分层联邦学习中非交互式可验证安全聚合方法

宋书汉[1,2]，田有亮[1,2,3]，王帅[1,2]

（1. 贵州大学公共大数据国家重点实验室，贵州 贵阳 550025；2. 贵州大学密码学与数据安全研究所，贵州 贵阳 550025；
3. 贵州大学大数据与信息工程学院，贵州 贵阳 550025）

摘 要： 联邦学习（FL）作为一种分布式机器学习（ML）框架，能够在保护数据隐私的同时实现模型协同训练。然而，FL 在隐私保护、参与方互信及恶意攻击等方面仍面临挑战，尤其是在分层联邦学习（HFL）中，中心服务器、中间层及终端设备的不可信性可能导致隐私泄露或恶意操纵。此外，恶意用户可能上传异常参数破坏训练进程，影响模型性能。因此，如何在 HFL 中高效实现安全验证和恶意检测，成为亟待解决的问题。针对 HFL 中的参与方无法互信、拜占庭攻击等问题，提出一种分层架构下的非交互式验证 FL 安全聚合方案。首先，基于承诺方案设计多层架构下的 FL 非交互式验证机制，允许各参与方进行相互验证。其次，基于零知识范围证明构造恶意更新的约束与检测方案，使服务器能检测并剔除恶意用户。再次，基于中国剩余定理（CRT）设计噪声掩码方案，在保证用户本地隐私的同时还支持用户的退出与重连；最后，安全性分析与实验评估表明，该方案能够以较高的效率实现安全相互验证以及恶意检测。

关键词： 联邦学习；零知识证明；可验证性；鲁棒性；隐私保护

中图分类号： TP309 **文献标志码：** A **DOI：** 10.19678/j.issn.1000-3428.0252018

Secure Aggregation Method for Hierarchical Federated Learning with Non-interactive Verification

SONG Shuhan[1,2], TIAN Youliang[1,2,3], WANG Shuai[1,2]

(1. State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, Guizhou, China;
2. Institute of Cryptography and Data Security, Guizhou University, Guiyang 550025, Guizhou, China;
3. College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, Guizhou, China)

【Abstract】 Federated Learning (FL), a distributed Machine Learning (ML) framework, can achieve collaborative model training while protecting data privacy. However, challenges remain in ensuring privacy protection, establishment of mutual trust among participants, and defending against malicious attacks. In Hierarchical Federated Learning (HFL), the potential untrustworthiness of central servers, intermediate layers, and edge devices poses risks of privacy leakage and malicious manipulation. Additionally, malicious users may upload abnormal parameters, disrupting the training process and affecting model performance. Therefore, efficient security verification and malicious detection are critical issues in HFL. To address the challenges of mutual distrust among participants and Byzantine attacks in HFL, this paper proposes a secure aggregation scheme with non-interactive verification under a hierarchical architecture. First, a mutual verification mechanism for HFL is designed based on a commitment scheme, allowing participants to perform mutual verification. Second, a constraint and detection scheme for malicious updates is constructed using non-interactive zero-knowledge range proofs, enabling the server to detect and filter out malicious users. Third, a noise masking scheme is designed based on the Chinese Remainder Theorem (CRT), supporting user exit and reconnection while ensuring users′ local privacy. Finally, security analysis and experimental evaluation demonstrate that the proposed scheme achieves secure mutual verification and malicious detection with high efficiency.

【Key words】 Federated Learning (FL); zero-knowledge proof; verifiability; robustness; privacy preserving

0 引言

近年来，随着数字化技术的快速发展，人工智能已经成为科技革命的核心驱动力之一[1]，并且已经广泛应用于医疗卫生[2]、交通系统[3]、智能驾驶[4]、工业生产[5]等场景中。但是传统的集中式机器学习

基金项目： 国家重点研发计划（2021YFB3101100）；国家自然科学基金（62272123，62272102）；贵州省高层次创新型人才基金（黔科合平台人才[2020]6008）；贵州省科技计划（黔科合平台人才[2020]5017，黔科合支撑[2022]一般 065）。

作者简介： 宋书汉，男，硕士研究生，主研方向为联邦学习、安全多方计算；田有亮（通信作者），教授、博士、博士生导师；王帅，博士研究生。

收稿日期： 2025-01-08 **修回日期：** 2025-03-20 **E-mail：** youliangtian@163.com

(ML)存在“数据孤岛”、隐私泄露、数据传输开销大等困境。为了解决上述集中式 ML 面临的问题与挑战,谷歌研究院的 KONEČNÝ 等[6]在 2016 年率先提出了联邦学习(FL)的概念。FL 是一种分布式的 ML 框架,其特点是允许分布式节点能够在不公开本地数据的情况下和其他节点协同训练模型。这使得 FL 能够在充分利用分散数据的同时确保包含用户隐私的数据信息留在本地,达到隐私保护的目的。

根据用户的数据分布特征可以将 FL 分为横向 FL 与纵向 FL[7],传统 FL 的架构通常是用户与中心服务器直接通信的模式。但在物联网(IoT)等边缘计算环境中,随着终端设备的数量增多,中心服务器与终端设备的通信开销将会大大增加,且终端设备的异构性与不稳定性可能会导致训练延迟等问题,这都为传统的 FL“用户-服务器”架构带来严峻挑战。为此,分层联邦学习(HFL)通过引入多层节点,将传统的“用户-服务器”架构扩展为“终端-边缘-服务器”架构。在 HFL 中,终端设备只需要与其对应的中间层服务器进行通信交互,从而降低中心服务器的负载压力,提升系统的可扩展性。例如,王汝言等[8]针对 IoT 中的设备异构问题,提出了一种资源高效的分层协同 FL 方法,提高了系统效率。LIM 等[9]提出了一种结合资源分配与激励机制的 HFL 框架,采用动态分簇等技术使得系统的通信开销大幅降低。

然而,FL 的分布式结构也引入了新的隐私安全问题,包括隐私泄露、参与方之间的互信问题及恶意攻击。研究发现,攻击者针对用户的局部模型参数发起重构攻击[10-11],可以推断出用户的隐私信息。此外,FL 系统中的服务器、用户、中间层等节点存在不可信的情况:一是恶意的服务器或中间层会尝试返回不正确的聚合结果,并获取局部模型参数的隐私[12];二是恶意用户会对训练过程进行攻击,破坏训练进程或是窃取其他用户的隐私信息,例如投毒攻击[13]、推理攻击[14]等;三是终端设备的不稳定性可能导致相关数据的传输失败、传输错误等问题。因此,如何在保证计算效率的同时增强 FL 的隐私保护、建立互相可信的验证机制,并有效防御恶意攻击,是当前研究的核心问题。

1 相关研究

针对 FL 中存在的隐私保护问题,近年来的研究工作主要聚焦于使用密码学相关技术设计安全聚合协议,如同态加密(HE)、安全多方计算(SMC)与差分隐私(DP)等,以确保用户本地更新参数的安全。WIBAWA 等[15]提出了一个使用了 HE 的医疗数据隐私信息保护 FL 算法,该算法结合多方计算协议来保护本地更新的模型免受对手的攻击。但是,基于 HE 的方法在 FL 的实际应用中通常会面对大量的幂运算,这会带来较大的计算开销,导致运行效率低、密文爆炸等问题。ZHANG 等[16]提出了一种增强的 SMC 方法,在基本信息传输到服务器之前,通过两轮分解对局部模型进行加密,为本地更新提供隐私保护。SMC 需要用户间频繁地进行数据传输,在 FL 中,边缘节点存在由于不稳定而掉线等情况,SMC 为了恢复掉线用户的信息,会带来更大的通信开销。CHAMIKARA 等[17]提出了一种用于工业环境的本地 DP FL 协议,该协议能够在不受信任实体的工业环境中运行,具有较强的隐私保护能力。但是,DP 会导致模型精度下降等问题,在对模型有高精度要求的场景中不适用。此外,BONAWITZ 等[18]提出了具有加性性质的双掩码机制,用于保护本地更新的隐私,服务器可以在进行正确的聚合后完全消除掩码,这被认为是一种更高效的保护方案。综上,本文采用了基于掩码的本地更新隐私保护方案。

在 FL 系统中,由于服务器、中间层和终端设备的潜在不可信性,参与方之间需要建立验证机制。GUO 等[19]提出了线性同态哈希(LHH)与承诺相结合的方案,实现服务器对用户上传的本地更新的验证。FU 等[20]提出了一种基于拉格朗日插值的聚合结果正确性验证机制。GAO 等[21]在 FU 等[20]工作的基础上,优化了其验证效率。LI 等[22]使用双线性配对设计了服务器对本地更新的验证机制,但是基于双线性配对的方法会导致 FL 的通信开销增大,尤其是在本地更新数据量比较大时。然而,现有基于 LHH 的验证机制仍存在缺点。例如,HAHN 等[23]提出的 LHH 方案虽能高效验证本地更新的完整性,但其用于验证的哈希值会直接向验证方暴露,有研究表明,这可能导致模型稀疏场景下的暴力猜解攻击,使得用户模型中的敏感信息泄露。因此,如何在保证验证效率的同时确保验证过程不泄露任何原始信息,成为切实需要解决的问题。针对这一问题,非交互式零知识证明(NIZK)因其独特的性质展现出显著优势:首先,NIZK 允许验证者无需与证明者交互即可验证断言的真实性,避免了上述双线性配对等方案由于多轮交互带来的高昂开销;其次,NIZK 中“零知识”的特性可确保验证过程中不泄露关于原始模型参数的任何信息,从而规避同态哈希

因暴露中间值导致的隐私风险。张首勋等[24]基于NIZK提出了一种去中心化的FL可信聚合方案,通信开销与效率比传统方案有所提升。AHMADI等[25]提出了一种NIZK与区块链相结合的FL方案,进一步降低了客户端的验证成本,提高了系统的可扩展性和效率。由此,本文选择引入NIZK技术,不仅能避免LHH技术潜在的安全问题,还能够将多轮交互的验证过程缩减为一次交互,缩减系统的通信开销。

模型投毒攻击是FL中常见的恶意攻击方式,攻击者的目的是上传恶意的梯度参数来破坏整个FL的训练任务,使其性能下降或出现错误。针对模型投毒攻击的防御思想主要是对模型参数进行检测,并剔除恶意参数。部分研究基于相似性算法来实现对恶意攻击的识别。魏立斐等[26]设计了一种异步FL的安全梯度聚合方案,基于余弦相似度与k-means算法实现对恶意客户端的识别。LI等[27]基于核密度估计方法来测量用户之间的相对分布,从而检测出恶意用户。部分研究聚焦于基于L范数对用户的本地更新加以限制的方法。SUN等[28]使用范数对用户的本地更新进行限制,实验证明该方法可以很好地防御恶意用户的模型替换攻击。LYCKLAMA等[29]针对范数约束防御相关的攻击进行了大量评估,并证明了其有效性与高效性。然而,基于相似性算法的方法存在一些不足之处:当恶意用户比例很大时,该方法的准确性可能会受到影响;当用户数量比较多时,相似度的计算通常会引入比较大的计算开销。与之相比,基于L范数的方法不依赖相关性,仅对模型参数本身进行约束,不受恶意用户比例的影响,同时,计算开销也不会因用户数量增多而增大。因此,本文将范数约束与零知识证明相结合,实现服务器对恶意参数的检测。

针对上述FL所面临的隐私保护、互信验证以及恶意攻击问题,本文主要的研究工作如下:

1)设计了一个多层架构的FL非交互式验证机制,允许FL中各参与方安全地进行互相验证,而不泄露相关隐私信息,在合理的安全假设下确保FL训练全过程的可信,且非交互式设计无需多次交互,大幅缩减通信开销。

2)设计了一个恶意用户参数的检测方案,约束用户上传的模型参数,允许服务器验证并识别恶意用户,确保本地更新的合法性,并提出了分层随机采样策略,显著降低开销。

3)对本文的机制与方案进行了安全性证明与实验评估,结果表明本文方案具有安全性与可行性。

2 背景知识

2.1 FL

FL的基本框架由N个用户$u_i(i=1,2,\cdots,N)$与云服务器CS组成。设参与FL训练的用户集合为$U=\{u_1,u_2,\cdots,u_i\}$,每个用户都拥有一个私有的数据集PD_i。在FL训练的第t轮训练中,U中的用户u_i从CS处接收最新的模型参数$\boldsymbol{w}^{t-1}$,并在本地基于PD_i利用随机梯度下降算法得到私有梯度$\boldsymbol{w}_i^t=\boldsymbol{w}_i^t-\eta_i\boldsymbol{g}_i^{t-1}$,其中,$\eta_i$是本地学习率,$\boldsymbol{g}_i^{t-1}=\nabla_w l_i$,$l_i$为$u_i$的损失函数。用户将私有梯度$\boldsymbol{w}_i^t$上传到$CS$后,$CS$通过联邦平均算法FedAvg[30]聚合得到全局参数$\boldsymbol{w}^t=\sum_{i=1}^{N}\boldsymbol{w}_i^t/N$,并下发给用户集$U$进行下一轮训练。

2.2 中国剩余定理(CRT)

CRT给出了一元线性同余方程组有解的判定条件,并用构造法给出了在有解情况下解的具体形式。

给定两两互质的正整数$f_1,f_2,\cdots,f_m$和任意整数$y_1,y_2,\cdots,y_m$,当以下方程同时成立时存在解C,且$C \bmod F$是range(F)内的唯一解,其中$F=\prod f_i,i\in[1,m]$。

$$C\equiv y_1(\bmod f_1)$$
$$C\equiv y_2(\bmod f_2)$$
$$\vdots$$
$$C\equiv y_m(\bmod f_m)$$

解C表示为:

$$C\equiv(y_1g_1st_1+y_2g_2st_2+\cdots+y_mg_mst_m)$$

式中:$g_i=F/f_i$;$g_i\equiv 0 \bmod f_j$(对于任意$j\neq i$);$g_ist_i\equiv 1 \bmod f_i$。

2.3 LHH

LHH是基于离散对数困难问题设计的具有加性同态性质的哈希算法方案。具体可以由3个多项式时间算法组成,表示为:

{LHH.Init,LHH.Hash,LHH.Eval}

具体为:

LHH.Init(1^k,1^d)

输入安全参数k和维度d,输出公共参数pp,其中包括了素数阶循环群$\mathbb{G}$、群的阶q、群的生成元g以及d个不同的元素$\{g_1,g_2,\cdots,g_d\}\in\mathbb{G}$。为表述方便,默认以下算法都隐含输入了$pp$。

LHH.Hash($\boldsymbol{x}$):输入维度为d的向量$\boldsymbol{x}$,输出$\boldsymbol{x}$的LHH值h。

$$h=\prod_{i=1}^{d} g_i^{x[i]} \in \mathbb{G} \tag{1}$$

LHH. Eval$(h_1,h_2,\cdots,h_n)$:输入 n 个 LHH 值,输出这些哈希值的线性组合 H。

$$H=\prod_{i=1}^{n} h_i \tag{2}$$

2.4 承诺方案

承诺方案允许参与方对信息进行承诺并保证其隐私性。传统的承诺方案一般分为两个阶段:承诺阶段和承诺揭示阶段[31]。在承诺阶段,证明者 P 对消息 m 进行承诺计算,得到承诺结果后将其发送给验证者 V,并保证信息的隐私性。在承诺揭示阶段,证明者 P 揭示被承诺的消息 m,验证者 V 进行承诺计算,并打开接收到的承诺值,将两者进行对比,若相等,则完成对承诺的正确验证。

承诺方案应满足以下 2 个基本性质:

1)隐藏性:在承诺未被揭示之前,验证者 V 不能得到关于消息 m 的任何信息。

2)绑定性:消息 m 在第一阶段得到的承诺值必须和其公开的承诺值保持一致。

2.5 Schnorr 协议

经典的 Schnorr 协议是一个交互式的身份验证协议,其安全性基于离散对数问题的困难性。利用 Fiat-Shamir 启发式将其转变为非交互式,非交互式的 Schnorr 协议执行方式如下:

1)A 选取一个随机数 $r_A \in \{1,2,\cdots,q-1\}$,计算 $R_A \equiv g^{r_A} \pmod p$,$e_A=\mathrm{Hash}(R_A,PK_A)$,并构造证明 $z_A=r_A+e_A \cdot sk_A$。

2)B 接收 z_A、R_A、e_A、PK_A,并核实:

$$g^{z_A} \stackrel{?}{=} R_A \cdot PK_A^{e_A} \tag{3}$$

2.6 NIZK

NIZK 系统可以允许证明者 P 向验证者 V 证明一个语句 $x \in \mathcal{L}$,而不透露任何关于 x 的信息。其中,$\mathcal{L}$ 是一个 NP 语言。NIZK 需要满足以下的性质[32]:

1)完备性。对任意 $x \in \mathcal{L}(|x|=\kappa)$及其证据 w,有:

$$\Pr[r \leftarrow_R \{0,1\}^{\mathrm{poly}(\kappa)};\pi \leftarrow \mathcal{P}(r,x,w):\mathcal{V}(r,x,\pi)=1]=1 \tag{4}$$

2)可靠性。如果 $x \notin \mathcal{L}$,那么对于任意的 $\mathcal{P}^*$,以下概率在 κ 上是可以忽略的:

$$\Pr[r \leftarrow_R \{0,1\}^{\mathrm{poly}(\kappa)};\pi \leftarrow \mathcal{P}^*(r,x,w):\mathcal{V}(r,x,\pi)=1] \tag{5}$$

3)零知识性。假设有一个多项式时间模拟器 S,它对于任意的 $x \in \mathcal{L}$ 及其证据 w,以下分布是不可区分的:

$$\{r \leftarrow_R \{0,1\}^{\mathrm{poly}(\kappa)};\pi \leftarrow \mathcal{P}(r,x,w):(r,x,\pi)\}$$
$$\{(r,x) \leftarrow S(x):(r,x,\pi)\} \tag{6}$$

本文采用了 Bulletproofs 零知识证明协议[33],并利用 Fiat-Shamir 启发式将其转变为非交互式。Bulletproofs 允许数据拥有者对数据进行范围证明,数据接收者对其进行验证,确保数据在某特定范围内,且不需要公开原始的数据信息。

3 系统模型

3.1 系统架构

系统主要由 4 个部分组成,分别是用户 U、收集器 DC、可信中心 TA、云服务器 CS,具体架构如图 1 所示。

1)用户 U:拥有本地私有数据集,且利用本地数据集参与 FL 的训练任务。在 FL 的每一轮训练中,U 在本地数据集上进行训练后得到本地梯度参数,对其进行加密、承诺、范围证明等操作后上传至对应的 DC;在更新阶段收到 CS 的梯度更新与聚合承诺后,对其聚合承诺进行验证,验证通过后接收该结果,并进行下一次训练。重复此过程直到训练任务完成。

2)收集器 DC:DC 作为 U 与 CS 中间的节点,作用是减少系统的通信复杂度,提高效率。每个 DC 都与一组特定的用户 U 相连接,负责收集 U 上传的数据,并对其进行预验证、聚合等操作,最后将通过验证的数据预聚合,并上传至 CS。此外,DC 还负责对 CS 的聚合结果的正确性进行验证,并将验证结果广播。

3)云服务器 CS:CS 具有强大的计算能力与数

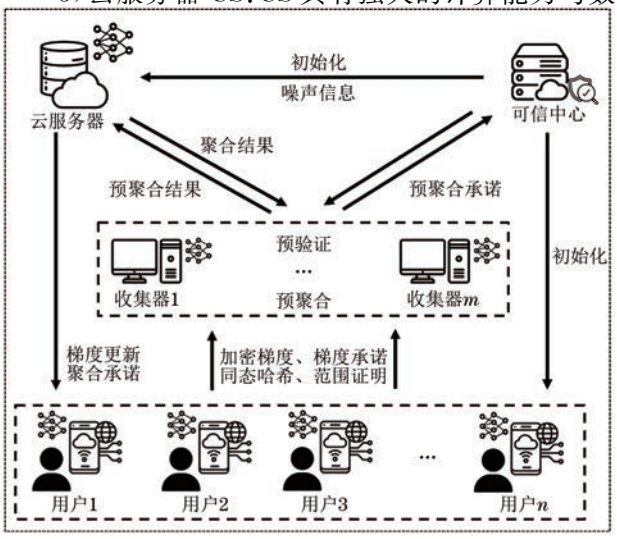

图 1 系统架构

Fig. 1 System architecture

据处理能力，负责控制整个 FL 训练任务。在收到 DC 的预聚合数据后，对其进行验证，验证通过后进行加密数据聚合。将聚合密文发送给 DC 并验证通过后，从 TA 处获取噪声信息，计算出模型更新，将更新与聚合承诺下发至用户。

4）可信中心 TA：TA 负责初始化参数与噪声信息的生成、分发和更新，并记录预聚合结果的承诺，用于后续 DC 对密文聚合结果的验证。实际上，TA 可以是 CS 上的某个可信执行环境，而不需要是独立的服务器。本文将其视为独立的服务器，便于描述与理解。

3.2 威胁模型与安全需求

在本文中，假定 DC 是半诚实的，这意味着它们会正确地执行 FL 训练协议，但是它们会尝试获取 U 的隐私信息，TA 是可信的。此外，本文假设各方不共谋。对于 U 和 CS，本文认为它们首先是半诚实的，但在此基础上会有以下特定的恶意行为，具体说明如下：

1）CS 的恶意行为：CS 出于两种动机会进行不正确的聚合。一是节省成本，进行错误聚合甚至随机返回结果，以节约计算资源；二是精心设计并篡改聚合结果，以诱导诚实的 U 提供更多包含隐私信息的数据。

2）U 的恶意行为：恶意的 U 会上传不正确的梯度参数，实施投毒攻击，以降低整体模型训练效果。

针对上述存在的安全威胁，本文需要满足以下的安全需求：

1）数据隐私性：U 的原始数据以及本地的梯度参数需要保密。

2）数据完整性与正确性：U 上传的数据与 CS 返回的数据需要确保其完整性与正确性，且双方都能对对方发送的数据进行验证。

3）数据安全性：U 要确保其上传的数据无恶意，且 CS 能对其进行验证，以防止恶意 U 的投毒攻击。

4 方案设计

4.1 符号与定义

首先介绍本文方案中使用的符号与定义，如表 1 所示。本文使用 U 来表示参与 FL 训练的用户集合，共有 m 个，$u_i \in U$ 表示该集合中具体的某一个用户。考虑到在 FL 训练的过程中，可能存在用户掉线的问题，因此，本文用 $U_t \subseteq U$ 表示在第 n 轮中成功上传了数据的用户，并有 $U \supseteq U_0 \supseteq U_1 \supseteq U_2 \supseteq U_3$。同时，本文使用 DC 来表示 FL 中的收集器集合，共有 t 个，$dc_j \in DC$ 表示该集合中具体的某一个收集器。

此外，本文使用粗体字母表示向量，例如 $\boldsymbol{w}$，该向量中的第 a 个元素用 $\boldsymbol{w}[a]$ 表示，其中 $a \in [1,n]$，n 为该向量的维度。

表 1 符号定义

Table 1 Symbol definition

参数	含义
u_i	第 i 个用户
m	用户数量
t	收集器数量
r_i	CS 为用户 u_i 分配的随机数
f_i	TA 为用户 u_i 分配的秘密值
dc_j	第 j 个收集器
$u_i \in dc_j$	用户 u_i 与收集器 dc_j 相连
$\boldsymbol{w}_i$	用户 u_i 的本地梯度更新
$\boldsymbol{k}_i$	用户 u_i 加密后的本地梯度更新
$\{pk_{ui}, sk_{ui}\}$	用户 u_i 的密钥对
Com_i	$\boldsymbol{w}_i$ 的承诺
$\boldsymbol{z}_i$	承诺证明辅助项
π_i	用户 u_i 计算的有关本地更新的证明
$\pi_{1\infty}$	$\boldsymbol{w}_i$ 的 L∞范式证明
π_{L2}	$\boldsymbol{w}_i$ 的 L2 范式证明
$B_{1\infty}, B_{L2}$	服务器规定的范围边界
K_j	C_j 的预聚合结果
K	CS 的聚合结果
$\boldsymbol{W}$	去除噪声后的聚合结果

4.2 功能模块

本节介绍了本文方案中所设计的功能模块，如无特殊说明，计算细节的相关内容在第 4.3 节中呈现。

4.2.1 相互验证机制

本文基于非交互式的 Schnorr 协议构造了一种承诺协议，并基于承诺与 LHH 设计了非交互式的相互验证机制。

1）承诺方案：主要用于服务器与用户之间的相互验证，双方都能对接收的模型参数的完整性进行验证。将承诺方案记作 COM，则 COM 可以定义为：

$$\mathrm{COM}=\{\mathrm{Com.gen},\mathrm{Prove.gen},\mathrm{Com.veri},\mathrm{AggCom.gen.AggCom.veri}\}$$

具体如下：

$\mathrm{Com.gen}(g,\boldsymbol{w}_i)\rightarrow Com_i$：输入用户的本地更新 $\boldsymbol{w}_i$ 与生成元 g，输出用户本地更新的承诺 Com_i。

$\mathrm{Prove.gen}(Com_i, pk_{ui}, \boldsymbol{z}_i)\rightarrow \pi_i$：输入用户的本地承诺 Com_i、用户的公钥 pk_{ui} 以及证明辅助项 $\boldsymbol{z}_i$，

输出供服务器验证的证明 π_i。

Com. veri(π_i)→(0,1):服务器输入证明 π_i,输出验证结果。

AggCom. gen(U,$\boldsymbol{w}_{\text{up}}$,$\tau$,$\pi_\tau$)→$\pi_{\text{agg}}$:服务器输入参与聚合任务的用户组 U、全局更新 $\boldsymbol{w}_{\text{up}}$、随机选取的用户编号 τ 及其证明 π_τ,输出全局更新的证明 π_{agg}。

AggCom. veri(π_{agg})→(0,1):用户输入证明 π_{agg},输出验证结果。

2)LHH:主要用于收集器对服务器和用户的数据进行预验证。虽然在第 1 章相关研究中提到,LHH 可能导致模型稀疏场景下的暴力猜解攻击,但是在本文的系统架构中,LHH 是用于加密后的梯度,在最坏的情况下,攻击者也只能得到加密梯度,而无法获取用户的隐私信息。其定义内容在第 2.3 节中。

基于上述机制,本文方案中主要的参与方用户 U、收集器 DC 以及服务器 CS 能够实现互相验证,确保参数的完整性与正确性。

4.2.2 掉线与重连

本文基于 CRT 来处理用户的掉线与重连问题。在本文方案中,掉线用户 U 的噪声总和由 TA 直接进行计算,确保最终聚合能够正确消除所有噪声。与传统的基于秘密共享的处理方式相比,本文方案能节省秘密共享中高昂的通信开销,且不会遭受共谋攻击的影响。

此外,掉线的用户 U 在任务进行期间可能会有重连的需求,利用 CRT 也能够支持用户重连回 FL。用户 U 掉线时仍保有自身的秘密值 f_i,其在重连后可以从 TA 处获取最新的参数 C_j^{new} 与 F_j^{new},并计算出自身的最新掩码:

$$y_i = C_j^{\text{new}} \bmod f_i$$
$$g_i = F_j^{\text{new}} / f_i$$
$$st_i = g_i^{-1} \bmod f_i \tag{7}$$

4.2.3 范围证明

本文基于 Bulletproofs 对用户 U 的本地更新进行 L∞与 L2 范围证明,Bulletproofs 支持将多个证明聚合为一个进行证明,这将节省很多验证开销。将范围证明记作 RangePF,则 RangePF 可以定义为:

RangePF={Range. gen,RangeAgg. veri}

具体为:

Range. gen($\boldsymbol{w}_i$)→($\pi_{\text{L}\infty}$,π_{L2}):基于用户的本地更新 $\boldsymbol{w}_i$ 生成其对应的范围证明 $\pi_{\text{L}\infty}$ 与 π_{L2}。

RangeAgg. veri($\pi_{\text{L}\infty}$,π_{L2})→(0,1):服务器对用户的范围证明进行聚合并验证,输出验证结果。

基于范围证明,服务器可以在不获取具体的单个用户本地更新的同时验证其是否在声称的范围内,并基于此判断用户本地更新的合法性。

4.2.4 分层随机抽样策略

由于 FL 中用户的本地更新参数量比较大,在生成 L∞范围证明时,如果要对本地更新中的每一个参数都进行一次范围证明,这将带来比较大的额外计算开销,对用户并不友好。事实上,并不需要对所有参数进行检查才能检出不符合范围的值,因为只要检测到一个离群值出现,就能确定恶意用户,所以本节提出了一种分层随机抽样策略。

该策略的核心思想为:将模型参数按照网络层级进行分层,并在每一层中进行统计学意义上的抽样验证。服务器基于给定的置信度和误差容限,计算出每层所需的最小样本量,再对每一层进行随机抽样,用户按照抽样结果进行范围证明。此外,用户在生成其范围证明之前,需要生成本地更新的承诺,服务器可以选择在用户上传承诺之后再执行分层随机抽样策略,利用承诺对于完整性的保证,更能确保该策略的安全性。

该策略能确保检出效果的同时极大减小参与范围证明的参数量,显著降低计算与通信开销。详细的安全性证明将在第 4.3 节中给出。

4.3 具体方案

4.3.1 第 0 轮:系统初始化

1)初始化。TA 执行 TA. Init($1^k 1^d 1^R$)初始化系统并生成公共参数用于后续的 LHH 计算、承诺计算、密钥计算与随机数生成。为简化表述,这些公共参数将隐含地作为相关计算的输入。

2)用户注册。u_i 执行 u_i. Register(CS,u_i),向 CS 注册,获取由 CS 生成的随机数 r_i;u_i 执行 u_i. Register(TA,u_i),向 TA 注册,获取由 TA 生成的秘密值 f_i,密钥对{pk_{ui},sk_{ui}},并确定所连接的收集器 dc_j。

3)用户选取。用户注册完毕后,CS 随机选择 q 个用户建立用户组 U_0,TA 从 U_0 中随机选取 m 个用户建立用户组 U_1,其中 $q>m$。U_1 中的用户参与本轮训练。

4)计算参数。对于 $u_i \in U_1$,TA 选择 $y_i < f_i$,其中,y_i 是任意整数。对于 $u_i \in dc_j$,计算式(8)并发送 $F=\{F_1,F_2,\cdots,F_t\}$ 至对应的 $u_i \in dc_j$。

$$F_j = \prod f_i \tag{8}$$

u_i 计算 $y_i g_i st_i$,其中:

$$g_i = F_j / f_i \tag{9}$$

$$st_i \equiv g_i^{-1} \bmod f_i \tag{10}$$

TA 计算：

$$\boldsymbol{C}=\{C_1,C_2,\cdots,C_t\} \tag{11}$$

$$C_j=\sum_{i=1}^{m} y_i g_i st_i \bmod F_j \tag{12}$$

5)广播初始模型。CS 设定 FL 训练目标，并广播初始模型。

4.3.2 第 1 轮：本地训练

1)本地训练。u_i 在本地训练，获得本地更新 $\boldsymbol{w}_i$。

2)承诺生成。u_i 计算其本地更新的承诺值 $Com_i=g^{\boldsymbol{w}_i}$，并构造出上述承诺的证明：

$$\pi_i=(\boldsymbol{z}_i,pk_{ui},Com_i) \tag{13}$$

式中：$\boldsymbol{z}_i=\boldsymbol{w}_i h_i+\boldsymbol{sk}_{ui}$，$\boldsymbol{sk}_{ui}$ 是将 sk_{ui} 拓展到与 $\boldsymbol{w}_i$ 维度相同的向量，sk_{ui} 是根据安全参数生成的随机数，$h_i=\mathrm{Hash}(Com_i,pk_{ui})$，$pk_{ui}=g^{sk_{ui}}$。

3)范围证明。u_i 生成范围证明，执行 Range.gen($\boldsymbol{w}_i$)→($\pi_{L\infty}$,π_{L2})。其中：$\pi_{L\infty}$ 是对 $\boldsymbol{w}_i$ 中的 $\boldsymbol{w}_i[a]$执行概率选取策略并进行范围证明，确保选取的每个 $\boldsymbol{w}_i[a]$都不超过 $B_{L\infty}$；π_{L2} 是 $\sum_{a=1}^{n}\boldsymbol{w}_i[a]^2$ 的范围证明，确保其不超过 B_{L2}^2。

4)梯度加密。u_i 对本地更新加密，获得加密上传数据：

$$\boldsymbol{k}_i=\boldsymbol{w}_i\oplus y_i g_i st_i\oplus \mathrm{PRG}(r_i) \tag{14}$$

式中：⊕表示将标量扩展为向量的形状进行相加。

5)哈希计算。u_i 计算加密上传数据的 LHH 值：

$$\mathrm{LHH}_i=\mathrm{LHH.Hash}(\boldsymbol{k}_i) \tag{15}$$

6)数据上传。u_i 将$\{\boldsymbol{k}_i,\mathrm{LHH}_i,\pi_i,\pi_{L\infty},\pi_{L2}\}$上传至其连接的 dc_j。

4.3.3 第 2 轮：收集器聚合

1)正确性预验证。dc_j 验证用户上传数据的正确性 $\mathrm{LHH.Hash}(\boldsymbol{k}_i)\overset{?}{=}\mathrm{LHH}_i$，去除验证不通过的用户，记验证通过的用户为 $U_{2,j}$，其中 $\sum_{j=1}^{t}U_{2,j}=U_2$。

2)数据预聚合。dc_j 聚合收到的加密数据：

$$\boldsymbol{K}_j=\sum \boldsymbol{k}_i \tag{16}$$

并计算其对应的 LHH 值：

$$\mathrm{LHH}_j=\mathrm{LHH.Hash}(\boldsymbol{K}_j) \tag{17}$$

3)数据上传。dc_j 将$\{\boldsymbol{K}_j,\mathrm{LHH}_j,U_{2,j},\pi_j\}$上传至 CS，其中，$\pi_j=\{\pi_1,\pi_2,\cdots,\pi_i\}$，$i\in U_{2,j}$。$dc_j$ 将 LHH_j 上传至 TA，TA 构建集合 $H_j=\{\mathrm{LHH}_1,\mathrm{LHH}_2,\cdots,\mathrm{LHH}_t\}$。

4.3.4 第 3 轮：服务器聚合

1)哈希验证。CS 对 dc_j 上传数据的正确性进行验证，计算 $\mathrm{LHH.Hash}(K_j)\overset{?}{=}\mathrm{LHH}_j$，丢弃验证不通过的 dc_j，记验证通过的 dc_j 为 DC_{pass}。

2)承诺验证。CS 对 u_i 上传数据的完整性进行验证 Com.veri(π_i)→(0,1)，即计算以下等式是否成立：

$$Com_i^{h_i}\,pk_{ui}=g^{\boldsymbol{w}_i h_i+\boldsymbol{sk}_{ui}}=g^{\boldsymbol{z}_i} \tag{18}$$

3)范围证明验证。CS 对 u_i 上传数据的安全性进行验证 RangeAgg.veri($\pi_{L\infty}$,π_{L2})→(0,1)，确保 u_i 的本地更新 $\boldsymbol{w}_i$ 在预定的合法范围内。

4)安全聚合。CS 聚合通过验证的数据：

$$\boldsymbol{K}=\sum_{j=1}^{t}\boldsymbol{K}_j=\sum_{i=1}^{m}\boldsymbol{k}_i=\sum_{i=1}^{m}\boldsymbol{w}_i\oplus\boldsymbol{C}\oplus\sum_{i=1}^{m}\mathrm{PRG}(r_i) \tag{19}$$

5)数据下发。CS 将 $\boldsymbol{K}$ 发送至 DC_{pass}。

4.3.5 第 4 轮：模型更新

1)正确性预验证。TA 首先将集合 H_j 发送至 DC_{pass}，其中的 dc_j 对 CS 的聚合结果进行正确性验证：

$$\mathrm{LHH.Hash}(\boldsymbol{K})\overset{?}{=}\mathrm{LHH.Eval}(\mathrm{LHH}_1,\mathrm{LHH}_2,\cdots,\mathrm{LHH}_t)$$

式中：LHH.Eval( )执行哈希值的线性组合。

若验证未通过，则中止本轮训练；若验证通过，则通知 CS 与 u_i。

2)掉线处理。CS 将 U_2 发送给 TA，TA 对比 U_1 与 U_2，得到 $U_{\mathrm{drop}}=U_2-U_1$，并计算掉线用户的噪声总和 M，并将 M 发送给 CS。

3)噪声去除。CS 去除聚合结果的噪声，得到 $\hat{\boldsymbol{W}}=\boldsymbol{K}\ominus C\oplus M\ominus\sum\mathrm{PRG}(r_i)$，并根据 FedAvg 算法计算出模型更新：

$$\boldsymbol{w}_{\mathrm{up}}=\frac{\hat{\boldsymbol{W}}}{|U_2|} \tag{20}$$

4)承诺生成。CS 计算：

$$\mathrm{AggCom.gen}(\pi_\tau,\tau,U_2,\boldsymbol{w}_{\mathrm{up}})$$

式中：π_τ 表示 U_2 中一个 $i=\tau$ 用户的相关参数，τ 为 TA 随机选取。

得到 π_{agg}：

$$\pi_{\mathrm{agg}}=\left\{\prod_{i\neq\tau}^{|U_2|}Com_i,\pi_\tau,U_2,\boldsymbol{w}_{\mathrm{up}}\right\} \tag{21}$$

5)更新全局模型。CS 将 π_{agg} 广播给 U_2 中的用户。

6)U_2 中的用户对服务器的更新 $\boldsymbol{w}_{\mathrm{up}}$ 进行验证 AggCom.veri(π_{agg})→{0,1}，即计算式(22)是否成立：

$$g^{\boldsymbol{w}_{\mathrm{up}}\cdot|U_2|}\,Com_\tau^{h_\tau-1}\,pk_\tau=g^{z_\tau}\prod_{i\neq\tau}^{|U_2|}Com_i \tag{22}$$

5 安全性分析

5.1 数据隐私性

上文提到,DC 和 CS 都是半诚实的,它们都会尝试从接收到的数据中推测包含用户隐私的信息。对于 DC 来说,它收到的加密数据包含了两重噪声加密,由于 U 生成噪声的秘密值 f_i 与随机数 r_i 分别由 TA 和 CS 生成,DC 对这两者都不可知,因此 DC 无法去除用户上传的加密数据中的噪声以窃取用户的相关隐私信息。对于 CS 来说,定义以下的游戏:

1)初始化:挑战者 C 生成 $F_1=\{f_1,f_2,\cdots,f_{2n}\}$,其中 $\gcd(f_i,f_j)=1,i,j\in[1,2n],i\neq j$。挑战者 C 计算 $F_2=\{f_1f_2,\cdots,f_{2n-1}f_{2n}\}$。对于任意 $i,j\in[1,2n],i\neq j,\gcd(f_{2i-1}f_{2i},f_{2j-1}f_{2j})=1$。敌手 A 从挑战者 C 处获取 $F=\prod_{i=1}^{2n}f_i$。

2)计算阶段 1:A 可以任意次地选择生成一个向量 $\boldsymbol{V}\in\mathbb{G}^d$,其中 $d<n$,并将其发送给 C。C 随机生成一个置换表 T。首先,C 对每个 f_i 都随机选取 $y_i<f_i$,并计算:

$$D=\sum_{i=1}^{2n}y_ig_ist_i \bmod F \tag{23}$$

$$x_i=D \bmod f_{2i-1}f_{2i} \tag{24}$$

式中,$i\in[1,n]$,$g_i=F/f_j$,$st_i\equiv g_i^{-1} \bmod f_i$。

其次,C 计算:

$$\boldsymbol{m}_{\text{mask}_0}=[1:d]\{T\times\{y_1g_1st_1,\cdots,y_{2n}g_{2n}st_{2n}\}\} \tag{25}$$

$$\boldsymbol{m}_{\text{mask}_1}=[1:d]\{T\times\{x_1h_1sd_1,\cdots,x_nh_nsd_n\}\} \tag{26}$$

式中:$g_i=F/f_j$,$i\in[1,2n]$,$j\in[1,n]$;$st_i\equiv g_i^{-1} \bmod f_i$;$h_j=F/(f_{2j-1}f_{2j})$;$sd_j\equiv h_j^{-1} \bmod f_{2j-1}f_{2j}$。

最后,C 将以下参数发送给 A:

$$\boldsymbol{V}_0=\boldsymbol{V}+\boldsymbol{m}_{\text{mask}_0},\boldsymbol{V}_1=\boldsymbol{V}+\boldsymbol{m}_{\text{mask}_1} \tag{27}$$

$$D=\sum_{i=1}^{2n}y_ig_ist_i \tag{28}$$

3)挑战阶段:C 生成一个向量 $\boldsymbol{M}\in\mathbb{G}^d$,并随机掷硬币 $b\in\{0,1\}$,然后向 A 发送 $\boldsymbol{M}_b$。

4)计算阶段 2:A 可以任意次地进行计算阶段 1 中的操作,获取不同的向量,并从 C 处获取带掩码的向量值。

5)猜测阶段:A 根据 $\{\boldsymbol{M}_b,D\}$ 猜测 b'。

定义 1 如果任意的多项式时间敌手 A 在选择性模型下不可区分性选择掩码攻击游戏中获胜的优势是可以忽略的,即:

$$\text{Adv}_{\text{scheme}}^{\text{Mask-IND}}(1^l)=\left|\Pr[b'\sim b]-\frac{1}{2}\right|\leqslant\text{negl}(l) \tag{29}$$

式中:$\text{negl}(l)$表示 l 中的一个可忽略函数。

因此,可以认为本文的加密方案是在选择性模型下具有不可区分性的选择掩码攻击安全的。

根据上述的游戏与定义,可以证明本文的加密方案是可以防止 CS 从加密数据中窃取用户的隐私信息的。从上述定义的游戏中可知,F_1 和 F_2 是 F 的两种因子分布的情况。C 向 A 发送 $\boldsymbol{M}_b$ 时,将根据 $F_1,F_2,\cdots,F_n$ 来生成加密的噪声。假设敌手赢得上述游戏的概率不可忽略,即敌手能够区分噪声的生成来源 F_i,但这与大整数分解的困难问题相矛盾。具体而言,根据 CRT 的特性,用户的加密上传数据 $\boldsymbol{k}_i=\boldsymbol{w}_i\oplus y_ig_ist_i\oplus\text{PRG}(r_i)$中,$y_ig_ist_i$ 的计算依赖于模数 f_i,若敌手能够从 $g_i=F/f_i$ 中求解 f_i,这等价于敌手完成了对大素数 F 的因子分解,然而这与大整数分解的数学困难问题相矛盾。因此,不存在敌手能够以不可忽略的概率赢得上述定义的游戏,本文的加密方案是可以防止服务器从加密数据中窃取用户隐私的。

5.2 数据完整性与正确性

上文提到,CS 除了会从加密数据中推测用户的隐私信息以外,还会有额外的恶意行为,即返回错误的聚合结果。本文方案采用了非交互式的承诺系统来实现 FL 中各实体间的互相验证。

本文基于 Schnorr 协议构造的 NIZK 系统的证据,形式为 $z_{\text{P}}=e_{\text{P}}\cdot r_{\text{P}}+sk_{\text{P}}$,其中,$r_{\text{P}}$ 是证明者 P 随机选取的数,$e_{\text{P}}=\text{Hash}(pk_{\text{P}},R_{\text{P}})$,$R_{\text{P}}=g^{r_{\text{A}}}$。接下来证明在 DLP 假设下该证据的构造是安全的:假设敌手 A 能进行哈希与证据询问。敌手 A 会按照以下步骤进行密钥消除攻击:

1)选择一个随机数 $r_{\text{P}*}$,并计算对应的 $R_{\text{P}*}$。

2)计算:

$$R_{\text{Pl}}=R_{\text{P}*}/pk_{\text{P}} \tag{30}$$

3)选定一个 $R_{\text{P}\cdot\text{guess}}$,发起哈希询问并获得:

$$e_{\text{P}\cdot\text{guess}}=\text{Hash}(pk_{\text{P}},R_{\text{P}\cdot\text{guess}}) \tag{31}$$

使得:

$$R_{\text{Pl}}=e_{\text{P}\cdot\text{guess}}\cdot R_{\text{P}\cdot\text{guess}} \tag{32}$$

4)伪造证据$\{R_{\text{P}\cdot\text{guess}},r_{\text{P}*}\}$并输出。

上述的 $R_{\text{P}\cdot\text{guess}}$ 只能由敌手 A 进行猜测,假设敌手 A 以均匀分布的概率随机抽取群上的元素进行哈希询问,那么对于一个确定的 R_{Pl} 值,敌手猜中 $R_{\text{P}\cdot\text{guess}}$ 并伪造证据通过验证的概率 $1/2^q$ 在 q 足够大时是可以忽略不计的。因此,以该形式构造的证

据在密钥消除攻击下是安全的。

此外，方案构造的非交互式证明系统包含了LHH，由于哈希函数的抗碰撞性，其证明也是不可伪造的，因此本文方案的证明系统能够保证数据的完整性与正确性。

5.3 分层随机抽样策略的安全性

上文提到，在生成 L∞范围证明时，采用分层随机抽样策略将显著降低开销。

抽样检测的安全性由检测失败的概率决定，记 $\mathcal{P}=\{P_1,P_2,\cdots,P_L\}$ 为本地更新的 L 层参数，对于每一层 $l\in[1,L]$，有 $P_l\in\mathbb{R}^{d_l}$，其中 d_l 为该层的参数数量。本地更新总参数量 $N=\sum_{l=1}^{L}d_l$。

引理 1（单层抽样保证） 对于任意层 l 中的参数集合 P_l，设 $N_l=|P_l|$ 为该层的参数总数，M_l 为越界参数的数量，$p_l=M_l/N_l$ 为越界参数的比例，n_l 为抽样数量，X_l 为抽样中发现的越界参数数量。

X_l 服从超几何分布 $H(N_l,M_l,n_l)$，其概率质量函数为：

$$P(X_l=k)=\frac{\binom{M_l}{k}\binom{N_l-M_l}{n_l-k}}{\binom{N_l}{n_l}} \tag{33}$$

定理 1（单层检测保证） 给定置信度 $1-\alpha$ 和误差容限 ε，当抽样数量满足：

$$n_l\geqslant\frac{n_0\cdot N_l}{n_0+N_l-1} \tag{34}$$

则：

$$P(|\hat{p}_l-p_l|\leqslant\varepsilon)\geqslant1-\alpha \tag{35}$$

式中：$z_{\alpha/2}$ 为标准正态分布的 $\alpha/2$ 分位数；$\hat{p}_l$ 为样本中观察到的越界比例；$n_0=\lceil z_{\alpha/2}^2\cdot p_l(1-p_l)\rceil/\varepsilon^2$。

证明 根据中心极限定理，$\hat{p}_l$ 近似服从正态分布，则对于给定的置信度 $1-\alpha$，有：

$$P\left(|\hat{p}_l-p_l|\leqslant z_{\alpha/2}\sqrt{\frac{p_l(1-p_l)}{n_l}\cdot\frac{N_l-n_l}{N_l-1}}\right)=1-\alpha \tag{36}$$

若误差不超过 ε，则：

$$z_{\alpha/2}\sqrt{\frac{p_l(1-p_l)}{n_l}\cdot\frac{N_l-n_l}{N_l-1}}\leqslant\varepsilon \tag{37}$$

那么可解得所需最小样本量：

$$n_l\geqslant\frac{n_0\cdot N_l}{n_0+N_l-1} \tag{38}$$

定理 2（多层检测保证） 对于多层抽样，因为每层独立进行抽样，且都满足定理 1，所以多层抽样满足：

$$P\left(\bigcap_{l=1}^{L}|\hat{p}_l-p_l|\leqslant\varepsilon\right)\geqslant\prod_{l=1}^{L}(1-\alpha)=(1-\alpha)^L \tag{39}$$

综上所述，当存在至少一层参数中的越界比例 $p_l\geqslant p_{\min}$ 时，分层抽样方案未能检测到任何越界参数的概率（即失败概率）满足：

$$P(\text{False})\leqslant\prod_{l=1}^{L}(1-p_l)^{n_l}\leqslant(1-p_{\min})^{\sum_{l=1}^{L}n_l} \tag{40}$$

6 实验分析

6.1 实验设置

本文方案的实验设置如下：采用单台电脑进行本地模拟，电脑配备 Intel i5-12400F 2.5 GHz CPU 以及 16 GB RAM，基于 Python 3.10 与 PyTorch 进行实验。实验采用了 CNN 模型与 Resnet-18 模型，其中 CNN 模型包含 2 层 5×5 的卷积层，使用 ReLU 作为激活函数，以及后续的全连接层与输出层；数据集采用了深度学习的经典数据集 MNIST 和 CIFAR-10。一些其他与实验相关的参数设置如表 2 所示，其中，Epoch 为每个用户在本地进行的迭代轮次，lr 为用户本地训练的学习率，Batch Size 为用户本地训练的批大小，E 为收集器的个数，N 为用户的个数。

表 2 参数设置

Table 2 Parameter setting

参数名	参数值
Epoch	5
lr	0.01
Batch Size	64
E	5
N	30

6.2 实验结果

本文的实验结果主要与 VREFL[34]、HFL[35]、RiseFL[36]、zkFL[37] 进行了对比。其中：VREFL 采用了掩码对模型参数进行加密，并基于 Merkle Tree 实现了可验证功能；HFL 则是未采用模型参数加密的方案；RiseFL 结合 Pedersen 承诺和可验证 Shamir 秘密共享设计了一个混合承诺机制，并基于 Σ-协议实现 L2 范数证明；zkFL 使用 zk-SNARK 生成零知识范围证明。上述方案的功能对比如表 3 所示，其中，“√”表示具备该功能，空白表示不具备该功能。此外，将本文方案与 VREFL、HFL 在不同模型与数据集下进行了训练准确率测试，并将本文

的承诺方案、零知识范围证明方案与 RiseFL、zkFL 进行了计算开销对比。

表 3 功能对比

Table 3 Function comparison

方案	可验证性	零知识证明	互相验证	范围证明
本文方案	√	√	√	√
VREFL	√			
HFL				
RiseFL	√	√		√
zkFL		√		√

6.2.1 服务器测试准确率

首先，本文基于 MNIST 数据集与 CNN 模型，分别在独立同分布(IID)与非独立同分布(Non-IID)数据划分的情况下进行了训练，并与 VREFL、HFL 进行了对比，绘制了服务器端的测试准确率的变化曲线，如图 2 与图 3 所示(彩色效果见《计算机工程》官网 HTML 版，下同)。

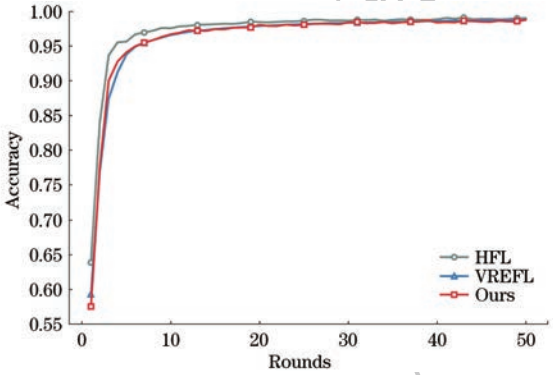

图 2 基于 CNN 与 MNIST IID 的测试准确率

Fig.2 Test accuracy based on CNN and MNIST IID

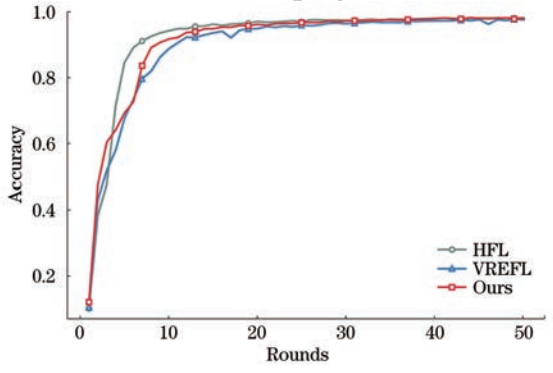

图 3 基于 CNN 与 MNIST Non-IID 的测试准确率

Fig.3 Test accuracy based on CNN and MNIST Non-IID

其次，本文基于 CIFAR-10 数据集与 Resnet-18 模型，分别在 IID 与 Non-IID 数据划分的情况下进行了训练，并与 VREFL、HFL 进行了对比，绘制了服务器端的测试准确率的变化曲线，如图 4 与图 5 所示。

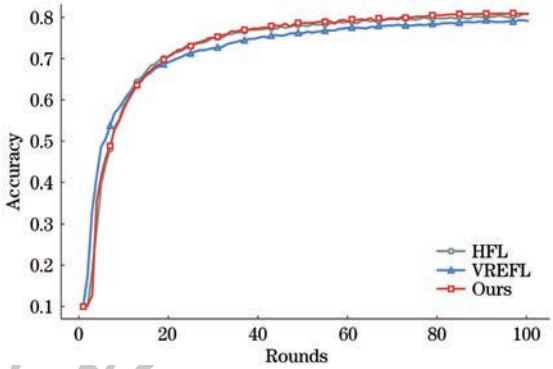

图 4 基于 Resnet-18 与 CIFAR-10 IID 的测试准确率

Fig.4 Test accuracy based on Resnet-18 and CIFAR-10 IID

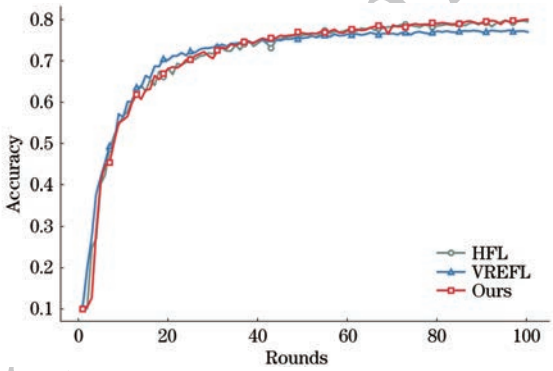

图 5 基于 Resnet-18 与 CIFAR-10 Non-IID 的测试准确率

Fig.5 Test accuracy based on Resnet-18 and CIFAR-10 Non-IID

实验结果表明，在基于 CNN 和 MNIST 以及基于 Resnet-18 和 CIFAR-10 两种实验方案下，无论是在 IID 还是 Non-IID 数据划分下，本文方案与其他两种方案相比，同样都能在 50 轮之后分别达到约 99%以及 80%的准确率，且收敛所需的轮数基本没有太大的差距。本文方案在上述测试准确率评估中均具有良好的模型性能，由此可以得知，本文方案在新增了非交互式可验证性与零知识范围证明的前提下，并不会影响 FL 任务整体的训练速度与效果，同样能在 FL 任务中达到较高的测试准确率。综上，本文方案并不会带来额外的训练速度与模型精度损失。

6.2.2 承诺协议开销

本节对承诺协议的计算开销与通信开销进行了实验，并分别从用户端和服务器端进行展示与分析。由于 VREFL 中并没有采用相似的承诺方案，且 HFL 不具备可验证功能，RiseFL 中设计了一种混合承诺机制，zkFL 中使用了 Pedersen 承诺，因此本节将着重评估本文方案与 RiseFL、zkFL 的开销。

本节的实验基于 CNN 模型分别在 MNIST 与 CIFAR-10 上进行，并采用 IID 数据划分。其中，用户在 CNN 上训练 MNIST 所得的本地更新含有

80 202 个参数，在 CNN 上训练 CIFAR-10 所得的本地更新含有 117 866 个参数。因为数据划分并不影响本地更新的模型参数量，所以本节实验仅在 IID 数据划分下进行。

图 6 展示了在不同安全参数设置下单个用户的承诺计算开销，图 7 展示了在不同安全参数设置下单个用户的通信开销。由图 6 的实验结果可知：在 MNIST 和 CIFAR-10 两种实验设置下，$k=1\ 024$ 时，承诺的计算开销为 0.488、0.633 s，仅比 $k=512$ 时分别增加了 0.129、0.163 s，但当 $k=2\ 048$ 时，计算开销分别大幅提升了 5、6 倍。图 7 的实验结果表明，通信开销随安全参数的变化规律与图 6 相似，$k=1\ 024$ 比 $k=512$ 的通信开销仅增加了不到 1 倍，而 $k=2\ 048$ 比 $k=1\ 024$ 的通信开销增加了 1 倍多，尤其在 CIFAR-10 上的增加更为显著。

因此，将安全参数设置为 $k=1\ 024$，能在确保安全性的同时，产生相对合理的计算与通信开销，达到比较好的效用平衡。后续的实验都将安全参数设置为 $k=1\ 024$。

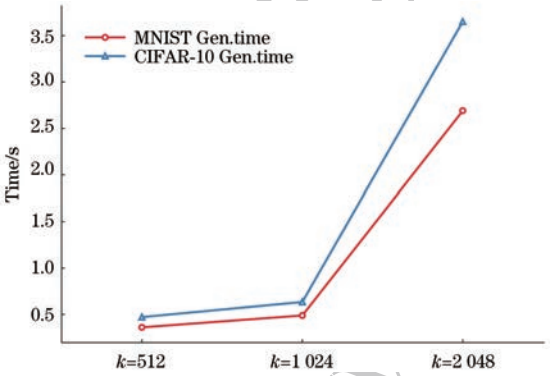

图 6 不同安全参数下的承诺生成开销

Fig. 6 Commitment generation overhead with different safe parameter

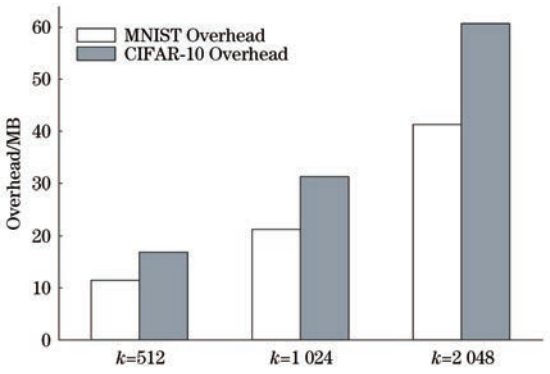

图 7 不同安全参数下的承诺通信开销

Fig. 7 Commitment communication overhead with different safe parameter

表 4 依次展示了用户生成承诺的开销、用户验证梯度更新承诺的开销、服务器生成梯度更新的承诺的开销以及服务器对所有用户上传的承诺进行验证的开销。其中，RiseFL 仅支持服务器对用户的单向验证，zkFL 中未涉及基于承诺的验证功能，“—”表示原文献中没有该指标值，下同。由表 4 的结果可知，本文方案不仅支持用户与服务器之间的相互验证，在开销上也具有一定的优势，本文的承诺方案在用户端带来的额外开销在可以接受的范围内，这对 FL 中的用户节点比较友好。

表 4 承诺计算开销

Table 4 Commitment compute overhead 单位：s

数据集	方案	User generation	User verification	Server generation	Server verification
MNIST	本文	0.49	11.99	10.96	78.25
	RiseFL	9.38	—	—	139.07
	zkFL	29.36	—	—	—
CIFAR-10	本文	0.63	19.40	17.29	95.78
	RiseFL	10.20	—	—	208.60
	zkFL	44.65	—	—	—

6.2.3 范围证明开销

本节对基于 Bulletproofs 的范围证明进行了实验，统计了在 MNIST 与 CIFAR-10 两个数据集下单个用户的范围证明生成与验证时间，如表 5 所示。之所以两种范围证明的开销相差较大，是因为两者基于的参数量差距较大。在应用本文提出的分层随机抽样策略之后（$\alpha=0.01$、$\varepsilon=0.01$），得到的参与生成 $\pi_{L\infty}$ 的参数数量在 $2^{10\sim12}$ 量级，而 π_{L2} 是对单个值进行证明。此外，RiseFL 基于 Σ-协议进行 L2 范数证明，zkFL 基于 zk-SNARK 生成零知识范围证明。

表 5 范围证明开销

Table 5 Range proof overhead 单位：s

数据集	方案	$\pi_{L\infty}$		π_{L2}	
		生成	验证	生成	验证
MNIST	本文	14.52	5.18	0.49	0.17
	RiseFL	—	—	135.86	207.80
	zkFL	476.98	309.23	—	—
CIFAR-10	本文	26.15	12.94	0.57	0.18
	RiseFL	—	—	137.00	207.95
	zkFL	717.20	438.32	—	—

得益于 Bulletproofs 的聚合证明能力，服务器可以将用户的范围证明进行聚合后再验证，与逐一验证相比，聚合验证的时间开销平均缩减了 60%，极大提高了验证效率。由表 5 的实验结果对比可知，本文方案在零知识证明的生成与验证开销上更

具有优势。

图 8 展示了采用分层随机抽样策略前后范围证明产生的通信开销对比。由于本地更新的参数量比较大，对其中所有参数进行范围证明将产生极大的开销，这对 FL 中的用户来说是不现实的。在应用分层随机抽样策略之后，由图 8 结果可知，范围证明的通信开销显著缩减，且缩减到了一个对用户合理可行的值。这证明本文的分层随机抽样策略具有现实意义。

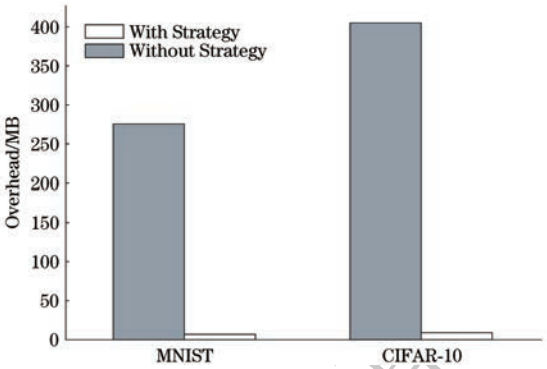

图 8 不同策略下的范围证明通信开销

Fig. 8 Range proof communication overhead with different strategy

6.2.4 投毒攻击与防御

本节对投毒攻击以及本文方案的拜占庭鲁棒性进行了实验。实验基于 CNN 模型分别在 MNIST 与 CIFAR-10 数据集的 IID 数据划分下进行。恶意用户的攻击目的在于破坏整体模型的训练效果，采用的投毒攻击方式有两种：一种是缩放攻击，攻击者会随机选取本地更新中的一些参数进行放大或者缩小；另一种是集中攻击，攻击者会将本地更新的参数集中到少量参数上。恶意用户将随机采取上述的攻击方式。

图 9 与图 10 展示了将恶意比例设置为 20%且不采取检测与防御措施时的效果，如图中红色曲线所示，在 20%的恶意用户攻击下，就能对整体的训练效果造成极大的破坏，导致训练最终无法收敛至预期准确率，因此可以认为在 30%与 40%的恶意比例下同样能造成有效攻击，图中仅展示恶意比例为 20%的攻击效果。在上述的有效攻击下，实验分别将恶意比例设置为 20%、30%、40%，记录了采取检测与防御措施后服务器的测试准确率变化曲线。实验结果表明，基于范围证明的恶意参数检测与防御方法能够准确检测并阻止恶意用户参与训练，在 MNIST 与 CIFAR-10 上训练的测试准确率分别能达到 99%与 80%左右，这证明了本文的防御方法在上述投毒攻击下的有效性。

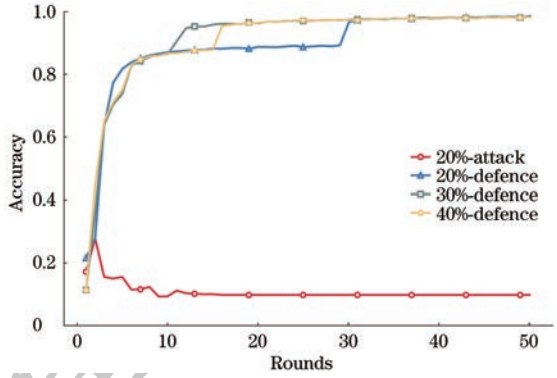

图 9 不同恶意比例下 MNIST 的测试准确率

Fig. 9 Test accuracy on MNIST with different malicious ratios

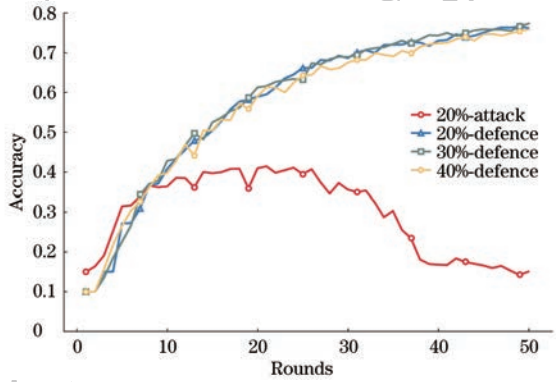

图 10 不同恶意比例下 CIFAR-10 的测试准确率

Fig. 10 Test accuracy on CIFAR-10 with different malicious ratios

7 结束语

本文基于非交互式 Schnorr 身份认证协议构造了一个承诺方案。首先，结合 LHH，实现了分层架构 FL 中各参与方的互相验证，确保 FL 训练全过程的可信。其次，基于 Bulletproofs 零知识证明设计了一个范围证明方案，并将其转化为非交互式，要求用户提供本地参数的范围证明，使服务器能够验证并识别恶意用户，确保本地参数的合法性，并提出了分层抽样策略，在确保安全性的同时显著优化范围证明带来的额外开销。最后，经过形式化的安全性分析，证明了本文的方案具备安全性，实验评估表明，本文的噪声掩码机制不影响 FL 的训练速度与精度，能以较高的效率进行相互验证与恶意检测，且恶意检测功能可以成功抵御训练过程中恶意用户的投毒攻击。未来的研究可以进一步优化零知识证明的计算与通信开销，以提高验证效率，同时可以探索针对更复杂攻击模式的安全防御策略，以提升系统的整体安全性。

参考文献

[1] LECUN Y, BENGIO Y, HINTON G. Deep learning[J].

Nature, 2015, 521(7553): 436-444.
[2] YU K H, BEAM A L, KOHANE I S. Artificial intelligence in healthcare [J]. Nature Biomedical Engineering, 2018, 2(10): 719-731.
[3] 江晟, 张仲义, 汪宗洋, 等. 基于改进 YOLOv7 的交通路口目标识别算法[J]. 吉林大学学报(理学版), 2024, 62(3): 665-673.
JIANG S, ZHANG Z Y, WANG Z Y, et al. Target recognition algorithm of traffic intersection based on improved YOLOv7 [J]. Journal of Jilin University (Science Edition), 2024, 62(3): 665-673. (in Chinese)
[4] AHISKA K, OZGOREN M K, LEBLEBICIOGLU M K. Autopilot design for vehicle cornering through icy roads [J]. IEEE Transactions on Vehicular Technology, 2017, 67(3): 1867-1880.
[5] GE Z Q, SONG Z H, DING S X, et al. Data mining and analytics in the process industry: the role of machine learning [J]. IEEE Access, 2017, 5: 20590-20616.
[6] KONEČNÝ J, MCMAHAN H B, RAMAGE D, et al. Federated optimization: distributed machine learning for on-device intelligence [EB/OL]. [2024-12-18]. https://arxiv.org/abs/1610.02527.
[7] ZHANG L, LI A R, PENG H Y, et al. Privacy-preserving data selection for horizontal and vertical federated learning [J]. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(11): 2054-2068.
[8] 王汝言, 陈伟, 张普宁, 等. 异构物联网下资源高效的分层协同联邦学习方法[J]. 电子与信息学报, 2023, 45(8): 2847-2855.
WANG R Y, CHEN W, ZHANG P N, et al. Resource-efficient hierarchical collaborative federated learning in heterogeneous Internet of Things [J]. Journal of Electronics & Information Technology, 2023, 45(8): 2847-2855. (in Chinese)
[9] LIM W Y B, NG J S, XIONG Z H, et al. Decentralized edge intelligence: a dynamic resource allocation framework for hierarchical federated learning [J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(3): 536-550.
[10] WANG Z B, SONG M K, ZHANG Z F, et al. Beyond inferring class representatives: user-level privacy leakage from federated learning [C] // Proceedings of the IEEE INFOCOM 2019. Washington D. C., USA: IEEE Press, 2019: 2512-2520.
[11] MALEKZADEH M, BOROVYKH A, GÜNDÜZ D. Honest-but-curious nets: sensitive attributes of private inputs can be secretly coded into the classifiers' outputs [C] // Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York, USA: ACM Press, 2021: 825-844.
[12] NGUYEN T, LAI P, TRAN K, et al. Active membership inference attack under local differential privacy in federated learning [EB/OL]. [2024-12-18]. https://arxiv.org/abs/2302.12685.
[13] CAO X Y, GONG N Z. MPAF: model poisoning attacks to federated learning based on fake clients [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Washington D. C., USA: IEEE Press, 2022: 3395-3403.
[14] HE X L, XU Y, ZHANG S C, et al. Enhance membership inference attacks in federated learning [J]. Computers & Security, 2024, 136: 103535.
[15] 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] // Proccedings of the European Interdisciplinary Cybersecurity Conference. New York, USA: ACM Press, 2022: 85-90.
[16] ZHANG C, EKANUT S, ZHEN L L, et al. Augmented multi-party computation against gradient leakage in federated learning [J]. IEEE Transactions on Big Data, 2024, 10(6): 742-751.
[17] CHAMIKARA M A P, LIU D X, CAMTEPE S, et al. Local differential privacy forFederated learning [C] // Proccedings of the European Symposium on Research in Computer Security. Berlin, Germany: Springer, 2022: 195-216.
[18] 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, USA: ACM Press, 2017: 1175-1191.
[19] GUO X J, LIU Z L, LI J, et al. VeriFL: communication-efficient and fast verifiable aggregation for federated learning [J]. IEEE Transactions on Information Forensics and Security, 2020, 16: 1736-1751.
[20] FU A M, ZHANG X L, XIONG N X, et al. VFL: a verifiable federated learning with privacy-preserving for big data in industrial IoT [J]. IEEE Transactions on Industrial Informatics, 2020, 18(5): 3316-3326.
[21] GAO S, LUO J J, ZHU J M, et al. VCD-FL: verifiable, collusion-resistant, and dynamic federated learning [J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 3760-3773.
[22] LI T, GAO C Z, JIANG L L, et al. Publicly verifiable privacy-preserving aggregation and its application in IoT [J]. Journal of Network and Computer Applications, 2019, 126: 39-44.
[23] HAHN C, KIM H, KIM M, et al. VerSA: verifiable secure aggregation for cross-device federated learning [J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(1): 36-52.
[24] 张首勋, 王玲玲, 耿克, 等. 融合零知识证明的去中心联邦学习可信聚合方案[J]. 计算机工程与应用, 2025, 61(8): 274-282.
ZHANG S X, WANG L L, GENG K, et al. Trustful aggregation scheme for decentralized federated learning based on zero knowledge proof [J]. Computer Engineering and Applications, 2025, 61(8): 274-282. (in Chinese)
[25] AHMADI M, NOURMOHAMMADI R. zkFDL: an efficient and privacy-preserving decentralized federated learning with zero knowledge proof [C] //Proceedings of the IEEE 3rd International Conference on AI in Cybersecurity (ICAIC). Washington D.C., USA: IEEE Press, 2024: 1-10.
[26] 魏立斐, 张无忌, 张蕾, 等. 基于本地差分隐私的异步横向联邦安全梯度聚合方案[J]. 电子与信息学报, 2024, 46(7): 3010-3018.
WEI L F, ZHANG W J, ZHANG L, et al. A secure gradient aggregation scheme based on local differential privacy in asynchronous horizontal federated learning [J]. Journal of Electronics & Information Technology, 2024, 46(7): 3010-3018. (in Chinese)
[27] LI X Y, QU Z, ZHAO S Q, et al. LoMar: a local defense against poisoning attack on federated learning [J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(1): 437-450.
[28] SUN Z T, KAIROUZ P, SURESH A T, et al. Can you really backdoor federated learning? [EB/OL]. [2024-12-18]. https://arxiv.org/abs/1911.07963.
[29] LYCKLAMA H, BURKHALTER L, VIAND A, et al. RoFL: robustness of secure federated learning [EB/OL]. [2024-12-18]. https://arxiv.org/abs/2107.03311.
[30] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C] // Proceedings of the International Conference on Artificial Intelligence and Statistics. [S. l.]:

PMLR, 2017: 1273-1282.
[31] 李顺东, 王道顺. 现代密码学: 理论, 方法与研究前沿[M]. 北京: 科学出版社, 2009.
LI S D, WANG D S. Modern cryptography: theory, methods and research frontiers [M]. Beijing: Science Press, 2009. (in Chinese)
[32] 杨波, 杨启良. 密码学中的可证明安全性[M]. 2 版. 北京: 清华大学出版社, 2024.
YANG B, YANG Q L. Provable security in cryptography [M]. 2nd ed. Beijing: Tsinghua University Press, 2024. (in Chinese)
[33] BÜNZ B, BOOTLE J, BONEH D, et al. Bulletproofs: short proofs for confidential transactions and more[C]//Proceedings of the IEEE Symposium on Security and Privacy (SP). Washington D.C., USA: IEEE Press, 2018: 315-334.
[34] YE H, LIU J Q, ZHEN H, et al. VREFL: verifiable and reconnection-efficient federated learning in IoT scenarios[J]. Journal of Network and Computer Applications, 2022, 207: 103486.
[35] LIU L M, ZHANG J, SONG S H, et al. Client-edge-cloud hierarchical federated learning[C]//Proceedings of the 2020 IEEE International Conference on Communications (ICC). Washington D.C., USA: IEEE Press, 2020: 1-6.
[36] ZHU Y Z, WU Y C, LUO Z J, et al. Secure and verifiable data collaboration with low-cost zero-knowledge proofs[EB/OL]. [2024-12-18]. https://arxiv.org/abs/2311.15310.
[37] WANG Z P, DONG N Q, SUN J H, et al. zkFL: zero-knowledge proof-based gradient aggregation for federated learning[J]. IEEE Transactions on Big Data, 2025, 11(2): 447-460.

文字编辑 陆燕菲
栏目编辑 赖玉玲