电 子 与 信 息 学 报
Journal of Electronics & Information Technology

FastProtector：一种支持梯度隐私保护的高效联邦学习方法

林 莉*①② 张笑盈① 沈 薇① 王万祥①
①(北京工业大学信息学部计算机学院 北京 100124)
②(可信计算北京市重点实验室 北京 100124)

摘 要：联邦学习存在来自梯度的参与方隐私泄露，现有基于同态加密的梯度保护方案产生较大时间开销且潜在参与方与聚合服务器合谋导致梯度外泄的风险，为此，该文提出一种新的联邦学习方法FastProtector，在采用同态加密保护参与方梯度时引入符号随机梯度下降(SignSGD)思想，利用梯度中正负的多数决定聚合结果也能使模型收敛的特性，量化梯度并改进梯度更新机制，降低梯度加密的开销；同时给出一种加性秘密共享方案保护梯度密文以抵抗恶意聚合服务器和参与方之间共谋攻击；在MNIST和CIFAR-10数据集上进行了实验，结果表明提出方法在降低80%左右加解密总时间的同时仍可保证较高的模型准确率。

关键词：联邦学习；梯度保护；低加密开销；共谋攻击
中图分类号：TP181; TP309; TP391 **文献标识码**：A **文章编号**：1009-5896(2022)00-0001-10
DOI: 10.11999/JEIT220161

FastProtector: An Efficient Federated Learning Method Supporting Gradient Privacy Protection

LIN Li①② ZHANG Xiaoying① SHEN Wei① WANG Wanxiang①
①(*College of Computer Science, Faculty of Information Technology, Beijing University of Technology, Beijing* 100124, *China*)
②(*Beijing Key Laboratory of Trusted Computing, Beijing* 100124, *China*)

Abstract: Federated learning has the problem of privacy leakage from the gradient. The existing gradient protection schemes based on homomorphic encryption incur a large time cost and the risk of gradient leakage caused by potential collusion between participants and aggregation server. A new federated learning method called FastProtector is proposed, where the idea of SignSGD is introduced when homomorphic encryption is used to protect participant gradients. Exploiting the feature that the majority of positive and negative gradients determine the aggregation result to still make the model convergent, the gradient is quantified and the gradient updating mechanism is improved, which can reduce the overhead of gradient encryption. Meanwhile, an additive secret sharing scheme is proposed to protect the gradient ciphertext against collusion attacks between malicious aggregation servers and participants. Experiments on MNIST and CIFAR-10 dataset show that the proposed method can reduce the total encryption and decryption time by about 80% while ensuring high model accuracy.

Key words: Federated learning; Gradient protection; Low encryption overhead; Collusion attacks

1 引言

当前数据融合需求迫切，机器学习技术在智慧医疗、智慧金融和智慧交通等领域得到前所未有的应用。然而，由于机器学习训练过程需要融合不同用户的本地数据，数据共享过程存在隐私泄露的风险。例如，医疗数据往往包含患者隐私，当多家医院共享数据时经常导致患者的隐私泄露[1–3]。为此，欧盟在2018年实施了《通用数据保护条例》(General Data Protection Regulation, GDPR)[4]；2020年《加利福尼亚州消费者隐私法案》(California Consumer Protection Act, CCPA)在美国加利福尼亚州正式生效[5]；我国在2017年实施《网络安

收稿日期：2022-02-22；改回日期：2022-11-16
*通信作者： 林莉 linli_2009@bjut.edu.cn
基金项目：国家自然科学基金项目(61502017)，北京市教委科技计划一般项目(KM201710005024)
Foundation Item: This work was supported by the National Natural Science Foundation of China (61502017) and the Scientific Research Common Program of Beijing Municipal Commission of Education (KM201710005024)

全法》[6]，2021年发布了《数据安全法》[7]和《个人信息保护法》[8]。随着这些法律法规的发布实施，不同组织间共享数据变得越来越困难，数据孤岛问题日益严重。

谷歌在2016年提出联邦学习解决数据孤岛问题[9]。联邦学习是一种分布式机器学习框架，多个参与方在本地训练，通过共享如梯度实现联合建模。尽管该方法可保证训练数据不出本地，但攻击者仍可通过这些共享的梯度反推出原始训练数据的内容[10, 11]，这会导致训练数据的隐私泄露，因而保护梯度尤为重要。现有工作提出了基于同态加密的梯度保护方法，例如，Zhou等人[12]利用Paillier加密方法在雾环境下保护参与方的梯度。Phong等人[13]提出了一种基于Paillier同态加密的联邦学习框架。Zhang等人[14]利用通过中国剩余定理降低了梯度加密开销，但由于中国剩余定理计算时要求被处理的数值不能过大，使得梯度取整时损失更多，会降低模型的准确率。Lohana等人[15]也利用Paillier同态加密保护上传的梯度，并通过只上传重要梯度来提高学习效率。上述方法虽安全性较高，但当需要对大量模型参数进行加密时开销往往较大，无法适应自动驾驶等实时性要求高的应用，故大部分现有方案都采用半同态加密[16]。此外，在现有方案中，参与方往往使用相同的一对加解密密钥，无法抵抗参与方与聚合服务器之间的共谋攻击[12]，Dong等人[17]结合秘密共享和Top-k梯度选择算法去实现在防止共谋攻击的同时验证服务端聚合结果的有效性，并验证该方法可提升联邦学习的通信效率，然而该方案中各方之间传输的是明文梯度共享，会带来隐私泄露风险[18]；Xia等人[19]针对纵向联邦学习场景提出了加性秘密共享方案，通过把明文计算转化成秘密份额计算以实现对参与方本地数据的隐私保护，但引入通信开销较大。Hao等人[20]利用差分隐私技术抵抗聚合服务器和参与方之间的共谋攻击，然而差分隐私技术会降低模型准确率[21]。Zhou等人[12]提出利用盲化技术抵抗聚合服务器和雾节点之间的共谋攻击，但前提是假设存在一个可信的盲化参数服务器来分发盲化参数。因此，如何提高加解密效率并能有效防止共谋攻击是联邦学习梯度隐私保护亟待解决的重要问题。

为此，本文提出一种支持梯度隐私保护的高效联邦学习方法FastProtector，主要贡献如下：

(1) 基于SignSGD思想[22]对梯度Paillier同态加密过程进行了简化，利用梯度中正负的多数决定聚合结果也能使模型收敛的特性，量化梯度并改进梯度更新机制，在增强梯度隐私保护的同时有效降低梯度加密的开销。

(2) 给出加性秘密共享技术，可抵抗梯度密文保护中聚合服务器和参与方之间潜在的共谋攻击。

(3) 我们已在MNIST和CIFAR-10两个数据集上进行了实验，结果表明本文方法可降低80%左右的加解密总时间，并且能确在降低加解密开销的同时保持良好的训练效果。

2 系统模型

本文主要针对多聚合服务器多参与方的复杂联邦学习应用[17]，场景如图1所示，包括特定聚合服务器、其他聚合服务器和参与方3类实体，其中特定聚合服务器和其他聚合服务器计算能力较强，负责生成密钥对，聚合参与方上传的梯度密文，并将生成的密钥对和聚合的梯度密文发给各个参与方。参与方在本地训练模型，在每轮训练中计算梯度和共享并对梯度共享进行加密，把加密的共享上传到聚合服务器，等待密钥对和聚合结果的下发。

这里特别指出，与文献[17]采用共享明文聚合不同，本文采用共享密文聚合，即通过加密梯度共享保证参与方之间、参与方与聚合服务器之间、聚合服务器之间只进行密文传输和聚合，增强对梯度共享的隐私保护。同时，为降低参与方的计算开销，本文选择一个计算能力较强的特定聚合服务器来完成该任务。

本文假设所有的聚合服务器和参与方都是诚实且好奇的，即会遵循协议执行过程但也会尝试根据收到的结果推断其他参与方的隐私信息。同时，本文也考虑各种潜在的共谋攻击，包括聚合服务器之间，参与方之间以及聚合服务器和参与方之间的共谋攻击。基于上述假设，本文目标是提出一种既能增强对梯度共享的隐私保护，又能抵抗多种共谋攻击的高效联邦学习方法。

3 方法设计

本节详细介绍提出FastProtector方法的方法架构、工作流程以及流程中各阶段涉及的核心算法。

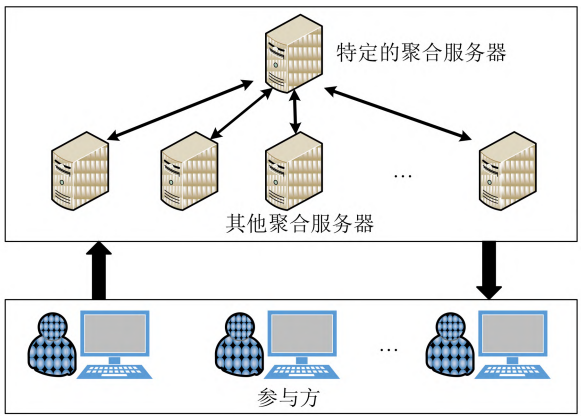


图 1 联邦学习场景

3.1 工作原理

FastProtector的实现需要在聚合服务器和参与方端部署不同的功能模块，如图2所示，特定聚合服务器上部署聚合服务器选择、密钥生成和梯度密文聚合模块；其他聚合服务器上部署密钥生成和共享加和模块；参与方端部署共享加密、基于SignSGD的共享生成、聚合梯度密文解密以及模型训练和测试模块。

如图3所示，本文提出方法分为初始化阶段、训练阶段、聚合阶段和更新阶段4个阶段。

(1) 初始化阶段。首先参与方们对特定聚合服务器发起请求，特定聚合服务器收到参与方请求后，会从所有聚合服务器中随机选择一个聚合服务器，基于采用Paillier加密算法生成密钥对，被选择的聚合服务器负责把密钥对分发给各个参与方，同时各个参与方在本地初始化训练所需参数，准备模型训练。

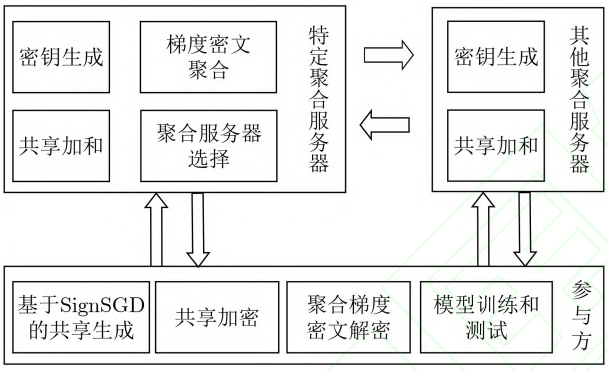


图 2 FastProtector方法架构

(2) 训练阶段。各个参与方在本地进行模型训练，然后基于SignSGD的思想，确定正负梯度量化的值，并对正负梯度量化结果值实施秘密共享算法将其分为多份共享，共享的份数与聚合服务器的个数相同，最后对生成的共享进行加密，并根据原始梯度中的正负将共享密文替换到对应位置，生成梯度共享密文。

(3) 聚合阶段。各个参与方将梯度共享密文依次上传给不同的聚合服务器，各个聚合服务器将收到的共享密文进行乘法运算，得到共享加和的结果，然后将结果统一发给特定聚合服务器，特定聚合服务器将收到的共享加和结果进行聚合，得到聚合梯度的密文，再下发给各个参与方。

(4) 更新阶段。各个参与方在本地对聚合梯度密文进行解密，并用解密后的聚合梯度更新模型，同时，准备下一轮模型训练，如果已达到预定的训练轮数，则训练终止。

下节开始将对各阶段核心算法进行详细介绍，为便于理解，先给出统一符号说明，具体如表1所示。

3.2 初始化阶段

该阶段主要由各个参与方完成训练所需参数的初始化，被特定聚合服务器选择的聚合服务器完成密钥对的生成和分发，初始化的参数和生成的密钥对后续用于训练阶段。不失一般性，设参与方的个数为$n(n\geq 3)$，聚合服务器的个数为$m(m\geq 3)$。

训练所需参数包括训练模型$\boldsymbol{M}$，学习率α和训练轮数epoch。

密钥对的生成过程包含以下几个步骤：

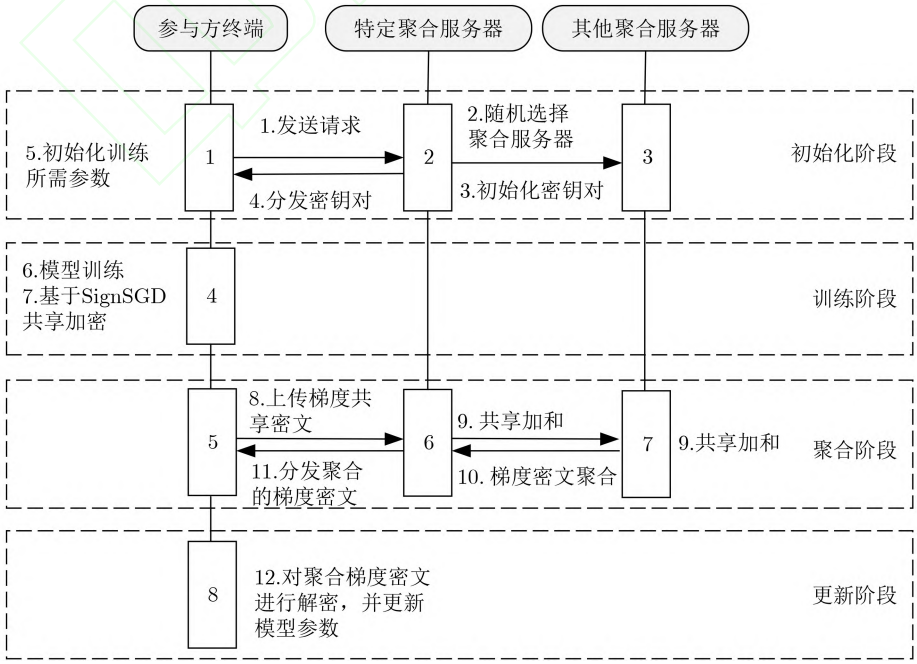


图 3 FastProtector的工作流程

表 1 变量符号说明

符号	类型	物理含义
n	标量	参与方个数
m	标量	聚合服务器个数
$\boldsymbol{M}$	张量	训练模型
α	标量	学习率
epoch	标量	训练轮数
pk	标量	公钥
sk	标量	私钥
$\boldsymbol{D}^i$	张量	参与方i的数据集
$\boldsymbol{D}^i_{\text{sub}}$	张量	参与方i的数据集的子集
$\boldsymbol{G}^i$	向量	参与方i的梯度，经过SignSGD方法处理后的梯度
$\boldsymbol{G}^i_j$	向量	参与方i的第j个梯度共享
$[[\boldsymbol{G}^i_j]]_{\text{pk}}$	列表	参与方i的第j个梯度共享密文
$[[\boldsymbol{G}^j_{\text{agg}}]]_{\text{pk}}$	列表	聚合服务器j的共享加和结果
$[[\boldsymbol{G}_{\text{agg}}]]_{\text{pk}}$	列表	特定聚合服务器下发的聚合梯度密文
pg	标量	正梯度量化的值
ng	标量	负梯度量化的值

(1) 初始化满足$gcd\,(p\cdot q,(p-1)\cdot(q-1))=1$的两个大素数$p$和$q$;

(2) 计算$n=p\cdot q$和$\lambda=lcm(p-1,\ q-1)$;

(3) 定义$\mathrm{L}(x)=(x-1)/n$;

(4) 取使$\mu=(\mathrm{L}(g^{\lambda}\bmod n^2))^{-1}\bmod n$存在的正整数 g，并保证$g<n^2$;

(5) 得到公钥$pk=(n,g)$和私钥$sk=(\lambda,\mu)$;

(6) 返回密钥对$(\mathrm{pk},\mathrm{sk})$。

3.3 训练阶段

该阶段主要是各个参与方完成模型训练和梯度共享加密。以参与方i为例，其过程如算法1所示。

3.4 聚合阶段

该阶段主要由各个参与方完成梯度共享密文的上传，聚合服务器完成共享加和，梯度密文的聚合以及聚合梯度密文的分发。共享加和以及梯度密文聚合的步骤如下。

(1)所有聚合服务器完成对参与方的共享的加和: $\left[\left[\boldsymbol{G}^j{}_{\text{agg}}\right]\right]_{\text{pk}}=\left[\left[\boldsymbol{G}^1_j\right]\right]_{\text{pk}}\cdot\ \left[\left[\boldsymbol{G}^2_j\right]\right]_{\text{pk}}\cdots\cdot\ \left[\left[\boldsymbol{G}^n_j\right]\right]_{\text{pk}}$, 其中$j=1,\ 2,\cdots,\ m$。

(2)特定聚合服务器收到其他聚合服务器发来的共享的加和结果后，完成梯度密文的聚合: $[[\boldsymbol{G}_{\text{agg}}]]_{\text{pk}}=\left[\left[\boldsymbol{G}^1_{\text{agg}}\right]\right]_{\text{pk}}\cdot\ \left[\left[\boldsymbol{G}^2_{\text{agg}}\right]\right]_{\text{pk}}\cdots\cdot\ \left[\left[\boldsymbol{G}^m_{\text{agg}}\right]\right]_{\text{pk}}$。

(3)最终，特定聚合服务器得到参与方梯度密文的聚合结果$[[\boldsymbol{G}_{\text{agg}}]]_{\text{pk}}$。

3.5 更新阶段

该阶段主要是各个参与方完成对聚合梯度密文的解密、模型的更新，其过程如算法2所示。

4 理论分析

本节从联邦学习聚合结果的正确性和梯度隐私保护能力两方面对提出方法FastProtector进行理论分析。

4.1 聚合结果的正确性

定理1 记特定聚合服务器下发的聚合梯度为$\boldsymbol{G}_{\text{agg}}$，参与方$i$经过Replace()方法处理后的梯度为$\boldsymbol{G}^i$，其中$i=1,\ 2,\cdots,\ n$且$n$为参与方个数，那么有$\boldsymbol{G}_{\text{agg}}=\boldsymbol{G}^1+\boldsymbol{G}^2+\cdots+\boldsymbol{G}^n$。

证明：特定聚合服务器下发的聚合梯度$\boldsymbol{G}_{\text{agg}}$应为所有参与方的梯度之和，根据算法1和算法2可知

$$
\begin{aligned}
\boldsymbol{G}_{\text{agg}} &= \mathrm{Decrypt}(\mathrm{pk},\mathrm{sk},[[\boldsymbol{G}_{\text{agg}}]]_{\text{pk}})=\mathrm{Decrypt}(\mathrm{pk},\mathrm{sk},\left[\left[G^1_{\text{agg}}\right]\right]_{\text{pk}}\cdot\left[\left[G^2_{\text{agg}}\right]\right]_{\text{pk}}\cdot\cdots\cdot\left[\left[\boldsymbol{G}^m_{\text{agg}}\right]\right]_{\text{pk}})\\
&=\mathrm{Decrypt}(\mathrm{pk},\mathrm{sk},\left[\left[\boldsymbol{G}^1_{\text{agg}}+G^2_{\text{agg}}+\cdots+\boldsymbol{G}^m{}_{\text{agg}}\right]\right]_{\text{pk}})\\
&=\mathrm{Decrypt}(\mathrm{pk},\mathrm{sk},\left[\left[\boldsymbol{G}^1_1+\boldsymbol{G}^2_1+\cdots+\boldsymbol{G}^n{}_1+\boldsymbol{G}^1_2+\boldsymbol{G}^2_2+\cdots+\boldsymbol{G}^n_2+\cdots+\boldsymbol{G}^1_m+\boldsymbol{G}^2_m+\cdots+\boldsymbol{G}^n_m\right]\right]_{\text{pk}})\\
&=\mathrm{Decrypt}(\mathrm{pk},\mathrm{sk},\left[\left[\boldsymbol{G}^1_1+\boldsymbol{G}^1_2+\cdots+\boldsymbol{G}^1_m+\boldsymbol{G}^2_1+\boldsymbol{G}^2_2+\cdots+\boldsymbol{G}^2_m+\cdots+\boldsymbol{G}^n_1+\boldsymbol{G}^n_2+\cdots+\boldsymbol{G}^n_m\right]\right]_{\text{pk}})
\end{aligned}
$$

证毕

算法1 模型训练和基于SignSGD的梯度共享加密

输入：数据集$\boldsymbol{D}^i$，模型$\boldsymbol{M}$，训练轮数epoch，正梯度量化的值pg，负梯度量化的值ng，公钥pk；

输出：梯度共享密文$\left[\left[\boldsymbol{G}^i{}_1\right]\right]_{\mathrm{pk}}, \left[\left[\boldsymbol{G}^i{}_2\right]\right]_{\mathrm{pk}}, \cdots, \left[\left[\boldsymbol{G}^i{}_m\right]\right]_{\mathrm{pk}}$。

(1) **for** ep=1 to epoch

(2) $\boldsymbol{D}^i{}_{\mathrm{sub}} = \mathrm{GetSubset}\left(\boldsymbol{D}^i\right)$; /*求解$\boldsymbol{D}^i$的随机子集*/

(3) $F^{\mathrm{i}}{}_{\mathrm{loss}} = \dfrac{1}{\mathrm{b}}\sum_{(\mathrm{x_l},\mathrm{y_l})\in \mathrm{D^i}_{\mathrm{sub}}} f(x_l, M, y_l)$; /*计算损失值*/

(4) $\boldsymbol{G}^i = \dfrac{\delta F^i{}_{\mathrm{loss}}}{\delta M}$; /*计算梯度*/

(5) $\boldsymbol{G}^i = \mathrm{Replace}\left(\boldsymbol{G}^i\right)$; /*将梯度$\mathrm{G}^i$中的正值替换为pg，负值替换为ng*/

(6) $\mathrm{pg}_{\mathrm{share}} = \mathrm{GetAdditiveShares}\left(\mathrm{pg}\right)$; /*计算得到pg的$m$份共享*/

(7) $\mathrm{ng}_{\mathrm{share}} = \mathrm{GetAdditiveShares}\left(\mathrm{ng}\right)$; /*计算得到ng的$m$份共享*/

(8) $\left[\left[\mathrm{pg}_{\mathrm{share}}\right]\right]_{\mathrm{pk}} = \mathrm{Encrypt}\left(\mathrm{pk}, \mathrm{pg}_{\mathrm{share}}\right)$; /*加密pg的$m$份共享*/

(9) $\left[\left[\mathrm{ng}_{\mathrm{share}}\right]\right]_{\mathrm{pk}} = \mathrm{Encrypt}\left(\mathrm{pk}, \mathrm{ng}_{\mathrm{share}}\right)$; /*加密ng的$m$份共享*/

(10) $\boldsymbol{G}_1^i = \boldsymbol{G}^i, \boldsymbol{G}_2^i = \boldsymbol{G}^i, \cdots, \boldsymbol{G}_m^i = \boldsymbol{G}^i$; /*复制$m$份相同的$\boldsymbol{G}^i$ */

(11) $\boldsymbol{G}_1^i{}' = \boldsymbol{G}_1^i{}'.\mathrm{tolist}(), \boldsymbol{G}_2^i{}' = \boldsymbol{G}_2^i{}'.\mathrm{tolist}(), \cdots, \boldsymbol{G}_m^i{}' = \boldsymbol{G}_m^i{}'.\mathrm{tolist}()$; /*将$\boldsymbol{G}_1^i{}', \boldsymbol{G}_2^i{}', \cdots, \boldsymbol{G}_m^i{}'$转换为列表类型*/

(12) $\mathrm{datanum} = \mathrm{len}(\boldsymbol{G}^i)$; /*获得G^i中元素个数*/

(13) **for** $\mathrm{num} = 0$ to $\mathrm{datanum} - 1$ /*根据G^i中的正负将共享密文替换到对应位置*/

(14) **if** $\boldsymbol{G}^i\,[\mathrm{num}] > 0$

(15) $G_1^i{}'\,[\mathrm{num}] = \left[\left[\mathrm{pg}_{\mathrm{share}}\right]\right]_{\mathrm{pk}}[0], G_2^i{}'\,[\mathrm{num}] = \left[\left[\mathrm{pg}_{\mathrm{share}}\right]\right]_{\mathrm{pk}}[1], \cdots, G_m^i{}'[\mathrm{num}] = \left[\left[\mathrm{pg}_{\mathrm{share}}\right]\right]_{\mathrm{pk}}[m-1]$;

(16) **else**

(17) $G_1^i{}'\,[\mathrm{num}] = \left[\left[\mathrm{ng}_{\mathrm{share}}\right]\right]_{\mathrm{pk}}[0], G_2^i{}'\,[\mathrm{num}] = \left[\left[\mathrm{ng}_{\mathrm{share}}\right]\right]_{pk}[1], \cdots, G_m^i{}'[\mathrm{num}] = \left[\left[ng_{\mathrm{share}}\right]\right]_{\mathrm{pk}}[m-1]$;

(18) **end if**

(19) **end for**

(20) $\left[\left[\boldsymbol{G}_1^i\right]\right]_{\mathrm{pk}} = \boldsymbol{G}_1^i{}', \left[\left[\boldsymbol{G}_2^i\right]\right]_{\mathrm{pk}} = \boldsymbol{G}_2^i{}', \cdots, \left[\left[\boldsymbol{G}_m^i\right]\right]_{\mathrm{pk}} = \boldsymbol{G}_m^i{}'$;

(21) Return $\left[\left[\boldsymbol{G}_1^i\right]\right]_{\mathrm{pk}}, \left[\left[\boldsymbol{G}_2^i\right]\right]_{\mathrm{pk}}, \cdots, \left[\left[\boldsymbol{G}^i{}_m\right]\right]_{\mathrm{pk}}$;

(22)**end for**

算法2 解密和更新

输入：聚合梯度密文$\left[\left[\boldsymbol{G}_{\mathrm{agg}}\right]\right]_{\mathrm{pk}}$，公钥pk，私钥sk，模型$\boldsymbol{M}$，学习率$\alpha$，训练轮数epoch；

输出：更新后的模型$\boldsymbol{M}$。

(1) **for** ep=1 to epoch

(2) $G'_{\mathrm{agg}} = \mathrm{Decrypt}(\mathrm{pk}, \mathrm{sk}, \left[\left[\boldsymbol{G}_{\mathrm{agg}}\right]\right]_{\mathrm{pk}})$; /*聚合梯度密文的解密*/

(3) $\boldsymbol{G}_{\mathrm{agg}} = \mathrm{torch.tensor}(\boldsymbol{G}_{\mathrm{agg}}')$; /*将$\boldsymbol{G}_{\mathrm{agg}}'$转换为张量类型*/

(4) $\boldsymbol{M} = \boldsymbol{M} - \alpha \cdot \boldsymbol{G}_{\mathrm{agg}}/n$; /*更新模型*/

(5) Return $\boldsymbol{M}$;

(6)**end for**

4.2 梯度隐私保护能力

定理2 已知参与方i的梯度共享密文为$\left[\left[\boldsymbol{G}_1^i\right]\right]_{\mathrm{pk}}$, $\left[\left[\boldsymbol{G}_2^i\right]\right]_{\mathrm{pk}}, \cdots, \left[\left[\boldsymbol{G}_m^i\right]\right]_{\mathrm{pk}}$，$i = 1, 2, \cdots, n$，那么单个聚合服务器(包括负责生成密钥对的聚合服务器和特定聚合服务器)或单个参与方都无法得到其他任意参与方完整的梯度$\boldsymbol{G}^i$，$i = 1, 2, \cdots, n$。

证明 由加性秘密共享算法性质知，参与方i的梯度为$\boldsymbol{G}^i = \boldsymbol{G}_1^i + \boldsymbol{G}_2^i + \cdots + \boldsymbol{G}_m^i$，$i = 1, 2, \cdots, n$。对于负责梯度聚合计算的单个聚合服务器、负责生成密钥对的聚合服务器和特定聚合服务器，由本文提出方法FastProtector可知，他们只可能拥有参与方i的梯度共享$\left[\left[\boldsymbol{G}_1^i\right]\right]_{\mathrm{pk}}, \left[\left[\boldsymbol{G}_2^i\right]\right]_{\mathrm{pk}}, \cdots, \left[\left[\boldsymbol{G}_m^i\right]\right]_{\mathrm{pk}}$中的一个，因此无法获得参与方$i$的完整梯度$\boldsymbol{G}^i$的值，$i = 1, 2, \cdots, n$。特别地，当拥有$\left[\left[\boldsymbol{G}_{\mathrm{agg}}\right]\right]_{\mathrm{pk}}$的特定聚

合服务器也负责生成密钥对pk和sk时，即他可通过对$[[\boldsymbol{G}_{\text{agg}}]]_{\text{pk}}$解密得到$\boldsymbol{G}_{\text{agg}}$，而根据定理1有$\boldsymbol{G}_{\text{agg}}=\boldsymbol{G}^1+\boldsymbol{G}^2+\cdots+\boldsymbol{G}^n$成立，但$n\geq 3$，因此无法通过$\boldsymbol{G}_{\text{agg}}$得到$\boldsymbol{G}^1,\boldsymbol{G}^2,\cdots,\boldsymbol{G}^n$的具体值，即无法得到参与方$i$的完整的梯度$\boldsymbol{G}^i$，$i=1,2,\cdots,n$。

根据定理1，有$\boldsymbol{G}_{\text{agg}}=\boldsymbol{G}^1+\boldsymbol{G}^2+\cdots+\boldsymbol{G}^n$成立。单个参与方能够获得聚合梯度密文$[[\boldsymbol{G}_{\text{agg}}]]_{\text{pk}}$，并拥有pk, sk以及他们自己的梯度$\boldsymbol{G}^i$，根据上述条件，令$\boldsymbol{G}'_{\text{agg}}=\boldsymbol{G}_{\text{agg}}-\boldsymbol{G}^i=\boldsymbol{G}^1+\cdots+\boldsymbol{G}^{i-1}+\boldsymbol{G}^{i+1}+\cdots+\boldsymbol{G}^n$，由于$n\geq 3$，因此$\boldsymbol{G}'_{\text{agg}}$是至少两个数的和，故无法通过$\boldsymbol{G}_{\text{agg}}$得到$\boldsymbol{G}^1,\boldsymbol{G}^2,\cdots,\boldsymbol{G}^n$的具体值，即无法得到任意参与方的完整梯度。

综上，任意单个聚合服务器(包括负责生成密钥对的聚合服务器和特定聚合服务器)或任意单个参与方都无法得到其他任意参与方的完整梯度。证毕

定理3 已知参与方i的梯度共享密文为$[[\boldsymbol{G}_1^i]]_{\text{pk}}$, $[[\boldsymbol{G}_2^i]]_{\text{pk}},\cdots,[[\boldsymbol{G}_m^i]]_{\text{pk}}$，$i=1,2,\cdots,n$，那么当$m-1$个聚合服务器(包括负责生成密钥对的聚合服务器和特定聚合服务器)和$n-2$个参与方共谋时，共谋者均无法得到任意未共谋参与方的完整梯度$\boldsymbol{G}^i$，$i=1,2,\cdots,n$。

证明 当$m-1$个聚合服务器(包括负责生成密钥对的聚合服务器和特定聚合服务器)和$n-2$个在线参与方共谋时，参与共谋的$m-1$个聚合服务器拥有$[[\boldsymbol{G}_{\text{agg}}^1]]_{\text{pk}},[[\boldsymbol{G}_{\text{agg}}^2]]_{\text{pk}},\cdots,[[\boldsymbol{G}_{\text{agg}}^{m-1}]]_{\text{pk}}$，pk, sk以及$[[G_{\text{agg}}]]_{\text{pk}}$，参与共谋的$n-2$个参与方拥有$\boldsymbol{G}_m^1,\boldsymbol{G}_m^2,\cdots,\boldsymbol{G}_m^{n-2}$，因此他们可以通过解密获得$\boldsymbol{G}_{\text{agg}}^1,\boldsymbol{G}_{\text{agg}}^2,\cdots,\boldsymbol{G}_{\text{agg}}^{m-1}$和$G_{\text{agg}}$，接着，可以通过$\boldsymbol{G}_{\text{agg}}^m=\boldsymbol{G}_{\text{agg}}-(\boldsymbol{G}_{\text{agg}}^1+\boldsymbol{G}_{\text{agg}}^2+\cdots+\boldsymbol{G}_{\text{agg}}^{m-1})$获得$\boldsymbol{G}^m{}_{\text{agg}}$，而根据定理1可知，$\boldsymbol{G}_{\text{agg}}^m=\boldsymbol{G}_m^1+\boldsymbol{G}_m^2+\cdots+\boldsymbol{G}_m^n$，因此他们可以通过$\boldsymbol{G}_m^{n-1}+\boldsymbol{G}_m^n=\boldsymbol{G}_{\text{agg}}^m-(\boldsymbol{G}_m^1+\boldsymbol{G}_m^2+\cdots+\boldsymbol{G}_m^{n-2})$获得$\boldsymbol{G}^{n-1}{}_m$与$\boldsymbol{G}^n{}_m$的和，然而，根据$\boldsymbol{G}_m^{n-1}$与$\boldsymbol{G}_m^n$的和并不能推出$\boldsymbol{G}_m^{n-1}$和$\boldsymbol{G}^n{}_m$中任意一个的具体值，而第$n-1$个参与方的梯度为$\boldsymbol{G}^{n-1}=\boldsymbol{G}_1^{n-1}+\boldsymbol{G}_2^{n-1}+\cdots+\boldsymbol{G}_m^{n-1}$，第$n$个参与方的梯度为$\boldsymbol{G}^n=\boldsymbol{G}^n{}_1+\boldsymbol{G}^n{}_2+\cdots+\boldsymbol{G}_m^n$，由于他们无法获得$\boldsymbol{G}_m^{n-1}$和$\boldsymbol{G}_m^n$的值，也就无法计算出$\boldsymbol{G}^{n-1}$和$\boldsymbol{G}^n$的值，即无法获得任意未共谋参与方完整的梯度值。因此，当$m-1$个聚合服务器(包括负责生成密钥对的聚合服务器和特定聚合服务器)和$n-2$个参与方共谋时，他们无法获得任意未共谋参与方完整的梯度值。证毕

5 实验评估

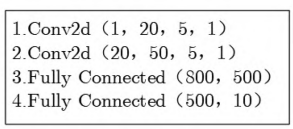
本节通过实验对提出方法进行评估，说明实验环境，给出模型准确率和加解密开销的实验结果及分析。

5.1 实验环境

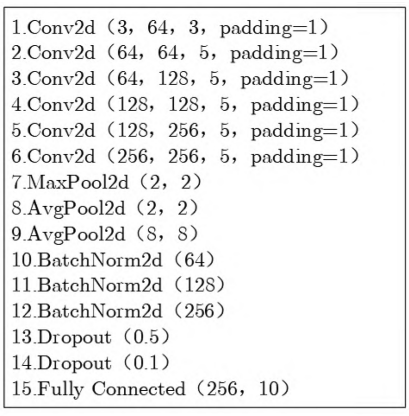
本文在Intel Core i7-9700F, 3.0 GHz CPU, 32 GB内存的机器上模拟3个聚合服务器和5个参与方，其中一个聚合服务器为特定聚合服务器。用PyTorch作为联邦学习的底层实现库，用Python3实现了联邦学习，采用多线程Socket实现联邦学习的通信。不失一般性，我们在经典数据集MNIST和CIFAR-10[13]上进行了实验，其中MNIST数据集包括60000个训练数据和10000个测试数据，采用的训练模型结构如图4(a)所示，总的梯度数量为431080。CIFAR-10数据集包括50000个训练数据和10000个测试数据，采用的训练模型结构如图4(b)所示，总的梯度数量为1148874。由于针对不同数据集的模型的收敛速度不同，因此，我们将MNIST和CIFAR-10的训练轮数分别设为800轮和2000轮，学习率均设为0.01。

5.2 模型准确率

本文针对MNIST和CIFAR-10数据集将本文方法FastProtector与原始联邦学习方法[9]和经典Pail-



(a) MNIST



(b) CIFAR-10

图 4 训练数据集的神经网络结构

lier加密联邦学习方法[13]比较模型准确率，其中原始联邦学习方法不进行加密操作，Paillier加密联邦学习方法直接对原始梯度进行加密。模型准确率通过计算测试集中预测正确的个数和总个数的比值得到，为了简化起见，我们只统计了200轮的训练结果，如图5、图6所示，经过200轮的模型训练，FastProtector与原始联邦学习方法以及Paillier加密联邦学习方法的模型准确率略有偏高但差异不大。

本文进一步统计了两个数据集上不同方法的最终模型准确率，如表2所示。不难看出，针对同一数据集，不同方法能够达到的最终模型准确率基本相同。

5.3 加解密开销

5.3.1 加密开销

本文针对MNIST和CIFAR-10数据集将本文提出方法FastProtector与经典Paillier加密方法[13]比较参与方的梯度加密开销。加密开销即基于SignSGD的共享生成和共享加密的总时间，实验结果如图7、图8所示。图7和图8分别针对两个数据集统计了64 bit长度密钥时不同训练轮数下的加密开销，

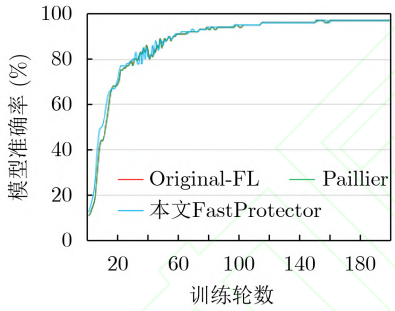


图 5 MNIST数据集上不同方法的模型准确率

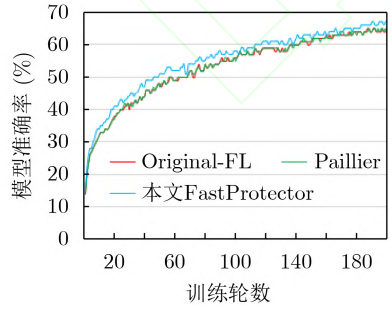


图 6 CIFAR-10数据集上不同方法的模型准确率

表 2 MNIST和CIFAR-10数据集上最终模型准确率(%)

方法	MNIST数据集 模型准确率	CIFAR-10数据集 模型准确率
Original-FL	99	84
Paillier	99	84
本文FastProtector	99	84

结果表明，对于两个不同数据集，随着训练的进行，FastProtector都比Paillier的加密时间更短，原因是FastProtector加密的共享数量少于Paillier加密的梯度数量。图9统计了64 bit长度密钥时不同梯度数量下的加密开销，结果表明，随着梯度数量的增加，Paillier和FastProtector的加密时间都呈线性增长，但FastProtector增长速度更慢且同样梯度数量下FastProtector的加密时间更短。图10统计了加密10000个梯度时不同密钥长度下的加密开销，结果表明，随着密钥长度的增加，Paillier和FastProtector的加密时间都呈线性增长，但FastProtector增长速度更慢且同样长度密钥下FastProtector的加密时间更短。

5.3.2 解密开销

本文针对MNIST和CIFAR-10数据集将FastProtector方法与经典Paillier加密方法[13]比较参与方梯度解密开销，密钥长度为64-bits。解密开销是

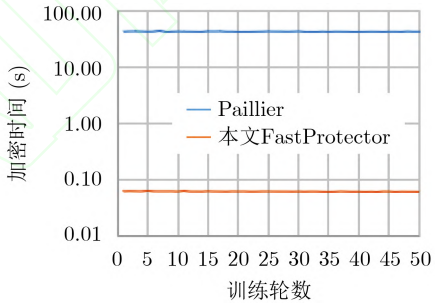


图 7 MNIST数据集上不同方法的加密开销

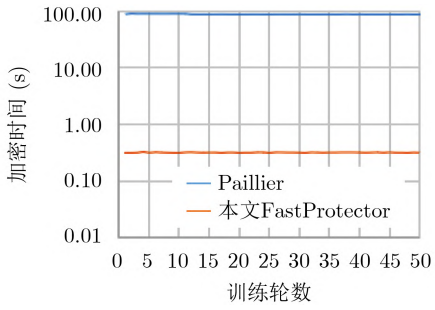


图 8 CIFAR-10数据集上不同方法的加密开销

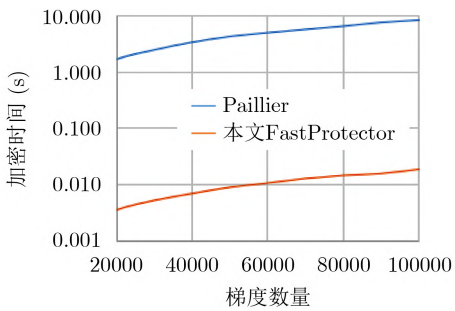


图 9 不同梯度数量的加密开销

指参与方对特定聚合服务器下发的聚合梯度密文解密的时间。实验结果如图11、图12所示，横坐标代表训练轮数，纵坐标代表解密时间，随着训练的进行，FastProtector的解密时间与经典Paillier解密方法几乎相同，因为都采用Paillier解密算法且解密梯度数量相同，因此时间相差不大。

综合5.3.1和5.3.2可知，FastProtector方法的加解密开销比经典Paillier方法要少得多，约降低80%左右, 这主要由FastProtector方法的加密开销降低实现。

5.4 训练时间

本文针对MNIST数据集测量参与方在本地完成若干轮训练的总时间开销，并与原始联邦学习方法、经典Paillier加密方法[13]进行比较，其中密钥长度为64-bits。对于FastProtector，训练一轮的总时间包括模型训练，梯度计算，基于SignSGD的共享生成，共享加密，共享密文上传，共享加和，共享加和传输，梯度密文聚合，聚合梯度密文下发，聚合梯度密文解密，模型更新的时间；对于原始联邦学习方法，训练一轮的总时间包括模型训练，梯度计算，梯度上传，梯度聚合，聚合梯度下发，模型更新的时间。实验结果如图13所示，FastProtector的一轮训练时间比原始的联邦学习方法要长，但增加的时间在可接受范围内，同时，FastProtector更加安全。FastProtector的一轮训练时间比Paillier加密方法要短得多，原因是FastProtector的共享加密时间比经典Paillier加密方法的梯度加密时间少很多，因此，本文提出方法FastProtector比经典Paillier加密方法效率高。

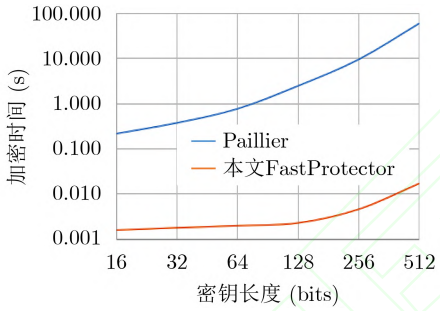


图 10 不同密钥长度的加密开销

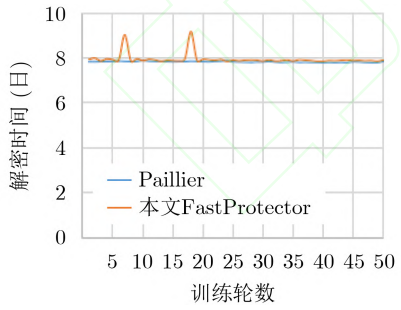


图 11 MNIST数据集上不同方法的解密开销

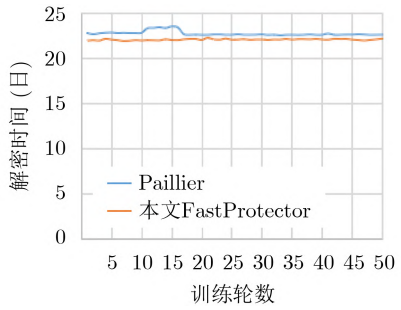


图 12 CIFAR-10数据集上不同方法的解密开销

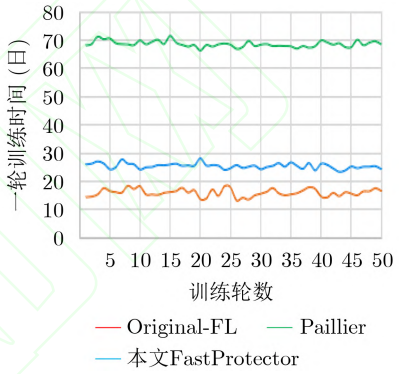


图 13 MNIST数据集上不同方法的一轮训练时间开销

6 结论与下一步工作

为解决联邦学习中的隐私泄露问题，本文提出了一种新的联邦学习方法FastProtector，采用同态加密并引入基于SignSGD思想的梯度加密方案来保护参与方上传的梯度，同时给出加性秘密共享算法来抵抗参与方和聚合服务器之间的共谋攻击，在MNIST和CIFAR-10数据集上的实验结果表明，FastProtector在降低梯度加解密开销的同时可保证较高的模型准确率，并能有效保护梯度，实现隐私保护、效率和训练效果之间的平衡。

本文提出方法FastProtector在使用秘密共享算法后，会导致参与方上传共享密文的数据量增加，虽可抵抗共谋攻击，但也会给参与方与聚合服务器之间增加额外通信开销；此外，本文提出的梯度加密方法只减少了加密时间，对解密时间没有影响，因此，在未来工作中，我们将进一步探索可抵抗共谋攻击并尽量减少额外通信开销的联邦学习方法以及可同时降低加密和解密开销的梯度加密算法。

参 考 文 献

[1] JEON J, KIM J, KIM J, *et al*. Privacy-preserving deep learning computation for geo-distributed medical big-data

platforms[C]. 2019 49th IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume, Portland, USA, 2019: 3–4. doi: 10.1109/DSN-S.2019.00007.

[2] LIU Yang, MA Zhuo, LIU Ximeng, *et al*. Privacy-preserving object detection for medical images with faster R-CNN[J]. *IEEE Transactions on Information Forensics and Security*, 2022, 17: 69–84. doi: 10.1109/TIFS.2019.2946476.

[3] VIZITIU A, NIȚĂ C I, PUIU A, *et al*. Towards privacy-preserving deep learning based medical imaging applications[C]. 2019 IEEE International Symposium on Medical Measurements and Applications, Istanbul, Turkey, 2019: 1–6. doi: 10.1109/MeMeA.2019.8802193.

[4] Intersoft consulting. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data and repealing Directive 95/46/EC (General Data Protection Regulation) [EB/OL]. https://gdpr-info.eu, 2020.

[5] DLA Piper. Data protection laws of the world: Full handbook[EB/OL]. https://www.dlapiperdataprotection.com, 2021.

[6] 中华人民共和国网络安全法(全文)[EB/OL]. http://www.zgyq.gov.cn/zwzxrdzt/xfzl/202208/t20220819_76128304.html, 2022.
Cybersecurity law of the People's Republic of China[EB/OL]. http://www.zgyq.gov.cn/zwzxrdzt/xfzl/202208/t20220819_76128304.html, 2022.

[7] 中华人民共和国数据安全法[EB/OL]. http://www.npc.gov.cn/npc/c30834/202106/7c9af12f51334a73b56d7938f99a788a.shtml, 2021.
Data security law of the People's Republic of China[EB/OL]. http://www.npc.gov.cn/npc/c30834/202106/7c9af12f51334a73b56d7938f99a788a.shtml, 2021.

[8] 中华人民共和国个人信息保护法[EB/OL]. http://www.npc.gov.cn/npc/c30834/202108/a8c4e3672c74491a80b53a172bb753fe.shtml, 2021.
Personal information protection law of the People's Republic of China[EB/OL]. http://www.npc.gov.cn/npc/c30834/202108/a8c4e3672c74491a80b53a172bb753fe.shtml, 2021.

[9] MCMAHAN H B, MOORE E, RAMAGE D, *et al*. Federated learning of deep networks using model averaging[EB/OL]. https://arxiv.org/abs/1602.05629v1, 2016.

[10] ZHU Ligeng, LIU Zhijian, and HAN Song. Deep leakage from gradients[EB/OL]. https://arxiv.org/abs/1906.08935, 2019.

[11] MA Chuan, LI Jun, DING Ming, *et al*. On safeguarding privacy and security in the framework of federated learning[EB/OL]. https://arxiv.org/abs/1909.06512, 2019.

[12] ZHOU Chunyi, FU Anmin, YU Shui, *et al*. Privacy-preserving federated learning in fog computing[J]. *IEEE Internet of Things Journal*, 2020, 7(11): 10782–10793. doi: 10.1109/JIOT.2020.2987958.

[13] PHONG L T, AONO Y, HAYASHI T, *et al*. Privacy-preserving deep learning via additively homomorphic encryption[J]. *IEEE Transactions on Information Forensics and Security*, 2018, 13(5): 1333–1345. doi: 10.1109/TIFS.2017.2787987.

[14] ZHANG Xianglong, FU Anmin, WANG Huaqun, *et al*. A privacy-preserving and verifiable federated learning scheme[C]. 2020 IEEE International Conference on Communications, Dublin, Ireland, 2020: 1–6. doi: 10.1109/ICC40277.2020.9148628.

[15] LOHANA A, RUPANI A, RAI S, *et al*. Efficient privacy-aware federated learning by elimination of downstream redundancy[J]. *IEEE Design & Test*, 2022, 39(3): 73–81. doi: 10.1109/MDAT.2021.3063373.

[16] MENG Dan, LI Hongyu, ZHU Fan, *et al*. FedMONN: Meta operation neural network for secure federated aggregation[C]. 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Yanuca Island, Fiji, 2020: 579–584. doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00073.

[17] 董业, 侯炜, 陈小军, 等. 基于秘密分享和梯度选择的高效安全联邦学习[J]. 计算机研究与发展, 2020, 57(10): 2241–2250. doi: 10.7544/issn1000-1239.2020.20200463.
DONG Ye, HOU Wei, CHEN Xiaojun, *et al*. Efficient and secure federated learning based on secret sharing and gradients selection[J]. *Journal of Computer Research and Development*, 2020, 57(10): 2241–2250. doi: 10.7544/issn1000-1239.2020.20200463.

[18] FANG Minghong, CAO Xiaoyu, JIA Jinyuan, *et al*. Local model poisoning attacks to Byzantine-Robust federated

learning[EB/OL]. https://arxiv.org/abs/1911.11815, 2021.

[19] 夏家骏, 鲁颖, 张子扬, 等. 基于秘密共享与同态加密的纵向联邦学习方案研究[J]. 信息通信技术与政策, 2021, 47(6): 19–26. doi: 10.12267/j.issn.2096-5931.2021.06.003.
XIA Jiajun, LU Ying, ZHANG Ziyang, *et al*. Research on vertical federated learning based on secret sharing and homomorphic encryption[J]. *Information and Communications Technology and Policy*, 2021, 47(6): 19–26. doi: 10.12267/j.issn.2096-5931.2021.06.003.

[20] HAO Meng, LI Hongwei, XU Guowen, *et al*. Towards efficient and privacy-preserving federated deep learning[C]. 2019 IEEE International Conference on Communications, Shanghai, China, 2019: 1–6. doi: 10.1109/ICC.2019.8761267.

[21] XIANG Liyao, YANG Jingbo, and LI Baochun. Differentially-private deep learning from an optimization perspective[C]. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, Paris, France, 2019: 559–567. doi: 10.1109/INFOCOM.2019.8737494.

[22] BERNSTEIN J, ZHAO J W, AZIZZADENESHELI K, *et al*. SignSGD with majority vote is communication efficient and fault tolerant[C]. The 7th International Conference on Learning Representations, New Orleans, USA, 2019: 1–20.

林 莉：女，博士，副教授，主要研究方向为云计算与边缘计算安全，隐私保护和人工智能安全.
张笑盈：男，硕士生，研究方向为隐私保护和联邦学习.
沈 薇：女，硕士生，研究方向为联邦学习安全与应用.
王万祥：男，硕士生，研究方向为联邦学习安全.

责任编辑：马秀强