基于同态加密的可验证隐私保护联邦学习方案

郭 显* 王典冬 冯 涛 成玉丹 蒋泳波
(兰州理工大学计算机与通信学院 兰州 中国 730050)

摘 要：现有基于同态加密的联邦学习安全和隐私保护方案中，仍面临着服务器伪造聚合结果或与用户合谋导致隐私数据泄露风险。针对上述问题，该文提出抗合谋的隐私保护和可验证联邦学习方案。首先，通过结合秘密共享算法实现密钥的生成和协作解密，并采用同态加密等密码学原语进一步保护模型，防止用户与服务器的合谋攻击。然后基于双线性聚合签名算法使每个用户能够独立验证服务器提供的聚合结果。同时，为了鼓励更多拥有高质量数据的用户参与进来，该文提出一种激励机制，为用户提供相应的奖励。安全性分析表明，该文方案对系统中存在的合谋攻击具有鲁棒性。最后，理论分析和实验验证结果表明该方案具有可靠性、可行性和有效性。

关键词：联邦学习；同态加密；隐私保护；可验证
中图分类号：TN918;TP181 **文献标识码**：A **文章编号**：1009-5896(2025)04-1113-13
DOI: 10.11999/JEIT240390

1 引言

激活数据要素价值，数据要素市场化是赋能新质生产力发展的关键，大规模数据已成为科技和商业发展的驱动力。然而，数据的隐私和安全问题也日益突出。为了做到数据的“可用不可见”，Google在2016年提出了联邦学习框架[1]，该框架允许多个用户之间协作去执行模型训练任务，而数据不会离开本地设备，以此来保护其隐私。然而，由于用户需要将本地梯度上传至服务器进行聚合，不可避免的将引起一些安全问题[2–6]，例如：合谋攻击[7]、模型逆向攻击[8]、服务器伪造聚合结果，这会导致用户本地数据的泄露，因此保护梯度参数显得尤为重要。

针对隐私泄露问题，大量基于密码学技术的隐私保护方案被提出。主要包括差分隐私、安全多方计算以及同态加密等方法。在差分隐私方面，研究人员通过在模型更新中引入噪声来防止攻击者重构用户敏感数据，有效避免用户敏感信息被推断或泄露。例如Wang等人[9]通过变分自动编码器重建了医疗数据，并向其添加了差分隐私噪声以抵抗推理攻击。但该方案容易产生大的隐私预算和过度的噪声扰动。Wei等人[10]提出的方法是对更新后用户模型进行加噪。Sun等人[11]提出使用局部差分隐私机制来处理深度神经网络中的局部权重，以适应不同层的权重范围。此外，为了防止过度的噪声扰动影响模型的准确度，Han等人[12]通过分析梯度更新值，权重参数值以及迭代过程中全局模型和局部模型的关系，来计算出每个模型的重要性系数，并根据重要性系数的大小将噪声添加到模型的参数中。Guo等人[13]则提出了一种动态的方法，根据迭代次数来调整隐私预算以动态地适应梯度下降联邦学习。Stevens等人[14]还将高斯噪声的添加与基于错误学习的掩码协议相结合，有效地降低了通信的复杂度。虽然使用差分隐私不会导致通信开销的增大且不考虑密钥问题，但仍面临着安全性和准确率之间权衡的挑战。安全多方技术也可用于安全的聚合用户本地模型更新而不泄露隐私数据。Bonawitz等人[15]提出了联邦学习的安全聚合，他们使用了Shamir秘密共享去解决用户掉线的问题，并采用对称加密和双重掩码保证了本地模型的数据安全，但他们聚合梯度所需的通信代价较大，给用户带来了沉重的负担。在Zheng等人[16]的系统设计中，通过轻量级、安全和弹性的聚合来实现联邦学习服务，并且可容忍用户掉线，同时支持模型压缩以提高通信效率，但该方案依赖硬件辅助的可信执行环境进行验证，这将导致额外的成本，难以广泛应用。鉴于模型准确性和通信开销的考虑，凭借在保障隐私数据安全的同时提供密文计算的优势，同态加密被广泛应用于隐私保护联邦学习方案中。Wibawa等人[17]提出了一种基于同态加密的医疗数据隐私保护联邦学习算法来保护深度学习模型免受对手的攻击。Wang等人[18]提出的机制能判断用户是否可信以及用户中途掉线问题，此外还集成了同态加密。但同态加密会导致大量的通信开销。为了降低通信开销， Zhang等人[19]将中国剩余定理(Chinese Re-

收稿日期：2024-05-17；改回日期：2025-03-24；网络出版：2025-03-31
*通信作者： 郭显 iamxg@163.com
基金项目：国家自然科学基金(61461027)，甘肃省自然科学基金(20JR5RA467)
Foundation Items: The National Natural Science Foundation of China (61461027), The Natural Science Foundation of Gansu Province(20JR5RA467)

mainder Theorem, CRT)和Paillier同态加密相结合去处理上传的梯度，并使用双线性聚合签名验证了服务器聚合结果的正确性。余晟兴等人[20]采用梯度选择方法对模型进行筛选，减少了需要上传的梯度数量。但上述所提出的方案中用于加解密和签名的密钥均需要可信第三方机构(Third Party Administrator, TPA)生成，并且所有用户往往使用一对相同的密钥，无法抵抗用户与服务器的合谋攻击。Ma等人[21]提出通过设置一个聚合的公钥，来对模型更新进行加密，对于聚合结果的解密则需要所有参与方协作进行。但该方案在加密时引入了误差，导致精度有所损失。

上述方案均假定由一个可信任的服务器诚实地执行联邦学习聚合操作。然而，在商业竞争和利益冲突中，聚合服务器可能恶意发送错误聚合结果。针对错误聚合结果问题，Ma等人[22]首次在隐私保护深度学习中引入了可验证性这一概念，通过利用ElGamal加密和聚合签名来构建其方案。然而，该方案中服务器和用户之间存在合谋的潜在风险且所有参与者都必须参加验证过程。当参与者数量较多时，验证机制导致成本较高。Xu等人[23]提出了支持可验证性的安全聚合协议VerifyNet，该协议将同态哈希函数与伪随机技术相结合来验证聚合结果。然而，随着用户数量的增长，用户的通信和计算成本将会显着增加。Shen等人[24]利用双线性聚合签名和可验证的秘密共享创建了一种验证用户数据完整性和身份的方法。可以有效消除部分用户的一些错误数据。但该方案的实现依赖于外部的高响应和低延迟雾节点来批量管理移动用户。

然而，现有基于同态加密的方案虽然能极大保护用户的隐私数据，但普遍存在通信开销较大、依赖TPA生成密钥、无法抵御用户与服务器之间的合谋攻击等问题。此外，随着用户数量增加，现有可验证方案的验证时间和计算成本将显著增长，严重影响系统效率。因此，设计一种高效可验证、抗合谋攻击且不依赖于TPA的隐私保护聚合方案成为亟待解决的问题。

本文针对现有方案的不足，提出了一种创新的隐私增强联邦学习聚合方案。本方案的主要贡献如下：

(1) 通过结合分布式密钥生成协议，交互生成多个密钥，使用户可使用自身私钥加密，解密需多个用户协作完成，摆脱对TPA的依赖，避免服务器与少于n–1个用户合谋且可容忍掉线。同时，通过对上传模型进行随机化处理以及 CRT降维处理后加密上传，在增强安全保护的同时还有效降低了通信开销。

(2) 针对服务器聚合结果验证问题，基于双线性聚合签名技术，设计了一种用户可独立验证聚合结果正确性的高效验证方案。

(3) 为了吸引更多拥有高质量数据的用户释放数据，本文设计了一种激励机制，根据训练数据的质量、数据丰富度等来计算出奖励，并在任务结束后给参与用户发放相应的奖励。

2 相关知识介绍

2.1 联邦学习

联邦学习是分布式机器学习，它能够使多个设备之间在不提供原数据的情况下协作进行模型的训练。各个参与设备首先从服务器上下载全局模型更新，使用本地数据和全局模型来对其模型进一步训练，然后将更新后的本地模型上传到服务器。服务器在接收到各个参与设备的参数后，进行参数更新以便于再次共享。

随机梯度下降(Stochastic Gradient Descent, SGD)可以简单的应用于分布式学习，但需要多次通信，为了解决这个问题McMahan等人[1]提出了联邦平均算法(Federated Averaging, FedAvg)，旨在平均不同设备上传的模型参数更新来优化全局模型。如式(1)所示，参与设备本地模型的更新为

$$\boldsymbol{W}_{t+1}^{k}=\boldsymbol{W}_{t}-\eta\nabla L_{k}(\boldsymbol{W}_{t}) \tag{1}$$

将本地模型$\boldsymbol{W}_{t+1}^{k}$上传到服务器后，服务器通过式(2)计算全局模型参数

$$\boldsymbol{W}_{t+1}=\sum_{k=1}^{K}\frac{M_{k}}{M}\boldsymbol{W}_{t+1}^{k} \tag{2}$$

其中，η为学习率，$L_k(\boldsymbol{W}_t)$表示第k个用户的损失函数，$\boldsymbol{W}_t$为第t轮的全局模型，M_k表示第k个用户数据总数，M表示所有用户数据总数。

2.2 分布式密钥生成协议

分布式密钥生成(Distributed Key Generation, DKG)协议[25]允许多个用户共同合作以生成1个密码系统的公钥和私钥。可验证秘密分享协议(Verifiable Secret Sharing, VSS)是DKG中重要的理论基础。

DKG协议通常包括3个阶段：分享阶段、验证阶段和密钥生成阶段。

(1) 分享阶段：每个用户随机选择秘密值s_i，并生成一个阈值为t的多项式：$f_i(x)=c_{i0}+c_{i1}x+c_{i2}x^2+\cdots+c_{it}x^t$，$c_{i0}=s_i$。接着计算分享给用户$j$的份额$f_i(j)$，$j=1,2,\cdots,n$，其中$n$表示用户的总数。并根据多项式计算出承诺值$C_{i0}=g^{c_{i0}}$，$C_{i1}=g^{c_{i1}}$，$\cdots$，$C_{it}=g^{c_{it}}$。

(2) 验证阶段：用户j在收到用户i分享的份额后，根据承诺值验证份额的正确性：$g^{f_i(j)}=\prod_{x=0}^{t}C_{ix}$。所有验证通过的用户组成集合$S$。

(3) 密钥生成阶段：通过每个用户$i\in S$公开的承诺值可计算出公钥$\mathrm{PK}=\prod_{i\in S}C_{i0}=\prod_{i\in S}g^{s_i}$。而私钥的值等于所有验证通过的用户的秘密值之和。为了保障秘密值不被公开，将通过选取$t+1$个用户，并使用拉格朗日插值法来间接计算出SK

$$\mathrm{SK}=\sum_{i\in R}\left(\sum_{i\in S}f_i(j)\prod_{i\in R,i\neq j}\frac{i}{i-j}\right) \tag{3}$$

其中$R\subseteq S$表示一组$t+1$个用户的集合。

2.3 双线性聚合签名

设g_1和g_2是乘法循环群G_1和G_2的生成元，并且存在映射关系$e:G_1\times G_2\to G_3$，对于消息m，有哈希函数$h:h(m)\in G_2$。算法共由5个部分构成：

(1) 密钥生成：用户随机选择私钥sk并计算公钥$\mathrm{pk}=g_1^{\mathrm{sk}}$。

(2) 签名：输入私钥sk和消息m，输出消息m的签名$\sigma:h(m)\to H$，$H^{\mathrm{sk}}\to\sigma\in G_2$。

(3) 验证：输入公钥pk，消息m和签名σ，然后计算$h(m)\to H$，判断$e(g_1,\sigma)=(\mathrm{pk},H)$等式是否成立。

(4) 聚合签名：设有多个用户$K_1,K_2,\cdots,K_n$，并对各自消息m_i签名生成σ_i，接着输出聚合签名$\sigma=\prod_{i=1}^{n}\sigma_i$。

(5) 验证聚合签名：输入聚合签名σ，以及$H_i=h(m_i)(i=1,2,\cdots,n)$，假设$K_i$的私钥是$\mathrm{sk}_i$，公钥为$\mathrm{pk}_i$，则判断等式$e(g_1,\sigma)=\prod_{i=1}^{n}e(\mathrm{pk}_i,H_i)$是否成立。

2.4 同态哈希

同态哈希函数[26]是一类具备同态性质的哈希函数，它能够将任意长度的数据值映射成固定长度的哈希值。同态哈希函数除了哈希函数具备的唯一性、单向性和抗碰撞性外，还具有同态性，对任意2个数据m_1, m_2，存在等式：$H(m_1+m_2)=H(m_1)\cdot H(m_2)$。

2.5 改进的ElGamal加密算法

ElGamal算法是Tather ElGamal在1985年提出的基于Diffie-Hellman密码交换协议的非对称加密算法[27]，它只适合乘法同态的性质。然而，在联邦学习系统中，通常需要实现对本地模型的加法聚合。文献[28]对ElGamal算法进行稍加修改以实现加法聚合，该算法由4个步骤组成。

(1) 初始化：由可信第三方根据安全参数k生成公共参数(q,g,G)，其中G是具有大素数q和生成元g的循环群。

(2) 密钥生成：用户随机选择一个数$u\in Z_q^*$作为私钥，计算$y=g^u\in G$作为公钥，其中公钥被用于加密，私钥被用于解密。

(3) 加密：对于加密的消息m，用户选择一个随机数$r\in Z_q^*$，并且计算$c_1=g^r$,$c_2=2^m y^r$，然后发送密文(c_1,c_2)给接收者。

(4) 解密：接收者在接收到密文之后，通过私钥u计算出明文

$$m=\log_2 2^m=\log_2(c_2/c_1^u)=\log_2(2^m y^r/(g^r)^u) \tag{4}$$

3 系统模型

该系统由服务器、用户、辅助节点和奖励分发机构4类角色构成。其架构如图1所示。服务器负责

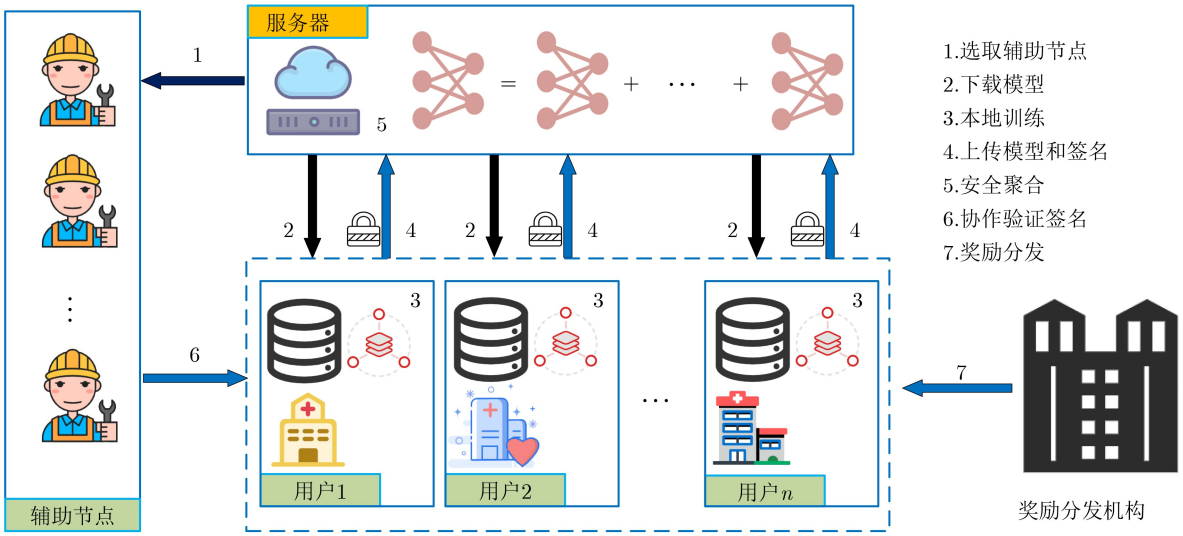


图 1 系统模型图

收集用户提交的签名和随机化后的本地模型，并执行一系列操作来完成模型的安全聚合。用户则负责在本地数据集上训练本地模型，并将随机化后的本地模型、签名及相关安全参数提交给服务器。最终，通过服务器和用户之间的相互合作，计算出全局模型。与传统的联邦学习框架不同，本文引入了辅助节点和奖励分发机构两类角色，旨在进一步增强本地模型的安全性，并吸引更多拥有高质量数据的用户参与。辅助节点由服务器选取负责每轮协助用户验证聚合结果的正确性，而奖励分发机构负责在训练结束后奖励的分发。

3.1 威胁模型

假设系统中的用户、辅助节点和奖励分发机构都正确遵守协议执行，但用户试图通过窃听信道中传输的信息来推断隐私信息，同时还可能会与服务器或辅助节点合谋，以满足他们对隐私数据的好奇心。此外，服务器还可能会恶意篡改最后的全局模型数据。本文旨在消除外部敌手、合谋攻击和服务器篡改聚合结果的问题。然而，模型训练中故意的破坏行为，如数据投毒、模型投毒等将暂时不考虑。

4 方案设计

本节将详细介绍本方案的方法架构、工作流程以及每个流程中运用到的核心算法。

4.1 工作原理

本文提出的方案包含6个主要阶段：初始化阶段、辅助节点选取阶段、签名和随机化模型阶段、聚合阶段、验证阶段以及激励机制阶段。若用户在前5个阶段中掉线，则其在该轮将不会得到任何奖励。

(1) 初始化阶段：所有用户在首次加入系统时，首先，基于分布式密钥生成协议生成密钥对。其次，服务器为用户生成所需参数。用户在收到参数后，并开始在本地进行模型训练。

(2) 辅助节点选取阶段：由服务器选取辅助节点。

(3) 签名和随机化模型阶段：用户在将本地模型上传至服务器之前，会利用辅助节点和其他用户的公钥以及自身的私钥，重新计算出新的私钥，并对本地模型签名。随后，用户利用公共矩阵和选取的随机向量，生成一个与模型维度匹配的随机模型，并将其叠加至本地模型之上。

(4) 聚合阶段：各个用户将随机化模型、模型签名和加密后的随机向量上传至服务器，服务器聚合模型签名得到聚合签名。同时聚合随机化模型和加密后的随机向量，并与用户协作解密出聚合结果。服务器将聚合结果和聚合签名下发给各个用户。

(5) 验证阶段：用户在收到聚合结果和聚合签名之后，通过辅助节点协作生成的验证密钥独立验证聚合结果的正确性。

(6) 激励机制阶段：在联邦学习结束后，根据用户在训练过程中的表现来发放奖励。

4.2 方案详细描述

(1) 初始化阶段：在系统初始化阶段，服务器确定一系列必要的参数，其中包括生成元g、初始化的全局模型$\boldsymbol{W}_{\text{global}}$、秘密分享的阈值$t$、双线性聚合签名中的同态哈希函数$h$及其生成元$g_1$，以及加密过程中使用的哈希函数$H$和生成元$g_2$。为了保障系统的稳健性，还需设定训练过程中允许退出和恶意共谋的用户的最大数量。此外，服务器和用户P_i各自生成密钥对$<\text{sk}_{\text{cs}},\text{pk}_{\text{cs}}>$，$<\text{sk}_i,\text{pk}_i>$，其中$\text{pk}_i=g^{\text{sk}_i}$, $\text{pk}_{\text{cs}}=g^{\text{sk}_{\text{cs}}}$，并将公钥广播至所有用户。如式(5)所示，计算会话密钥$k_{i,j}$

$$k_{i,j}=\text{pk}_i{}^{\text{sk}_j}=\text{pk}_j{}^{\text{sk}_i}=g^{\text{sk}_i\text{sk}_j} \tag{5}$$

接着，各用户基于分布式密钥生成协议，计算用于后续的隐私保护聚合过程的主私钥和主公钥。具体操作如下：

步骤1 份额和承诺值计算。P_i选取一个秘密值s_i，并将其分为n份，然后生成一个t次多项式$f_i(x)$

$$f_i(x)=c_{i0}+c_{i1}x+c_{i2}x^2+\cdots+c_{it}x^t \tag{6}$$

其中$c_{i0}=s_i$，分享的份额$s_{i\to j}=f_i(x_j)$、承诺值$C_{i0}=g^{c_{i0}}$, $C_{i1}=g^{c_{i1}}$,$\cdots$, $C_{it}=g^{c_{it}}$，根据承诺值计算的公共多项式为：$F_i(x)=C_{i0}\cdot C_{i1}^{x}\cdot C_{i2}^{x^2}\cdots C_{it}^{x^t}$。

步骤2 份额的发送及验证。P_i通过P_j公钥将份额$s_{i\to j}$加密并随着承诺值一同发送给P_j。P_j收到加密后的份额后通过私钥解密，并使用接收到的承诺值去验证份额$s_{i\to j}$的正确性，当且仅当$g^{s_{i\to j}}=F_i(j)$，则认为接收到的份额是有效的。将所有验证通过的用户记为集合S_1。

步骤3 群私钥和群公钥计算。P_j根据接收到的份额$s_{i\to j}$计算自己的群私钥和群公钥$<\text{gsk}_j,\text{gpk}_j>$

$$\text{gsk}_j=\sum_{p_j\in S_1}s_{i\to j},\text{gpk}_j=g_2{}^{\text{gsk}_j} \tag{7}$$

步骤4 主公钥和主私钥计算。通过每个用户收到的秘密份额，可生成整个系统的主公钥和主私钥$<\text{mpk},\text{msk}>$

$$\text{mpk}=\prod_{P_i\in S_1}C_{i0}=\prod_{P_i\in S_1}g_2{}^{s_i},\text{msk}=\sum_{P_i\in S_1}s_i \tag{8}$$

但为了保障用户选取的秘密值s_i不被公开，可采用拉格朗日插值法计算出主私钥msk，其中$R\subseteq S_1$表示一组$t+1$个用户的集合

$$\mathrm{msk}=\sum_{j\in R}\mathrm{gsk}_j\prod_{\upsilon\in R,j\neq\upsilon}\frac{\upsilon}{\upsilon-j} \tag{9}$$

(2) 辅助节点选取阶段：服务器向空闲节点发送协作任务请求，并基于响应节点的性能评估结果，采用动态加权算法选取特定数量的高可信节点作为辅助节点aui。将被选取成为辅助节点集群记为S_{au}。

(3) 签名和随机化模型阶段：集合S_1中的用户在将本地模型$\boldsymbol{W}_i$提交到服务器之前，为了防止本地模型被外部敌手、服务器和用户相互勾结的其他用户窥探，需要对本地模型随机化以达到隐私保护的目的。并根据式(10)计算随机化模型

$$\boldsymbol{V}_i=\boldsymbol{W}_i+\sum_{j\in S_1,j\neq i}(-1)^{i>j}\mathrm{CK}_{i,j}+\varphi\boldsymbol{B}_i(\mathrm{mod}\ R) \tag{10}$$

其中，φ是预先设置好的公共比例因子，$\mathrm{CK}_{i,j}=H(k_{i,j})$，其中$H$为哈希操作，若$i>j$则$(-1)^{i>j}=-1$否则为1。此外$\boldsymbol{B}_i$是与本地模型$\boldsymbol{W}_i$维度匹配的随机整数模型。服务器在聚合了随机化模型之后需要消掉$\boldsymbol{B}_i$来恢复$\boldsymbol{W}_i$，则需要通信$\boldsymbol{B}_i$。为了降低传输过程中的通信开销，假设$\boldsymbol{B}_i$的维度为$d_1\times 1$，将其划分为：$\boldsymbol{B}_i=\boldsymbol{A}_{d_1\times d_2}\times\boldsymbol{Y}_i$，其中维度为$d_1\times d_2$的矩阵$\boldsymbol{A}$为所有用户和服务器的公共矩阵，$\boldsymbol{Y}_i$的维度为$d_2\times 1$，且$d_2<d_1$, $d_2>>|S_2|$。由于$\boldsymbol{B}_i$的维度大于$\boldsymbol{Y}_i$，因此传输$\boldsymbol{Y}_i$通信开销更小。如图2所示将$\boldsymbol{Y}_i$通过CRT处理后得到$\overline{\boldsymbol{Y}_i}$。在处理过程中，用户$P_i$将$\boldsymbol{Y}_i$除$k$划分为$\mu$个相等的块，若$\boldsymbol{Y}_i$不能被$k$整除将使用0填充。

用户P_i在完成模型的随机化之后，利用改进的ElGamal同态加密算法和系统主公钥${g_2}^{\mathrm{msk}}$对向量$\overline{\boldsymbol{Y}_i}$进行加密。每个用户$P_i$分别选取一个随机数$r_i\in Z_q^*$，并根据式(11)去计算密文$(c_{i1},c_{i2})$。同时用户$P_i$对$\boldsymbol{W}_i$的签名为：$\sigma_i={g_1}^{K_i}h_i$，其中私钥$K_i=\sum_{j\in S_{\mathrm{au}}}k_{i,j}+\sum_{j\in S_1,i\neq j}(-1)^{i>j}k_{i,j}$，如果$i>j$则$(-1)^{i>j}=-1$否则为1，$h_i=h(\boldsymbol{W}_i)$，其中$h$表示同态哈希操作。然后将元组$(i,\sigma_i,c_{i1},c_{i2})$发送给服务器。其中$k_{i,j}$为初始化时生成的会话密钥

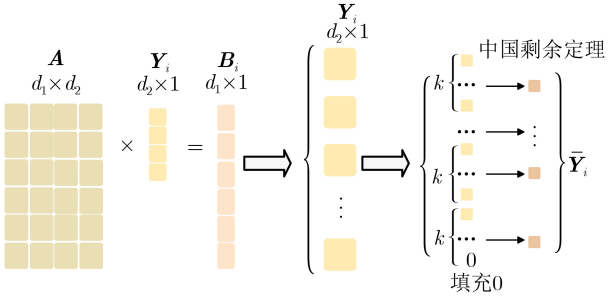


图 2 数据预处理

$$\mathrm{Enc}_{\mathrm{mpk}}(\overline{\boldsymbol{Y}_i})=(c_{i1},c_{i2})=({g_2}^{r_i},2^{\overline{\boldsymbol{Y}_i}}{g_2}^{\mathrm{msk}\times r_i}) \tag{11}$$

(4) 聚合阶段：服务器从用户集合S_1中收集到随机化模型、密文和签名$(\boldsymbol{V}_i,\sigma_i,c_{i1},c_{i2})$之后，开始聚合这些随机化模型。将上传过程中服务器成功收到响应的用户集合记为S_2。若在上传的过程中无用户掉线，即$S_1=S_2$，则服务器直接聚合$\{\boldsymbol{V}_i\}_{i\in S_1}$。否则，服务器找出$S_1$中存在而$S_2$中不存在的用户(记为$S_1\backslash S_2$)，接着服务器将$S_1\backslash S_2$中用户的身份列表广播给所有用户。当用户$P_i$收到该列表后，主动根据式(12)去计算出两个值$\boldsymbol{\beta}_i$和$\gamma_i$并发送给服务器

$$\begin{aligned}\boldsymbol{\beta}_i&=\sum_{b\in S_1\backslash S_3,b\neq i}(-1)^{i>b}\mathbf{CK}_{i,b},\gamma_i\\&=\sum_{b\in S_1\backslash S_2,b\neq i}(-1)^{i>b}k_{i,b}\end{aligned} \tag{12}$$

其中，如果$i>b$则$(-1)^{i>b}=1$否则为−1。随后，服务器将对在线用户集合S_2上传的随机化模型$\{V_i\}_{i\in S_2}$进行聚合操作

$$\begin{aligned}\boldsymbol{\theta}&=\sum_{i\in S_2}\boldsymbol{V}_i+\boldsymbol{\beta}_i=\sum_{i\in S_2}\boldsymbol{W}_i+\sum_{j\in S_1,j\neq i}(-1)^{i>j}\mathbf{CK}_{i,j}\\&\quad+\varphi\sum_{i\in S_2}\boldsymbol{B}_i+\sum_{b\in S_1\backslash S_2,b\neq i}(-1)^{i>b}\mathbf{CK}_{i,b}\\&=\sum_{i\in S_2}\boldsymbol{W}_i+\varphi\boldsymbol{A}\sum_{i\in S_2}\boldsymbol{Y}_i\end{aligned} \tag{13}$$

由式(13)可知，只要知道了用户上传的$\boldsymbol{Y}_i$值，就可以很容易通过$\boldsymbol{\theta}-\varphi\boldsymbol{A}\sum_{i\in S_2}\boldsymbol{Y}_i$计算出聚合结果$\sum_{i\in S_2}\boldsymbol{W}_i$。

随后，服务器对上传密文和签名进行聚合操作，根据式(14)、式(16)，可计算由每个用户$P_i\in S_2$发送的加密模型信息和签名。其中r_i为各个用户选取的随机数，r为r_i的和

$$c_1=\prod_{i\in S_2}c_{i1}=\prod_{i\in S_2}{g_2}^{r_i}={g_2}^{r} \tag{14}$$

$$c_2=\prod_{i\in S_2}c_{i2}=2^{\sum_{i\in S_2}\overline{\boldsymbol{Y}_i}}{g_2}^{\mathrm{msk}\times r} \tag{15}$$

$$\sigma=\prod_{i\in S_2}\sigma_i{g_1}^{\gamma_i} \tag{16}$$

由式(14)、式(15)可知，只要知道了主私钥msk，就可以解密出聚合密文(c_1,c_2)，并得到用户上传的聚合结果$\sum_{i\in S_2}\overline{\boldsymbol{Y}_i}$，然而主私钥不被单个用户和服务器所知道，它需要$t+1$个用户去协助生成，以根据秘密重构算法完成解密。

服务器从集合S_2中随机选取$t+1$个在线的用户记为$S_3(S_3\subseteq S_2)$，如式(17)所示去计算δ_i和$c_1^{\delta_i}$，

其中，x_i是秘密份额$(x_i, f(x_i))$的一部分，在初始阶段时生成。用户P_i计算δ_i并发送给服务器，服务器再计算$c_1^{\delta_i}$并将其发送给对应的用户$P_i \in S_3$

$$\delta_i = \prod_{p_j \in S_3, i \neq j} \frac{-x_j}{x_i - x_j}, c_1^{\delta_i} = {g_2}^{r\delta_i} \tag{17}$$

当用户P_i接收到$c_1^{\delta_i}$后，使用其群私钥gsk_i计算得到λ_i，这个λ_i仅自身知道，如式(18)所示，然后将λ_i发送给服务器

$$\lambda_i = (c_1^{\delta_i})^{\mathrm{gsk}_i} = {g_2}^{\prod\limits_{p_j \in S_3, i \neq j} \frac{-x_j}{x_i - x_j} \times \mathrm{gsk}_i \times r} \tag{18}$$

服务器在收到$\{\lambda_i\}_{i \in S_3}$后，根据式(19)进行聚合

$$\begin{aligned} \prod_{i \in S_3} \lambda_i &= \prod\nolimits_{i \in S_3} {g_2}^{\prod\limits_{p_j \in S_3, i \neq j} \frac{-x_j}{x_i - x_j} \times \mathrm{gsk}_i \times r} \\ &= {g_2}^{[\sum\limits_{i \in S_3} \prod\limits_{p_j \in S_3, i \neq j} \frac{-x_j}{x_i - x_j} \times \mathrm{gsk}_i] \times r} = {g_2}^{\mathrm{msk} \times r} \end{aligned} \tag{19}$$

至此，服务器可以通过式(20)、式(21)计算出参数之和

$$\overline{\boldsymbol{Y}_g} = \sum_{i \in S_2} \overline{\boldsymbol{Y}_i} = \log_2 \frac{c_2}{\prod\limits_{i \in S_3} \lambda_i} \tag{20}$$

$$\boldsymbol{Y}_g = \bar{\boldsymbol{Y}}_g \bmod m_i,\ i = 1, 2, \cdots, k \tag{21}$$

最终，已知$\boldsymbol{\theta}$、比例因子φ和公共矩阵$\boldsymbol{A}$的条件下，服务器可计算出聚合结果$\boldsymbol{W}_g$

$$\boldsymbol{W}_g = \sum_{i \in S_2} \boldsymbol{W}_i = \boldsymbol{\theta} - \varphi \boldsymbol{A} \boldsymbol{Y}_g \tag{22}$$

(5) 验证阶段：用户根据聚合结果$\boldsymbol{W}_g$和式(23)独立判断聚合结果正确性的操作为

$$e(g_1, \sigma) = e\left(g_1, \prod_{i=1}^{S_3} {g_1}^{K_i} h_i {g_1}^{\gamma_i}\right) = e(g_1, {g_1}^{K} h(\boldsymbol{W}_g)) \tag{23}$$

因为 $K = \sum_{i \in S_2} (K_i + \gamma_i) = \sum_{i \in S_{\mathrm{au}}} \left(\sum_{j \in S_2, j \notin S_{\mathrm{au}}} k_{i,j}\right)$，因此辅助节点$\mathrm{aui} \in S_{\mathrm{au}}$计算出自己的$K_{\mathrm{aui}} = \sum_{j \in S_2, j \notin S_{\mathrm{au}}} k_{i,j}$然后将${g_1}^{K_{\mathrm{aui}}}$广播出去。各个用户$P_i$在收到${g_1}^{K_{\mathrm{aui}}}$之后进行聚合${g_1}^{K} = \prod_{i \in S_{\mathrm{au}}} {g_1}^{K_{\mathrm{aui}}}$。在验证完成过后，根据式(1)更新全局梯度，如果不满足终止条件，则用户继续本地训练为下一轮联邦学习做准备。同时设计了激励机制来吸引更多拥有高质量数据的用户加入到该框架中来。

(6) 激励机制阶段：在激励机制中，辅助节点参与协作计算将获得基本奖励$\mathrm{Reward}_{\mathrm{basic}}$，而用户的奖励由两部分组成，即基本奖励$\mathrm{Reward}_{\mathrm{basic}}$和额外奖励$\mathrm{Reward}_{\mathrm{extra}}$。对于额外奖励的计算，将通过以下方法实现。用户$P_i$持有的本地数据集具有如下形式$D_i^l = \{D_i^1, D_i^2, \cdots, D_i^m\}$，服务器的测试数据集表示为$D_{\mathrm{s}}^l = \{D_{\mathrm{s}}^1, D_{\mathrm{s}}^2, \cdots, D_{\mathrm{s}}^m\}$，其中$m$表示数据集中数据类型，$D_i^m, D_{\mathrm{s}}^m$表示不同类型中的数据集。用$D_{\mathrm{same}} = D_i^l \cap D_s^l$表示测试集与用户数据集中相同数据类型的数据集，$|D_{\mathrm{same}}|$表示集合$D_{\mathrm{same}}$中的元素的数量。

当用户本地数据集样本过少时，训练出来的模型容易过拟合，使用这个模型对数据进行预测时，预测的精度将偏低。为减少模型更新的轮数，应尽量避免过多聚合这种精度偏低的模型。本文将用户本地数据与测试集的相关程度和对测试集的预测精度联系起来。对于用户P_i，在联邦学习中上传的本地模型的平均预测精度(表示为Avg_i)可通过式(24)计算得出，其中avg_j表示相同类型数据集的预测准确度

$$\mathrm{Avg}_i = \frac{1}{|D_{\mathrm{same}}|} \sum_{j=1}^{|D_{\mathrm{same}}|} \mathrm{avg}_j \tag{24}$$

此外还考虑了模型每轮损失。模型损失从侧面反映了模型优化的好坏程度，故使用模型损失作为评价训练数据质量的标准，用户P_i在第k轮的数据质量的计算公式如下，其中loss_i^k是用户P_i在第k轮的模型损失，$\mathrm{loss_avg}^k$是第k轮的平均模型损失

$$q_i^k = \frac{\mathrm{loss\_avg}^k}{\mathrm{loss}_i^k} \tag{25}$$

当联邦学习结束时，根据式(26)、式(27)计算得出用户P_i提供的局部模型的评估准确度以及数据质量

$$\mathrm{Acc} = \frac{1}{\mathrm{sucess\_num}} \sum_{i=1}^{\mathrm{sucess\_num}} \mathrm{Avg}_i \tag{26}$$

$$Q_i = \frac{1}{\mathrm{sucess\_num}} \sum_{i=1}^{\mathrm{sucess\_num}} q_i^k \tag{27}$$

其中$\mathrm{sucess_num}$表示用户在FL训练过程期间服务器下发测试集后用户成功上传预测精度和数据质量的次数。接着根据相关类型的数据集定义了用户P_i训练的局部模型的丰富度

$$\mathrm{Richneas} = \frac{|D_{\mathrm{same}}|}{|D_i^l|} \tag{28}$$

其中$|D_i^l|$表示用户P_i的本地数据集中元素的数量。同时还将用户本地数据集和服务器的测试数据集之间相同类型的数据特征差异定义为Similarity，即

$$\mathrm{Similarity}_i = \frac{1}{|D_{\mathrm{same}}|} \sum_{i}^{|D_{\mathrm{same}}|} \left(\mathrm{AF}(D_i^j) - \mathrm{AF}(D_{\mathrm{s}}^j)\right)^2 \tag{29}$$

其中AF$(\cdot)$表示某种数据集类型的平均特征值，因此在联邦学习中，用户的额外奖励可通过式(30)进行计算

$$\text{Reward}_{\text{extra}}=\left(\frac{\text{Richness}\times\text{Acc}+Q_i}{\rho+\text{Similarity}_i}\right)\times\text{Reward}_{\text{basic}} \tag{30}$$

5　安全性分析

在本文所提方案中，所有的用户和辅助节点都正确执行协议，因此他们可以诚实地执行安全的聚合过程。但服务器有可能会去篡改聚合后的结果，还可能会串通其他用户试图去窃取诚实用户的隐私数据。在此基础上，分析了该方案的安全性，包括随机化模型的安全性、同态加密的安全性。

5.1 随机化模型的安全性证明

本方案中，$\boldsymbol{B}_i$是用户每轮选取的随机向量，用于在真实输入$\boldsymbol{W}_i$上叠加一个随机值，进而实现对模型参数的屏蔽。屏蔽后的结果具有随机性，使得服务器观察到的是被随机化处理的模型参数。在聚合模型中，只要来自所有客户端的随机化结果之和等于所有真实的输入之和，攻击者就无法区分真实的输入和随机化后的结果。因为随机化后的结果有效地隐藏了每个用户的真实输入，所以攻击者能观察到的仅是随机化后的输入的总和，而不是任何单独的真实输入。在本文的系统框架中，用户上传给服务器的随机化模型为

$$\boldsymbol{V}_i=\boldsymbol{W}_i+\sum_{j\in S_1,j\neq i}(-1)^{i>j}\mathbf{CK}_{i,j}+\varphi\boldsymbol{B}_i(\text{mod } R) \tag{31}$$

从式(31)可知，只要S_1中存在至少两个诚实的参与方，诚实用户上传的模型更新仍然受到由其他诚实用户的之间建立的相互密钥$\mathbf{CK}_{i,j}$所产生的数值的保护。

下面将重新介绍集合符号$S_3\subseteq S_2\subseteq S_1$和$S_{\text{au}}$所代表的含义。$S_1$表示系统初始阶段通过验证的用户集合，$S_2$表示用户从本地上传模型参数时未掉线的用户集合。$S_3$表示从$S_2$随机选取$t+1$在线用户组成的集合。$S_{\text{au}}$表示为被选择作为辅助节点的集合。为了简化描述，本文用cs表示为服务器，$\mathcal{A}(\mathcal{A}\subseteq S_1)$表示为参与合谋的用户集合，ch表示诚实方，$\text{view}_{\mathcal{A}}^{t}\{\boldsymbol{W}_S,S_1,S_2,S_3\}$是一个随机变量，它代表了在给定Shamir秘密共享算法中的门限t的情况下，合谋方在整个联邦学习过程中能获得的所有视图。

定理1　仅由用户合谋的情况下。假设有一个PPT模拟器Sim试图在$\mathcal{A}$的帮助下打破随机化模式下的语义安全。如果给定$t,S,\boldsymbol{W}_S,S_1,S_2,S_3,S_{\text{au}}$，其中$|S|\geq t$, $\mathcal{A}\subseteq S$, $|\text{ch}|\geq 2$，则由模拟器Sim获取的视图与$\mathcal{A}$的视图不可区分，“$\equiv$”表示不可区分$\text{view}_{\mathcal{A}}^{t}\{\boldsymbol{W}_S,S_1,S_2,S_3,S_{\text{au}}\}\equiv\text{view}_{\text{Sim}}^{t}\{\boldsymbol{W}_S,S_1,S_2,S_3,S_{\text{au}}\}$

证明　在这个威胁模型中，本文只考虑集合S_1中用户的合谋，而服务器被认为是绝对诚实的。在真实和理想的安全仿真模型中，首先构造模拟器Sim，以模拟理想视图，模拟器的构造如下：

(1) 对于每个用户$i\in\text{ch}$，Sim随机选择$\overline{\mathbf{CK}}_{i,j}\leftarrow\mathbb{Z}_R$和$\overline{B_i}\leftarrow\mathbb{Z}_R$，并计算参数

$$\overline{\boldsymbol{V}_i}=\sum_{j\in\text{ch},j\neq i}(-1)^{i>j}\overline{\mathbf{CK}}_{i,j}+\varphi\overline{\boldsymbol{B}}_i(\text{mod } R) \tag{32}$$

(2) 对于合谋用户$\mathcal{A}$，Sim直接使用其提供的真实输入$V_{\mathcal{A}}$。

(3) Sim输出敌手视图$\{\overline{\boldsymbol{V}_i}\}_{i\in\text{ch}}$。

其中，对于诚实用户，其随机项$\sum_{j\in\text{ch},j\neq i}(-1)^{i>j}\mathbf{CK}_{i,j}+\varphi\boldsymbol{B}_i$在模$R$下均匀分布，与Sim生成的$\overline{\boldsymbol{V}_i}$统计不可区分。而根据式(31)，所有用户的随机化参数之和使得敌手无法分解出单个$\boldsymbol{W}_i$。此外，由于$|\text{ch}|\geq 2$，敌手无法通过合谋获取足够的$\mathbf{CK}_{i,j}$或随机向量$\boldsymbol{B}_i$来解构非合谋用户的$\boldsymbol{W}_i$。因此，定理1成立，协议在存在至少两个诚实用户时满足语义安全性。

定理2　用户与服务器的合谋的情况下。假设有一个PPT模拟器Sim试图在$\mathcal{A}$的帮助下打破我们随机化模式下的语义安全。如果给定$t,S,\boldsymbol{W}_S,S_1,S_2,S_3,S_{\text{au}}$，其中$|S|\geq t$, $\mathcal{A}\subseteq S\cup\{\text{cs}\}$, $|\text{ch}|\geq 2$，则由模拟器Sim获取的视图与$\mathcal{A}$的视图不可区分$\text{view}_{\mathcal{A}}^{t}\{\boldsymbol{W}_S,S_1,S_2,S_3,S_{\text{au}}\}\equiv\text{view}_{\text{Sim}}^{t}\{\boldsymbol{W}_S,I,S_1,S_2,S_3,S_{\text{au}}\}$，$\boldsymbol{I}=\sum_{i\in S_2\backslash\mathcal{A}}\boldsymbol{W}_i$。

给定合谋方$\mathcal{A}$的真实输入和所有诚实在线的用户$S_2\backslash\mathcal{A}$的真实输入之和，那么，除了给定视图和$\mathcal{A}$的视图之外，模拟器Sim仍然不能学习任何单个诚实客户端的真实输入。

证明 (1)首先，模拟器Sim利用$\mathcal{A}$得到与$\mathcal{A}$相同的视图。

(2)其次，如果$|\mathcal{A}|\geq t$，合谋方能解密出主公钥mpk则模拟器Sim能获得诚实用户的单个$\boldsymbol{B}_i$，然后模拟每个诚实用户$P_i\in S_2\backslash\mathcal{A}$，并用随机选择的数$\mathbf{CK}'_{i,j}$去代替$\mathbf{CK}_{i,j}$，根据DDH假设，在私钥未知的情况下，$\mathbf{CK}'_{i,j}$和$\mathbf{CK}_{i,j}$是不可区分的，故模拟器Sim的视图不比合谋方$\mathcal{A}$多。

5.2 模型签名过程中的安全性证明

定理3　辅助节点与用户合谋的情况下，诚实用户的签名仍然受到保护。

证明　在本方案中，用户P_i签名消息的私钥

$$K_i = \sum_{j\in S_{\text{au}}} k_{i,j} + \sum_{j\in S_1, j\neq i} (-1)^{i>j} k_{i,j} \tag{33}$$

其中$\sum_{j\in S_1,i\neq j}(-1)^{i>j}k_{i,j}$可以被划分为两个部分，即

$$\sum_{j\in \text{ch},i\neq j} (-1)^{i>j} k_{i,j} + \sum_{j\in \mathcal{A},i\neq j} (-1)^{i>j} k_{i,j} \tag{34}$$

如果集合$\mathcal{A}$中的所有用户与辅助节点中的所有用户合谋，合谋方将能够得到

$$\sum_{j\in S_{\text{au}}} k_{i,j} + \sum_{j\in \mathcal{A},i\neq j} (-1)^{i>j} k_{i,j} \tag{35}$$

但诚实用户对消息签名的私钥仍与其他诚实用户之间建立的相互密钥$\sum_{j\in \text{ch},i\neq j}(-1)^{i>j}k_{i,j}$所保护。故好奇的用户仍然不能获得诚实用户签名的私钥。

5.3 模型上传过程中的安全性证明

在上节安全性的分析已经证明了随机化机制可以保护用户的真实输入$\boldsymbol{W}_i$。在本节中，如果给定真实的输入$\boldsymbol{W}_i$，讨论服务器和合谋的用户集合能否通过打破同态加密的语义安全来解密出诚实用户的单个模型。

在给定加密密文$(g^{r_i}, 2^{\overline{\boldsymbol{Y}_i}} g^{\text{msk}\times r_i})$，攻击者恢复用户$P_i$上传的向量$\overline{\boldsymbol{Y}_i}$需解决以下计算难题：

(1) Shamir方案的安全性：确保攻击者无法通过少于t+1份的密钥片段重构msk。

(2) 离散对数问题：攻击者在给定g^r的情况下，求解r是困难的，即

$$\forall \text{PPT } \mathcal{A}, \Pr[\mathcal{A}(g, g^r) = r] \leq \text{negl}(\lambda) \tag{36}$$

其中，λ为安全参数，$\text{negl}(\lambda)$表示可忽略函数。

(3)计算Diffie-Hellman问题：在给定$(g^{r_i}, g^{\text{msk}})$的情况下，计算$g^{\text{msk}\times r_i}$也是计算困难的，即

$$\forall \text{PPT } \mathcal{A}, \Pr[\mathcal{A}(g, g^r, g^{\text{msk}}) = g^{\text{msk}\cdot r}] \leq \text{negl}(\lambda) \tag{37}$$

这意味着攻击者无法直接恢复$\overline{\boldsymbol{Y}_i}$。

由于$\mathcal{A} \leq t$，攻击者无法重构msk，因此无法直接计算密文中的$2^{\overline{\boldsymbol{Y}_i}} g^{\text{msk}\times r_i}$。此外，若攻击者试图从密文中恢复$\overline{Y_i}$，则需要求解$g^{\text{msk}\times r_i}$，该计算等价于求解计算Diffie-Hellman问题，与计算困难假设矛盾。因此，攻击者无法在可行时间内恢复$\overline{\boldsymbol{Y}_i}$。此外，由定理1可得出，即使敌手通过$\mathcal{A} \geq t+1$重构msk，但只要存在至少2个诚实用户，敌手无法计算得到用户的单个模型。

6 实验分析

6.1 数据集

为了对建议的方案进行有效评估，本文使用的数据集是Mnist数据集，该数据集包含了60 000张训练图像和10 000张测试图像。这些图像都是黑白的，大小为28×28像素，每张图像都代表了0～9的一个数字，并且每个数字都有对应的标签。然后，为了更好地模拟高质量数据和低质量数据，本文将训练集重新划分为多个不同的子集，每个用户拥有的数据都是独立的，不可泄漏的，这保证了图像的安全性和隐私性。此外，对于每个用户持有的训练集，还将其划分为两个部分：90%用作训练模型，10%用于充当评估模型的验证集。并使用CNN卷积神经网络对模型进行训练。

6.2 方案比较

本文比较了5种方案，表1列出了是否依赖第三方的支持，是否使用同态加密以及是否支持独立可验证的功能比较。

6.3 对比结果与分析

在本节中，将通过性能和评估结果去分析本文所提出方案的有效性和效率。使用python语言实现了联邦学习框架中局部模型的训练以及梯度的更新。在处理器为Intel i7-12700H(2.3 GHz)CPU、运行内存为16 GB RAM 和操作系统为Windows的个人笔记本电脑上运行联邦学习，在同一服务器上运行多个用户程序，用户将自己的程序封装以供服务器调用。服务器负责参数的更新，聚合和分发。为了保证5种方案对比的公平性，实验均采用相同的软硬件环境、相同的数据集、预处理方法和分发方法。初始化阶段的密钥生成时间如表2所示。

(1)精确性：在每个训练阶段，服务器与所有用户都进行一次通信，并将Batch Size使用bs简化表示，bs的取值范围为(10, 50, 100, 150)，学习率设置为0.001。用户的数量为10个，如图3所示，通过对比实验可以看出，当bs取不同的值时，模型的

表 1 5种方案的功能比较

方案	不依赖于TPA	同态加密	独立可验证
文献[1]	√	×	×
文献[18]	×	√	×
文献[21]	√	√	×
文献[22]	×	√	√
本文	√	√	√

表 2 初始化阶段生成时间

用户数量	时间(s)
10	0.008
30	0.230
50	1.060
100	8.000

准确率以及收敛速度均不相同。当bs的设置为100时，获得了最好的收敛速度和模型准确度。因此，在接下来的对比实验中将bs的值设置为100。然后对5种方案进行了比较，如图4所示，本方案在测试集中获得了较好的精度，同时因为高质量数据的加入，使得本方案的收敛速度快于其他方案。文献[21]在加密时对明文添加了误差所以导致最后的模型精度有所下降。

(2) 通信时间：对这5种方案的训练时间进行了比较。如图5所示，随着通信轮次的增多，训练时间也越长。利用这个对比实验说明了同态加密会带来额外的时间开销，但它可以保证用户上传参数的安全性。本方案将模型参数拆分后再使用CRT进一步处理，使其传输更小的梯度参数，接着使用同态加密去保证参数的安全性。此外，还使用了双线性聚合签名来验证聚合结果的正确性，故本方案对比文献[1]和文献[21]增加了通信时间是合理的。

(3) 验证时间：通过去改变用户在联邦学习中退出的比例来分析用户验证聚合结果的时间。图6为随着用户掉线率的增加用户验证聚合结果正确性的时间花销。对于每一个中途退出的用户，服务器必须重新计算验证所需要的参数，被指定为辅助节点的用户也必须重新计算验证时所需要的密钥。从图6中可以清楚的看出，用户的验证时间不会随着用户数量的增加而增加。即使在30个用户的情况下，并且退出率达到了30%，验证时间也只比没有用户退出的情况多增加了1%。其次，通过图7表明，随着用户数量的增加，文献[22]的验证开销呈线性增长，而本文提出方案验证时间基本保持不变。

(4) 激励机制的评估结果如图8所示，实验中，将任务参与者获得的基本奖励值设为100，ρ的值设为0.5。在图8(a)中，数据丰富度Richness和相似度Similarity分别固定在0.5和0.07，随着数据质量的提高，任务参与者获得的总奖励会越来越高。在图8(b)中，数据质量和相似度Similarity分别固定在1和0.07，随着数据丰富度Richness的增加，任务参与者获得的总奖励会越来越高。在图8(c)中，数据质量和丰富度分别固定在1和0.5。数据之间的特征分布越相似，相似度Similarity值越小，因此任务参与者获得的总奖励会随着数据相似度的增加而减

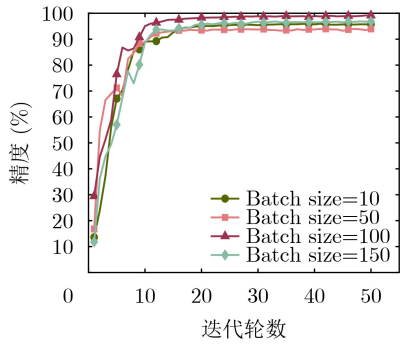


图 3 在不同Batch size中精度随通信回合数的变化

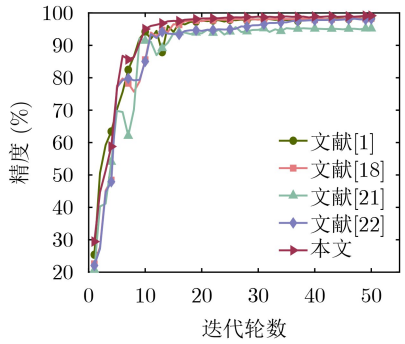


图 4 5种方案的精度随通信轮数的增加而增加

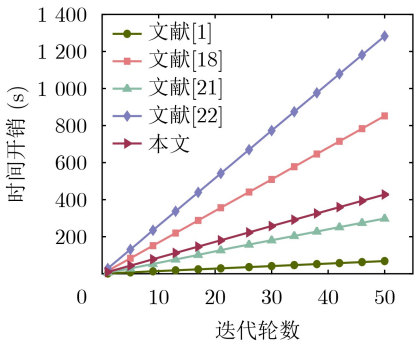


图 5 运行时间随着通信轮数的增加而增加

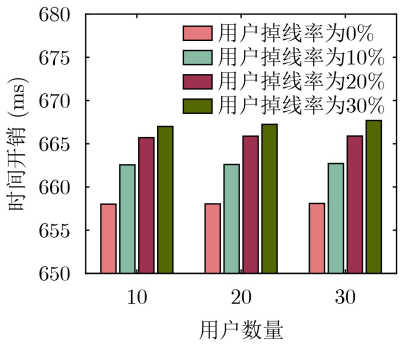


图 6 用户数量对验证时间的影响

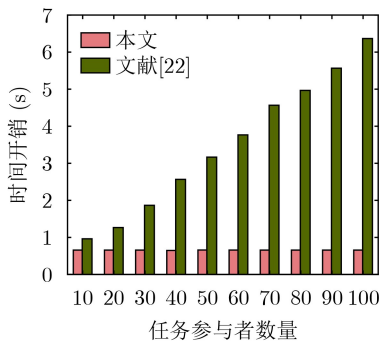
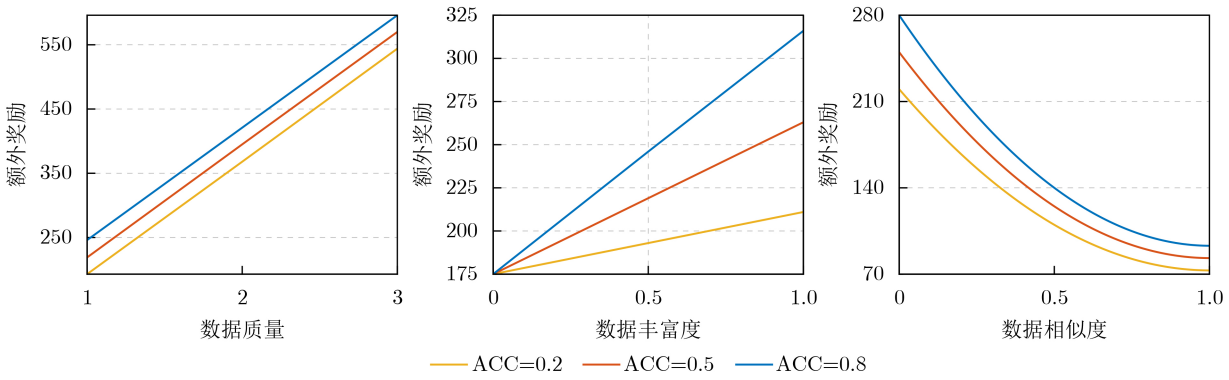


图 7 验证时间对比



(a) 数据质量对奖励的影响 (b) 数据丰富度对奖励的影响 (c) 数据相似度对奖励的影响

图 8 激励机制的评估结果

少。此外，图8还表明，任务参与者获得的总奖励与模型预测精度相关，将局部模型的评估准确度ACC设置成不同值，以区分高质量用户和低质量用户，从图中可看出，ACC值越高，其总奖励越高，这更加有效激励高质量任务参与者积极释放数据。通过实验结果表明，该激励机制能够根据训练数据的特性对局部训练和局部模型进行综合评价和激励。

7 结束语

本文提出一种基于同态加密和可验证的隐私保护联邦学习方案，并通过优化模型参数处理，有效缓解了同态加密带来的巨大的计算开销问题。因此，本方案在保障数据隐私的同时，避免了通信开销的显著增长。此外，本方案不仅引入了分布式密钥生成协议以消除对可信第三方机构的依赖，还结合Diffie-Hellman密钥交换协议和Shamir秘密共享算法，实现用户独立验证服务器提供的聚合结果的能力，同时支持用户中途退出以及用户合谋。为了进一步促进拥有高质量数据的用户对数据的释放，本文还提出了一个激励机制，旨在通过合理的奖励策略吸引更多拥有高质量数据的用户的加入。实验表明，本方案在模型收敛速度和预测精度方面均表现出色，同时用户验证聚合结果的时间也不会随着用户数量的增加而增加。然而，本文未考虑恶意用户的情况，例如个别用户可能上传错误或具有欺骗性的模型更新，从而影响全局模型的准确性和公平性。在今后的工作中，将进一步研究如何在保护数据隐私的同时，有效识别并抵御恶意用户的攻击。

参 考 文 献

[1] MCMAHAN B, MOORE E, RAMAGE D, *et al*. Communication-efficient learning of deep networks from decentralized data[C]. The 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA, 2017: 1273–1282.

[2] RODRÍGUEZ-BARROSO N, JIMÉNEZ-LÓPEZ D, LUZÓN M V, *et al*. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges[J]. *Information Fusion*, 2023, 90: 148–173. doi: 10.1016/j.inffus.2022.09.011.

[3] 孙钰, 严宇, 崔剑, 等. 联邦学习深度梯度反演攻防研究进展[J]. 电子与信息学报, 2024, 46(2): 428–442. doi: 10.11999/JEIT230541.

SUN Yu, YAN Yu, CUI Jian, *et al*. Review of deep gradient inversion attacks and defenses in federated learning[J]. *Journal of Electronics & Information Technology*, 2024, 46(2): 428–442. doi: 10.11999/JEIT230541.

[4] ZHANG Pengfei, CHENG Xiang, SU Sen, *et al*. Task allocation under geo-indistinguishability via group-based noise addition[J]. *IEEE Transactions on Big Data*, 2023, 9(3): 860–877. doi: 10.1109/TBDATA.2022.3215467.

[5] FENG Jun, YANG L T, REN Bocheng, *et al*. Tensor recurrent neural network with differential privacy[J]. *IEEE Transactions on Computers*, 2024, 73(3): 683–693. doi: 10.1109/TC.2023.3236868.

[6] FENG Jun, YANG L T, ZHU Qing, *et al*. Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment[J]. *IEEE Transactions on Dependable and Secure Computing*, 2020, 17(4): 857–868. doi: 10.1109/TDSC.2018.2881452.

[7] ALAZAB M, RM S P, PARIMALA M, *et al*. Federated learning for cybersecurity: Concepts, challenges, and future directions[J]. *IEEE Transactions on Industrial Informatics*, 2022, 18(5): 3501–3509. doi: 10.1109/TII.2021.3119038.

[8] 王冬, 秦倩倩, 郭开天, 等. 联邦学习中的模型逆向攻防研究综述[J]. 通信学报, 2023, 44(11): 94–109. doi: 10.11959/j.issn.1000-436x.2023209.

WANG Dong, QIN Qianqian, GUO Kaitian, *et al*. Survey on model inversion attack and defense in federated learning[J]. *Journal on Communications*, 2023, 44(11):

94–109. doi: 10.11959/j.issn.1000-436x.2023209.

[9] WANG Xiaoding, HU Jia, LIN Hui, *et al*. Federated learning-empowered disease diagnosis mechanism in the internet of medical things: From the privacy-preservation perspective[J]. *IEEE Transactions on Industrial Informatics*, 2023, 19(7): 7905–7913. doi: 10.1109/TII.2022.3210597.

[10] WEI Kang, LI Jun, DING Ming, *et al*. User-level privacy-preserving federated learning: Analysis and performance optimization[J]. *IEEE Transactions on Mobile Computing*, 2022, 21(9): 3388–3401. doi: 10.1109/TMC.2021.3056991.

[11] SUN Lichao, QIAN Jianwei, and CHEN Xun. LDP-FL: Practical private aggregation in federated learning with local differential privacy[C]. The Thirtieth International Joint Conference on Artificial Intelligence, 2021: 1571–1578.

[12] HAN Liquan, FAN Di, LIU Jinyuan, *et al*. Federated learning differential privacy preservation method based on differentiated noise addition[C]. 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 2023: 285–289. doi: 10.1109/ICCCBDA56900.2023.10154864.

[13] GUO Shengnan, WANG Xibin, LONG Shigong, *et al*. A federated learning scheme meets dynamic differential privacy[J]. *CAAI Transactions on Intelligence Technology*, 2023, 8(3): 1087–1100. doi: 10.1049/cit2.12187.

[14] STEVENS T, SKALKA C, VINCENT C, *et al*. Efficient differentially private secure aggregation for federated learning via hardness of learning with errors[C]. 31st USENIX Security Symposium (USENIX Security 22), Boston, USA, 2022: 1379–1395.

[15] BONAWITZ K, IVANOV V, KREUTER B, *et al*. Practical secure aggregation for privacy-preserving machine learning[C]. The 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, USA, 2017: 1175–1191. doi: 10.1145/3133956.3133982.

[16] ZHENG Yifeng, LAI Shangqi, LIU Yi, *et al*. Aggregation service for federated learning: An efficient, secure, and more resilient realization[J]. *IEEE Transactions on Dependable and Secure Computing*, 2023, 20(2): 988–1001. doi: 10.1109/TDSC.2022.3146448.

[17] WIBAWA F, CATAK F O, KUZLU M, *et al*. Homomorphic encryption and federated learning based privacy-preserving CNN training: Covid-19 detection use-case[C]. The 2022 European Interdisciplinary Cybersecurity Conference, Barcelona, Spain, 2022: 85–90. doi: 10.1145/3528580.3532845.

[18] WANG Bo, LI Hongtao, GUO Yina, *et al*. PPFLHE: A privacy-preserving federated learning scheme with homomorphic encryption for healthcare data[J]. *Applied Soft Computing*, 2023, 146: 110677. doi: 10.1016/j.asoc.2023.110677.

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

[20] 余晟兴, 陈钟. 基于同态加密的高效安全联邦学习聚合框架[J]. 通信学报, 2023, 44(1): 14–28. doi: 10.11959/j.issn.1000-436x.2023015.

YU Shengxing and CHEN Zhong. Efficient secure federated learning aggregation framework based on homomorphic encryption[J]. *Journal on Communications*, 2023, 44(1): 14–28. doi: 10.11959/j.issn.1000-436x.2023015.

[21] MA Jing, NAAS S A, SIGG S, *et al*. Privacy-preserving federated learning based on multi-key homomorphic encryption[J]. *International Journal of Intelligent Systems*, 2022, 37(9): 5880–5901. doi: 10.1002/int.22818.

[22] MA Xu, ZHANG Fangguo, CHEN Xiaofeng, *et al*. Privacy preserving multi-party computation delegation for deep learning in cloud computing[J]. *Information Sciences*, 2018, 459: 103–116. doi: 10.1016/j.ins.2018.05.005.

[23] XU Guowen, LI Hongwei, LIU Sen, *et al*. VerifyNet: Secure and verifiable federated learning[J]. *IEEE Transactions on Information Forensics and Security*, 2020, 15: 911–926. doi: 10.1109/TIFS.2019.2929409.

[24] SHEN Xiaoying, LUO Xue, YUAN Feng, *et al*. Verifiable privacy-preserving federated learning under multiple encrypted keys[J]. *IEEE Internet of Things Journal*, 2024, 11(2): 3430–3445. doi: 10.1109/JIOT.2023.3296637.

[25] SCHINDLER P, JUDMAYER A, STIFTER N, *et al*. EthDKG: Distributed key generation with Ethereum smart contracts[J]. *Cryptology ePrint Archive*, 2019.

[26] YUN A, CHEON J H, and KIM Y. On homomorphic signatures for network coding[J]. *IEEE Transactions on Computers*, 2010, 59(9): 1295–1296. doi: 10.1109/TC.2010.73.

[27] ELGAMAL T. A public key cryptosystem and a signature scheme based on discrete logarithms[J]. *IEEE Transactions on Information Theory*, 1985, 31(4): 469–472. doi: 10.1109/TIT.1985.1057074.

[28] ZHANG Li, XU Jianbo, VIJAYAKUMAR P, *et al*. Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system[J]. *IEEE Transactions on Network Science and Engineering*, 2023, 10(5): 2864–2880. doi: 10.1109/TNSE.2022.3185327.

郭　显：男，教授，博士，研究方向为密码学基础理论与应用、安全协议设计与分析、大数据安全等.

王典冬：男，硕士生，研究方向为隐私保护和联邦学习.
冯 涛：男，研究员，博士，研究方向为网络与信息安全、密码学、工业互联网等.
成玉丹：女，讲师，博士，研究方向为网络与信息安全.
蒋泳波：男，副教授，博士，研究方向为网络与信息安全、下一代网络体系结构、工业控制网络安全等.

责任编辑：余 蓉

A Verifiable Privacy Protection Federated Learning Scheme Based on Homomorphic Encryption

GUO Xian WANG Diandong FENG Tao CHENG Yudan JIANG Yongbo

(*School of Computer and Communication*, *Lanzhou University of Technology*, *Lanzhou* 730050, *China*)

Abstract:

Objective The growing reliance on data in today's digital age highlights the importance of effective data management across industries. Federated Learning (FL), an innovative approach, facilitates data collaboration and joint model development while maintaining privacy. However, existing homomorphic encryption-based security schemes for FL present several limitations. In some cases, FL servers may falsify aggregation results, leading to inaccurate models and subsequent issues, such as decision-making errors and erosion of trust in the system. Furthermore, servers may collude with users to steal private data, resulting in privacy breaches and potential misuse, including illegal marketing or cyberattacks. These issues undermine public confidence in data security and limit the broader adoption of FL, thus impeding innovation and efficiency gains. Many current schemes also depend heavily on trusted Third PArties (TPA) for key generation, introducing high communication overhead and diminishing model training efficiency, which discourages users from sharing data. This study proposes an optimized solution utilizing a distributed key generation protocol to prevent collusion and reduce third-party dependency. It integrates the Chinese Remainder Theorem (CRT) to lower communication costs and introduces auxiliary nodes to ensure aggregation accuracy. Additionally, an incentive mechanism is designed to encourage users to share high-quality private data. Collectively, these measures address key challenges in existing systems, offering a safer, more efficient, and reliable framework for the widespread adoption of FL.

Methods The proposed FL scheme is collusion-resistant, privacy-preserving, and verifiable, integrating a distributed key generation protocol to achieve interactive key generation. This method enables users to encrypt data using their private keys while requiring collaborative decryption from multiple participants, thereby eliminating the reliance on TPA. It effectively prevents server collusion involving fewer than n–1 users and incorporates a fault-tolerant mechanism to address potential user disconnections. Enhanced data security and reduced communication overhead are achieved by employing randomized model processing, combined with CRT based dimensionality reduction prior to encryption. Specifically, each user superimposes a random model of identical dimensions onto their local model, uploads the randomized model to the server for aggregation, and then decomposes the random model into a public matrix and a low-dimensional vector. After applying CRT to reduce the vector's dimensionality, homomorphic encryption is performed, reducing the data that must be encrypted and uploaded. The scheme also introduces auxiliary nodes and utilizes a bilinear aggregate signature algorithm, enabling each user to independently verify the aggregation results provided by the server, ensuring correctness and verifiability. Additionally, an incentive mechanism based on data characteristics, such as quality and richness, encourages participation from users with high-quality data. By dynamically calculating and distributing rewards after task completion, the mechanism effectively promotes the active sharing of high-quality data by users.

Results and Discussions The proposed scheme is comprehensively evaluated through extensive experiments. The results demonstrate improvements in both model accuracy and training efficiency. As shown in (Fig. 4), the scheme achieves slightly higher accuracy on the MNIST dataset compared to FedAvg and three other approaches. This improvement is attributed to the incentive mechanism, which effectively encourages

participation from users with high-quality data. Additionally, the preprocessing steps involving model randomization and CRT-based dimensionality reduction prior to encryption enhance communication efficiency, as evidenced by the time overhead comparison in (Fig. 5). The experimental evaluation of the designed verification scheme, shown in (Fig. 6), reveals that the verification time of user will not increase with the increase of the number of users. Even with 30 users, the verification time increases by only 1%, despite a 30% dropout rate, when compared to scenarios with no user dropouts. (Fig. 7) further confirms the verification time advantages of the proposed scheme over other verification approaches. Finally, the reward allocation mechanism demonstrates desirable fairness characteristics, as shown in (Fig. 8), where users contributing high-quality data consistently receive proportionally greater rewards throughout the training process.

Conclusions　The proposed privacy-preserving FL scheme, based on homomorphic encryption and verifiable mechanisms, effectively reduces the computational overhead associated with homomorphic encryption through optimized model parameter processing. This ensures data privacy while preventing excessive communication costs. Additionally, the framework incorporates a distributed key generation protocol to eliminate reliance on trusted third-party institutions and integrates the Diffie-Hellman key exchange protocol with Shamir's secret sharing algorithm. This combination enables users to independently verify aggregation results provided by the server while supporting user dropouts and preventing collusion. To further encourage data contributions from users with high-quality data, an incentive mechanism is introduced, employing rational reward strategies to attract such users. Experimental results demonstrate excellent performance in model convergence speed and prediction accuracy, with verification time for aggregation results remaining stable regardless of the number of users. However, this study does not address potential malicious behaviors, such as individual users uploading erroneous or deceptive model updates that could compromise global model accuracy and fairness. Future work will focus on developing mechanisms to identify and mitigate attacks from malicious users while maintaining data privacy protection.

Key words: Federated Learning (FL); Homomorphic encryption; Privacy protection; Verifiability