联　　盟　　规　　范

PPCA 8 —2023

隐私计算 跨平台互联互通

开放协议 第 3 部分：基于半同态的联邦线性回归

Privacy-preserving computation cross-platform interconnection—

Open protocol Part 3: Federated linear regression based on partially homomorphic encryption

2023-07-26 发布　　2023-07-26 实施

隐私计算联盟　发 布

目　次

前　言

本文件是《隐私计算 跨平台互联互通》系列文件之一，该系列标准的结构和名称如下：

——开放协议 第1部分：ECDH-PSI；

——开放协议 第2部分：SS-LR；

——开放协议 第3部分：PHE-FLR。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本文件由隐私计算联盟提出并归口。

本文件起草单位：中国信息通信研究院、联通数字科技有限公司、北京百度网讯科技有限公司、北京数牍科技有限公司。

本文件主要起草人：孙林、苏亮、徐文静、荆博、王虹妍、单进勇、白桂东、白玉真、袁博、王思源。

引　言

当前多方安全计算、联邦学习等隐私计算技术快速发展，越来越多的产品从试点部署阶段转入落地应用，市场竞争火热。但是，不同技术厂商提供的产品和解决方案在设计原理和功能实现之间存在较大差异，使得部署于不同平台的隐私计算参与方之间无法跨平台完成同一计算任务，为实现与部署于不同平台的多个合作方之间的数据融合，用户往往不得不部署多套产品以逐一适配。作为促进跨机构间数据共享融合的关键技术，隐私计算有望成为支撑数据流通产业的基础设施，但高额的应用成本不利于隐私计算技术的推广应用。因此，解决不同产品之间的技术壁垒，实现计算任务在跨平台间的互联互通已成为产业内的迫切需求。

版权声明

隐私计算 跨平台互联互通
开放协议 第3部分：基于半同态的联邦线性回归

1 范围

本文件规定了异构隐私计算平台进行跨平台的基于半同态的联邦线性回归（PHE-FLR）的算法协议和传输层实现参考。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1

隐私计算 privacy-preserving computation
在保证数据提供方不泄露原始数据的前提下，对数据进行分析计算的一系列信息技术，保障数据在流通与融合过程中的“可用不可见”。

3.2

开放协议 open protocol
通过定义算法执行流程中交互信息，各平台独立开发算法来实现平台间算法的互通。

4 缩略语

下列缩略语适用于本文件。
FLR：联邦线性回归（Federated Linear Regression）
PHE：半同态加密（Partially Homomorphic Encryption）
PHE-FLR：基于半同态的联邦线性回归（Federated Linear Regression based on Partially Homomorphic Encryption）

5 概述

5.1 PHE-FLR 算法概述

线性回归是一种拥有两个或两个以上自变量的回归算法，可以用来分析自变量和因变量之间的线性关系。本协议针对的是两个参与方纵向建模的场景，同一个样本的特征输入

分散在不同的参与方手中，每个参与方都无法单独完成训练，需要合作来完成建模过程。整个训练过程默认采用梯度下降法进行训练，梯度和损失值都在同态加密后进行计算。

5.2 PHE-FLR 算法流程

算法流程包含两个阶段，第一阶段为算法协商握手阶段，第二阶段为主体运行阶段。

算法协商握手阶段，用来确定加密算法类型、学习率，正则化方法，模型收敛终止条件等算法运行所需的公共信息。

算法主体运行阶段，通过多轮迭代来更新模型，直到模型收敛终止训练，两方需要共同计算梯度和代价函数的计算。

5.2.1 损失函数计算

损失函数为$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y}^{(i)}-y^{(i)}\right)^2+\mathrm{L}$，其中:

1）$\hat{y}^{(i)}$为第i个样本的预测值，$y^{(i)}$为第i个样本的真实值。

2) L为正则化损失项，如果为L1正则化，L=$\frac{\lambda}{m}(|\theta_1|+\ldots+|\theta_n|+|b|)$，如果为L2正则化，则L=$\frac{\lambda}{2m}(\theta_1^2+\ldots+\theta_n^2+b^2)$。

3) $\left(\hat{y}^{(i)}-y^{(i)}\right)^2=\left(\hat{y}^{(i_A)}+\hat{y}^{(i_B)}-y^{(i)}\right)^2=\hat{y}^{(i_A)2}+\left(\hat{y}^{(i_B)}-y^{(i)}\right)^2+2*\hat{y}^{(i_A)}*\left(\hat{y}^{(i_B)}-y^{(i)}\right)$，$y^{(i_A)}=\sum_{k=1}^{n_A}\theta_k*x_{ik}$为A利用本地样本特征计算的部分预测值，$y^{(i_B)}=\sum_{k=1}^{n_B}\theta_k*x_{ik}+b$ 为B利用本地样本特征计算的部分预测值，b为偏置值。

5.2.2 梯度计算和权重更新方法

对于L1正则化，$\theta_j=\theta_j-\alpha*\frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}^{(iA)}+\hat{y}^{(iB)}-y^{(i)}\right)*x_{ij}+\frac{\lambda}{m}sign\left(\theta_j\right)$，对于L2正则化，$\theta_j=\theta_j-\alpha*\frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}^{(iA)}+\hat{y}^{(iB)}-y^{(i)}\right)*x_{ij}+\frac{\lambda}{m}\theta_j$

其中，α为学习率，λ为正则化参数，sign(θ)为θ的符号，即：θ>0,sign(θ)=1; θ<0,sign(θ)=-1; otherwise, sign(θ)=0

5.2.3 两方计算步骤

假设有A和B两个参与方参与训练，且B方为标签方，双方纵向建模，包含以下步骤：

第一步：A/B两方各自生成随机同态公私钥对后，双方互相交换同态公钥。

第二步：A/B两方各自计算本方的部分预测值，再用本方同态公钥进行加密，各方计算预测值如下：

A方计算同态加密结果 $\mathrm{encByPk_A}\left(\hat{y}^{(\mathrm{i_A})}\right)$, $\mathrm{encByPk_A}\left(\hat{y}^{(\mathrm{i_A})2}\right)$及$\mathrm{encByPk_A}(\mathrm{L_A})$，其中$L_A$是A方的正则损失项，$\mathrm{encByPk_A}$表示用A方公钥加密。

B方计算同态加密结果 $encByPk_B\left(\hat{y}^{(i_B)}-y^{(i)}\right)$，$encByPk_B\left(\left(\hat{y}^{(i_B)}-y^{(i)}\right)^2\right)$及$encByPk_B(L_B))$，其中$L_B$是B方的正则损失项，$\mathrm{encByPk_B}$表示用B方公钥加密。

第三步：A/B两方交换同态加密后的部分预测值。

第四步：A/B两方将本地梯度和损失值用对方同态公钥进行加密，与上一步接收到的加密值一起计算梯度和损失值，为保护数据隐私，计算结果都加上同态加密后的随机数进行混淆。其中，两方各自进行梯度和损失值计算的公式如下：

A方进行的加密梯度计算： $encGradForA = encByPk_B\left(\left(\hat{y}^{(iA)} + \hat{y}^{(iB)} - y^{(i)}\right) * x_i\right) = encByPk_B\left(\hat{y}^{(i_A)} * x_{iA}\right) + HE\left(encByPk_B\left(\hat{y}^{(i_B)} - y^{(i)}\right), x_{i_A}\right) + encByPk_B(randA)$

A 方进行的损失值计算： $encCostForA = encByPk_B\left(\hat{y}^{(i_A)2}\right) + encByPk_B\left(\left(\hat{y}^{(i_B)} - y^{(i)}\right)^2\right) + HE\left(encByPk_B\left(\hat{y}^{(i_B)} - y^{(i)}\right), 2\hat{y}^{(i_A)}\right) + encByPk_B(randA\_L) + encByPk_B(L_A) + encByPk_B(L_B)$

B方进行加密梯度计算： $encGradForB = encByPk_A\left(\left(\hat{y}^{(iA)} + \hat{y}^{(iB)} - y^{(i)}\right) * x_i\right) = HE\left(encByPk_A\left(\hat{y}^{(i_A)}\right), x_{i_B}\right) + encByPk_A\left(\left(\hat{y}^{(i_B)} - y^{(i)}\right) * x_{i_B}\right) + encByPk_A(randB)$

B 方进行的损失值计算： $encCostForB = encByPk_A\left(\left(\hat{y}^{(i_B)} - y^{(i)}\right)^2\right) + encByPk_A\left(\hat{y}^{(i_A)2}\right) + HE\left(encByPk_A\left(\hat{y}^{(i_A)}\right), 2\left(\hat{y}^{(i_B)} - y^{(i)}\right)\right) + encByPk_A(randB\_L) + encByPk_A(L_A) + encByPk_A(L_B)$

其中，randA、randA_L、randB、randB_L 是 A/B 两方各自产生的随机数，要求至少104 比特级别，HE 代表同态加密，损失值用于后续判定是否满足迭代结束条件。

第五步：A/B两方交换混淆后的加密梯度和损失值。

第六步：A/B两方都用本地私钥解密上一步接收的梯度和损失值。

第七步：A/B两方互相交换解密后的梯度和损失值。

第八步：A/B两方得到解密后的梯度和损失值后，去除混淆随机数，恢复和更新梯度和损失值。

其中，更新损失值的公式见5.2.1，更新梯度和权重的公式见5.2.2。

第九步：A/B两方各自判定训练是否达到收敛，如果收敛，本方发送停止消息后结束训练任务，对方收到停止消息后也结束训练；如果没有收敛，重复执行第二步至第八步。

其中，收敛条件包含：

1）迭代轮数超过双方协商的最大迭代轮数。

2）两次迭代的损失值差值小于双方协商的差值阈值。

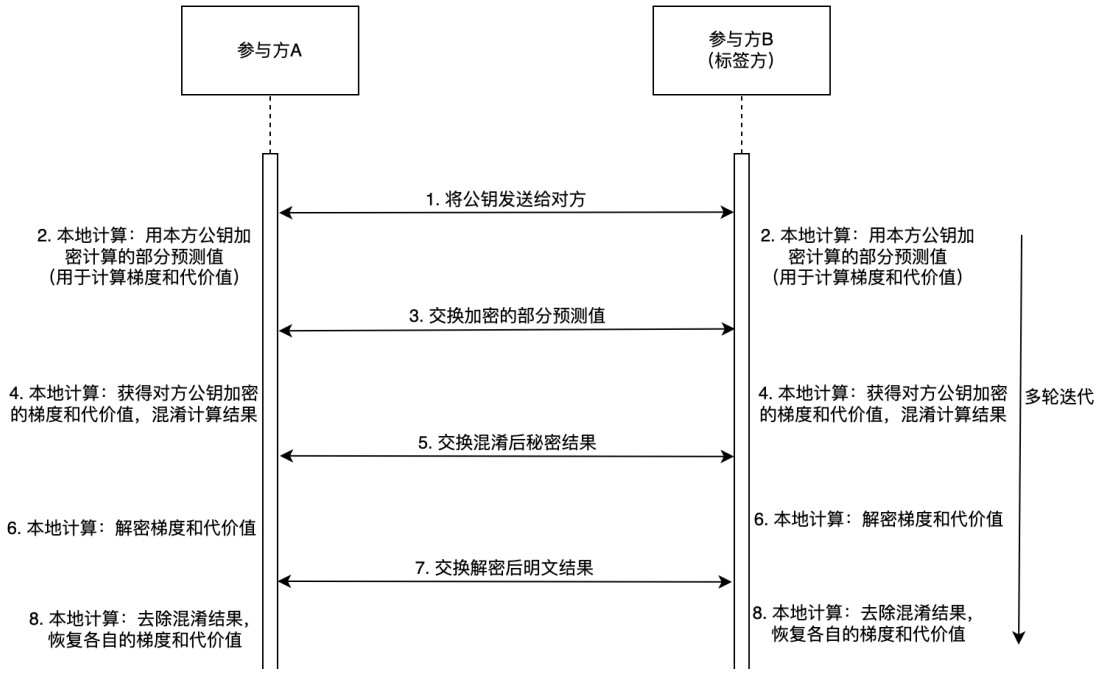

图1 PHE-FLR算法流程图

6 协议配置

6.1 安全参数

计算安全等级：112比特级别。

6.2 密码算法

6.2.1 Paillier 算法

Paillier是一种半同态加密算法，它可以在不解密的情况下对密文进行加法运算，本协议Paillier的秘钥长度为2048位。

6.2.2 Paillier 的加解密精度

算法运算过程中梯度和损失值都为浮点数，同态加密只能处理整数，可以根据精度取值将浮点数做处理，例如精度取值为6，要加密的梯度值grad为浮点数，可将grad转换为grad’ = round(grad*10^6) 进行同态加密运算，解密后再除以10^6。

7 算法协商握手

7.1 HandshakeRequest 消息

HandshakeRequest包括算法协商握手请求的基本信息，其数据结构如表1所示。

表 1 HandshakeRequest 数据结构

属性名称	数据类型	数据说明	示例	数据备注
algo_method	string	采用的密码算法，包含密码算法和安全强度，默认 paillier_2048	“paillier_2048”	必选
learning_rate	float	梯度下降每次迭代的学习率	0.01	必选
update_method	string	样本更新方式，可选 mini_batch(批量更新),full_batch（全量更新），默认 mini_batch	“mini_batch”	必选
batch_size	int32	梯度下降的批量样本集大小，只在 update_method 选取 min_batch 生效	100	必选
loss_diff	float	损失值的差值阈值，两次迭代之间的差值小于该阈值，迭代终止	0.0001	必选
max_iterations	int32	迭代的最大轮数，超过最大轮数迭代终止，-1 为不限制轮数	20	必选
phe_precison	int32	精度取值，同态加密的浮点数转换为整数时所乘的 10 的幂次	5	必选
regularizer	string	正则化方法,支持 L1、L2,默认 L2	“l2”	必选
regularizer_scale	float	正则化系数	0.5	必选

7.2 HandshakeResponse 消息

HandshakeResponse消息结构包括算法协商握手响应的基本信息，其数据结构如表2所示。

表 2 HandshakeResponse 数据结构

属性名称	数据类型	数据说明	示例	数据备注
header	ResponseHeader	握手请求响应头。ResponseHeader 的格式见表 3	见表	必选
algo_method	string	对方决策下来的密码算法	“paillier_2048”	必选
learning_rate	float	对方决策下来的学习率	0.01	必选
update_method	string	对方决策下来的样本更新方式	“mini_batch”	必选
batch_size	int32	对方决策下来的梯度下降的批量样本集大小	100	必选
loss_diff	float	对方决策下来的 loss 函数的差值	0.0001	必选
max_iterations	int32	对方决策下来的迭代的最大轮数	20	必选
phe_precison	int32	对方决策下来的精度取值	5	必选

属性名称	数据类型	数据说明	示例	数据备注
regularizer	string	对方决策下来的正则化方法	“l2”	必选
regularizer_scale	float	对方决策下来的正则化系数	0.5	必选

ResponseHeader消息结构包括的基本信息如表3所示

表 3 ResponseHeader 数据结构

属性名称	数据类型	数据说明	数据备注
error_code	Int32	握手响应的结果，其值域如表 4 所示。	必选
error_msg	string	用户自定的消息字符串	可选

error_code的取值和说明如表4所示。

表 4 error_code 的值域表

数值	数据说明
0	成功
31100000	GENERIC_ERROR，通用错误
31100001	UNEXPECTED_ERROR，状态不符合预期错误
31100002	NETWORK_ERROR，网络通信错误
31100100	INVALID_REQUEST，非法请求
31100101	OUT_OF_RESOURCE，运行资源不满足
31100200	HANDSHAKE_REFUSED，握手拒绝
31100201	UNSUPPORTED_VERSION，不支持的版本
31100202	UNSUPPORTED_ALGO，不支持的算法
31100203	UNSUPPORTED_PARAMS，不支持的算法参数

8 算法主体运行

8.1 传输同态公钥的消息

传输本方同态公钥给对方时，发送本条消息，数据结构如表 5 所示。

表 5 传输同态公钥的消息数据结构

属性名称	数据类型	数据说明	示例	数据备注
type	int32	标识消息的类型，取值 5	5	必选
home_pubkey	bytes	同态公钥	需要传输的实际信息	必选

8.2 传输本地计算部分预测值的消息

已完成本地部分预测值计算后，发送本条消息，数据结构如表6所示。

表 6 传输本地计算部分预测值的消息数据结构

属性名称	数据类型	数据说明	示例	数据备注
type	int32	标识消息的类型，取值 8	8	必选
loop_round	int32	迭代轮数	2	必选
part_bytes	bytes	加密的部分预测值	需要传输的实际信息	必选

8.3 传输用对方同态公钥加密的本轮最终梯度和损失值的消息

已接收完本地和其他方的部分加密梯度和损失值，并计算完成本轮最终的加密梯度和损失值，发送本条消息，数据结构如表7所示。

表 7 传输同态公钥加密的本轮最终梯度和损失值的消息数据结构

属性名称	数据类型	数据说明	示例	数据备注
type	int32	标识消息的类型，取值 10	10	必选
loop_round	int32	迭代轮数	2	必选
enc_grad_from_other	bytes	完整加密梯度	需要传输的实际信息	必选
enc_cost_from_other	bytes	完整加密损失值	需要传输的实际信息	必选

8.4 传输用对方同态私钥解密的本轮最终梯度和损失值的消息

已接收到其他方最终的加密梯度和损失值，并用本方私钥完成解密后，发送本条消息，数据结构如表 8 所示。

表 8 传输用对方同态私钥解密的本轮最终梯度和损失值的消息数据结构

属性名称	数据类型	数据说明	示例	数据备注
type	int32	标识消息的类型，取值 12	12	必选
loop_round	int32	迭代轮数	2	必选
grad_bytes	bytes	解密梯度	需要传输的实际信息	必选
cost_bytes	bytes	解密损失值	需要传输的实际信息	必选

8.5 任务停止消息

触发模型迭代停止条件时，发送本条消息，数据结构如表 9 所示。

表 9 任务停止消息数据结构

属性名称	数据类型	数据说明	示例	数据备注
type	int32	标识消息的类型，取值 14	14	必选
loop_round	int32	迭代轮数	2	必选
stopped	int32	是否停止，1 为停止，0 为继续执行	1	必选

参 考 文 献

[1]ISO/IEC 18033-6:2019 IT Security techniques— Encryption algorithms —Part 6: Homomorphic encryption
[2]机器学习术语表 https://developers.google.cn/machine-learning/glossary/?hl=zh-CN