Reliable and Privacy-Preserving Federated Learning with Anomalous Users

ZHANG Weiting[1], LIANG Haotian[2], XU Yuhua[2], ZHANG Chuan[2]

(1. Beijing Jiaotong University, Beijing 100091, China;
2. Beijing Institute of Technology, Beijing 100081, China)

DOI: 10.12142/ZTECOM.202301003

https://kns.cnki.net/kcms/detail/34.1294.TN.20230210.1505.002.html, published online February 10, 2023

Manuscript received: 2022-11-01

Abstract: Recently, various privacy-preserving schemes have been proposed to resolve privacy issues in federated learning (FL). However, most of them ignore the fact that anomalous users holding low-quality data may reduce the accuracy of trained models. Although some existing works manage to solve this problem, they either lack privacy protection for users' sensitive information or introduce a two-cloud model that is difficult to find in reality. A reliable and privacy-preserving FL scheme named reliable and privacy-preserving federated learning (RPPFL) based on a single-cloud model is proposed. Specifically, inspired by the truth discovery technique, we design an approach to identify the user's reliability and thereby decrease the impact of anomalous users. In addition, an additively homomorphic cryptosystem is utilized to provide comprehensive privacy preservation (user's local gradient privacy and reliability privacy). We give rigorous theoretical analysis to show the security of RPPFL. Based on open datasets, we conduct extensive experiments to demonstrate that RPPEL compares favorably with existing works in terms of efficiency and accuracy.

Keywords: federated learning; anomalous user; privacy preservation; reliability; homomorphic cryptosystem

Citation (IEEE Format): W. T. Zhang, H. T. Liang, Y. H. Xu, et al., "Reliable and privacy-preserving federated learning with anomalous users," *ZTE Communications*, vol. 21, no. 1, pp. 15–24, Mar. 2023. doi: 10.12142/ZTECOM.202301003.

1 Introduction

With the popularity of big data techniques, machine learning has promoted wide applications in artificial intelligence fields, such as the smart IoT[1-2], smart industry[3-4], and autonomous driving[5-6]. Nowadays, due to the emergence of data protection regulations, like General Data Protection Regulation (GDPR)[7] and California Consumer Privacy Act (CCPA)[8], users pay increasing attention to data privacy. Data privacy significantly hinders training data collection, which limits the development of machine learning. Federated learning (FL), as a collaborative machine learning paradigm, is considered a promising solution to the challenges and has attracted tremendous attention from industry and academia. Specifically, a typical framework of FL consists of a server and some users (i.e., data owners). In FL, to preserve data privacy, users only share the trained local models' parameters instead of sharing raw data.

In spite of the benefits, there are two challenges in designing such an FL scheme. The first one is that the gradient attack may lead to privacy leakage. Specifically, in the gradient attack, adversaries utilize user-shared model parameters to infer sensitive information from training data. Thus far, some works[9-10] have been proposed to utilize the gradient leak attack to compromise user privacy. For instance, ZHU et al.[10] introduced a gradient inversion attack scheme to reconstruct sensitive information from public shared gradients, where adversaries launch attacks by iteratively optimizing the dummy inputs and the corresponding labels. Followed by Ref. [10], some gradient attack schemes have been proposed[11-12]. For instance, to enhance the performance of gradient inversion attacks, ZHAO et al.[11] proposed a simple and effective gradient inversion attack. Their scheme improves the effectiveness of recovering label information by combining the mathematical analysis of the gradients. Subsequently, YIN et al.[12] extended the gradient inversion attack into FL applications that are more practical, e.g., high-resolution images with large batch-size. If gradient attacks are not considered well in designing FL schemes, user privacy will incur serious threats. Therefore, users will be reluctant to participate in these applications, which significantly hinders the development of FL. The sec-

This work was supported in part by the Fundamental Research Funds for Central Universities under Grant No.2022RC006, in part by the National Natural Science Foundation of China under Grant Nos.62201029 and 62202051, in part by the BIT Research and Innovation Promoting Project under Grant No. 2022YCXZ031, in part by the Shandong Provincial Key Research and Development Program under Grant No. 2021CXGC010106, and in part by the China Postdoctoral Science Foundation under Grant Nos.2021M700435, 2021TQ0042, 2021TQ0041, BX20220029 and 2022M710007.
ZHANG Weiting and LIANG Haotian contribute equally in this work.
Corresponding author: ZHANG Chuan

ond challenge is that users with low-quality data decrease the performance of FL. In practical applications, the data quality of different users is usually uneven due to various reasons (e.g., device quality and education level)[13]. For example, users with high-quality devices usually own superior data, while users with low-quality devices have poorer data. If anomalous users are not identified in the training process, they will impair the performance of FL and even lead to the unavailability of FL models. Thus, it is also crucial to identify anomalous users and reduce their negative influence on the FL training process.

In recent years, to deal with the gradient attacks and preserve user privacy in FL, some solutions[14-16] have been proposed. Particularly, based on their cryptographic tools, these schemes can be categorized into three classes, i.e., secure multi-party computation (SMC) based schemes, homomorphic encryption (HE) based schemes, and differential privacy (DP) based schemes. DP-based FL schemes address the privacy leakage issues by adding noise[14]. However, the introduction of noise unavoidably reduces the model accuracy, hindering the applications of FL. To preserve user privacy, some SMC-based schemes[15] are proposed without compromising model accuracy. However, frequent user interaction introduces tremendous resource overhead to users and the server. To make a trade-off among the model's accuracy, user privacy, and resource overhead, some HE-based FL schemes are proposed[16].

Unfortunately, most existing privacy-preserving FL schemes ignore anomalous users. To address the challenge, several works[17-18] have been proposed to identify anomalous users and reduce their impacts. Specifically, ZHAO et al.[17] utilized the differential privacy technique and function mechanism to enable privacy-preserving FL. In their scheme, the server is allowed to access each user's data quality for identifying anomalous users. However, in practice, the user's data quality should be private. Once the data quality is disclosed to the server, it will lead to discrimination in the training process, which significantly reduces the users' enthusiasm to participate in FL. To preserve data quality information when identifying anomalous users, XU et al.[18] designed a framework to support privacy-preserving FL by introducing a non-colluding two-cloud model. In their scheme, additively homomorphic cryptosystem and YAO's garbled circuits are utilized to evaluate user data quality without compromising user privacy. It is hard to find two non-colluding clouds in practice, thereby limiting its implementation in real-world applications. Moreover, it also ignores the problem of user collusion. In FL, users may collude with each other to infer others' sensitive information. Therefore, a privacy-preserving FL scheme with anomalous user identification and user collusion resistance deserves to be investigated.

To solve the challenges, we propose a reliable and privacy-preserving FL (RPPFL) scheme based on the single-cloud model. The comparison results of RPPFL and other existing works are shown in Table 1. To identify anomalous users, RPPFL evaluates data quality without compromising user privacy. Particularly, we epitomize the contributions as follows:

▼Table 1. Comparison of RPPFL and other existing works

	User Privacy Preservation	Robust to User Instability	Support for Anomalous Users	Collusion Resistance	Server Setting
PPDL[16]	√	×	×	×	Single-cloud
PPML[19]	√	√	×	√	Single-cloud
SecProbe[17]	√	×	√	√	Single-cloud
PPFDL[18]	√	√	√	×	Two non-colluding clouds
RPPFL	√	√	√	√	Single-cloud

PPDL: privacy-preserving deep learning
PPFDL: privacy-preserving federated deep learning
PPML: privacy-preserving machine learning
RPPFL: reliable and privacy-preserving federated learning

- We first discover the challenges in designing a privacy-preserving FL scheme that supports anomalous identification. Then, to resolve these challenges, we design a reliable and privacy-preserving FL scheme named RPPFL, which is also resilient to user collusion attacks.
- We adopt the truth discovery technique to evaluate data quality. Subsequently, we utilize the (p, t) threshold Paillier cryptosystem to strengthen RPPFL to protect user privacy from being exposed and defend against user collusion attacks.
- Formal analysis proves the security of RPPFL. Then, based on the open datasets MNIST and CIFAR-10, extensive experiments are conducted to demonstrate that RPPFL is practically efficient and effective.

In this paper, the remainder is established as follows. In the next section, we illustrate the related models and security requirements of our construction. The preliminaries are reviewed in Section 3, and the detailed construction is presented in Section 4. Section 5 provides the security analysis. The experiments are given in Section 6, and Section 7 discusses the related works. Section 8 concludes the paper.

2 Models and Security Requirements

We first present the system model and threat model of RPPFL. After that, based on the threat model, we give the security requirements. To have a better understanding, we list some frequently used notations that appear in RPPFL, which is shown in Table 2.

2.1 System Model

As we can see in Fig. 1, the system model of RPPFL consists of an aggregation server and several users.

- The aggregation server is an entity with strong computing and storage capabilities. To reduce the anomalous users' negative impacts on the accuracy of the model, the aggregation server is allowed to identify users' data quality (i.e., user reliability). Then, with the user's reliability and local gradients, the aggregation server aggregates the global gradients in a privacy-preserving manner. Subsequently, global gradients

will be sent to the users.

• The users are entities holding different datasets that can be utilized to train FL models. To get models with better performance, they cooperate in training models with the help of an aggregation server. Instead of sharing datasets directly, they share the gradients of local models. To protect gradient privacy, users first encrypt local gradients with an additively homomorphic cryptosystem. Then, users send them to the aggregation server and update local models after receiving global gradients from the aggregation server.

2.2 Threat Model

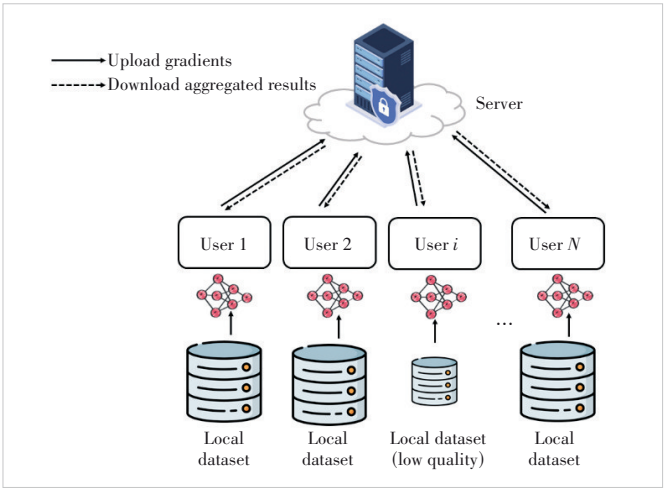
In our scenario, like previous works[20-21], we presume that the aggregation server and all users are honest-but-curious. That is, the server will faithfully obey the designed procedures to accomplish its task. However, it may try to retrieve others' sensitive information using prior acquired knowledge. Besides, we presume that the aggregation server will not collude with users and there are at most $t-1$ users colluding. Then, we mainly consider the following two adversaries.

1) The aggregation server may try to deduce users' local gradients and reliability according to the information it acquired.

2) The user may try to infer the information of his/her reliabilities according to the information he/she acquired.

▼Table 2. Frequently used notations

Notation	Meaning
n	A large positive integer
$\mathbb{Z}_n$	The set of integers modulo n
$\mathbb{Z}_n^*$	The multiplicative group of reversible elements of $\mathbb{Z}_n$
N	The number of users
K	The number of the selected users
M	The number of gradient types
M_f	A big integer of the magnitude of 10
x_m^k	The m-th gradient of the k-th user
$\widetilde{x_m^k}$	The integer corresponding to the enlargement of x_m^k
x_m^*	The aggregated result of the m-th gradient
R_k	The reliability (indicates the data quality) of the user k
$\mathbb{C}$	The coefficient used to amplify users' reliability
sk_k	The secret key of the selected user k
sk_{N+1}	The secret key of the aggregation server
$\mathrm{Enc}_{\mathrm{pk}}(\cdot)$	The ciphertext encrypted by a public key
r_k	The random value selected by the user k



▲Figure 1. System model of reliable and privacy-preserving federated learning (RPPFL)

2.3 Security Requirements

On the basis of system and threat models, we have developed the following security requirements.

1) User's local gradient privacy. To effectively preserve user privacy, the user's local gradients should be sent to the aggregation server in the ciphertext, which prevents the adversary (e.g., the server) from recovering the user's sensitive information from the shared gradients and global parameters.

2) Privacy protection of reliability for users. To ensure the fairness of the FL process, all information related to the reliability of the user should be kept secret and unavailable to any participant, even to the user itself.

3 Preliminaries

In this section, we will illustrate the preliminaries about truth discovery, FL, and the additively homomorphic cryptosystem.

3.1 Truth Discovery

Truth discovery aims at estimating ground truth data from numerous heterogeneous data. In general, it is composed of two main steps: weight update and truth update.

1) Weight update

In this step, the weight of each user is computed based on the distance between their provided data and the ground truths. Without losing generality, we here assume the ground truths are fixed. Typically, each user's weight w_k can be computed as $w_k = f\left(\sum_{m=1}^{M} d(x_m^k, \boldsymbol{x}_m^*)\right)$, where f denotes a monotonically decreasing function, and $d(x_m^k, \boldsymbol{x}_m^*)$ is a distance function (i.e., the Euclidean distance). Therefore, if the provided data from a specific user are close to the ground truth, the user's weight will be assigned to a higher value.

2) Truth update

In this step, on the basis of each user's weight, the ground truth is estimated according to Eq. (1):

$$\boldsymbol{x}_m^* = \frac{\sum_{k=1}^{K} x_m^k \cdot w_k}{\sum_{k=1}^{K} w_k}. \tag{1}$$

In the case of continuous data, $\boldsymbol{x}_m^*$ means the estimated

ground truth. As for the categorical data, $\boldsymbol{x}_m^*$ represents a probability vector. Each element in the vector means the probability of a specific answer being the truth[22].

3.2 Federated Learning

As a collaborative learning paradigm, FL intends to train models based on data from distributed users. The basic training process of FL is shown below.

1) Selecting users

Assume there exist N users, each holding a local dataset $\mathcal{D}_j, j \in [1,N]$, which is derived from the whole training dataset $\mathcal{D}=\{(u_i,v_i); i = 1,2,\cdots,M\}$, where $\mathcal{D} = \bigcup_{j \in [1,N]} \mathcal{D}_j$. For each epoch $t \in \{1,2,\cdots\}$ in FL, the aggregation server chooses K users at random, where $K < N$.

2) Local training

Each selected user $k, k \in [1, K]$, randomly chooses a small batch of dataset B^k. Then, they leverage stochastic gradient descent (SGD), a commonly used optimization algorithm, to calculate gradients over their local datasets. Specifically, we let $\boldsymbol{u}_i^k$ and $\boldsymbol{v}_i^k$ denote the feature vector and its corresponding label in B^k, respectively, and θ_t^k denotes the parameters of the model in the current epoch. The loss function, indicating the distance between prediction results and real labels, can be denoted as $L(\theta_t^k,\boldsymbol{u}_i^k,\boldsymbol{v}_i^k)$. Then, the gradient can be calculated as Eq. (2):

$$\nabla_{\theta_t^k} = \nabla L\left(B^k,\theta_t^k\right) = \frac{\sum_{<u_i,v_i> \in B^k} \nabla L\left(\theta_t^k,\boldsymbol{u}_i^k,\boldsymbol{v}_i^k\right)}{\left|B^k\right|}. \tag{2}$$

After that, $\nabla_{\theta_t^k}$ will be transmitted to the aggregation server.

3) Global aggregation

After receiving local gradients from all selected users, the aggregation server will aggregate the global gradients as Eq. (3):

$$\text{Global} = \frac{\sum_{k=1}^{K} \nabla_{\theta_t^k}}{K}. \tag{3}$$

Finally, the global gradients will be transmitted to the users to update their local model as:

$$\theta_{t+1}^k = \theta_t^k - \eta \cdot \text{Global}, \tag{4}$$

where η denotes the learning rate.

3.3 Additively Homomorphic Cryptosystem

The cryptosystem in RPPFL is on the basis of the (p,t)-threshold Paillier cryptosystem[22]. As a typical asymmetric cryptosystem, it utilizes the public key (pk) to encrypt the plaintexts and secret key (sk) to recover the plaintexts. Note that (p,t)-threshold Paillier cryptosystem splits the secret key into p parts, i.e., $(\text{sk}_1,\text{sk}_2,\cdots,\text{sk}_p)$, and sends them to p different parties. In (p,t)-threshold Paillier cryptosystem-based applications, any entity cannot decrypt the ciphertexts alone. That is, the ciphertext can only be decrypted if at least t entities cooperate together. Moreover, even if some users are dropped off during the process because of the insatiability, the ciphertext can still be recovered.

We use $\text{Enc}_{\text{pk}}(\cdot)$ to denote the ciphertexts encrypted by the public key. Then, assuming $m \in \mathbb{Z}_n$ denotes a plaintext, its corresponding ciphertext can be calculated as follows:

$$C = \text{Enc}_{\text{pk}}(m) = g^m r^n \bmod n^2, \tag{5}$$

where $r \in \mathbb{Z}_n^*$ is a randomly selected value and should be kept secret. For decryption, each party l, $l \in [1,p]$, requires to compute the partial decryption c_l according to Eq. (6) with the secret key sk_l,

$$c_l = c^{2\Delta sk_l}, \tag{6}$$

where we denote $\Delta = p!$. Based on the algorithm in Ref. [23], these partial decryptions can be composed together for decrypting the ciphertext C in order to recover the plaintext m.

Then, we further present additively homomorphic properties of our adapted cryptosystem. Specifically, given the ciphertexts of two plaintexts, $m_1,m_2 \in \mathbb{Z}_n$ are encrypted with the same public key:

$$\begin{aligned} C_1 &= \text{Enc}_{\text{pk}}(m_1) = g^{m_1} r_1^{\ n} \bmod n^2, \\ C_2 &= \text{Enc}_{\text{pk}}(m_2) = g^{m_2} r_2^{\ n} \bmod n^2. \end{aligned} \tag{7}$$

We have

$$\begin{aligned} \text{Enc}_{\text{pk}}(m_1 + m_2) &= \text{Enc}_{\text{pk}}(m_1) \cdot \text{Enc}_{\text{pk}}(m_2) \\ &= g^{m_1 + m_2}(r_1 r_2)^n \bmod n^2, \end{aligned} \tag{8}$$

$$\text{Enc}_{\text{pk}}(b \cdot m_1) = \text{Enc}_{\text{pk}}(m_1)^b = g^{bm_1} r_1^{bn} \bmod n^2, \tag{9}$$

where b denotes a constant.

4 Scheme Design and Details

In this section, we first illustrate the approach that we utilize to handle anomalous users. Then, we give the details of our proposed RPPFL.

4.1 Approach to Handling Anomalous Users

To decrease the negative influence of anomalous users on the trained model in federation learning, here we describe the mechanism Me_{AU}, which is inspired by the truth discovery[24]. In RPPFL, we assume that the data from different users are independently and equally distributed. We assume that each user holds M categories of gradients (in Section 3.2) after train-

ing on their local dataset. The m-th gradient of the k-th user can be represented as x_m^k, where $m \in [1, M], k \in [1, K]$. We use $\boldsymbol{x}_m^*$ to denote the global m-th gradient of K selected users. Additionally, we let R_k represent the reliability (indicates the data quality) of the user k. Me_{AU} mainly includes two phases: updating the user's reliability and updating global gradients.

1) Update user's reliability

The user's reliability will be given a high value when the calculated gradient is close to the global gradient from the server. Specifically, given the global gradient $\boldsymbol{x}_m^*$, the reliability of user k is calculated as follows:

$$R_k = f\left(\sum_{m=1}^{M} d\left(x_m^k, x_m^*\right)\right), \tag{10}$$

where f denotes a monotonically decreasing function, and $d(\cdot)$ denotes a function that measures the distance between the local gradients and global gradients. In RPPFL, we use the same method as in Ref. [18], and formulate Eq. (10) as:

$$R_k = \frac{\mathbb{C}}{\sum_{m=1}^{M} d\left(x_m^k, x_m^*\right)}, \tag{11}$$

where $\mathbb{C}$ is used to amplify users' reliability, which is calculated according to Eq. (12):

$$\mathrm{C} = \chi^2_{\left(1-\frac{\alpha}{2}, |M|\right)}, \tag{12}$$

where χ denotes the Chi-squared distribution, and α represents its corresponding significance level. It is noteworthy that when the value of α and M (the number of gradients) is determined, the coefficient $\mathbb{C}$ can be regarded as a constant. On the basis of some proposed works[18, 25–26], for users with high-quality data for training, the obtained gradients are always consistent in the direction of the vector with high probability. To guarantee the convergence of training, the direction of the local gradient x_m^k is always required as the same with the global gradient x_m^*. Thus, we compute $d(x_m^k, x_m^*) = (x_m^k - x_m^*)^2$ if x_m^k and x_m^* are both positive or negative. If not, we set $d(x_m^k, x_m^*)$ to a large positive integer (illustrated in Section 4.2).

2) Update global gradients

With the reliability of each user given, the aggregated result of m-gradient is calculated as

$$x_m^* = \frac{\sum_{k=1}^{K} R_k x_m^k}{\sum_{k=1}^{K} R_k}. \tag{13}$$

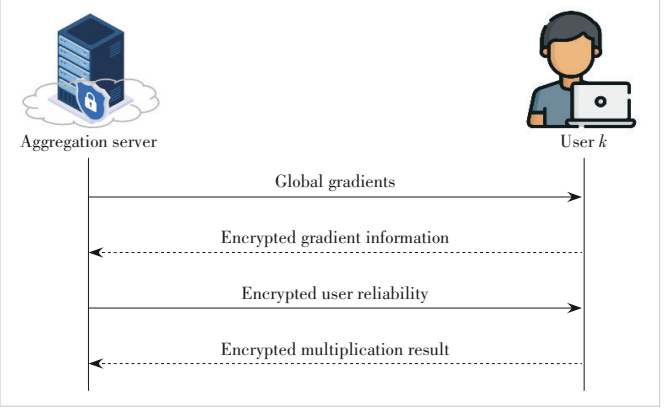
Note that we do not directly remove these anomalous users. The reason is that the reliability information is kept secret from all participants, even the users themselves, to prevent discrimination during the training phase. The existence of low-quality data is inevitable. In some rare cases where all users are normal, there is still the possibility that the trained model will be overfitted in the actual prediction. Based on the above facts, RPPFL tolerates gradients from anomalous users but ensures that the global gradients are mainly contributed by normal users. However, ensuring that each participant in federated learning is unaware of users' reliability will inevitably increase the difficulty of reducing the impacts of low-quality data.

4.2 Reliable and Privacy-Preserving Federated Learning

As shown below, we first briefly summarize the main process of RPPFL, i.e., reliability identification and gradient aggregation, and then give its details. The workflow of RPPFL is displayed in Fig. 2, and the protocol framework is shown as Protocol 1. We assume that a trusted third party (TTP) has executed the (p,t)-threshold Paillier cryptosystem before running the reliable and privacy-preserving federated learning protocol, where $p = N + 1$ and $t = K + 1$. The secret keys $(\mathrm{sk}_1, \mathrm{sk}_2, \cdots, \mathrm{sk}_N)$ are sent to N different users, respectively, and sk_{N+1} is sent to the aggregation server. Besides, the public key is distributed to all entities.

• Reliability identification. In this step, each selected user first calculates the Euclidean distance between its local gradients and the global gradients from the aggregation server. These calculation results will be encrypted using the public key and then transmitted to the aggregation server. With these ciphertexts, the aggregation server calculates the reliability of each user while protecting data privacy. Ultimately, the encrypted reliability will be sent to the corresponding user for the following procedure.

• Gradient aggregation. In this phase, each user calculates the product of their gradient and reliability in the encryption domain. These ciphertexts are transmitted to the server. With the help of K selected users, the server decrypts these received ciphertexts and subsequently updates



▲ **Figure 2. Workflow of reliable and privacy-preserving federated learning (RPPFL)**

the global models.

Note that the additively homomorphic cryptosystem is defined over the integer ring. However, the gradient often consists of many floating-point numbers in real-world federated learning. We define a big integer M_f, which is a magnitude of 10. Before utilizing homomorphic encryption on the gradient x_m^k, we calculate $\lfloor M_f \cdot x_m^k \rfloor$, which we denote as $\widetilde{x_m^k}$. $\widetilde{x_m^k}$ is the rounded version of the gradient for encryption, and the original approximated result can be easily recovered by simply dividing $\widetilde{x_m^k}$ with M_f. Unless otherwise mentioned, we also use this format to represent other rounded values in the remaining parts of the paper. Then, for each negative integer x_m^k, we use the trick adopted in Ref. [27] by simply replacing it with its inverse in the cryptosystem.

The update of the global models in federated learning lasts for several iterations. Here, we give the calculation procedure in one of the iterations.

1) Reliability identification

Step 1: The aggregation server first selects K users and sends the global gradient $\{x_m^*\}_{m=1}^M$ to them. If it is in the first iteration, $\{x_m^*\}_{m=1}^M$ is the random value initialized by the aggregation server; otherwise, $\{x_m^*\}_{m=1}^M$ is derived in the previous iteration. Upon receiving $\{x_m^*\}_{m=1}^M$, the user $k, k \in [1,K]$, calculates:

$$\mathrm{D} = \sum_{m=1}^{M} d\left(x_m^k, x_m^*\right), \tag{14}$$

and obtains its reciprocal, i.e., $\mathbb{D}^{-1}$. Then, to preserve the privacy of $\mathbb{D}^{-1}$, the user $k, k \in [1,K]$, chooses a random value $r_k \in \mathbb{Z}_n^*$ and encrypts it as follows:

$$\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right) = g^{\widetilde{\mathbb{D}^{-1}}} r_k^n \bmod n^2. \tag{15}$$

When the encryption is completed, each user sends $\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right)$ to the aggregation server.

Step 2: After receiving $\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right)$ from all selected K users, the aggregation server calculates the reliability of each user in ciphertexts as

$$\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right) = \mathrm{Enc}_{\mathrm{pk}}\left(\left\lfloor M_f \cdot \frac{1}{\mathbb{D}} \right\rfloor \cdot \left\lfloor M_f \cdot \mathbb{C} \right\rfloor\right) = \mathrm{Enc}_{\mathrm{pk}}\left(\left\lfloor M_f \cdot \frac{1}{\mathbb{D}} \right\rfloor\right)^{\left\lfloor M_f \cdot \mathbb{C} \right\rfloor} = g^{\tilde{\mathbb{C}} \widetilde{\frac{1}{\mathbb{D}}}} r_k^{\tilde{\mathbb{C}} n} \bmod n^2, \tag{16}$$

where the aggregation server calculates $\mathbb{C}$ and keeps it secretly. $\left[\widetilde{\cdot}\right]$ denotes the product of two rounded values. After that, the aggregation server transmits the encrypted reliability $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)$ to user $k, k \in [1,K]$.

Protocol 1. Reliable and privacy-preserving federated learning

Input:

K selected users, M types of gradients, local gradients $\{x_m^k\}_{m,k=1}^{M,K}$, initialized global gradients $\{x_m^*\}_{m=1}^M$, and coefficient $\mathbb{C}$

Output:

Global gradients $\{x_m^*\}_{m=1}^M$

1. The aggregation server sends $\{x_m^*\}_{m=1}^M$ to each user k.
2. Each user k computes the local gradients.
3. Each user k computes $\mathrm{Enc}_{\mathrm{pk}}(\mathbb{D}^{-1})$, where $\mathbb{D}^{-1} = 1/\sum_{m=1}^{M} d(x_m^k, x_m^*)$.
4. Each user k sends $\mathrm{Enc}_{\mathrm{pk}}(\mathbb{D}^{-1})$ to the aggregation server.
5. The aggregation server computes $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}])$ for each user k.
6. The aggregation server sends $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}])$ back to each user k.
7. Each user k computes the product of local gradients and their reliability $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}] \cdot \widetilde{x_m^k})$, $m \in [1,M]$.
8. Each user k sends $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}] \cdot \widetilde{x_m^k})$, $m \in [1,M]$ to the aggregation server.
9. The aggregation server computes $\mathrm{Enc}_{\mathrm{Global}}$ and $\mathrm{Enc}_{\mathrm{pk}}\left([\sum_{k=1}^{K} \widetilde{R_k}]\right)$.
10. The aggregation server computes $\{x_m^*\}_{m=1}^M$ according to Eqs. (19) and (20).
11. Repeat steps 3 – 7 until the convergence criteria in FL is reached.

2) Gradient aggregation

Once the reliability of each user has been obtained, the next step is to update the global gradients according to the reliability and local gradients of all selected users.

Step 1: After receiving $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)$ from the aggregation server, the user k calculates the product of local gradients and their reliability in ciphertexts

$$\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right) = \mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)^{\widetilde{x_m^k}} = g^{\widetilde{x_m^k}\left[\widetilde{R_k}\right]} r_k^{\widetilde{x_m^k} n} \bmod n^2. \tag{17}$$

Then, $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)$ will be transmitted to the aggregation server.

Step 2: When the aggregation server receives the ciphertexts $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)$, $k \in [1,K]$, from all selected users, it aggregates them in ciphertexts according to the homomorphic property of the (p,t)-threshold Paillier cryptosystem.

$$\mathrm{Enc}_{\mathrm{Global}} = \prod_{k=1}^{K} \mathrm{Enc}_{pk}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right) = g^{\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)}\left(\prod_{k=1}^{K} r_k\right)^n \bmod n^2 = \mathrm{Enc}_{pk}\left(\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)\right). \tag{18}$$

After that, $\mathrm{Enc}_{\mathrm{Global}}$ is sent to K selected users. Each user k uses their secret key sk_k to partially decrypts $\mathrm{Enc}_{\mathrm{Global}}$ and then sends them to the aggregation server. The aggregation server first obtains the partial decryption with its secret key sk_{N+1}. Then, based on $K+1$ partially decrypted ciphertexts, the aggregation server recovers the plaintexts $\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)$. Similarly, the aggregation server can also calculate the summation of each user's reliability, i. e., $\sum_{k=1}^{K}\left[\widetilde{R_k}\right]$. Therefore, the global gradients can be updated as:

$$\widetilde{x_m^*} = \frac{\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)}{\sum_{k=1}^{K}\left[\widetilde{R_k}\right]}, \tag{19}$$

which will be sent to K users to update their local models. Note that x_m^* can be recovered by calculating

$$x_m^* = \left\lfloor \widetilde{x_m^*} / \left(M_f\right) \right\rfloor. \tag{20}$$

Reliability identification and gradient aggregation are performed iteratively until the convergence criteria are fulfilled.

5 Security Analysis

Based on the threat model in Section 2.2, the potential threats mainly come from the entities (i.e., users and the aggregation server). Thus, the objective of RPPFL is to protect the user's local gradient and the user's reliability from being exposed to any entity in RPPFL. Furthermore, it should also be resilient to the user collusion attack. Here, we prove the security of RPPFL by giving Theorem 1, followed by the corresponding proof.

Theorem 1. Assuming that the aggregation server is non-colluding with users and there are at most $t-1$ users colluding, neither the user's local gradient nor the user's reliability will be leaked to any entity in RPPFL.

Proof. First, we prove that each user cannot infer their own reliability from the information they have acquired and the ciphertexts returned by the aggregation server. Next, we show that the aggregation server cannot infer each user's local gradient and reliability from the information it holds and the ciphertexts returned by the user.

The user knows the ciphertexts $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)$, $\mathrm{Enc}_{\mathrm{Global}}$, and plaintexts $\{x_m^*\}_{m=1}^{M}$, $\mathbb{D} = \sum_{m=1}^{M} d\left(x_m^k, x_m^*\right)$. Since there are at most $t-1$ users colluding, the user cannot recover the secret key (sk), from sk_k. Additionally, the (p,t)-threshold Paillier cryptosystem has already been demonstrated to defend against chosen-plaintext attacks[22]. Therefore, the user cannot decrypt these ciphertexts. With the global gradient $\{x_m^*\}_{m=1}^{M}$, the user calculates $\mathbb{D}$ locally. However, since $\mathbb{C}$ is only known by the aggregation server. Without knowing $\mathbb{C}$, it is impossible for the user to acquire its reliability.

For the aggregation server, it knows the ciphertexts $\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right)$, $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot x_m^k\right)$, and plaintexts $\mathbb{C}$, $\sum_{k=1}^{K}\left(R_k \cdot x_m^k\right)$, $\sum_{k=1}^{K} R_k$. Since the (p,t)-threshold Paillier cryptosystem has been demonstrated to defend against chosen-plaintext attacks, the aggregation server cannot recover the secret key, and thus cannot decrypts these ciphertexts. As for $\mathbb{C}$, without the plaintexts $\mathbb{D}$, the aggregation server cannot obtain the users' reliabilities. Although the aggregation server knows the sum of K users' reliabilities, i. e., $\sum_{k=1}^{K} R_k$, it is impossible to identify the individual reliability of each user without knowing other information. Similarly, it is also impossible to separate the individual reliability and model weight from $\sum_{k=1}^{K}\left(R_k \cdot x_m^k\right)$.

Therefore, RPPFL can prevent the user's local gradient and reliabilities from disclosing to other entities. Moreover, for the user collusion attack, the properties of the Paillier cryptosystem ensure the safety of the scheme when there are no more than $t-1$ users colluding.

6 Experiments

In this section, we perform experiments to observe the performance of RPPFL. The FL framework is built via PyTorch with Cuda 10.2, which runs on the server with two Nvidia Tesla-P40 GPUs for hardware and RedHat for the operating system. For the cryptosystem, we utilize the Paillier library for implementation, and the running environment is Java 18.0. Moreover, we choose MNIST and CIFAR-10 as the datasets in FL, which are commonly used in many scenarios. As for the users in FL, they are all equipped with the same convolutional neural network (CNN) to calculate local gradients with the use of their local data. The model in the experiments is inspired by LeNet widely used in various situations. Finally, as for the hyper-parameters, the learning rate is set to 0.001, while the batch size is 128.

6.1 Accuracy Performance

In this part, we observe the accuracy performance of RPPFL. As mentioned before, many attributes influence the model's accuracy. Here, we mainly focus on the impact of the number of users and the number of gradients per user. With-

out losing generality, we set the dataset $\mathcal{D}_i$ for each user k in the same size. Meanwhile, to construct low-quality data for anomalous users, we replace a fixed proportion of their original data with random noises $\epsilon \in [0,1]$. The ratio of the replaced data is set to 20% in our experiments.

1) Number of users

We first illustrate the influence of the number of users that take part in the training process. To better demonstrate the performance of RPPFL, we take two related works[18, 28] for comparison.

Fig. 3 displays the comparison of accuracy based on a different number of users, where the number of gradients for each user is set to 2 500. The figure demonstrates that the increment in the number of users in RPPFL does improve the model accuracy because more data from corresponding users contribute to the trained model. Moreover, for both the MNIST dataset in Fig. 3(a) and CIFAR-10 dataset in Fig. 3(b) , the accuracy of RPPFL is about the same as PPFDL in Ref. [18] and outperforms that in Ref. [28]. Therefore, we can reach the conclusion that RPPFL can ensure the aggregation gradients are mainly contributed by users with data of high quality.

2) Number of gradients per user

We then discuss the influence of the number of gradients for each user on accuracy performance.

Fig. 4 demonstrates that the model accuracy will also improve when the number of gradients increases. It is evident that more involved gradients in the FL training procedure will boost the convergence rate and make the model more accurate. From Figs. 4(a) and 4(b), the performance of RPPFL is still better than the schemes in Refs. [28] and [18]. In conclusion, RPPFL ensures that the user with high-quality data is rewarded with high reliability and guarantees that the aggregation result is mainly contributed by these users.

6.2 Efficiency

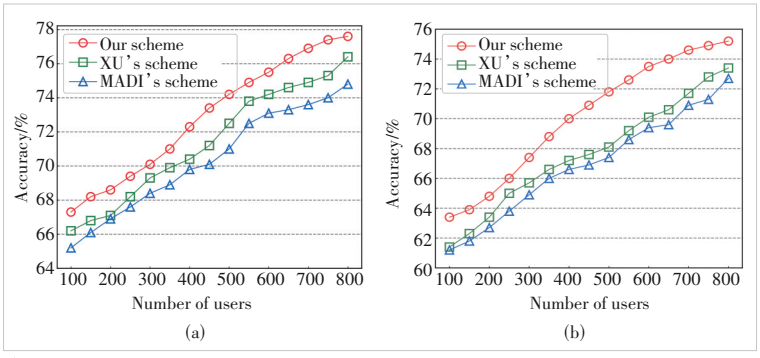
In this part, we observe the efficiency performance of RPPFL. For simplicity, we here only discuss and visualize the efficiency in the aggregation phase of FL. To keep fairness, we test the schemes in Refs. [28] and [18] on the same platform (hardware and software) for RPPFL. Specifically, the CNN network is the same for every user, and other hyper-parameters remain the same.

Fig. 5(a) demonstrates the computational cost for different user numbers, while Fig. 5(b) presents the one for different gradient numbers per user. It can be observed that with the growth of the number of users and the number of gradients per user, the aggregation time increases for all the schemes. Moreover, RPPFL has better efficiency than the one in Ref. [28]. As we can see, the RPPFL is moderately inferior to the one in Ref. [18]. It is because the PPFDL in Ref. [18] adopts a two-cloud model, where the computational costs are shared between the two cloud servers, while RPPFL is established on a single cloud model. However, PPFDL requires two non-colluding cloud servers, which is not practical in real-world scenarios compared with RPPFL.

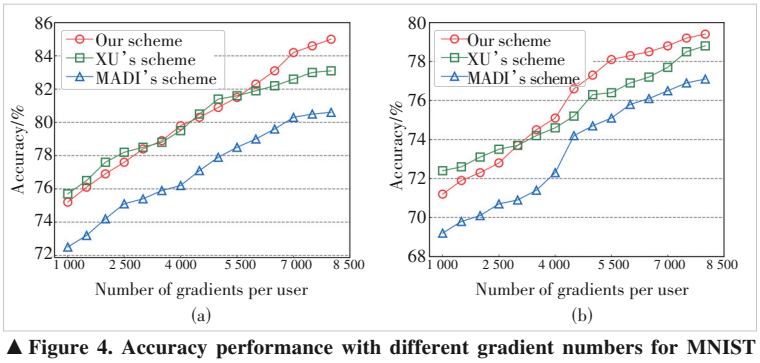
7 Related Works

In this section, we illustrate some related works of privacy-preserving federated learning.

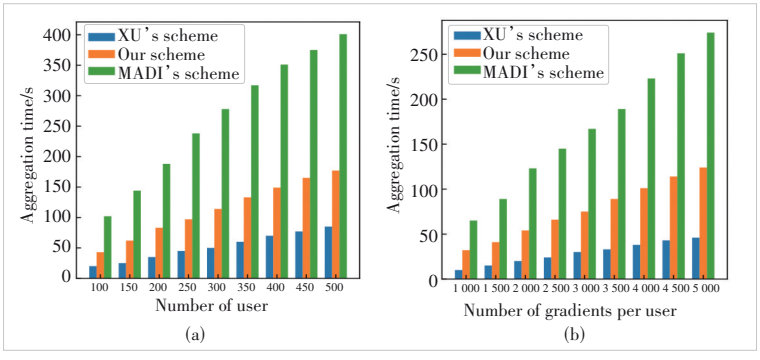
Since the proposal of the original FL, many schemes have been designed to preserve data privacy in FL based on privacy-preserving techniques. These techniques can be mainly divided into three categories: differential privacy, secure multi-party computation, and homomorphic encryption. As for the differential privacy, the authors in Ref. [29] proposed a



▲ Figure 3. Accuracy performance with different user numbers for MNIST and CIFAR-10 datasets



▲ Figure 4. Accuracy performance with different gradient numbers for MNIST and CIFAR-10 datasets



▲ Figure 5. Computational costs for different schemes

mechanism that set different proportions of selected parameters to preserve data privacy while preserving training accuracy. In 2016, ABADI et al.[30] leveraged differential privacy with a moderate privacy budget to learn models of deep neural networks. When it comes to secure multi-party computation, the authors in Ref. [19] proposed a safe and practical aggregation protocol in the FL training process. SMC was adopted to ensure the privacy of the users' gradients shared with the aggregation server. In 2018, JAYARAMAN et al.[31] introduced a distributed learning method that combines DP with SMC. Moreover, because the users' access to power and network bandwidth is always under a particular constraint in real-world scenarios, secret sharing and key exchange protocols are also considered to enhance the robustness of FL. Authors in Ref. [32] proposed a scheme leveraging the secret key-sharing technique to protect privacy in FL while verifying the integrity of aggregation results. For homomorphic encryption, in 2018, PHONE et al.[16] presented a system for privacy-preserving collaborative deep learning. It utilizes Learning with Errors (LWE)-based homomorphic encryption to secure the privacy of publicly shared model parameters among the participants. Furthermore, the authors in Ref. [20] designed high-efficiency protocols by adopting secure two-party computation, which was established on the two-server model (non-collusion). In 2021, MADI et al.[28] presented a scheme with a combination of homomorphic encryption and verifiable computing. The aim was to execute a federated averaging operator directly in the ciphertext and prove that the operator is correctly executed.

In conclusion, homomorphic encryption can be applied for privacy-preserving federated learning according to its property of addition and multiplication in the ciphertext domain. However, the enormous computational burden is unacceptable in scenarios that exist plenty of users or training data with large dimensions. Although SMC is better that HE in terms of computational costs, it always needs many interactions among entities. This brings a high communication burden and a lack of robustness. Compared with the other two techniques, differential privacy performs better in cost. But a balance between privacy and accuracy should always be considered. Ref. [33] demonstrated that if the model accuracy was acceptable, adversaries could still reconstruct the user's private data. Authors in Ref. [34] successfully leveraged a generative adversarial network (GAN) to violate data privacy even if all shared parameters were protected by differential privacy. Therefore, combining the advantages of different privacy-preserving mechanisms while overcoming their drawback has raised much concern for researchers.

Moreover, all these solutions mentioned above fail to consider the problem of anomalous users. To tackle this problem, SecProbe was proposed[17] as the first solution to handling anomalous users in collaborative deep learning while protecting data privacy. It utilized techniques based on DP to perturb the objective function of the target network. However, Ref. [34] showed that the current mechanism of DP can hardly reach an acceptable balance between security and accuracy. XU et al.[18] designed PPFDL with the leverage of additively homomorphic cryptosystem and garbled circuits. However, their system structure is based on the two-cloud model, and it requires two non-colluding cloud servers. Therefore, such limitation makes their scheme impractical in many real-world situations like edge computing. Moreover, their PPFDL is also vulnerable to user collusion attacks.

8 Conclusions

In this paper, we propose RPPFL, a reliable and privacy-preserving federated learning scheme. RPPFL uses a truth discovery technique to identify each user's reliability according to their data quality and thereby reduce the contribution of anomalous users on the global models. Specifically, we leverage an additively homomorphic cryptosystem to enrich the truth discovery technique to provide comprehensive privacy protection (e.g., model privacy and data quality privacy) and user collusion resistance. Security analysis demonstrates the security of RPPFL. Experimental results of two different real-world datasets indicate that RPPFL has acceptable performance on both accuracy and efficiency. For future work, considering that the user may infer data information of others with the global gradients, we will focus on designing a reliable and privacy-preserving federated learning scheme that can protect the privacy of gradients on both the aggregation server side and the user side.

References

[1] WANG J S, LIU Y, ZHANG W T, et al. ReLFA: resist link flooding attacks via renyi entropy and deep reinforcement learning in SDN-IoT [J]. China communications, 2022, 19(7): 157 – 171. DOI: 10.23919/JCC.2022.07.013
[2] KANG J W, LI X D, NIE J T, et al. Communication-efficient and cross-chain empowered federated learning for artificial intelligence of things [J]. IEEE transactions on network science and engineering, 2022, 9(5): 2966 – 2977. DOI: 10.1109/TNSE.2022.3178970
[3] ZHANG W T, YANG D, WU W, et al. Spectrum and computing resource management for federated learning in distributed industrial IoT [C]//Proceedings of 2021 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2021: 1 – 6. DOI: 10.1109/ICCWorkshops50388.2021.9473515
[4] ZHANG W T, YANG D, WU W, et al. Optimizing federated learning in distributed industrial IoT: A multi-agent approach [J]. IEEE journal on selected areas in communications, 2021, 39(12): 3688 – 3703. DOI: 10.1109/JSAC.2021.3118352
[5] PENG H X, SHEN X M. Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks [J]. IEEE journal on selected areas in communications, 2021, 39(1): 131 – 141. DOI: 10.1109/JSAC.2020.3036962
[6] PENG H X, WU H Q, SHEN X S. Edge intelligence for multi-dimensional resource management in aerial-assisted vehicular networks [J]. IEEE wireless communications, 2021, 28(5): 59 – 65. DOI: 10.1109/MWC.101.2100056
[7] European Union. General data protection regulation [EB/OL]. [2022-10-28]. https://gdpr-info.eu/
[8] State of California Department of Justice. California consumer privacy act [EB/OL]. [2022-10-28]. https://oag.ca.gov/privacy/ccpa
[9] SONG C Z, RISTENPART T, SHMATIKOV V. Machine learning models that

remember too much [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 587 – 601. DOI: 10.1145/3133956.3134077
[10] ZHU L G, LIU Z J, HAN S. Deep leakage from gradients Advances [EB/OL]. [2022-10-28]. https://doi.org/10.1007/978-3-030-63076-8_2
[11] ZHAO B, K R MOPURI, H BILEN. iDLG: improved deep leakage from gradients [EB/OL]. [2022-10-28]. https://arxiv.org/abs/2001.02610
[12] YIN H X, MALLYA A, VAHDAT A, et al. See through gradients: image batch recovery via gradinversion [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2021: 16332 – 16341
[13] ZHANG C, ZHAO M Y, ZHU L H, et al. FRUIT: a blockchain-based efficient and privacy-preserving quality-aware incentive scheme [J]. IEEE journal on selected areas in communications, 2022, 40(12): 3343 – 3357. DOI: 10.1109/JSAC.2022.3213341
[14] OUADRHIRI A E, ABDELHADI A. Differential privacy for deep and federated learning: a survey [J]. IEEE access, 2022, 10: 22359 – 22380. DOI: 10.1109/ACCESS.2022.3151670
[15] PEYVANDI A, MAJIDI B, PEYVANDI S, et al. Privacy-preserving federated learning for scalable and high data quality computational-intelligence-as-a-service in Society 5.0 [J]. Multimedia tools and applications, 2022, 81(18): 25029 – 25050. DOI: 10.1007/s11042-022-12900-5
[16] 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
[17] ZHAO L C, WANG Q, ZOU Q, et al. Privacy-preserving collaborative deep learning with unreliable participants [J]. IEEE transactions on information forensics and security, 2020, 15: 1486 – 1500. DOI: 10.1109/TIFS.2019.2939713
[18] XU G W, LI H W, ZHANG Y, et al. Privacy-preserving federated deep learning with irregular users [J]. IEEE transactions on dependable and secure computing, 2022, 19(2): 1364 – 1381. DOI: 10.1109/TDSC.2020.3005909
[19] 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. ACM, 2017: 1175 – 1191. DOI: 10.1145/3133956.3133982
[20] MOHASSEL P, ZHANG Y P. SecureML: a system for scalable privacy-preserving machine learning [C]//Proceedings of 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 19 – 38. DOI: 10.1109/SP.2017.12
[21] ZHENG Y F, DUAN H Y, WANG C. Learning the truth privately and confidently: encrypted confidence-aware truth discovery in mobile crowdsensing [J]. IEEE transactions on information forensics and security, 2018, 13(10): 2475 – 2489. DOI: 10.1109/TIFS.2018.2819134
[22] MIAO C L, JIANG W J, SU L, et al. Cloud-enabled privacy-preserving truth discovery in crowd sensing systems [C]//Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. ACM, 2015: 183 – 196. DOI: 10.1145/2809695.2809719
[23] DAMGARD I, JURIK M. A generalisation, a simplification and some applications of paillier's probabilistic public-key system [M]. Public Key Cryptography. Berlin, Heidelberg: Springer Berlin, 2001: 119 – 136. DOI: 10.1007/3-540-44586-2_9
[24] LI Y L, GAO J, MENG C S, et al. A survey on truth discovery [J]. ACM SIGKDD explorations newsletter, 2016, 17(2): 1 – 16. DOI: 10.1145/2897350.2897352
[25] SMITH V, CHIANG C K, SANJABI M, et al. Federated multi-task learning [EB/OL]. [2022-10-28]. https://arxiv.org/abs/1705.10467
[26] WANG L P, WANG W, LI B. CMFL: mitigating communication overhead for federated learning [C]//Proceedings of 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019: 954 – 964. DOI: 10.1109/ICDCS.2019.00099
[27] XU G W, LI H W, TAN C, et al. Achieving efficient and privacy-preserving truth discovery in crowd sensing systems [J]. Computers & security, 2017, 69: 114 – 126. DOI: 10.1016/j.cose.2016.11.014
[28] MADI A, STAN O, MAYOUE A, et al. A Secure Federated Learning framework using Homomorphic Encryption and Verifiable Computing [C]//Proceedings of 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS). IEEE, 2021: 1 – 8. DOI: 10.1109/RDAAPS48126.2021.9452005
[29] SHOKRI R, SHMATIKOV V. Privacy-preserving deep learning [C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015: 1310 – 1321. DOI: 10.1145/2810103.2813687
[30] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy [C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 308 – 318. DOI: 10.1145/2976749.2978318
[31] JAYARAMAN B, WANG L X, EVANS D, et al. Distributed learning without distress: privacy-preserving empirical risk minimization [C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. ACM, 2018: 6346 – 6357. DOI: 10.5555/3327345.3327531
[32] XU G W, LI H W, LIU S, 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
[33] JAYARAMAN B, EVANS D. Evaluating differentially private machine learning in practice [EB/OL]. [2022-10-28]. https://arxiv.org/abs/1902.08874
[34] HITAJ B, ATENIESE G, PEREZ-CRUZ F. Deep models under the GAN: information leakage from collaborative deep learning [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 603 – 618. DOI: 10.1145/3133956.3134012

Biographies

ZHANG Weiting received his PhD degree in communication and information systems from Beijing Jiaotong University, China in 2021. From Nov. 2019 to Nov. 2020, he was a visiting PhD student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an associate professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include industrial Internet of Things, edge intelligence, and machine learning for wireless networks.

LIANG Haotian received his BS degree from Lanzhou University, China in 2022. He is currently working towards his master's degree in the School of Cyberspace Science and Technology, Beijing Institute of Technology, China. His research interests include machine learning security, Internet of Things security, and cloud security.

XU Yuhua is currently an undergraduate student in School of Computer Science and Technology, Beijing Institute of Technology, China. She is currently working at the research laboratory of advanced network and data security at the School of Cyberspace Science and Technology, Beijing Institute of Technology. Her research interests include applied cryptography and blockchain.

ZHANG Chuan (chuanz@bit.edu.cn) received his PhD degree in computer science from Beijing Institute of Technology, China in 2021. From Sept. 2019 to Sept. 2020, he worked as a visiting PhD student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an assistant professor at the School of Cyberspace Science and Technology, Beijing Institute of Technology, China. His research interests include secure data services in cloud computing, applied cryptography, machine learning, and blockchain.