Privacy-Preserving Incentive Mechanism Design for Federated Cloud-Edge Learning

Tianyu Liu, *Graduate Student Member, IEEE*, Boya Di, *Member, IEEE*, Peng An, *Member, IEEE*, and Lingyang Song, *Fellow, IEEE*

***Abstract*—To avoid the original private data uploading in cloud-edgecomputing, the federated learning (FL) scheme is recently proposed which enhances the privacy preservation. However, the attacks against the uploaded model updates in FL can still cause private data leakage which demotivates the privacy-sensitive participating edge devices. To address this issue, we aim to design a privacy-preserving incentive mechanism for the federated cloud-edge learning (PFCEL) system such that 1) the privacy-sensitive edge devices are motivated to contribute to the local training and model uploading, 2) a trade-off between the private data leakage and the model accuracy is achieved. We first model the data leakage quantitatively from an adversarial perspective, and then formulate the incentive design problem as a three-layer Stackelberg game, where the interaction between the edge servers and edge devices is further formulated as an optimal contract design problem. Extensive theoretical analysis and numerical evaluations demonstrate the effectiveness of our designed mechanism in terms of privacy preservation and system utility.**

***Index Terms*—Cloud-edge computing, federated learning, differential privacy, incentive mechanism.**

I. Introduction

With the fast development of Internet of Things (IoT), the massive edge devices are producing huge amounts of data, which incurs demands for low data processing latency. The long transmission latency of the conventional cloud computing scheme deteriorates the performance of the latency-sensitive data driven applications. Facing these challenges, cloud-edge computing [1] has been proposed, which allows the data to be processed locally at the edge devices (ED) and then partially offloaded to the nearby edge servers (or access points (AP)). The outputs of the APs are uploaded to the cloud center (CC) for further processing or aggregation. Such a three-layer cloud-edge scheme [2], consisting of CC, APs and EDs, enables the computing task offloading from lower layers (EDs) to upper layers (APs and CC), thereby minimizing the overall latency. However, it also suffers severe private data leakage since the uploaded original private data of ED users are entirely exposed to the APs.

To address the data privacy concerns, the federated learning (FL) scheme proposed by Google [3] has drawn attention recently. In the FL, the EDs can train the models locally and only need to upload the model updates to the upper layers, rather than the original private data. In this way, the privacy preservation of the local data is greatly enhanced without degrading the model accuracy[1] of the data-driven applications such as image classification [4], face recognition [5] and health care [6]. In more detail, in the federated cloud-edge learning (FCEL) system, each AP connects with some EDs within its wireless connection range and conducts partial model aggregation using the model updates from these EDs. Then the partial models of the APs are aggregated in the CC to obtain the global model for specific data-driven applications.

Although the FCEL system enhances the data privacy preservation through local data processing, there still exists potential private data leakage. This is because the private data outputs (i.e., the model updates) may be attacked by leveraging the sensitive information contained in these outputs. For example, in [7], a curious server can use generative adversarial networks (GAN) [8] to generate fake data that have similar distributions as real data. In [9], the original data can be partially recovered by mimicking the learning process and minimizing the difference between the fake loss function gradients and the real gradients contained within the transmitted model updates. Faced with such privacy risks, the privacy-sensitive data owners (EDs) may not participate in the FL tasks.

To motivate these privacy-sensitive ED users to actively participate in the FL task, we propose to apply the differential privacy (DP) [10] based method to the FCEL system, which allows the ED users to upload model updates with noise perturbations for privacy preservation. In principle, DP makes it indistinguishable whether a certain data record exists in the training dataset through perturbation mechanisms. Compared to other privacy preserving methods such as secure multi-party computation (MPC) [11] and homomorphic encryption [12], DP fits the FL scheme more since it avoids the high computation and communication overhead, especially when the number of participating devices is huge. Besides, we can observe that when applying the

[1] In the image classification task, model accuracy denotes the percentage of correct class predictions that the model achieves on test dataset.

Manuscript received April 4, 2021; revised June 25, 2021; accepted July 17, 2021. Date of publication July 27, 2021; date of current version September 16, 2021. This work was supported by the National Natural Science Foundation of China under Grants 61931019, 61625101, 61941101, and 62032003. Recommended for acceptance by Prof. Bo Ji. *(Corresponding author: Lingyang Song.)*

Tianyu Liu, Boya Di, and Lingyang Song are with the Department of Electronics, Peking University, Beijing 100871, China (e-mail: liuty@pku.edu.cn; diboya@pku.edu.cn; lingyang.song@pku.edu.cn).

Peng An is with the Beijing Wondersoft Technology Corp., Ltd, Beijing 100080, China (e-mail: anpeng@wondersoft.cn).

Digital Object Identifier 10.1109/TNSE.2021.3100096

DP based method, the added perturbations on data outputs may influence the model training, and thus, there exists a trade-off between private data leakage and model accuracy. The existing research on DP-FL [13], [14] mainly focus on the influential factors on the convex loss function convergence under noise perturbations, while we study how to achieve the trade-off based on non-convex loss function convergence analysis and private data leakage analysis. To achieve a satisfying trade-off, we aim to design an incentive mechanism[2] such that 1) the EDs are motivated to participate in the FL task while minimizing their own private data leakage, 2) the CC obtains a global model with target accuracy contributed by the model updates of EDs and APs.

In the literature, most existing works focus on the incentive mechanism of the two-layer federated edge learning scheme to motivate the participating devices to contribute more data and resources ([17]–[21]). Only some initial works consider the three-layer FL system with an emphasis on the centralized resource allocation and task scheduling ([22], [23]). In [17], a reputation-based worker selection scheme is proposed to incentivize mobile devices with high-quality data to participate in the learning task. In [18], a deep reinforcement learning-based incentive mechanism is proposed to motivate edge nodes to provide more training data contribution. In [19], hierarchical incentive mechanisms are proposed to solve the incentive mismatches between workers and model owners, as well as among model owners in the FL task over mobile crowdsensing. In [20], differential privacy is applied to preserve the FL worker cost privacy in the incentive mechanism and the local model privacy. In [21], a Stackelberg-game-based incentive mechanism is proposed between the base stations and the FL users. In [22], an algorithm is proposed for efficient resource scheduling in the three-layer FL task. In [23], a multi-layer federated learning protocol is proposed to improve the efficiency of the FL task. Besides, in [24], a multi-layer hierarchical learning framework named fog learning is proposed which can reduce network resource costs and model training times through local model aggregations at different network layers.

However, the above works either have not considered the additional private data leakage from the uploaded model updates or have not depicted the hierarchical interactions among the CC-AP-ED layers, i.e., the updated model exchanging between adjacent layers. In contrast, we aim to develop a privacy-preserving incentive mechanism for the federated cloud-edge learning system (PFCEL) to achieve a privacy-accuracy trade-off, where new challenges have arisen. *First*, each ED user has different privacy sensitivity. Thus, how to quantitatively measure the data leakage of different ED users is a challenge. *Second*, the privacy sensitivity of each ED user is not known to each other. In this case, it is not trivial to incentivize them given incomplete information. *Third*, it is challenging to deal with the conflict between the system utility maximization and the personal utility of each node in different layers, where the two-layer mechanisms no longer apply and a three-layer one is needed.

To address the above challenges, we propose a privacy-preserving incentive mechanism where a three-layer Stackelberg game (TLSG) ([25], [26]) is formulated. Specifically, the interaction between EDs and APs is formulated as an optimal contract design problem such that the APs can motivate the EDs to actively participate in the FL task without any prior knowledge of the privacy sensitivity of the EDs. By estimating the strategies of the EDs and APs, the CC then determines the monetary incentives for the lower layers such that the global model accuracy can be optimized.

The main contributions of this paper are summarized as follows:

- We consider the privacy-preserving federated cloud-edge learning system and propose an incentive mechanism to motivate the edge devices with privacy concerns to actively participate in the computing task.
- For the measurement of privacy preservation, we propose a quantitative private data leakage model from an adversarial perspective that draws the relationship between the level of data leakage and the privacy budget.
- The designed incentive mechanism achieves a trade-off between the private data leakage and the model accuracy. Both theoretical analysis and simulation results are provided to verify the effectiveness of the PFCEL scheme compared to the traditional two-layer FL-MEC ones.

The remainder of this paper is organized as follows. In Section II, we introduce the federated cloud-edge learning system, and propose the privacy model and data leakage model. In Section III, we formulate the TLSG and contract problems. The contract design and TLSG optimization are presented in Section IV and Section V respectively. Section VI presents the numerical evaluation results and Section VII concludes the article.

II. System Model

In this section, we first introduce the PFCEL system in Section II-A. Then we model the privacy preservation and the corresponding data leakage in Sections II-B and II-C, respectively.

A. System Description

As shown in Fig. 1, we consider the PFCEL system consisting of one CC, M APs and N_0 EDs. The CC is in wired connections with M APs. Each AP$_i$ serves as an edge server, and has wireless connections with N_i EDs, satisfying $N_0 = \sum_{i=1}^{M} N_i$. The nodes of three layers together conduct the FL task as follows. The CC releases the FL task (e.g. deep learning based image classification task), distributes an initial deep model to the EDs, and recruits the EDs to participate in the FL task (training a local model) using their local data. Then the APs and the CC conduct partial and global model aggregations periodically.

[2] The monetary incentives are widely used in scenarios such as financial risk control [15] and data broker platforms [16]. In financial risk control, the banks provide monetary rewards to other financial institutions for the usage of the private data outputs to obtain a federated risk assessment model. In data broker platforms, the brokers provide monetary incentives to the users in exchange for private data and charge the users for using the collected data.

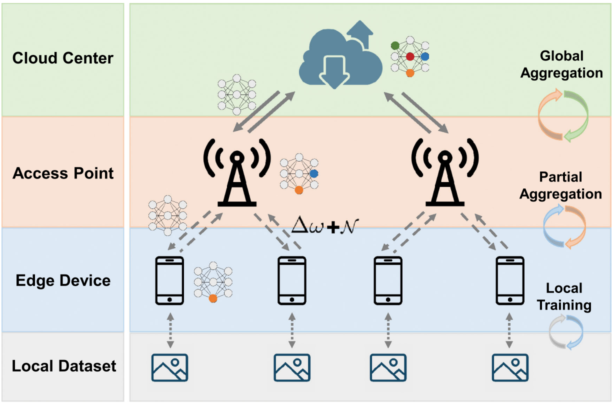


Fig. 1. The privacy-preserving federated cloud-edge learning system.

The above FL process consists of three iterations with different time scales (see their relations in Fig. 2) as below:

- *Local training iteration*: each ED conducts local model training based on the local data samples. Denote $\mathcal{D}_{i,j}$ as the local dataset of $\text{ED}_{i,j}$, where $\text{ED}_{i,j}$ is the j^{th} ED in connection with AP_i. The local dataset $\mathcal{D}_{i,j}$ has $D_{i,j}$ data samples. During local training iterations, the task of $\text{ED}_{i,j}$ is to minimize the loss function $f_{i,j}(\omega_{i,j})$ of model parameters $\omega_{i,j}$, which is defined as:

$$\min_{\omega} f_{i,j}(\omega_{i,j}) \triangleq f_{i,j}(\omega_{i,j}; \mathcal{D}_{i,j}), \tag{1}$$

  For example, in the image classification task, the form of $f_{i,j}(\omega_{i,j})$ is the commonly used cross entropy loss [27].

- *Partial aggregation iteration*: every T_0 local training iterations, each AP receives the model updates from EDs, conducts partial model aggregation, and distributes the aggregated new partial model to the EDs. The loss function $F_i(\omega_i)$ of AP_i is expressed by:

$$F_i(\omega_i) = \sum_{j=1}^{N_i} \frac{D_{i,j}}{\sum_{j=1}^{N_i} D_{i,j}} f_{i,j}(\omega_{i,j}). \tag{2}$$

- *Global aggregation iteration*: every T_1 partial aggregation iterations, the CC receives the model updates from the APs, conducts global model aggregation, and distributes the aggregated new global model to the APs and EDs. There are totally T_2 global aggregation iterations. The loss function $F(\omega)$ of CC is expressed by:

$$F(\omega) = \sum_{i=1}^{M} \frac{\sum_{j=1}^{N_i} D_{i,j}}{\sum_{i=1}^{M} \sum_{j=1}^{N_i} D_{i,j}} F_i(\omega_i). \tag{3}$$

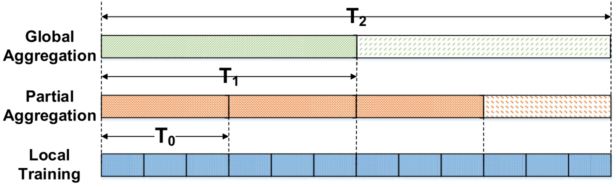


Fig. 2. The time scales of different iterations.

The model parameters ω can be updated via different methods. For example, we adopt the gradient descent method in the example of image classification. Denote e as the local training iteration index and η as the learning rate. During each local training iteration, the model parameters are updated on an ED by:

$$\omega_{i,j}^{e+1} = \omega_{i,j}^{e} - \eta \nabla f_{i,j}(\omega_{i,j}^{e}), \tag{4}$$

where $\nabla f_{i,j}(\omega_{i,j})$ denotes the loss function gradients.

Denote t as the partial aggregation iteration index. During each partial aggregation, the model parameters on AP_i are updated by:

$$\omega_i^{t+1} = \omega_i^t + \sum_{j=1}^{N_i} \frac{D_{i,j}}{\sum_{j=1}^{N_i} D_{i,j}} \Delta\omega_{i,j}^t, \tag{5}$$

where $\Delta\omega_{i,j}$ is the local model update, and $\Delta\omega_{i,j}^t = -\eta \sum_{e=(t-1)T_0}^{tT_0} \nabla f_{i,j}(\omega_{i,j}^e)$, i.e., the range of e for $\Delta\omega_{i,j}^t$ is from $e = (t-1)T_0$ to $e = tT_0$.

Denote s as the global aggregation iteration index. During each global aggregation, the model parameters on the CC are updated by:

$$\omega^{s+1} = \omega^s + \sum_{i=1}^{M} \frac{\sum_{j=1}^{N_i} D_{i,j}}{\sum_{i=1}^{M} \sum_{j=1}^{N_i} D_{i,j}} \Delta\omega_i^s, \tag{6}$$

where $\Delta\omega_i$ is the partial model update, $\Delta\omega_i^s = \sum_{t=(s-1)T_1}^{sT_1} \Delta\omega_i^t$, and $\Delta\omega_i^t = \omega_i^{t+1} - \omega_i^t$. The range of t for $\Delta\omega_i^s$ is from $t = (s-1)T_1$ to $t = sT_1$, and the range of s is from $s = 1$ to $s = T_2$.

B. Privacy Model

In the PFCEL system, a major concern is about the privacy of the ED local data. When the uploaded model updates are acquired by the curious servers, the private images of the users may be recovered [9]. We consider to apply the differential privacy (DP) based method for privacy preservation to the local datasets. By applying a permutation mechanism on the ED datasets, DP makes it hard to distinguish between different local datasets. Next we show the definition of (ϵ, δ)-DP:

Definition 1 ((ϵ, δ)-Differential Privacy [10]): A randomized mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-Differential Privacy, if for all measurable sets $\mathcal{S} \subseteq \mathcal{R}$ and for any two adjacent datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}$,

$$Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq \exp(\epsilon) Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta, \tag{7}$$

where ϵ is the privacy budget and δ is the failure probability.

According to (7), a smaller privacy budget ϵ implies that it is harder to distinguish between datasets $\mathcal{D}$ and $\mathcal{D}'$, leading to a lower data leakage risk. In our system, different EDs have different privacy budgets $\{\epsilon_{i,j}\}$. In order to guarantee (ϵ, δ)-DP, we apply the Gaussian mechanism that provides a way to add a Gaussian noise perturbation $n \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$ to the parameters, where d is the dimension of the model parameters, as shown by the following lemma:

Lemma 1 (Gaussian Mechanism): If running T model aggregations, the Gaussian perturbation mechanism with scale $\sigma = \Delta_2 \sqrt{2\ T\, log(1.25/\delta)}/\epsilon$ is (ϵ, δ)-DP, where $\Delta_2 = max_{\mathcal{D},\mathcal{D}'}\ \ ||s(\mathcal{D}) - s(\mathcal{D}')||_2$ is the l_2-sensitivity of the parameters.

For clipped parameters $\|\omega\| \leq L$, the l_2-sensitivity is $\Delta_2 = 2\ L/D$ [13], where D is the size of the local datasets and we assume that the sizes $\{D_{i,j}\}$ are the same $D = D_{i,j}$. Substitute Δ_2 into σ, and the Gaussian noise standard deviation is expressed as:

$$\sigma = \frac{2\ L\sqrt{T\, log(1.25/\delta)}}{D\epsilon} = \frac{2cL\sqrt{T}}{D\epsilon}, \tag{8}$$

where $c = \sqrt{2log(1.25/\delta)}$.

To evaluate the influence of the added noise on the FL task, we consider the convergence upper bound of the task, which is defined as the speed of model training until the updated parameters ω converge to stable values. Then we have the following theorem which provides the formulation of the convergence upper bound.

Theorem 1 (Convergence Upper Bound): Under assumptions that the loss function $F(\omega)$ is β-Lipschitz and ρ-Lipschitz smooth, the convergence upper bound A in our system is given by:

$$A = \frac{2}{T(2\eta E - \rho\eta^2 E^2)} \left(\Theta + \frac{2c\beta dL}{ND} \sqrt{\frac{2\ T^3}{\pi}} \sum_{j=1}^{N} \frac{1}{\epsilon_j} + \frac{2\rho c^2\ d^2\ L^2\ T^2}{N^2 D^2} \sum_{j=1}^{N} \frac{1}{{\epsilon_j}^2} \right), \tag{9}$$

where T is the number of aggregations, E is the number of training iterations, and N is the number of participating EDs. $\Theta = F(\omega_0) - F(\omega^*)$ is the difference between the initial loss function value $F(\omega_0)$ and the optimal loss function value $F(\omega^*)$.

Proof: See proof in Appendix A. ■

Theorem 1 provides a relationship between the Gaussian noise (represented by ϵ) and the convergence upper bound of the FL task loss function from a two-layer perspective. We note that the parameters N, T and E in (9) differ in the AP-ED partial model convergence analysis and CC-ED global model convergence analysis. For AP-ED, $N = N_i$, $T = T_1 T_2$ and $E = T_0$. For CC-ED, $N = N_0$, $T = T_2$ and $E = T_0 T_1$. For simplicity, we express A in the following form:

$$A = \lambda_1 + \lambda_2 \sum_{j=1}^{N} \frac{1}{\epsilon_j} + \lambda_3 \sum_{j=1}^{N} \frac{1}{{\epsilon_j}^2}, \tag{10}$$

where $\lambda_1 = \frac{2\Theta}{T(2\eta E - \rho\eta^2 E^2)}$, $\lambda_2 = \frac{4c\beta dL}{NDT(2\eta E - \rho\eta^2 E^2)} \sqrt{\frac{2\ T^3}{\pi}}$ and $\lambda_3 = \frac{4\rho c^2\ d^2\ L^2\ T}{N^2 D^2 (2\eta E - \rho\eta^2 E^2)}$.

C. Data Leakage

After introducing the privacy model, a derived problem is how to evaluate the effectiveness of the privacy preservation. We look into this problem from an adversarial perspective, i.e., we assume that the edge server (AP) is a curious agent who tries to make use of the received ED user information (model updates) to retrieve the user private information (images). A typical method is to recover a batch of images using the parameter gradients within the model updates updated by the EDs ([9], [28]).

In the method [9], a batch of original images can be recovered pixel-wise by minimizing the gradient match loss between the real parameter gradients $\nabla F(\omega)$ and the fake parameter gradients $\nabla F(\omega')$, which is expressed as:

$$x^* = \min_{x'}(\|\nabla F(\omega') - \nabla F(\omega)\|^2), \tag{11}$$

where x' is a fake image and x^* is a recovered image.

The EDs calculate $\nabla F(\omega)$ to update the model parameters using their private data, while the curious APs update the dummy inputs and labels to minimize the gradient match loss $\mathcal{L} = \|\nabla F(\omega') - \nabla F(\omega)\|^2$ and recover the images. After some iterations, the gradient match loss comes to the minimum and the dummy inputs (fake images) become recovered images that are almost the same as the original images.

In order to preserve the data privacy against the curious APs, an effective way for the EDs is to add Gaussian noise permutation to the parameter gradients before transmission, as introduced in Subsection II-B. We measure the level of data leakage with respect to the scale of added Gaussian noise. To reveal the detailed relationship between data leakage level and the Gaussian noise, we conduct further experiments under different choices of Gaussian noise standard deviation σ.

From our experiments on a rich set of images from MNIST [29] dataset, we draw the following conclusions. Firstly, the added Gaussian noise standard deviation σ has an approximate scale threshold σ_m, beyond which the original image recovery is no longer successful. In our experiment on MNIST, $\sigma_m \approx 0.074$. Generally, the threshold σ_m is different for various scenarios and adapts to the corresponding FL tasks, datasets and models. If the added noise scale is larger than the threshold, the curious APs can not recover the images. However, large noise scale may deteriorate the convergence bound or even make the FL task unable to converge. A properly chosen noise scale can help to preserve privacy and ensure the convergence simultaneously. Secondly, the gradient match loss $\mathcal{L}$ has an exponential relation with σ (see Fig. 3). In the general case, the relation between $\mathcal{L}$ and σ is denoted as:

$$\mathcal{L}(\sigma) = {b_1}' \exp({b_2}'\sigma), \tag{12}$$

where ${b_1}'$ and ${b_2}'$ are constant coefficients.[3] With the growth of σ, the perturbation impact of the noise on the parameter

[3] The coefficients ${b_1}'$, ${b_2}'$ and the threshold σ_m can be obtained through experiments on the datasets of CC [30] or on the public datasets that do not contain private information of ED users [31].

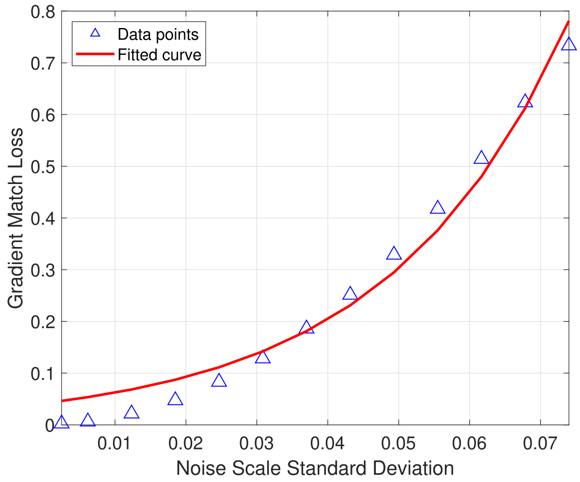


Fig. 3. The exponential relationship between the gradient match loss and the noise scale standard deviation σ. In our experiments, the constant coefficients are $b_1{}' = 0.042$ and $b_2{}' = 39.5$.

gradients grows much faster, as the noise gradually overwhelms the valid information contained in the gradients.

Substituting (8) into (12), we obtain the relation between $\mathcal{L}$ and ϵ, which is expressed as:

$$\mathcal{L}(\epsilon) = b_1{}'\exp\left(\frac{2b_2{}'cL\sqrt{T}}{D\epsilon}\right) = b_1\exp\left(\frac{b_2}{\epsilon}\right), \tag{13}$$

where $b_1 = b_1{}'$ and $b_2 = \frac{2b_2{}'cL\sqrt{T}}{D}$.

As the gradient match loss is negatively related to the effectiveness of the original image recovery and the level of data leakage, we define the data leakage level as the reciprocal of $\mathcal{L}(\epsilon)$, i.e., $l(\epsilon) = 1/\mathcal{L}(\epsilon)$.

III. Problem Formulation

In this section, we first formulate the PFCEL system utility, then introduce the contract design problem for the AP-ED layers, and finally introduce the three-layer Stackelberg game framework for the CC-AP-ED layers.

A. System Utility

The goal of the FL task is to obtain a global model with the desired accuracy within a certain time period while preserving the ED user's data privacy. To achieve this goal, the system utility is related to the global model satisfaction level G and the ED data leakage level $l(\epsilon_{i,j})$, which is expressed as:

$$U^{Sys} = G - \sum_{i=1}^{M}\sum_{j=1}^{N_i}\theta_{i,j}l(\epsilon_{i,j}), \tag{14}$$

where $\theta_{i,j}$ is a coefficient that is introduced in Section III-B detailedly and the global model satisfaction level G is denoted as:

$$G = C - A(\epsilon, N_0, T_2), \tag{15}$$

where the global model convergence bound A is given by (9), and C is a constant. G has a negative relation with the global convergence bound, as smaller convergence bound means closer to the optimum, which provides higher satisfaction.

However, the nodes (CC, APs and EDs) in the system are selfish and only care about their own utilities. To motivate the selfish nodes to actively participate in the FL task, we consider designing an incentive mechanism where the system utility can be improved by each node maximizing its own utility. In other words, the utility optimization of each node is consistent with the utility optimization of the system.

Specifically, the CC provides monetary incentives (i.e., rewards) to the APs which compensate for the incentives provided to the EDs by the APs. Based on the received rewards from the CC, the APs adjust their incentive strategies, i.e., how to pay the EDs. Based on the received rewards from the APs, the EDs also adjust their strategies, i.e., what privacy budgets ϵ to choose. The received rewards of EDs compensate for their data leakage, and different rewards will influence EDs' strategies of choosing different privacy budgets. There exists a three-layer leader-follower relationship among the CC-AP-ED nodes, based on which we formulate the incentive mechanism design problem as a three-layer Stackelberg game. Within the TLSG framework, we further formulate the AP-ED relationship based on contract theory. Next we demonstrate the formulation of the incentive mechanism design problems by introducing the utility functions of different nodes.

B. Contract Formulation

In this section, we formulate the incentive mechanism design problem between the APs and EDs with the target of designing the optimal contracts $\{(R, \epsilon)\}$. An AP delegates the FL task to multiple EDs, whose types $\{\theta\}$ are unobservable to the AP, and offers a menu of contracts $\{(R, \epsilon)\}$ to the EDs. If an ED chooses to accept the contract (R, ϵ), then he will provide the AP with local model update under privacy budget ϵ, and in return, the AP pays the reward R to the ED. If an ED declines to accept any contract, we assume that the ED signs a contract of $(0, 0)$. Due to the competitive relationship among the APs, each AP designs the contracts in a way that more EDs are motivated to participate in the task and the AP itself can receive more incentive from the CC.

We use the contract type $\theta \in [\underline{\theta}, \bar{\theta}]$ to describe the privacy preference of ED users. A larger θ means the user cares more about privacy. The incentive I_i received from the CC is defined as:

$$\begin{aligned} I_i &= a_i(C_i - A_i(\epsilon_i, N_i, T_1T_2)) \\ &= a_i(C_i - \lambda_{1,i}) - N_i\int_{\underline{\theta}_i}^{\bar{\theta}_i} p_i(\theta)\left(\frac{a_i\lambda_{2,i}}{\epsilon_i(\theta)} + \frac{a_i\lambda_{3,i}}{\epsilon_i{}^2(\theta)}\right)\mathrm{d}\theta, \end{aligned} \tag{16}$$

where A_i is the partial model convergence bound given by (9), a_i is a coefficient, C_i is a constant, and the convergence bound in discrete form is transformed into a continuous form as the contracts are continuous functions with respect to the contract type θ.

Given the incentive I_i received, the optimal contract formulation strategy for AP$_i$ can be obtained by:

$$\max_{(R(\theta),\epsilon(\theta))} \quad U_i^{AP} = I_i - N_i \int_{\underline{\theta}_i}^{\bar{\theta}_i} p_i(\theta) R_i(\theta) \mathrm{d}\theta = \tag{17}$$

$$a_i(C_i - \lambda_{1,i}) - N_i \int_{\underline{\theta}_i}^{\bar{\theta}_i} p_i(\theta) \left(\frac{a_i \lambda_{2,i}}{\epsilon_i(\theta)} + \frac{a_i \lambda_{3,i}}{{\epsilon_i}^2(\theta)} + R_i(\theta) \right) \mathrm{d}\theta,$$

$$s.t. \quad R_i(\theta) - \theta l(\epsilon_i(\theta)) \geq R_i(\tilde{\theta}) - \theta l(\epsilon_i(\tilde{\theta})), \quad \theta \neq \tilde{\theta}, \tag{17a}$$

$$R_i(\theta) - \theta l(\epsilon_i(\theta)) \geq 0, \tag{17b}$$

$$\epsilon_i(\theta) \geq 0, \tag{17c}$$

$$\sum_{j=1}^{N_i} R_{i,j} \leq I_i, \tag{17d}$$

where $p_i(\theta)$ is the probability density function. The first constraint follows Incentive Compatibility (IC), the second constraint follows Incentive Rationality (IR), the third constraint ensures that ϵ is within reasonable range, and the fourth constraint ensures that the total rewards are less than the total incentive. The IC and IR constraints are further defined as below:

Definition 2 (Incentive Compatibility): Each ED of type θ only chooses the contract designed for its type, i.e. $(R_i(\theta), \epsilon(\theta))$, instead of any other type of contracts, to maximize its utility:

$$R_i(\theta) - \theta l(\epsilon_i; \theta) \geq R_i(\tilde{\theta}) - \theta l(\epsilon_i; \tilde{\theta}), \quad \theta \neq \tilde{\theta}. \tag{18}$$

Definition 3 (Incentive Rationality): Each ED only participates in the learning task when its utility is not less than zero:

$$R_i(\theta) - \theta l(\epsilon_i(\theta)) \geq 0. \tag{19}$$

After receiving a menu of contracts $\{(R_i(\theta), \epsilon_i(\theta))\}$ from AP$_i$, the optimal privacy budget choice strategy for ED$_{i,j}$ can be obtained by:

$$\max_{\epsilon_{i,j}} \quad U_{i,j}^{ED} = R_{i,j} - \theta_{i,j} l(\epsilon_{i,j}), \; s.t. \quad \epsilon_{i,j} \geq 0, \tag{20}$$

where $R_{i,j}$ is the received reward according the selected contract, and $\theta_{i,j}$ is the privacy preference of the ED user.

C. Three-Layer Stackelberg Game

Now we have formulated the incentive mechanism design of the AP-ED layers with contract theory, and the continuous form of the global model satisfaction level G is:

$$G = C - \left({\lambda_1}' + \sum_{i=1}^{M} N_i \int_{\underline{\theta}_i}^{\bar{\theta}_i} p_i(\theta) \left(\frac{{\lambda_2}'}{\epsilon_i(\theta)} + \frac{{\lambda_3}'}{{\epsilon_i}^2(\theta)} \right) \mathrm{d}\theta \right) \tag{21}$$

Then for the CC, based on its prediction of the EDs' and APs' strategies, the optimal incentive strategy can be obtained by:

$$\begin{aligned} \max_{\{a_i\}} \quad U^{CC} &= G - \sum_{i=1}^{M} I_i \\ &= C - \lambda_1' - \sum_{i=1}^{M} a_i(C_i - \lambda_{1,i}) + \sum_{i=1}^{M} N_i \int_{\underline{\theta}_i}^{\bar{\theta}_i} p_i(\theta) \\ &\left(\frac{a_i \lambda_{2,i} - {\lambda_2}'}{\epsilon_i(\theta)} + \frac{a_i \lambda_{3,i} - {\lambda_3}'}{{\epsilon_i}^2(\theta)} \right) \mathrm{d}\theta \end{aligned} \tag{22}$$

$$s.t. \quad I_i - N_i \int_{\underline{\theta}}^{\bar{\theta}} p_i(\theta) R_i^*(\theta) \mathrm{d}\theta \geq 0, \tag{22a}$$

$$R_i^*(\theta) = \underset{R_i(\theta)}{\operatorname{argmax}} \, \mathbb{E}[U_i^{AP}],$$

$$G - \sum_{i=1}^{M} I_i \geq 0, \tag{22b}$$

$$\sum_{i=1}^{M} a_i = 1, \tag{22c}$$

$$0 \leq a_i \leq 1, \tag{22d}$$

where the first constraint ensures that the AP utility is not less than zero, the second constraint ensures that the CC utility is not less than zero, the third constraint ensures that the sum of the incentive assignment coefficients is 1 and the fourth constraint ensures that the incentive assignment coefficients are between 0 and 1.

The TLSG framework is demonstrated in Fig. 4. Based on the utility functions of the nodes, we can define the pure strategy Stackelberg equilibrium of the TLSG, which consists of the following three concepts: the optimal privacy budget choice strategy of the EDs, the optimal contract formulation strategies of the APs and the optimal incentive strategy of the CC. For the Stackelberg equilibrium, no one will deviate from his chosen strategy after considering the opponents' choices. With the utility functions of the nodes, we can rewrite the system utility as the sum of all nodes, which is denoted as:

$$U^{Sys} = U^{CC} + \sum_{i=1}^{M} U_i^{AP} + \sum_{i=1}^{M} \sum_{j=1}^{N_i} U_{i,j}^{ED}. \tag{23}$$

IV. Contract Design for Edge-Device Layers

In this section, we first simplify the constraints of the contract optimization problem, then solve the reformulated problem by optimal control methods.

A. Constraint Reformulation

As defined in section III-B, the contract offered by the AP is formed as $(R(\theta), \epsilon(\theta))$. To find the optimal menu of contracts $\{(R^*(\theta), \epsilon^*(\theta))\}$, we first simplify the constrains, i.e. IC (Definition 2) and IR (Definition 3). IC actually refers to a set of constraints with different combinations of θ and $\tilde{\theta}$, and IR refers to a set of constraints with different θ. Following a similar approach in [32] and [33], we can reduce the set of IC constraints.

According to Equation (18), for any $(\theta, \tilde{\theta}) \in [\underline{\theta}, \bar{\theta}]^2$, the following two inequalities hold:

$$R(\theta) - \theta l(\epsilon(\theta)) \geq R(\tilde{\theta}) - \theta l(\epsilon(\tilde{\theta})), \tag{24}$$

$$R(\tilde{\theta}) - \tilde{\theta} l(\epsilon(\tilde{\theta})) \geq R(\theta) - \tilde{\theta} l(\epsilon(\theta)). \tag{25}$$

Add the above two inequalities, and we have:

$$(\theta - \tilde{\theta})(l(\epsilon(\tilde{\theta})) - l(\epsilon(\theta))) \geq 0. \tag{26}$$

Equation (26) implies that $l(\epsilon(\theta))$ should be a non-increasing function of θ. Besides, both $R(\theta)$ and $l(\epsilon(\theta))$ are differentiable everywhere. Given θ, Equation (24) implies that the function $y(\tilde{\theta}) = R(\tilde{\theta}) - \theta l(\epsilon(\tilde{\theta}))$ reaches the maximum at $\tilde{\theta} = \theta$, and θ must satisfy the following two conditions:

$$\frac{\mathrm{d}R(\theta)}{\mathrm{d}\theta} - \theta \frac{\mathrm{d}l(\epsilon(\theta))}{\mathrm{d}\theta} = 0, \tag{27}$$

$$\frac{\mathrm{d}^2 R(\theta)}{\mathrm{d}\theta^2} - \theta \frac{\mathrm{d}^2 l(\epsilon(\theta))}{\mathrm{d}\theta^2} \leq 0. \tag{28}$$

Differentiating Equation (27), we have:

$$\frac{\mathrm{d}^2 R(\theta)}{\mathrm{d}\theta^2} - \frac{\mathrm{d}l(\epsilon(\theta))}{\mathrm{d}\theta} - \theta \frac{\mathrm{d}^2 l(\epsilon(\theta))}{\mathrm{d}\theta^2} = 0. \tag{29}$$

Then by the difference between Equation (28) and (29), we can get:

$$-\frac{\mathrm{d}l(\epsilon(\theta))}{\mathrm{d}\theta} \geq 0. \tag{30}$$

Now we have reduced the set of IC constraints to a differential equation (27) and a monotonicity constraint (30). The continuous form of the ED utility (Equation (20)) is given by:

$$U(\theta) = R(\theta) - \theta l(\epsilon(\theta)). \tag{31}$$

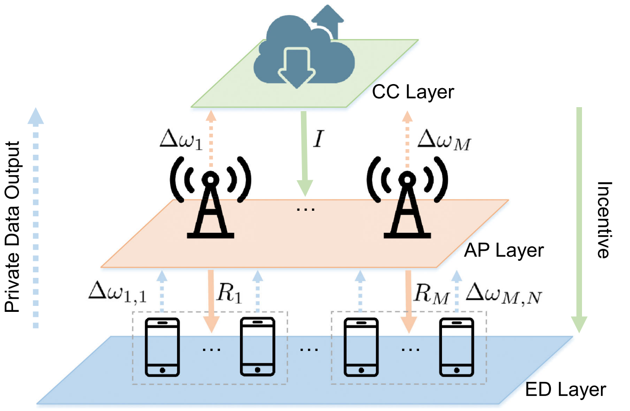


Fig. 4. The CC-AP-ED three-layer Stackelberg game framework.

Differentiating the ED utility and applying (27), we have:

$$\dot{U}(\theta) = \frac{\mathrm{d}R(\theta)}{\mathrm{d}\theta} - l(\epsilon(\theta)) - \theta \frac{\mathrm{d}l(\epsilon(\theta))}{\mathrm{d}\theta} = -l(\epsilon(\theta)). \tag{32}$$

To optimize $R(\theta)$, we can instead find the optimal $U(\theta)$ and $\epsilon(\theta)$, and acquire $R(\theta)$ by $R(\theta) = U(\theta) + l(\epsilon(\theta))$. Based on Equation (32), the set of IR constraints can be simplified to $U(\bar{\theta}) \geq 0$ and the function reaches the optimum at $\bar{\theta}$:

$$U^*(\bar{\theta}) = 0. \tag{33}$$

Based on the simplified IC and IR constraints and omitting the index i for simplicity, the optimization problem of U^{AP} can be rewritten as:

$$\max_{(U(\theta),\epsilon(\theta))} \quad U^{AP} = a(C - \lambda_1) - N \int_{\underline{\theta}}^{\bar{\theta}} p(\theta) \Bigg(\frac{a\lambda_2}{\epsilon(\theta)} + \frac{a\lambda_3}{\epsilon^2(\theta)} + U(\theta) + \theta l(\epsilon(\theta)) \Bigg) \mathrm{d}\theta, \tag{34}$$

$$s.t. \quad -\frac{\mathrm{d}l(\epsilon(\theta))}{\mathrm{d}\theta} \geq 0, \tag{34a}$$

$$\dot{U}(\theta) = -l(\epsilon(\theta)), \tag{34b}$$

$$U^*(\bar{\theta}) = 0, \tag{34c}$$

$$0 \leq \epsilon(\theta) \leq 1. \tag{34d}$$

B. Optimal Control Solutions

The optimization problem of U^{AP} with new constraints fits the formation of the optimal control problem [34], and we apply the optimal control methods to optimize U^{AP}. In our problem, $\epsilon(\theta)$ is the control variable and $U(\theta)$ is the state variable. Then the *Hamiltonian* is defined as:

$$\begin{aligned} H(U(\theta),\epsilon(\theta),q(\theta),\theta) = -p(\theta)\bigg(&\frac{a\lambda_2}{\epsilon(\theta)}+\frac{a\lambda_3}{\epsilon^2(\theta)}+U(\theta) \\ &+\theta l(\epsilon(\theta))\bigg) - q(\theta)l(\epsilon(\theta)), \end{aligned} \tag{35}$$

where $q(\theta)$ is the co-state variable.

According to the *Pontryagin Minimum principle* [34], the optimal solution $(U^*(\theta),\epsilon^*(\theta))$ should satisfy the following five conditions:

$$\dot{U}^*(\theta) = \frac{\partial H(U^*(\theta),\epsilon^*(\theta),q^*(\theta),\theta)}{\partial q(\theta)} = -l(\epsilon(\theta)), \tag{36}$$

$$\dot{q}^*(\theta) = -\frac{\partial H(U^*(\theta),\epsilon^*(\theta),q^*(\theta),\theta)}{\partial U(\theta)} = p(\theta), \tag{37}$$

$$\frac{\partial H(U^*(\theta),\epsilon(\theta),q^*(\theta),\theta)}{\partial \epsilon(\theta)} = 0, \tag{38}$$

$$H(U^*(\theta),\epsilon^*(\theta),q^*(\theta),\theta) \geq H(U^*(\theta),\epsilon(\theta),q^*(\theta),\theta), \tag{39}$$

$$q^*(\underline{\theta}) = 0. \tag{40}$$

From conditions (37) and (40), we know that $q^*(\theta) = P(\theta)$, where $P(\theta)$ is the cumulative density function. We assume that the ED user privacy preference θ follows uniform distribution[4] within $[\underline{\theta},\bar{\theta}]$, then the probability density function is:

$$p(\theta) = \frac{1}{\bar{\theta}-\underline{\theta}}, \quad \theta \in [\underline{\theta},\bar{\theta}], \tag{41}$$

and the cumulative density function is given by:

$$P(\theta) = \frac{\theta-\underline{\theta}}{\bar{\theta}-\underline{\theta}}, \quad \theta \in [\underline{\theta},\bar{\theta}]. \tag{42}$$

Given the assumption of the uniform distribution, the optimal control variable $\epsilon^*(\theta)$ can be derived by maximizing the Hamiltonian following the first order condition (38):

$$-p(\theta)\left(-\frac{a\lambda_2}{\epsilon^2(\theta)}-\frac{2a\lambda_3}{\epsilon^3(\theta)}\right) - (\theta p(\theta)+q(\theta))b_1 e^{-\frac{b_2}{\epsilon(\theta)}}\frac{b_2}{\epsilon^2(\theta)} = 0, \tag{43}$$

then we solve the above equation and obtain the expression of $\epsilon(\theta)$:

$$\epsilon(\theta) = \frac{2b_2\lambda_3}{2\lambda_3\ W(2b_3(\theta+b_4)-b_3\underline{\theta}))-b_2\lambda_2}, \tag{44}$$

where $W(\cdot)$ denotes the *Lambert W function* [35], b_4 is a constant coefficient, and b_3 is a simplified coefficient:

$$b_3 = \frac{b_1 b_2^2 \exp\left(\frac{b_2\lambda_2}{2\lambda_3}\right)}{2a\lambda_3}, \tag{45}$$

Besides, the data leakage $l(\epsilon(\theta))$ is expressed by:

$$l(\epsilon(\theta)) = b_1 \exp\bigg(-W(2b_3(\theta+b_4)-b_3\underline{\theta}))+\frac{b_2\lambda_2}{2\lambda_3}\bigg). \tag{46}$$

By conducting the integration of the first order derivative (Equation (36)) and following the boundary condition (Equation (33)), we have the expression of $U(\theta)$:

$$\begin{aligned} U(\theta) = \int_{\underline{\theta}}^{\bar{\theta}} -l(\epsilon(\theta))\mathrm{d}\theta = -\frac{b_1\exp(\frac{b_2\lambda_2}{2\lambda_3})}{4b_3}(&W^2(2b_3(\theta+b_4) \\ &- b_3\underline{\theta}) + 2\ W(2b_3(\theta+b_4)-b_3\underline{\theta})) + b_5, \end{aligned} \tag{47}$$

where b_5 is a constant coefficient.

To this end, we have got the optimal solution, $U(\theta)$ (47) and $\epsilon(\theta)$ (44), to the reformulated optimization problem of U^{AP} (34). And we can derive the expression of $R(\theta)$:

$$\begin{aligned} R(\theta) &= U(\theta)+\theta l(\epsilon(\theta)) \\ &= \frac{b_1\exp(\frac{b_2\lambda_2}{2\lambda_3})}{4b_3}(W^2(2b_3(\theta+b_4)-b_3\underline{\theta}) \\ &\quad + 2\ W(2b_3(\theta+b_4)-b_3\underline{\theta})) \\ &\quad + b_1\theta\exp\bigg(-W(2b_3(\theta+b_4)-b_3\underline{\theta}))+\frac{b_2\lambda_2}{2\lambda_3}\bigg)+b_5, \end{aligned} \tag{48}$$

then $R(\theta)$ (48) and $\epsilon(\theta)$ (44) are the optimal solution to the original optimization problem of U^{AP} (17). And U_{AP} is:

$$\begin{aligned} U^{AP} = a(C-\lambda_1) - N\Bigg(&(4c_1-c_2+c_3)+(c_2-2c_1)W(x) \\ &+c_1\ W^2(x)-\frac{4c_1-c_2}{W(x)} \\ &+\frac{b_1\exp(\frac{b_2\lambda_2}{2\lambda_3})x(W^3(x)+2W(x)-2)}{8{b_3}^2W(x)} \\ &+\frac{1}{2b_3}\left(\frac{x}{2b_3}\left(W(x)+\frac{1}{W(x)}-1\right)\right. \\ &\left.\left.-\frac{W(x)(W(x)+2)}{2b_4}\right)\right)\bigg|_{\underline{\theta}}^{\bar{\theta}}, \end{aligned} \tag{49}$$

where $x = 2b_3(\theta+b_4)$, $c_1 = \frac{1}{{b_2}^2}$, $c_2 = \frac{\lambda_2-\lambda_3}{b_2\lambda_3}$ and $c_3 = \frac{{\lambda_2}^2}{4{\lambda_3}^2}-\frac{\lambda_2}{2\lambda_3}$.

[4] The assumption of the uniform distribution can be easily extended to other distributions as well.

V. Cloud Layer Strategy Optimization

In this section, we first present the CC utility optimization algorithm, then analyze the Nash Equilibrium of the three-layer Stackelberg game.

A. Optimization Algorithm

We have obtained the optimal contract $(R(\theta), \epsilon(\theta))$ in the last section, which is the solution to the sub-problem between APs and EDs. In fact, the solution to the sub-problem is based on the pre-determined incentive assignment strategy of the CC, represented by the assignment coefficients $\{a_i\}$. The CC holds an expectation about the strategies (contracts) of APs and we can substitute the expected $\epsilon(\theta)$ function (44) into the CC utility (22):

$$\begin{aligned} U^{CC} &= C - {\lambda_1}' - \sum_{i=1}^{M} a_i (C_i - \lambda_{1,i}) \\ &\quad - \sum_{i=1}^{M} N_i \int_{\underline{\theta}_i}^{\bar{\theta}_i} \frac{1}{\bar{\theta}_i - \underline{\theta}_i} \left(\frac{(a_i \lambda_{2,i} - {\lambda_2}')(2\lambda_{3,i} W(x) - b_{2,i}\lambda_{2,i})}{2 b_{2,i} \lambda_{3,i}} \right. \\ &\quad \left. + \frac{(a_i \lambda_{3,i} - {\lambda_3}')(2\lambda_{3,i} W(x) - b_{2,i}\lambda_{2,i})^2}{4 {b_{2,i}}^2 {\lambda_{3,i}}^2} \right) \mathrm{d}\theta, \end{aligned} \tag{50}$$

where the result of the third term integration is:

$$\begin{aligned} Int = & \Bigg((4d_1 - d_2 + d_3) + (d_2 - 2d_1) W(x) \\ & + d_1\ W^2(x) - \frac{4d_1 - d_2}{W(x)} \Bigg) \Bigg|_{\underline{\theta}_i}^{\bar{\theta}_i}, \end{aligned} \tag{51}$$

and the replacements $x = 2b_{3,i}(\theta + b_{4,i}) - b_{3,i}\underline{\theta}$, $d_1 = \frac{a_i \lambda_{3,i} - {\lambda_3}'}{{b_{2,i}}^2}$, $d_2 = \frac{\lambda_{2,i}{\lambda_3}' - {\lambda_2}'\lambda_{3,i}}{b_{2,i}\lambda_{3,i}}$, and $d_3 = \frac{2\lambda_{2,i}\lambda_{3,i}{\lambda_3}' - a_i {\lambda_{2,i}}^2 \lambda_{3,i} - {\lambda_{2,i}}^2 {\lambda_3}'}{4{\lambda_{3,i}}^2}$ are for simplicity.

The optimization of the CC utility is NP-hard due to the non-convexity, and we propose an iterative algorithm for optimizing the U^{CC} by gradient ascent (see Algorithm 1). The partial derivatives of U^{CC} with respect to a_i is represented by $\frac{\partial U^{CC}}{\partial a_i}$, which is given by:

$$\begin{aligned} \frac{\partial U^{CC}}{\partial a_i} &= -(d_i - \lambda_{1,i}) \\ & - d_4 N_i * \left[\left(\frac{4\lambda_{3,i}}{{b_2}^2 a_i} - \frac{1}{{a_i}^2} \left(\frac{4(a_i\lambda_{3,i} - {\lambda_3}')}{{b_2}^2} \right) \right) \right. \\ & + \frac{{\lambda_3}'}{{b_{2,i}}^2 {a_i}^2} \left(W^2\left(\frac{d_4}{a_i}\right) - 2\ W\left(\frac{d_4}{a_i}\right) - \frac{4}{W\left(\frac{d_4}{a_i}\right)} \right) + \\ & \frac{{\lambda_3}' - \lambda_{3,i} a_i}{{b_{2,i}}^2 {a_i}^2 (W\left(\frac{d_4}{a_i}\right) + 1)} \left(2\ W^2\left(\frac{d_4}{a_i}\right) - 2W\left(\frac{d_4}{a_i}\right) + \frac{4}{W\left(\frac{d_4}{a_i}\right)} \right) \\ & \left. + d_2 d_4 \left(\frac{1}{{a_i}^2 (W^2\left(\frac{d_4}{a_i}\right) + W\left(\frac{d_4}{a_i}\right))} - \frac{1}{{a_i}^2 W\left(\frac{d_4}{a_i}\right)} \right) \right] \Bigg|_{\underline{\theta}_i}^{\bar{\theta}_i}, \end{aligned} \tag{52}$$

where $d_4 = \frac{b_{1,i} {b_{2,i}}^2 \exp\left(\frac{b_{2,i}\lambda_{2,i}}{2\lambda_{3,i}}\right)(\theta_i + b_{4,i})}{\lambda_{3,i}}$.

Algorithm 1. The CC Utility Optimization Algorithm

```
Input: Initialized $\{a_i\}$ and other parameters.
Output: Optimized $\{a_i\}$.
for $k$ in range $(1, K)$ do
    if The differences of $\{a_i\}$ between two adjacent iterations are
    all within the threshold $\{|a_i^k - a_i^{k-1}| \leq \nu\}$. then
        Break.
    end if
    Randomly shuffle the order of the $M$ parameters $\{a_i^k\}$.
    for $i$ in range $(1, M)$ do
        Choose the coefficient $a_i^k$ following the shuffled order.
        Fix other coefficients $\{a_{-i}^k\}$.
        for $r$ in range $(1, R)$ do
            if $a_i^{k,r-1}$ is out of the normal rangethen
                Set $a_i^{k,r-1}$ back to the normal range.
                Break.
            end if
            Calculate the partial gradient of $U^{CC}$ w.r.t. $a_i$:
            $\nabla_{a_i^{k,r-1}} U^{CC} = \frac{\partial U^{CC}}{\partial a_i^{k,r-1}}$.
            Conduct a step of gradient ascent
            $a_i^{k,r} = a_i^{k,r-1} + \eta \nabla_{a_i^{k,r-1}} U^{CC}$.
            if The difference is within the threshold
            $|a_i^{k,r} - a_i^{k-1,r-1}| \leq \nu$ then
                Break.
            end if
        end for
    end for
    Rearrange the values of $\{a_i^{k,R}\}$ by $a_i = \frac{a_i^{k,R}}{\sum_{i=1}^{M} a_i^{k,R}}$ to ensure that
    $\sum_{i=1}^{M} a_i^{k,R} = 1$.
    Replace the old values $\{a_i^k\}$ with the new values
    $\{a_i^{k+1} = a_i^{k,R}\}$.
end for
```

During each iteration, we randomly choose a coefficient a_i and fix other coefficients $\{a_{-i}\}$. Then we update a_i by gradient ascent until the partial gradient with respect to a_i is within the threshold. The same process goes for all the M coefficients. The procedure is repeated until all the partial gradients are within the threshold, which means the coefficients converge to stable values, and we obtain the optimized coefficients $\{a_i\}$.

B. Nash Equilibrium Analysis

In the first stage of the Stackelberg game between the CC and APs, the CC announces its incentive assignment strategy. Then in the second stage, each AP strategizes its contract design to maximize its own utility, which is a non-cooperative game. Given the overall incentive, there exists a stable strategy for each AP such that the utility of an AP will only decrease if the AP unilaterally changes its strategy from the optimal strategy under Nash Equilibrium:

Definition 4 (Nash Equilibrium): A set of strategies $\{G_1^{NE}, G_2^{NE}, \ldots, G_M^{NE}\}$ is a Nash Equilibrium of the second stage if for any AP$_i$,

$$U_i^{AP}(G_i^{NE}, G_{-i}^{NE}) \geq U_i^{AP}(G_i, G_{-i}^{NE}), \tag{53}$$

where G_i denotes the strategy of AP$_i$, G_i^{NE} is the strategy under Nash Equilibrium, and G_{-i} denotes the strategies of all other APs.

To study the Nash Equilibrium of the second stage game in the Stackelberg game between the CC and APs, we consider the discrete form of U^{AP}:

$$U^{AP}(G_i, G_{-i}) = \frac{G_i}{\sum_{i=1}^{M} G_i} I - \sum_{j=1}^{N_i} R_{i,j}, \tag{54}$$

$$G_i = C_i - A_i, \tag{55}$$

$$A_i = \lambda_{1,i} + \lambda_{2,i} \sum_{j=1}^{N_i} \frac{1}{\epsilon_{i,j}} + \lambda_{3,i} \sum_{j=1}^{N_i} \frac{1}{\epsilon_{i,j}{}^2}, \tag{56}$$

where I is the total incentive provided by the CC. The original strategies of the APs are the contract menus provided to the EDs which can not be directly represented, so we use the intermediate result of the strategies (i.e. the model profit $G_i = C_i - A_i$) for analysis instead.

Then we derive the first-order derivative of $U^{AP}(G_i, G_{-i})$:

$$\frac{\partial U^{AP}(G_i, G_{-i})}{\partial G_i} = \frac{\sum_{/i} G_i}{(\sum_{i=1}^{M} G_i)^2}, \tag{57}$$

and the second-order derivative of $U^{AP}(G_i, G_{-i})$ is given by:

$$\frac{\partial^2 \, U^{AP}(G_i, G_{-i})}{\partial G_i{}^2} = -\frac{2\sum_{/i} G_i}{(\sum_{i=1}^{M} G_i)^3} \leq 0. \tag{58}$$

As the second-order derivative is below zero, we know that the utility reaches the maximum when AP$_i$ chooses the strategy which achieves G_i. When all the APs choose their strategies that satisfy the above conditions and maximize their utilities, the strategies come to a stable state, which follows the Nash Equilibrium.

Proposition 1 (Convergence of the CC utility optimization algorithm): After a limited number of iterations, the coefficients $\{a_i\}$ obtained by applying the algorithm converge to stable values and the strategies of APs reach NE.

Proof: Convergence of the optimization algorithm depends on Step 15 ($a_i^{k,r} = a_i^{k,r-1} + \eta \nabla_{a_i^{k,r-1}} U^{CC}$) of the algorithm. As the CC utility gradually approaches the maximum, the absolute value of the partial gradients $\nabla_{a_i^{k,r-1}}$ approach zero. When the differences $|a_i^{k,r} - a_i^{k,r-1}|$ between iterations of all $\{a_i\}$ are within the threshold μ, the coefficients converge to stable values. ■

VI. Numerical Evaluation

TABLE I
The Parameter Settings for Numerical Evaluation

Parameters	Values
Number of Access Points M	$M = 5$
Number of Edge Devices N_i and N_0	$N_i = 10$, $N_0 = 50$
The aggregation frequency T_1 and T_2	$T_1 = 10$, $T_2 = 6$
The data leakage parameter b_1	$b_1 = 0.04$
The data leakage parameters $b_2{}'$ and b_2	$b_2{}' = 39.5$, $b_2 = 1.375$
The constant coefficients b_4 and b_5	$b_4 = 0.03$, $b_5 = 0.035$
Learning rate η	$\eta = 0.01$
Fail probability of differential privacy δ	$\delta = 0.1$
β-Lipschitz loss function	$\beta = 0.5$
ρ-Lipschitz smooth loss function	$\rho = 0.5$
Dataset size D	$D = 900$
Parameter dimension d	$d = 100$
The clipping parameter L	$L = 1$
The range of θ $[\underline{\theta}, \bar{\theta}]$	$\underline{\theta} = 0.0$, $\bar{\theta} = 1.0$

In Section IV and V, we have presented the optimal contract design and the optimization of TLSG respectively. In this section, we conduct numerical experiments based on the previous theoretic analysis, and evaluate the incentive mechanism and privacy preservation results. The DP-FedAvg [36] algorithm, which applies DP to the two-layer FedAvg scheme and adds perturbation on local model updates, is used as the benchmark. Before the numerical evaluation, we summarize the parameter settings in Table I.

A. Incentive Mechanism Evaluation

In Section IV, we solved the optimal contract design problem and obtained the expressions of $R(\theta)$ (48), $\epsilon(\theta)$ (44) and $U(\theta)$ (32) that compose the contract. We first check the IC (Definition 2) and IR (Definition 3) constraints that a reasonable contract must follow, by sampling a set of contract types θ and drawing the corresponding lines of the ED utility over different types:

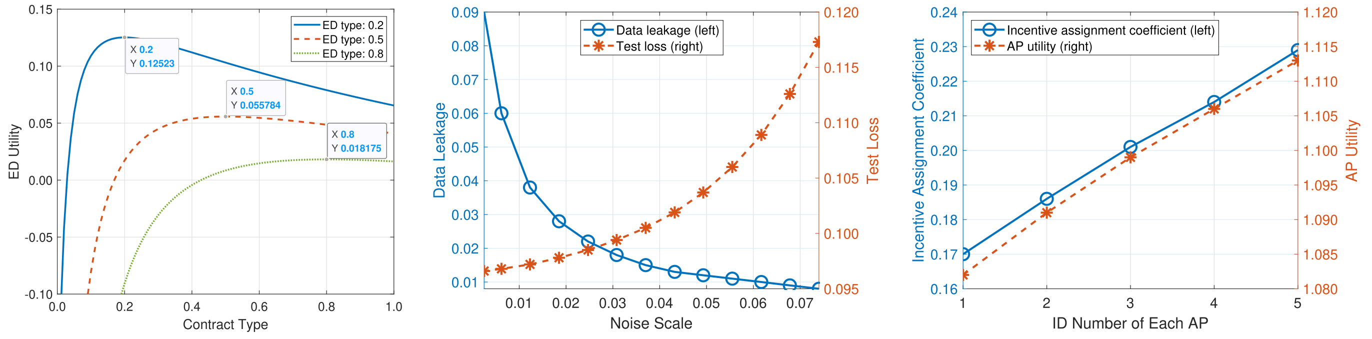
$$U(\tilde{\theta}) = R(\tilde{\theta}) - \theta l(\epsilon(\tilde{\theta})), \tag{59}$$

where θ represents the contract type and $\tilde{\theta}$ is the variable.

As shown in Fig. 5(a), the ED utility reaches the maximum when $\tilde{\theta} = \theta$, represented by the three mark points of different lines. Therefore, each ED of type θ maximizes its utility by choosing the contract designed for its type, which follows the IC constraint. When choosing the designed contracts, the ED utilities are all larger than zero, which follows the IR constraint. We can further note that, for EDs with high privacy preference, the utility is below zero when they choose a contract designed for EDs with low privacy preference, because the data leakage overwhelms the received reward under this condition.

Then we demonstrate the trade-off between private data leakage and model accuracy, where the model accuracy is represented by the global model test loss that is negatively correlated with the model accuracy. As shown in Fig. 5(b), the designed incentive mechanism achieves the privacy-accuracy trade-off which is tunable by the added noise scales. With the growth of the noise scale, the private data leakage decreases while the test loss increases.

To evaluate the optimization results of algorithm 1, we demonstrate the incentive assignment coefficients (the left y-



(a) (b) (c)

Fig. 5. (a) The ED utility with different contract types. (b) The trade-off between the private data leakage and the model accuracy (represented by the test loss, which is negatively correlated with model accuracy). (c) The incentive assignment coefficient (left) and the utility (right) of each AP.

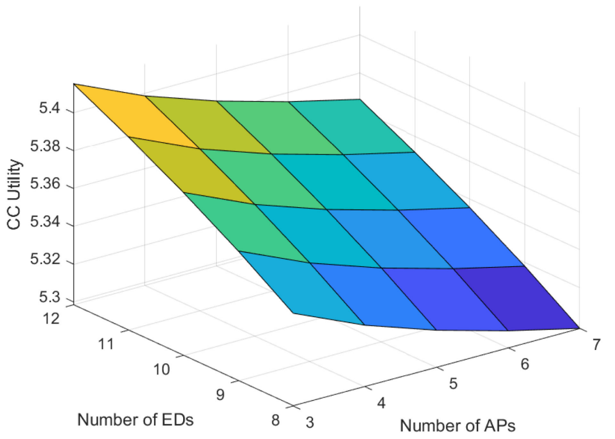


Fig. 6. The CC utility with different M and N_i settings.

axis in Fig. 5(c)) and the utilities (the right y-axis in Fig. 5(c)) of different APs. The APs are in connection with different numbers of EDs, i.e. $N_i = [8, 9, 10, 11, 12]$ and the total number of EDs is $N_0 = 50$. As the number of participating EDs grows, the incentive assignment coefficients and AP utilities both increase and have similar trends. This is because more EDs can provide more private data contributions to the FL task.

In Fig. 6, we show the performance of the proposed incentive mechanism in terms of the CC utility with different M and N_i. With the growth of M, the CC utility decreases because more incentives are provided to APs which brings more cost. With the growth of N_i, the CC utility increases because more contributions are provided by EDs.

B. PFCEL Evaluation

We evaluate the performance of PFCEL on the MNIST dataset. For comparison, we also conduct experiments with DP-FedAvg, under the same noise scales for privacy preservation. The experiments on DP-FedAvg only have CC and EDs. The number of EDs for DP-FedAvg is the same as that of PFCEL ($N_0 = 50$). The evaluation metrics include the accuracy and loss on both training datasets and test datasets, where the accuracy is defined as:

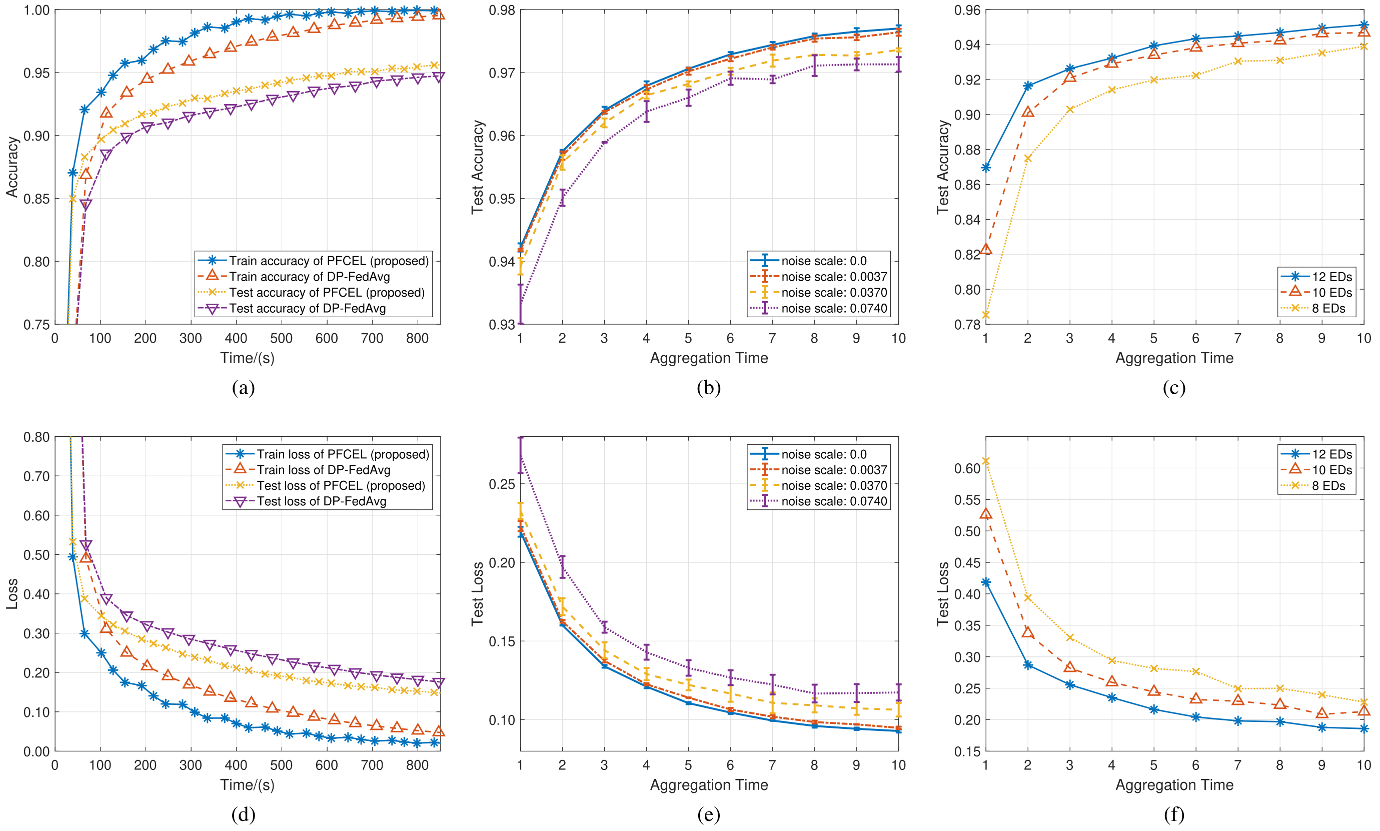
$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}, \tag{60}$$

and the loss is defined by Equation (3).

In Fig. 7(a), we compare the global model train accuracy and test accuracy of PFCEL and DP-FedAvg. We can note that PFCEL converges faster and achieves higher accuracy than DP-FedAvg. In Fig. 7(d), we compare the global model train loss and test loss of PFCEL and DP-FedAvg. The results also demonstrate that PFCEL converges faster and achieves lower loss than DP-FedAvg. Besides, the train accuracy is higher than the test accuracy, which is because of the generalization property of the model [37].

In Fig. 7(b) and Fig. 7(e), we compare the global model test accuracy and test loss of PFCEL with different noise scales. The lines denote the average values, while the error bars represent the maximum and minimum values. As demonstrated, the test accuracy decreases and the test loss increases with the growth of noise scales. We can note that the test accuracy and test loss under noise perturbations are close to those with zero noise scales, which show that PFCEL not only preserves privacy but also ensures the model performance. Besides, we observe from the error bars that the variance of the model accuracy increases with the growth of noise scales.

In Fig. 7(c) and Fig. 7(f), we show the training efficiency of PFCEL by comparing the global model test accuracy and test loss with different N_i under non-i.i.d. setting (each ED with 5 categories of images instead of the whole 10 categories in MNIST). With the growth of N_i, the test accuracy increases and the test loss decreases, because more participating EDs can decrease the non-i.i.d. degree of the whole data set.



(a) (b) (c)

(d) (e) (f)

Fig. 7. (a) The comparisons of global model train accuracy and test accuracy between PFCEL and DP-FedAvg. (b) The comparison of PFCEL global model test accuracy with different noise scales. (c) The comparison of PFCEL global model test accuracy with different number of EDs. (d) The comparisons of global model train loss and test loss between PFCEL and DP-FedAvg. (e) The comparison of PFCEL global model test loss with different noise scales. (f) The comparison of PFCEL global model test loss with different number of EDs.

VII. Conclusion

In this paper, we have designed a privacy-preserving incentive mechanism for the federated cloud-edge learning system. The upper layers in the system offer monetary incentives to the lower layers, to compensate for the model updates provided by the lower layers using private data. We have formulated the problem as a three-layer Stackelberg game and the sub-problem as an optimal contract design problem. The optimal strategies of each layer are derived by applying optimal control method and gradient ascent algorithm. Through a comprehensive set of theoretical analysis and numerical evaluations, we draw the following conclusions:

- Given the same level of privacy preservation, i.e. the same scale of noises, the proposed PFCEL system achieves faster convergence and higher accuracy than conventional methods within the same training time on both training dataset and test dataset.
- The designed incentive mechanism achieves a trade-off between the private data leakage and the model accuracy, which is tunable by the added noise scales.
- To guarantee the target model accuracy, a moderate scale of noises (i.e., $\sigma < 0.0370$ in our scenario) is essential since a too large scale can result in a large variance of the model accuracy.

References

[1] J. Ren, G. Yu, Y. He, and G. Y. Li, “Collaborative cloud and edge computing for latency minimization,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.

[2] P. Wang, Z. Zheng, B. Di, and L. Song, “HetMEC: Latency-optimal task assignment and resource allocation for heterogeneous multi-layer mobile edge computing,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4942–4956, Oct. 2019.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105, .

[5] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, “Deep face recognition: A survey,” in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images*, 2018, pp. 471–478.

[6] A. Esteva *et al.*, “A guide to deep learning in healthcare,” *Nature Med.*, vol. 25, no. 1, pp. 24–29, 2019.

[7] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the gan: Information leakage from collaborative deep learning,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 603–618.

[8] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 27, pp. 2672–2680.

[9] L. Zhu, Z. Liu, and S. Han, “Deep leakage from gradients,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14774–14784.

[10] C. Dwork *et al.*, “The algorithmic foundations of differential privacy,” *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2014.

[11] W. Du and M. J. Atallah, “Secure multi-party computation problems and their applications: A review and open problems,” in *Proc. Workshop New Secur. Paradigms*, 2001, pp. 13–22.

[12] C. Gentry *et al.*, A fully homomorphic encryption scheme. Ph.D. dissertation, Stanford Univ., Stanford, Stanford, CA, USA, 2009.

[13] K. Wei *et al.*, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, Apr. 17, 2020.

[14] K. Wei *et al.*, “User-level privacy-preserving federated learning: Analysis and performance optimization,” *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2021.3056991.

[15] F. AI, “Data monetization in ai marketplaces,” 2020. [Online]. Available: https://medium.com/@federata/data-monetization-in-ai-marketplaces-49b76a3e591c

[16] C. Niu, Z. Zheng, S. Tang, X. Gao, and F. Wu, “Making big money from small sensors: Trading time-series data under pufferfish privacy,” in *Proc. IEEE INFOCOM -IEEE Conf. Comput. Commun.*, 2019, pp. 568–576.

[17] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, “Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[18] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, “A learning-based incentive mechanism for federated learning,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[19] W. Y. B. Lim *et al.*, “Hierarchical incentive mechanism design for federated machine learning in mobile networks,” *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9575–9588, Oct. 2020.

[20] C. Ying, H. Jin, X. Wang, and Y. Luo, “Double insurance: Incentivized federated learning with differential privacy in mobile crowdsensing,” in *Proc. Int. Symp. Reliable Distrib. Syst.*, 2020, pp. 81–90.

[21] L. U. Khan *et al.*, “Federated learning for edge networks: Resource optimization and incentive mechanism,” *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Oct. 2020.

[22] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, “HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[23] W. Wu, L. He, W. Lin, and R. Mao, “Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1539–1551, Jul. 2021.

[24] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, “From federated to fog learning: Distributed machine learning over heterogeneous wireless networks,” *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 41–47, Dec. 2020.

[25] T. Basar and G. J. Olsder, “Dynamic noncooperative game theory,” *Soc. Ind. Appl. Math.*, 1998.

[26] J. Hu, Z. Zheng, B. Di, and L. Song, “Multi-layer radio network slicing for heterogeneous communication systems,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2378–2391, Oct.-Dec. 2020.

[27] Z. Zhang and M. R. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in *Proc. 32nd Conf. Neural Inf. Process. Syst.*, 2018, pp. 8778–8788.

[28] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, “Inverting gradients-how easy is it to break privacy in federated learning?,” in *Adv. Neural Inform. Process. Syst.*, pp. 16937–16947, 2020.

[29] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[30] B. Liu, Y. Li, Y. Liu, Y. Guo, and X. Chen, “PMC: A privacy-preserving deep learning model customization framework for edge computing,” *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 4, pp. 1–25, 2020.

[31] J. Wu *et al.*, “Hierarchical personalized federated learning for user modeling,” in *Proc. Web Conf.*, 2021, pp. 957–968.

[32] J.-J. Laffont and D. Martimort, *The Theory of Incentives: The Principal-Agent Model*. Princeton: Univ. Press, Princeton, NJ, USA, 2009.

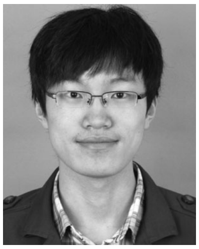
[33] L. Xu, C. Jiang, Y. Chen, Y. Ren, and K. R. Liu, “Privacy or utility in data collection? A contract theoretic approach,” *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1256–1269, Oct. 2015.

[34] D. E. Kirk, *Optimal Control Theory: An Introduction*. Courier Corporation, New York, NY, USA, 2012.

[35] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, “On the lambert w function,” *Adv. Comput. Math.*, vol. 5, no. 1, pp. 329–359, 1996.

[36] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, “Learning differentially private recurrent language models,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[37] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge: MIT Press, 2016.



Tianyu Liu (Graduate Student Member, IEEE) received the M.S. degree from the School of Software and Microelectronics, Peking University, Beijing, China, in 2018. He is currently working toward the Ph.D. degree with the School of Electrical Engineering and Computer Science, Peking University. His research interests include edge computing, federated learning, and data privacy.

Boya Di (Member, IEEE) received the B.S. degree in electronic engineering from Peking University, Beijing, China, in 2014 and the Ph.D. degree from the Department of Electronics, Peking University, China, in 2019. She was a Postdoc Researcher with Imperial College London, London, U.K., and is currently an Assistant Professor with Peking University. Her current research interests include reconfigurable intelligent surfaces, multiagent systems, edge computing, vehicular networks, and aerial access networks. One of her journal papers is currently listed as ESI highly cited papers. Since June 2020, she has been an Associate Editor for the IEEE Transactions on Vehicular Technology. She was also a TPC Member in GlobeCom 2016, GlobeCom 2020, ICCC 2017, ICC 2016, ICC 2018, and VTC 2019.

Peng An (Member, IEEE) received the B.S. and M.S. degrees from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2004 and 2012, respectively. He is currently the General Manager of R&D Center of Beijing Wondersoft Technology Corp., Ltd, Beijing, China. His research interests include network security, with an emphasis on data security and mechanism design in data security management systems.

Lingyang Song (Fellow, IEEE) received the Ph.D. degree from the University of York, York, U.K., in 2007. He was a Postdoctoral Research Fellow with the University of Oslo, Oslo, Norway, and Harvard University, Cambridge, MA, USA, until rejoining Philips Research U.K., in March 2008. In May 2009, he joined the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, as a Full Professor. He has authored or coauthored extensively, wrote six text books, and is co-inventor of a number of patents (standard contributions). His main research interests include cooperative and cognitive communications, physical layer security, and wireless ad hoc or sensor networks. He was the recipient of nine paper awards in IEEE journal and conferences, including IEEE Journal on Selected Areas in Communications 2016, IEEE WCNC 2012, ICC 2014, Globecom 2014, and ICC 2015. He is currently on the Editorial Board of the IEEE Transactions on Wireless Communications and Journal of Network and Computer Applications. He was the TPC Co-Chair for the International Conference on Ubiquitous and Future Networks (ICUFN2011/2012), symposium Co-Chair in the International Wireless Communications and Mobile Computing Conference (IWCMC 2009/2010), IEEE International Conference on Communication Technology (ICCT2011), and IEEE International Conference on Communications (ICC 2014, 2015). He was the recipient of 2012 IEEE Asia Pacific Young Researcher Award. Since 2015, he has been an IEEE ComSoc Distinguished Lecturer. He was the recipient of the K. M. Stott Prize for excellent research.