c_{i-1}\oplus e_i$的结果即可. 该算法主要的复杂度显然在于第2步，需要$O(k^2)$次乘法. 当有了二进制密文的加法之后，二进制密文乘法的结果可以通过多个二进制密文相加的结果得到，不再赘述.

**算法18 全同态加密二进制密文的加法**

**输入**:$\{a\}=\{a_{k-1}\},\{a_{k-2}\},\cdots,\{a_0\}$和$\{b\}=\{b_{k-1}\},\{b_{k-2}\},\cdots,\{b_0\}$

**输出**:$\{a+b\}$

1. 对于$0\leqslant i<k$,计算$\{e_i\}=\{a_i\}\oplus\{b_i\}$和$\{g_i\}=\{a_i\}\{b_i\}$.

2. 对于$0\leqslant i\leqslant j<k$,计算$g_{[i,j]}=\{g_i\}\prod_{k>i}^{j}\{e_k\}$.

3. 计算第$0\leqslant j<k$位的进位比特$c_j=\sum_{i=0}^{j}\{g_{[i,j]}\}$.

4. 对于$0\leqslant i\leqslant k$,计算并输出$o_i=\{a_i\}\oplus\{b_i\}\oplus\{c_{i-1}\}$,无定义项默认为0.

5. 输出$\{a+b\}=o_{k-1},o_{k-2},\cdots,o_0$.

对于一般的非线性函数，一般的处理方法是多项式逼近，例如文献[49]中把逻辑回归函数展开为1次多项式就获得了不错的结果. 然而，对于人工智能领域的非线性函数，当其输入和输出的比特长度有限时，查表的方法可能效率更高. 假设函数$f$的输入为$k$比特，输出为$k'$比特，文献[35]中给出了查表法的一般步骤.

算法19主要的计算都在第2步，仅当输入二进制密文对应的明文与序号$i$相等时，$b_i$才为1，其他情况为0. 该算法需要的乘法次数为$O(k2^k)$. 文献[50]中给出了一个更为有效的查表计算方法，需要的乘法次数为$O(2^k)$.

有些非线性函数具有迭代的数值计算的算法，例如使用牛顿-拉夫逊迭代求方程的根，牛顿迭代法求平方根等. 在全同态加密中，迭代运算的好处在于精度与计算能力之间可以权衡. 文献[49]采用牛顿-拉夫逊迭代计算逻辑回归训练模型的最优解，在只进行1次迭代时就获得了可以接受的模型精度. 测试牛顿迭代法求平方根，当输入范围在1到81万之间时，设初值为300，

**算法19　全同态加密中的查表算法**

**输入:**$k$比特输入$k'$比特输出的函数$f$,二进制密文$\{a_{k-1}\},\{a_{k-2}\},\cdots,\{a_0\}$

**输出:**二进制密文$\{o_j\}, j\in\{0,1,\cdots,k'-1\}$

1. 对于$0\leqslant i\leqslant 2^k-1$,计算$f(i)$的输出比特$o_{k'-1},o_{k'-2},\cdots,o_0$.

2. 对于输入的二进制密文$\{a_{k-1}\},\cdots,\{a_0\}$,计算$2^k$个密文,其中第$i$个密文计算方法如下:首先把$i$按照二进制展开为$(i_{k-1},i_{k-2},\cdots,i_0)$,计算$\{b_i\}=\prod_{i_j=0}^{k-1}\{b_{i_j}\}$,其中$i_{i_j}=1$时,$\{b_{i_j}\}=\{a_{i_j}\}$,否则$\{b_{i_j}\}=(1-\{a_{i_j}\})$.

3. 对于$0\leqslant i\leqslant 2^k-1$,对每一个输出比特$o_{i_j}, j\in\{0,1,\cdots,k'-1\}$,计算$o_{i_j}\{b_i\}$.

4. 对于$j\in\{0,1,\cdots,k'-1\}$,计算$\{o_j\}=\sum_{i=0}^{2^k-1}o_{i_j}\{b_i\}$.

5. 输出二进制密文$\{o_j\}, j\in\{0,1,\cdots,k'-1\}$.

迭代8次,可以得到一个近似的平方根.

与在明文上直接计算相比,全同态加密算法目前依旧需要付出较大的存储和时间代价. 研究者们从人工智能模型的角度,给出了一些全同态加密算法友好的人工智能模型,这些模型只需要加法和较少次数的乘法,或者只需要极少的比较等. 文献[51]定义了一个二分类的线性平均分类器,只包含加法和乘法操作,在同一加密密钥的假设下,可以完成分类器的训练和预测. 文献[35]给出了一个比特级别的逻辑回归模型,以适应其逐比特加密的全同态加密密文. 文献[34]评估了平凡贝叶斯分类器在全同态加密算法下的表现,该分类器只需要加法、乘法和比较运算. 文献[1]给出了一个8层的神经网络,仅包含加法和乘法,能完成预测. 文献[52]采用泰勒级数展开来近似医学领域的Cox风险比例回归模型.

表5总结了前述单中心模型下、通过各种加密方案可以执行的计算. 可以看到,对于$k$比特的整数,在全同态加密下其需要的乘法总量还是比较大的,其连乘的次数是$O(k)$级别的,并且人工智能算法的精度越高,全同态加密下的计算代价越大. 半同态加密和功能加密在单中心模型下仅能进行对应方案所支持的运算,因此在一些只需要简单运算的场景中可以使用,例如联邦学习场景. 需要特别说明的是,表5中没有列出基于数据扰动和SGX的隐私保护方案,这是因为这些方案的运算可以看成明文上的运算;同时表5也没有列出全同态加密算法能够支持的所有运算,例如没有单独列出求平方根的运算,这是因为我们给出的运算已经可以表明全同态加密算法在单中心模型下的复杂度.

### 3.4　现实模型

现实模型中没有服务提供方,对某个函数$F$的计算全部由参与者完成,模型描述如下.

**表5　单中心模型下基于加密方案的若干计算**

| 加密方案 | 全同态加密 | 半同态加密 | 功能加密 |
|---|---|---|---|
| 加法 | 同态 | 加法同态 | 特定内积函数 |
| 乘法 | 同态 | 乘法同态 | 特定内积函数 |
| 比较 | $O(k)$ | - | - |
| 二进制密文加 | $O(k^2)$ | - | - |
| 非线性函数 | $O(k2^k)$ | - | - |
| 方程求根 | 数值计算 | - | - |

**定义4**　(现实模型)现实模型中的参与者记为$P_1,P_2,\cdots,P_n$. 对于$1\leqslant i\neq j\leqslant n$,参与者$P_i$与$P_j$通过交换消息完成某个计算函数$(y_1,y_2,\cdots,y_n)=F(x_1,x_2,\cdots,x_n)$,获得输出$y_i$.

参与者本身有不同的置信假设. 通常有半诚实假设和恶意假设. 半诚实假设下攻击者诚实的执行协议,但是推断其他参与方的秘密输入;恶意假设下攻击者任意执行协议,以获取其他参与方的秘密输入.

当前现实模型下的计算有多种实现方法.

首先,多中心和对称双中心模型下的计算可以转化为现实模型下的计算. 当多中心和对称双中心模型下的服务提供方同时充当参与方时,每一个服务提供方既是参与方又是服务提供方,就成为现实模型下的计算. 此时,对服务提供方的置信假设转化为对参与方的置信假设,可以完成表3中的各种运算. 例如文献[53]使用Garbled电路在两个参与者$P_1$和$P_2$的私有基因库数据上计算最小等位基因频率MAF(Minor Allele Frequency)和$\chi^2$统计. 文献[54]使用Garbled电路完成了ID3算法[55]属性集合为空和叶子节点判定时的加和、比较等运算,并将Garbled电路和乘法同态加密作为基本构造模块,完成了ID3算法寻找最大增益属性的步骤.

其次,当参与者数量为2时,通过服务提供方同时充当参与方,可以完成非对称双中心模型下的计算. 文献[56]中,参与方$P_1$直接把自己的公钥给参与方$P_2$,形成了以$P_1$为CSP的非对称双中心模型,同时参与方$P_2$直接把自己的公钥给参与者$P_1$,又形成了以$P_2$为CSP的非对称双中心模型,因此双方可以完成表4中的各种运算.

最后,参与者$P_1,P_2,\cdots,P_n$可以模拟CSP服务提供方并指定某个$P_i$为SP,$1\leqslant i\leqslant n$,从而能够根据非对称双中心模型执行表4中的各种运算. 文献[57]给出了一个无可信第三方的门限加密系统. 以分布式生成ElGamal加密密钥为例,给定循环群$G=<g>$,在半诚实模型下,该协议可以简化描述如下:

之后参与者$P_1,P_2,\cdots,P_n$可以采用个人合成份额对消息进行部分解密,并通过指数上的拉格朗日插值恢复原始消息. 当考虑恶意攻击者时,算法20需要增加

**算法20 分布式门限加密密钥生成算法**

**输入:**$(G, g, q)$,Shamir秘密分享

**输出:**参与者$P_i$输出合成份额$x_i$与合成公钥pk

1. 参与者$P_i$随机选择个人私钥$s_i$,计算并广播个人公钥$\mathrm{pk}_i=g^{s_i}$.

2. 参与者$P_i$采用$t$门限的Shamir秘密分享得到份额$s_{ij}^{(t)}$,$1\leqslant j\leqslant n$,然后将份额$s_{ij}^{(t)}$秘密的发送给参与者$P_j$,$1\leqslant j\leqslant n$.

3. 参与者$P_i$计算个人合成份额$x_i=\sum_{j=1}^{n}s_{ji}^{(t)}$和合成公钥$\mathrm{pk}=\prod_{j=1}^{n}\mathrm{pk}_j$.

4. 参与者$P_i$输出合成份额$x_i$和合成公钥pk.

承诺和多项式估值验证的步骤.

## 4 未来研究方向

通过对各种计算模型的分析,我们看到多中心模型下主要的计算代价是通信量,单中心模型下主要的计算代价是计算量,双中心模型介于两者之间,有多种工具使其计算代价在计算量和通信量之间灵活转化,现实模型可以根据参与者的数量,等价的转化为其他计算模型来衡量计算代价. 然而考虑到人工智能模型参数和数据的规模,提升算法的效率和安全性依旧是未来一段时间内主要的研究方向. 对于算法效率和安全性的提升,一条发展路径是提升单个密码学工具的效率和安全性,另外一条发展路径则是混合多种密码学工具,以提升整体的效率和安全性.

(1)提升单个密码学工具的效率和安全性. 对于基于秘密份额的计算,文献[58]给出了抗主动攻击者的常数轮安全多方计算协议的高效构造方法,文献[59]给出了一个安全多方计算的平台,融合了当前30种安全多方计算协议,可以快速的对比各种算法的安全性和效率. 对于基于全同态加密的计算,人们不断提升全同态加密的速度,特别是bootstrapping算法的速度,文献[50]包含高效的bootstrapping算法,一次计算的时间约0.01 s. 文献[60]给出了CKKS算法GPU版本的bootstrapping算法,其单个明文对应的平均计算时间约8 μs. 对于基于OT的计算,文献[61]给出了相关不经意传输COT(Correlated Obviously Transfer)的高速实现. 对基于Garbled电路的实现,文献[62]给出了改进的"认证与门三元组"协议,提高了多方认证的Garbled电路协议的效率.

(2)提升混合计算的效率和安全性. 基于秘密份额的计算和基于全同态的计算具有转化算法;在两方和三方情况下,基于秘密份额的计算和基于Garbled电路的计算、基于OT的计算也具有转化算法. 这些算法的存在使混合计算成为可能. 文献[63]混合使用了加性秘密分享、安全多方计算和Garbled电路技术,对一个4层的CNN进行了隐私保护的预测,结果比文献[1]的计算方法在时间上有明显的优势. 文献[64]混合使用了同态加密和Garbled电路技术,对多个神经网络模型进行了隐私保护的预测,结果显示比文献[1]、文献[21]、文献[63]中的方法在计算时间上有明显的优势. 文献[65]给出了一个自动化的编程框架,可以自动化的选择不同的密码算法来完成隐私保护,结果显示其自动化生成的隐私保护协议与专门设计的协议具有可比性. 文献[66]给出了一个推荐系统,来自动化的选择不同的同态加密体制和参数,以满足应用的需求.

总之,对于能够完成加法和乘法这两种基本运算的密码学工具,每种工具在效率和安全性上的提升都是值得关注的. 同时鉴于每种密码学工具都有其优点和缺点,提升混合计算的效率和安全性对实际的人工智能隐私保护具有重要意义.

## 5 结论

本文综述了基于密码学技术的人工智能隐私保护计算模型及在每种计算模型下完成的主要计算. 表6列出了本文给出的所有算法及部分典型文献在4类计算模型中的分布,其中符号"-"表示未涉及或者不适用.

**表6 本文的算法及部分典型文献**

| 密码学工具 | | 多中心模型 | 双中心模型 | 单中心模型 | 现实模型 |
|---|---|---|---|---|---|
| 份额 | Shamir秘密分享 | 算法5,6 | - | 算法16 | 算法20 |
| | 加性秘密分享 | 文献[15,16]<br>算法4, 6 | 文献[16,21]<br>算法4 | 算法15 | - |
| | 任意接入结构 | 文献[14]<br>算法<br>1~3, 6~9 | - | - | - |
| 密文 | 半同态 | - | 文献[31]<br>算法<br>10~14 | 文献<br>[41,42] | 文献[56]<br>算法20 |
| | 全同态 | 文献[59] | 文献[21,34] | 文献[50]<br>算法<br>17~19 | - |
| | 功能加密 | - | 文献[36,37] | 算法15 | - |
| Garbled电路 | | 算法7 | 文献[29,30]<br>算法14 | - | 文献[53,<br>54] |
| OT | | 算法8,9 | 文献[21] | - | - |

本文的计算模型包括多中心模型、双中心模型、单中心模型和现实模型. 在每种模型下,对于当前能够单独完成加法和乘法的四类密码学工具,给出了20个典型的算法、对于这些算法之外涉及的技术进展给出了典型文献. 这些算法涉及不同计算模型下的加法、乘法、比较等基本运算,也包括二进制加法、安全联邦学习算法、非线性运算的查表运算等. 本文分析了每个算

法的主要计算或通信代价,阐述了部分算法的基本原理,包括秘密份额上的符号提取,全同态密文上的比较和二进制加法,半同态加密密文上的比较等算法.

**参考文献**


[1] DOWLIN N, GILAD-BACHRACH R, LAINE K, et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy[C]//Proceedings of the 33rd International Conference on International Conference on Machine Learning. Cambridge: MIT Press, 2016: 19-24.

[2] WU Y, CAI S, XIAO X, et al. Privacy preserving vertical federated learning for tree-based models[J]. VLDB Endowment, 2020, 13(11): 2090-2103.

[3] YAO A C. Protocols for secure computations[C]// Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. Piscataway: IEEE, 1982: 160-164.

[4] GOLDREICH O, MICALI S, WIGDERSON A. How to play any mental game[C]//Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing[C]. New York: Association for Computing Machinery, 1987: 218-229.

[5] DAVID E, VLADIMIR K, MIKE R. A pragmatic introduction to secure multi-party computation[J]. Foundations and Trends in Privacy and Security, 2018, 2(2-3): 70-246.

[6] 纪守领,杜天宇,李进锋,等. 机器学习模型安全与隐私研究综述[J]. 软件学报, 2021, 32(1): 41-67.
JI S L, DU T Y, LI J F, et al.Security and pdvacy of machine learning models: A survey[J]. Journal of Software, 2021, 32(1): 41-67. (in Chinese)

[7] 谭作文,张连福.机器学习隐私保护研究综述[J]. 软件学报, 2020, 31(7): 2127-2156.
TAN Z W, ZHANG L F. Survey on privacy preserving techniques for machine learning[J]. Journal of Software, 2020, 31(7): 2127-2156. (in Chinese)

[8] 刘睿瑄, 陈红, 郭若杨, 等. 机器学习中的隐私攻击与防御[J]. 软件学报, 2020, 31(3): 866-892.
LIU R X, CHEN H, GUO R Y, et al. Survey on privacy attacks and defenses in machine learning[J]. Joumal of Software, 2020, 31(3): 866-892. (in Chinese)

[9] AL-RUBAIE M, CHANG J M. Privacy-preserving machine learning: Threats and solutions[J]. IEEE Security Privacy, 2019, 17(2): 49-58.

[10] TANUWIDJAJA H C, CHOI R, KIM K. A survey on deep learning techniques for privacy-preserving[C]//Proceedings of the Second International Conference on Machine Learning for Cyber Security. Switzerland: Springer Nature, 2019: 29-46.

[11] OGUNSEYI T B, YANG C. Survey and analysis of cryptographic techniques for privacy protection in recommender systems[C]//Proceedings of the 4th International Conference on Cloud Computing and Security. Switzerland: Springer Nature, 2018: 691-706.

[12] DUGAN T, ZOU X. A survey of secure multiparty computation protocols for privacy preserving genetic tests[C]// Proceedings of the First International Conference on Connected Health: Applications, Systems and Engineering Technologies. Piscataway: IEEE, 2016: 173-182.

[13] MOHASSEL P, RINDAL P. ABY3: A mixed protocol framework for machine learning[C]// Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2018: 35-52.

[14] ITO M, NONMEMBER A, SAITO A, et al. Secret sharing scheme realizing general access structure[J]. Electronics and Communications in Japan, Part 3, 1989, 72(9): 56-64.

[15] DAN B. Sharemind: Programmable Secure Computations with Practical Applications[D]. Tartu: University of Tartu, 2013.

[16] DAMGÅRD I, PASTRO V, SMART N, et al. Multiparty computation from somewhat homomorphic encryption [C]//Proceedings of the 32nd Annual International Cryptology Conference. New York: Springer Verlag, 2012: 643-662.

[17] CRAMER R, DAMGÅRD I, ISHAI Y. Share conversion, pseudorandom secret-sharing and applications to secure computation[C]//Proceedings of Second Theory of Cryptography Conference. Berlin: Springer, 2005: 342-362.

[18] DAMGÅRD I, NIELSEN J B. Scalable and unconditionally secure multiparty computation[C]// Proceedings of the 27nd Annual International Cryptology Conference. Berlin: Springer, 2007: 572-590.

[19] REISTAD T I, TOFT T. Secret sharing comparison by transformation and rotation[C]// Proceedings of Information Theoretic Security - Second International Conference. Berlin: Springer, 2007: 169-180.

[20] YU C H. Sign modules in secure arithmetic circuits[EB/OL]. (2011)[2021]. http://eprint.iacr.org/2011/539.

[21] MOHASSEL P, ZHANG Y. SecureML: A system for scalable privacy-preserving machine learning[C]//Proceedings of IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2017: 19-38.

[22] ASHAROV G, LINDELL Y, SCHNEIDER T, et al. More efficient oblivious transfer extensions[J]. Journal of Cryptology, 2017, 30(2): 805-858.

[23] DAMGÅRD I, FITZI M, KILTZ E, et al. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation[C]//Proceedings of Third Theory of Cryptography Conference. Berlin: Springer, 2006: 285-304.

[24] FURUKAWA J, LINDELL Y, NOF A, et al. High-throughput secure three-party computation for malicious adversaries and an honest majority[C]//Proceedings of the Advances in Cryptology EUROCRYPT 2017, Part II. Cham: Springer, 2017: 225-255.

[25] ERKIN Z, VEUGEN T, TOFT T, et al. Generating private recommendations efficiently using homomorphic encryption and data packing. IEEE Transactions on Information Forensics and Security, 2012, 7(3): 1053-1066.

[26] SHARMA S, CHEN K. Confidential boosting with random linear classifiers for outsourced user-generated data [C]//In Computer Security-Proceddings Part I of the 24th European Symposium on Research in Computer Security. Cham: Springer, 2019: 41-65.

[27] SMART N P, VERCAUTEREN F. Fully homomorphic SIMD operations[J]. Designs, Codes and Cryptography, 2014, 17(1): 57-81.

[28] BOST R, POPA R A, TU S, et al. Machine learning classification over encrypted data[C]//Proceedings of the 22nd Annual Network and Distributed System Security Symposium. Reston: The Internet Society, 2015: 1-14.

[29] ROQUES O, VIPIN R. Secure multi-party computation [DB/OL]. (2020-11-26) [2021-1-2]. https://github.com/ojroques/garbled-circuit.

[30] NIKOLAENKO V, WEINSBERG U, IOANNIDIS S, JOYE M, BONEH D, TAFT N. Privacy-preserving ridge regression on hundreds of millions of records[C]//Proceedings of the 2013 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2013: 334-348.

[31] GONZÁLEZ-SERRANO F J, AMOR-MARTÍN A, CASAMAYÓN-ANTÓN J. Supervised machine learning using encrypted training data[J]. International Journal of Information Security, 2018, 17(4): 365-377.

[32] BRESSON E, CATALANO D, POINTCHEVAL D. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications[C]//Proceedings of 9th International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2003: 37-54.

[33] ZHANG X, CHEN X, LIU J, et al. DeepPAR and DeepDPA: Privacy preserving and asynchronous deep learning for industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2020, 16(3): 2081-2090.

[34] KIM S, OMORI M, HAYASHI T, et al. Privacy preserving naive bayes classification using fully homomorphic encryption[C]//Proceedings of the 25th International Conference on Neural Information Processing. Cham: Springer, 2018: 349-358.

[35] CRAWFORD J L H, GENTRY C, HALEVI S, et al. Doing real work with FHE: The case of logistic regression [C]//Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. New York: Association for Computing Machinery, 2018: 1-12.

[36] MARC T, STOPAR M, HARTMAN J, et al. Privacy enhanced machine learning with functional encryption[C]// Proceedings Part I of the 24th European Symposium on Research in Computer Security. Cham: Springer, 2019: 3-21.

[37] SANS E D, GAY R, POINTCHEVAL D. Reading in the dark: Classifying encrypted digits with functional encryption[EB/OL]. (2018-2-22)[2021-1-3]. https://eprint.iacr.org/2018/206.2021.

[38] OHRIMENKO O, SCHUSTER F, FOURNET C, et al. Oblivious multi-party machine learning on trusted processors [C]//Proceedings of the 25th USENIX Conference on, Security Symposium. New York: ACM, 2016: 619-636.

[39] LIU K, KARGUPTA H, RYAN J. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining[J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(1): 92-106.

[40] CHOTARD J, DUFOUR S E, GAY R, et al. Decentralized multi-client functional encryption for inner product [C]//Proceedings Part II of the 24th International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2018:703-732.

[41] DJATMIKO M, FRIEDMAN A, BORELI R, et al. Secure evaluation protocol for personalized medicine[C]// Proceedings of the 13th Workshop on Privacy in the Electronic Society. New York: Association for Computing Machinery, 2014: 159-162.

[42] WANG J, ARRIAGA A, TANG Q, et al. Facilitating privacy-preserving recommendation-as-a-service with machine learning[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Se-

curity. New York: Association for Computing Machinery, 2018: 2306-2308.

[43] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2017: 1175-1191.

[44] PHONG L T, AONO Y, HAYASHI T, et al. Privacy-preserving deep learning: Revisited and enhanced[C]//Proceedings of the Applications and Techniques in Information Security - 8th International Conference. Singapore: Springer, 2017: 100-110.

[45] MA X, ZHANG F G, CHEN X F, et al. Privacy preserving multi-party computation delegation for deep learning in cloud computing[J]. Information Sciences, 2018, 459: 103-116.

[46] CHAI D, WANG L, CHEN K, et al. Secure federated matrix factorization[J]. IEEE Intelligent Systems, 2021, 36 (5):11-20.

[47] ZHANG X, CHEN X F, LIU J, et al. DeepPAR and DeepDPA: Privacy-preserving and asynchronous deep learning for industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2019, 16(3): 2081-2090.

[48] YIN H, MALLYA A, VAHDAT A, et al See through gradients: Image batch recovery via gradinversion[C]//Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 16332-16341.

[49] BONTE C, VERCAUTEREN F. Privacy-preserving logistic regression training[J]. BMC Medical Genomics. 2018,11(Suppl 4):86 13-21.

[50] ILARIA C, NICOLAS G, MARIYA G, et al. TFHE: Fast fully homomorphic encryption over the torus[J]. Journal of Cryptology. 2020, 33(1): 34-91.

[51] THORE G, KRISTIN L, MICHAEL N. ML Confidential: Machine learning on encrypted data[C]// Proceedings of the 15th International Conference on Information Security and Cryptology 2012. Berlin: Springer, 2012: 1-21.

[52] JOPPE W B, KRISTIN L, MICHAEL N. Private predictive analysis on encrypted medical data[J]. Journal of Biomedical Informatics, 2014, 50: 234-243.

[53] CONSTABLE S D, TANG Y Z, WANG S, et al. Privacy-preserving GWAS analysis on federated genomic datasets [J]. BMC Medical Informatics and Decision Making, 2015, 15(Suppl 5): S2 1-9.

[54] LINDELL Y, PINKAS B. Privacy preserving data mining [C]//Proceedings of the 20th Annual International Cryptology Conference. Berlin: Springer, 2000: 36-54.

[55] QUINLAN J R. Induction of decision trees[J]. Machine Learnin, 1986, 1(1): 81-106.

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

[57] PEDERSEN T P. A threshold cryptosystem without a trusted party [C]//Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques in Advances in Cryptology 1991. Berlin: Springer, 1991: 522-526.

[58] CARMIT H, MUTHURAMAKRISHNAN V, MOR W. The price of active security in cryptographic protocols [C]//Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology 2020. Cham: Springer, 2020:184-215.

[59] MARCEL K. MP-SPDZ: A versatile framework for multi-party computation[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2020: 1575-1590.

[60] JUNG W, KIM S, AHN J H, et al. Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021, 2021(4): 114-148.

[61] YANG K, WENG C, LAN X, et al. Ferret: Fast extension for correlated ot with small communication[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2020: 1607-1626.

[62] YANG K, WANG X, ZHANG J. More Efficient mpc from improved triple generation and authenticated garbling[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2020: 1627-1646.

[63] RIAZI M S, WEINERT C, TKACHENKO O, et al. Chameleon: A hybrid secure computation framework for machine learning applications[C]//Proceedings of the 2018 on Asia Conference on Computer and Communications Security. New York: Association for Computing Machinery, 2018: 707-721.

[64] JUVEKAR C, VAIKUNTANATHAN V, CHANDRAKASAN A. GAZELLE: A low latency framework for secure neural network inference[C]//Proceedings of the 27th USENIX Conference on Security Symposium 2018. New York: ACM, 2018: 1651-1668.

[65] CHANDRAN N, GUPTA D, RASTOGI A, et al. EzPC: Programmable and efficient secure two-party computation for machine learning[C]//Proceedings of the IEEE European Symposium on Security and Privacy 2019. Piscataway: IEEE, 2019: 496-511.

[66] SHAIK I, SINGH A K, NARUMANCHI H, et al. A recommender system for efficient implementation of privacy preserving machine learning primitives based on TFHE [C]//Proceedings of the 4th International Symposium on Cyber Security Cryptology and Machine Learning. Cham: Springer, 2020: 193-218.

## 作者简介

**田海博** 男,1979年出生,河北深州人. 主要研究方向为密码学应用,包括人工智能、区块链的隐私保护等.

E-mail: tianhb@mail.sysu.edu.cn

**梁岫琪** 男,1997年出生,广东肇庆人. 主要研究方向为人工智能隐私保护.

E-mail: liangxq8@mail2.sysu.edu.cn