IMFL: An Incentive Mechanism for Federated Learning With Personalized Protection

Mengqian Li, Youliang Tian, *Senior Member, IEEE*, Junpeng Zhang, Zhou Zhou, Dongmei Zhao, and Jianfeng Ma, *Member, IEEE*

***Abstract*—Federated learning (FL) allows clients to keep local data sets and train collaboratively by uploading model gradients, which achieves the goal of learning from fragmented sensitive data. Although FL prevents clients' data sets from being shared directly, local private information may be leaked through gradients. To mitigate this problem, we combine game theory to design an FL scheme (incentive mechanism for the FL) based on the incentive mechanism and differential privacy (DP). First, we explore three DP variants, all of which are resistant to deep leakage from gradients (DLG) but differ in their level of privacy protection. In addition, we perform the convergence analysis of the FL model based on DP. Then, with the assistance of game theory, we analyze the natural state of the server and clients in the FL process and formulate the utility function of both sides under the case of considering the attack. Finally, we establish the optimization problem as a Stackelberg game and solve for the optimal strategy of the server and clients by deriving the Nash equilibrium to achieve personalized protection. Theoretical proof demonstrates that both types of entities can achieve optimal actions by maximizing their utility functions upon reaching the Nash equilibrium. Besides, extensive experiments are conducted on real-world data sets to demonstrate that the IMFL is efficient and feasible.**

***Index Terms*—Differential privacy (DP), federated learning (FL), privacy preserving, Stackelberg game.**

Manuscript received 31 December 2023; revised 28 February 2024; accepted 6 April 2024. Date of publication 12 April 2024; date of current version 26 June 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3101100; in part by the National Natural Science Foundation of China under Grant 62272123 and Grant 62062020; in part by the Central Government Guides Local Science and Technology Development Found Projects under Grant 216Z0701G; in part by the Science and Technology Program of Hebei Province under Grant 22567606H; in part by the Project of High-Level Innovative Talents of Guizhou Province under Grant [2020]6008; in part by the Science and Technology Program of Guizhou Province under Grant [2020]5017 and Grant [2022]065; and in part by the Science and Technology Program of Guiyang under Grant [2022]2-4. *(Corresponding author: Youliang Tian.)*

Mengqian Li, Youliang Tian, and Zhou Zhou are with the State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China (e-mail: mqli88@163.com; youliangtian@163.com; mitzhouzhou@163.com).

Junpeng Zhang is with the School of Cyber Engineering, Xidian University, Xi'an 710071, China, and also with the Hebei Key Laboratory of Network and Information Security, Hebei Normal University, Shijiazhuang 050024, China (e-mail: zhangjunpeng@hebtu.edu.cn).

Dongmei Zhao is with the Hebei Key Laboratory of Network and Information Security, College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang 050024, China (e-mail: zhaodongmei666@126.com).

Jianfeng Ma is with the State Key Laboratory of Integrated Services Network, School of Cyber Engineering, Xidian University, Xi'an 710071, China (e-mail: jfma@mail.xidian.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3387973

I. Introduction

With the global acceleration of the digital economy, propelled by rapid advancements in 5G, artificial intelligence (AI), and the Internet of Things (IoT), data has become a critical strategic resource affecting global competition. As a novel data fusion framework, federated learning (FL) overcomes the bottleneck of traditional centralized computing architecture like machine learning (ML) [1]. It enhances data quality and availability, unlocking the intrinsic value of data resources. Technically, FL [2] is a distributed collaborative ML paradigm where the data training task is placed on multiple local clients with private data sets. Then, the central server aggregates the updated parameters to train the global model [3]. This approach circumvents direct access to raw data sets, preserving high utility and potentially mitigating risks of private information leaks. With its practical and innovative concept, FL has been deployed in various domains, including smart healthcare [4], [5], smart transportation [6], smart industry [7], and so on.

The distributed characteristic provides a degree of data privacy protection in FL. However, it is insufficient to protect against some internal and external attacks, for example, deep gradient leakage attacks (DLG) [8] and membership inference attacks [9]. A malicious adversary could attack the training model, deviating the target model from the correct result, affecting availability [10], and possibly recovering the original image of the training data set of participants by intercepting the gradient. To enhance data privacy in FL, differential privacy (DP) techniques [11] are utilized to mask interaction parameters between the server and clients. Uploading noisy model gradients prevents malicious adversaries from deducing client's private information during model training. Secure aggregation by the server cancels out noise from client submissions, ensuring FL model convergence. This fuzzing does not generate high communication or computational overheads compared to encryption methods, such as homomorphic encryption (HE) and secure multiparty computation (SMC). Besides, one advantage of this technology is the ability to autonomously adjust the strength of privacy protection, providing personalized DP protection. Recent works [12], [13] focus on employing DP techniques to safeguard sensitive information in FL training. However, these studies overlook the varying levels of privacy protection that DP can offer across different FL locations, such as record-level and client-level protection. In this article, we conduct a detailed analysis of DP applications in FL.

An initial significant challenge for DP, while providing lightweight privacy protection, is ensuring the utility of the data. DP contains a parameter that depicts the strength of information protection, known as privacy budget, which is used to measure the effectiveness of privacy protection. As privacy budgets increase, noise scale decreases while data utility improves. Previous works on DP [14], [15] have primarily focused on privacy protection against attackers, neglecting data utility. However, with the widespread adoption of DP, privacy protection now extends beyond single privacy concerns to encompass data utility [16], [17], [18]. Existing research [19], [20] incentivizes clients to contribute high-quality data, some incorporating reputation mechanisms [21], [22] to enhance data utility. Nonetheless, no work considers adversary attack behavior, hindering the achievement of an effective balance between data privacy and utility in the presence of adversaries. Therefore, addressing the tradeoff between privacy and data utility while leveraging DP techniques to thwart attacks becomes an urgent concern.

To tackle the above problems, we introduce an incentive mechanism for the FL scheme (IMFL) scheme integrated with DP and formulate a game theory model to optimize the balance between privacy and utility. We supplement the traditional FL system with an incentive mechanism, which aims at motivating clients to incorporate DP noise and upload high-quality model gradients. In this system, we consider the server as *honest-but-curious*, posing a potential threat to model training, which is capable of stealing client model training information. In local training, each client adds personalized noise to the gradients with DP before uploading. However, the DP technique would reduce data utility, and thus we take the incentive mechanism into account by constructing respective utility functions for the server and the client, as well as solving them with the Stackelberg game to derive optimal solutions for both sides. In addition, we analyze the above game model and prove that satisfy the Nash equilibrium. Eventually, the experimental simulation results show that the proposed scheme can enable the clients and server to make privacy-preserving and incentive decisions, respectively, proving the effectiveness of the proposed approach. In general terms, the main contributions of this article are highlighted as follows.

1) We propose an FL incentive mechanism framework leveraging the DP technique, named IMFL, which motivates clients to contribute high-quality local models while safeguarding privacy, aiming to maximize data utility while defending against gradient attacks.
2) We are the first to investigate DP in detail, distinguishing three DP mechanisms, BDP, EDP, and CDP. Then, we show experimentally that all three DP variants are resilient against DLG but provide different privacy protection levels, where the BDP and EDP guarantee record-level protection, and the CDP offers client-level protection.
3) We take advantage of game theory to analyze the natural state of the server and clients. To break the dilemma of both sides, combine the Stackelberg game with an incentive mechanism to find an equilibrium of privacy and utility, which provides personalized protection.
4) We prove the convergence of the FL system based on three DP mechanisms and evaluate the IMFL through extensive simulation experiments, proving the effectiveness and feasibility of this scheme.

The remainder of this article is organized as follows. Section II introduces the preliminaries. Section III presents the IMFL model and the three DP mechanisms. Section IV depicts the problem statement and the Stackelberg game. Section V carries out Nash equilibrium derivation. Section VI demonstrates the experimental results and the related work is reviewed in Section VII. Finally, we conclude this article in Section VIII.

II. Preliminary Knowledge

In this section, we review some basic notions of DP, game theory, and mutual information (MI).

A. Differential Privacy

DP as a probabilistic mechanism can be adapted to provide privacy preservation of distributed data processing systems to prevent information leakage. The definitions of DP are expressed as follows [23].

Definition 1 [$(\varepsilon, \delta)-DP$]: A randomized mechanism $\mathcal{M}: \mathcal{D} \to \mathcal{O}$ with a domain $\mathcal{D}$ of possible all clients training data set and range $\mathcal{O} \subseteq \text{Range}(\mathcal{M})$ satisfies $(\varepsilon, \delta)-$DP for any two adjacent databases $\mathcal{D}, \mathcal{D}' \subseteq \mathcal{D}$ that differ with only a single record $\|\mathcal{D} - \mathcal{D}'\|_1 \leq 1$, we have the following inequation:

$$\Pr[\mathcal{M}(\mathcal{D}) \in O] \leq e^{\varepsilon} \Pr\left[\mathcal{M}\left(\mathcal{D}'\right) \in O\right] + \delta \tag{1}$$

where ε is a positive number serving as privacy budget and $\delta \in [0, 1]$ is the term of relaxation, signifying the probability of an exceptionally rare adverse event. If $\delta = 0$, $\mathcal{M}$ achieves pure ε-DP.

Definition 2 (Gaussian Mechanism): Given the training data set $\mathcal{D}$ and a query function f, the randomized algorithm with Gaussian noise can be presented as $\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + \mathcal{N}(0, \sigma^2 I)$, where $\mathcal{N}(0, \sigma^2 I)$ is the Gaussian distribution that satisfies $\sigma \geq (c \cdot \Delta f/\varepsilon)$ with $c^2 > 2\ln(1.25/\delta)$.

Definition 3 (L_2-Sensitivity): For any two adjacent databases $\mathcal{D}, \mathcal{D}' \subseteq \mathcal{D}$, the l_2-sensitivity of a query function is

$$\Delta f = \max \|f(\mathcal{D}) - f\left(\mathcal{D}'\right)\|_2. \tag{2}$$

B. Nash Equilibrium

The Nash equilibrium, a cornerstone of game theory, will be utilized in our system to conduct theoretical analysis of both clients and the server.

Definition 4 (Nash Equilibrium): Give n players and a game $G = \{S_1, \ldots, S_n; U_1, \ldots, U_n\}$, a strategy profile is a Nash equilibrium if any player has no incentive to deviate from his/her respective strategies after they have considered and anticipated the other player's rational choices or strategies

$$\forall s_i \in S_i, U_i\left(s_i, s_{-i}^*\right) \leq U_i\left(s_i^*, s_{-i}^*\right) \tag{3}$$

where S_i is the set of pure strategies which are available for player i, and U_i is the utility function. s_i is the strategy chosen

from the player i own strategies set. The s_{-i} means the other $(n-1)$ players' possible chosen strategies besides player i.

C. Mutual Information

MI is a concept in information theory used to quantify the amount of information that one random variable carries about another.

Definition 5 (MI): Let X and Y be random variables on the discrete space $\mathcal{X}$ and $\mathcal{Y}$ with mass probability $p(X(i)) = \Pr\{X = X(i)\}, X(i) \in \mathcal{X}$ and $p(Y(j)) = \Pr\{Y = Y(j)\}, Y(j) \in \mathcal{Y}$, respectively. The joint probability mass function of two random variables, X and Y, can be expressed as $p(X(i), Y(j))$. Hence, the MI is defined as

$$I(X;Y) = \sum_{i=1}^{n}\sum_{j=1}^{n} p(X(i),Y(j)) \cdot \log\left(\frac{p(X(i),Y(j))}{p(X(i)) \cdot p(Y(j))}\right) \quad (4)$$

when the variables X and Y are independent of each other, there is $p(X(i), Y(j)) = p(X(i)) \cdot p(Y(j))$. The MI is directly related Shannon's entropy of the variables through the following equations:

$$I(X;Y) = \begin{cases} H(X) - H(X|Y|) \\ H(Y) - H(Y|X|) \\ H(X) + H(Y) - H(X,Y) \end{cases} \quad (5)$$

where $I(X;Y)$ is also unbounded $[0, \infty)$. In the case of jointly continuous random variables, the MI expression is

$$I(X;Y) = \int\int p(X(i),Y(j)) \log\left(\frac{p(X(i),Y(j))}{p_{\mathcal{X}}(X(i)) \cdot p_{\mathcal{Y}}(Y(j))}\right). \quad (6)$$

III. System Model

In this section, we provide an elaborate explanation of the IMFL. Additionally, we introduce the threat model pertinent to this scenario. Then, we describe three DP mechanisms in IMFL to defend against threat attacks. For ease of understanding, the symbols and descriptions involved in the paper are shown in Table III.

A. FL With Incentive Mechanism

In recent years, FL [24] has revolutionized traditional ML model training with its distributed nature. This integration of AI technology has shown superior performance across diverse domains in modern applications. Moving forward, we will provide an overview of the entire IMFL system.

The IMFL system consists of N clients and one server. Each client $i \in N$ holds their own private local data set $\mathcal{D}_i = \{(x_m, y_m)\}_{m=1}^{|\mathcal{D}_i|}$, where $x_m \in \mathbb{R}^d$ is a multidimensional vector with d features and $y_m \in \mathbb{R}$ is the corresponding label. During the process of model training, the server and clients interact through model gradients without sharing the local data set. The training steps of IMFL in Fig. 1 is as follows.

1) *Initialization:* The server initializes the global model ω^1 and incentive mechanism protocol, then broadcasts them to k randomly selected clients.
2) *Local Training:* The clients who are selected to perform training tasks receive a global model and incentive mechanism protocol. Then, the client i calculates gradients $\nabla\mathcal{L}(\omega_i^t)$ through a private data set and upload them with DP noise to the server to update the new global model.
3) *Global Aggregating:* After the server collects all model gradients from local clients, it performs global aggregation and derives a new global model for the next iteration as

$$\begin{aligned} \omega^{t+1} &= \omega^t - \eta\left(\sum_{i\in k} p_i\left(\nabla\mathcal{L}\left(\omega_i^t\right) + n_i^t\right)\right) \\ &= \omega^t - \eta\sum_{i\in k} p_i \hat{g}_i^t. \end{aligned} \quad (7)$$

4) *Incentive Mechanism:* The server collects local gradients from the selected clients to train the global model. Since clients mask their truth gradients with personalized DP before uploading, the server employs an incentive mechanism to encourage clients to contribute more realistic gradients by assigning correspondingly high rewards to clients with less noise addition. The server pays each client i a reward R_i. Thus, the incentive mechanism is represented as

$$R = R_1 + R_2 + \cdots + R_k \quad (8)$$

where R_i means the rewards of the client i.

During the FL training process, the above steps will be repeated until the optimization problem of the global loss function converges or achieves the desired model accuracy.

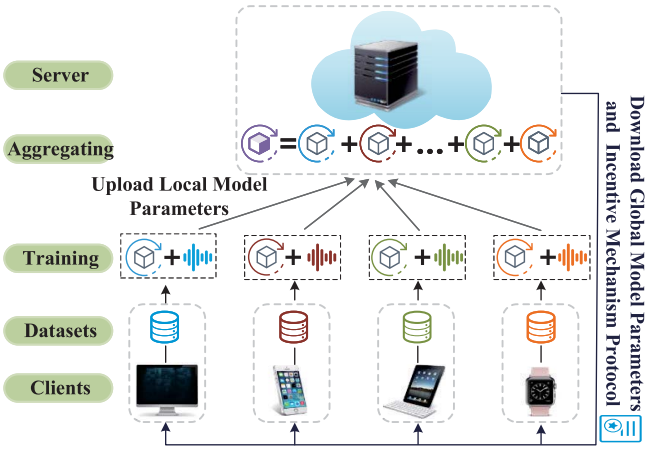

Fig. 1. Incentive mechanism FL system architecture.

TABLE I
Description of Main Notations

Notation	Description
N	The total number of clients participant in FL training
k	The number of clients who are selected for FL training
$\mathcal{D}_i$	The private dataset of the client i
ω,ω_i	The weights of the global model and local model
g,g_i	The gradients of the global model and local model
$\hat{g}_i$	The distorted gradients of the client i
$\mathcal{L}(\omega_i)$	The loss function of the client i
W	The fixed revenue of the server
r_i,R_i	The reward rate and reward of the client i
z	The reconstruct loss function
ε_i	The privacy budget of the client i

B. Threat Model

We suppose the server is *honest-but-curious* aiming to disclose some other private data set information but correctly execute training tasks. For example, in DLG, the attacker receives gradients from other clients participating in the training and recovers the original information from the training data set by continuously optimizing the reconstructed Func.(9) to find images that lead to a similar change in model prediction as the ground truth [25]

$$\arg\min_{x\in[0,1]^d} 1-\frac{\langle\nabla\mathcal{L}(x,y),\nabla\mathcal{L}(x^*,y)\rangle}{\langle\nabla\mathcal{L}(x,y)\rangle\cdot\langle\nabla\mathcal{L}(x^*,y)\rangle}+\nu TV(x). \quad (9)$$

Furthermore, in FL, an external adversary may compromise a client in a constrained manner. While this adversary lacks access to private training data, it can intercept local gradients destined for the server and execute locally stored programs. This scenario facilitates a white-box gradient leakage attack on the compromised client. In essence, whether internal or external, a privacy disclosure risk exists if the adversary acquires the genuine gradients.

C. DP in FL

In this article, we focus on DP solutions, where the client adds noise to the local model gradients. According to this, we classify DP into three types according to where noise is injected into clients, offering varying levels of privacy protection against attacks. In particular, record-level protection implies defending against the malicious server who extracts records of interest and reveals the client's identity by these records in FL. Client-level protection, on the other hand, can only prevent the malicious server from inferring the identity of clients and can not protect the record information about the data set. The definitions and characteristics of the three types of DP are as follows.

1) BDP: The BDP is defined as the DP noise added to FL in each batch iteration, which can be presented as $\hat{g}_b^{j,l} \leftarrow \bar{g}_b^{j,l} + \text{Gau}(c\cdot\Delta f/\varepsilon_i)$. BDP provides record-level privacy protection and requires the client to add artificial noise during *batch* training, which could prevent untrusted servers and clients from stealing samples and defend against infected clients performing white-box gradient attacks. We illustrate the details of the BDP for incentive mechanisms in Algorithm 1. In the proposed algorithm, the server interacts with k clients for T rounds. The client performs E epochs of local training, where each epoch contains $(|\mathcal{D}_i|/\mathcal{B})$ batch iterations. Moreover, the change in the noise addition location does not affect the complexity of the algorithm. Therefore, the complexity of the algorithms for FL incentives based on all three DP is $O(kTE(|\mathcal{D}_i|/\mathcal{B})$.

2) EDP: We defined the EDP as adding DP noise after the batch ending in each epoch iteration, i.e., $\hat{g}^j \leftarrow \hat{g}^j + \text{Gau}(c\cdot\Delta f/\varepsilon_i)$. EDP is also capable of providing record-level privacy protection by requiring the client to perform noise addition at each *epoch*. With the support of EDP, FL offers the possibility of reducing communication overhead while protecting the raw data. However, the degree of protection provided against attacks with the same privacy budget is somewhat weaker than BDP since EDP only adds noise at the end of each epoch iteration, and the amount of noise is much less. This algorithm can effectively prevent inference attacks on the server, but it cannot prevent attackers from performing gradient leakage attacks on each training sample during the client's local training process.

3) CDP: In this article, we named the conventional local DP as CDP, which applies the noise to the gradients after the local iteration ends and before uploading to the server, such as $\hat{g}_i^t \leftarrow \hat{g}_i^t + \text{Gau}(c\cdot\Delta f/\varepsilon_i)$. CDP provides client-level privacy preservation that requires the addition of noise before uploading to the server. Compared with the traditional central DP [26], CDP defends against external adversaries during the uplink transmission and internal adversaries, such as the honest-but-curious server.

Algorithm 1: IMFL With BDP

Input: local data set $\mathcal{D}_i$, where $(x_m, y_m) \in \mathcal{D}_i$, initial global model parameters ω^1, learning rate η, clipping norm C, privacy budget ε, maximum global round T, local iteration E, local batchsize $\mathcal{B}$, total payment R.

Output: global model ω.

$k \leftarrow$ randomly select k clients with probability κ
for *each round t from 1 to T* **do**
  Client** i **do
  // receive initial parameters
  $\omega_i^1 \leftarrow \omega^1$, $\varepsilon_i \leftarrow \varepsilon$, $r_i^1 \leftarrow 1$
  for *each local iteration j from 1 to E* **do**
    for *each batch l from 1 to* $\frac{|\mathcal{D}_i|}{\mathcal{B}}$ **do**
      for *b in* $\{1,\dots\mathcal{B}\}$ **do**
        $g_b^{j,l} = \nabla\mathcal{L}\left(\omega_i^{t,j},(x_b,y_b)\right)$
        // clip gradients by L_2 norm
        $\bar{g}_b^{j,l} \leftarrow g_b^{j,l}/\max\left(1, \|g_b^{j,l}\|/C\right)$
        // add Gaussian noise
        $\hat{g}_b^{j,l} = \bar{g}_b^{j,l} + Gau\left(\frac{c\cdot\Delta f}{\varepsilon_i}\right)$
      $\hat{g}^{j,l} = \frac{1}{\mathcal{B}}\sum_{b=1}^{\mathcal{B}}\hat{g}_b^{j,l}$
    $\omega_i^{t,j+1} = \omega_i^{t,j} - \eta\hat{g}^{j,l}$
  $\hat{g}_i^t = \left(\omega_i^t - \omega_i^{t-1}\right)/\eta$
  Upload $\hat{g}_i^t$ and ε_i to the server
  Server do
  // collect local update gradients from k participants
  $\hat{g}_i^t, i = 1, \dots k$
  // aggregation
  $\hat{g}^t = p_i\sum_{i=1}^{k}\hat{g}_i^t$
  // update global model
  $\omega^{t+1} = \omega^t - \eta\hat{g}^t$
// calculate rewards
$R_i = r_i\cdot\varepsilon_i | i = 1, \dots k$
$R = \sum_{i=1}^{k} R_i$
return ω

D. Privacy Analysis

In the FL framework, adversaries obtain private information from gradients. The combination of DP and FL can mitigate the risks of data leakage during distributed learning. It enables secure and robust FL sharing by adding noise to mask raw local update gradients before uploading. According to Definition 1, we give some vital lemmas about the DP of our proposed system.

Lemma 1: We suppose that the client has neighboring data sets $\mathcal{D}_i$ and $\mathcal{D}_i'$, only differing in one example at iteration t. In the model training process, we apply the gradient clipping technique, which can ensure that $|\nabla\mathcal{L}(w_i^t, \mathcal{D}_i)| \leq C$. Based on (2), the sensitivity Δf can be expressed as

$$\begin{aligned}\Delta f &= \max\left\|\nabla\mathcal{L}\left(\omega_i^t, \mathcal{D}_i\right) - \nabla\mathcal{L}\left(\omega_i^t, \mathcal{D}_i'\right)\right\|_2 \\ &= \max\left\|\frac{1}{h}\sum_{l=1}^{h}\frac{1}{\mathcal{B}}\sum_{b=1}^{\mathcal{B}}\nabla\mathcal{L}\left(\omega_i^t, x_b\right)\right. \\ &\qquad \left.-\frac{1}{h}\sum_{l=1}^{h}\frac{1}{\mathcal{B}}\sum_{b=1}^{\mathcal{B}}\nabla\mathcal{L}\left(\omega_i^t, x_b{}'\right)\right\|_2 \\ &= \frac{1}{h}\cdot\frac{1}{\mathcal{B}}\max\left\|\nabla\mathcal{L}\left(\omega_i^t, x_b\right) - \nabla\mathcal{L}\left(\omega_i^t, x_b'\right)\right\|_2 \\ &= \frac{1}{|\mathcal{D}_i|}\cdot 2C \end{aligned} \tag{10}$$

where $h = (|\mathcal{D}_i|/\mathcal{B})$.

Lemma 2: For each client $i \in N$ who participates in model training, the protection mechanism, which adds noise to the gradient, satisfies DP at each iteration.

Proof: See Appendix A. ∎

E. Convergence Analysis

In this section, we analyze the theoretical results on the convergence performance of FL by taking BDP as an example. Before proceeding with the analysis, we lay down the following formal assumptions.

Assumption 1: $\mathcal{L}(\omega_i)$ is convex.

Assumption 2: $\mathcal{L}(\omega_i)$ satisfies the Polyak–Lojasiewicz condition, there exists a positive parameter τ, which means that $\mathcal{L}(\omega) - \mathcal{L}(\omega^*) \leq (1/2\tau)|\nabla\mathcal{L}(\omega)|^2$, where ω^* is the optimal result.

Assumption 3: For any ω_i and $\omega_i' \in \mathbb{R}^d$, $\mathcal{L}(\omega_i)$ is Lipschitz continuous, and there exists a constant μ, which implies $|\mathcal{L}(\omega_i) - \mathcal{L}(\omega_i')| \leq \mu|\omega_i - \omega_i'|$.

Assumption 4: For any i and ω_i, there is $|\nabla\mathcal{L}(\omega_i) - \nabla\mathcal{L}(\omega)| \leq \rho_i$ and $\mathbb{E}\{\rho_i\} = \rho$, where ρ_i is the divergence metric.

Using the above four bounded assumptions, we now derive the expected increment in the convergence upper bound of the algorithm.

Theorem 1: With the $(\varepsilon, \delta) - DP$ protection level, after the T iteration of Algorithm 1, the convergence upper bound can be expressed as

$$\begin{aligned}&\mathbb{E}\left\{\mathcal{L}\left(\omega^T\right) - \mathcal{L}\left(\omega^*\right)\right\} \\ &\leq \gamma^T\mathbb{E}\left\{\mathcal{L}\left(\omega^0\right) - \mathcal{L}\left(\omega^*\right)\right\} + \gamma^T \\ &\quad + \sum_{t=0}^{T-1}\gamma^{T-1}\left(\frac{\mu\eta^2(N-k)\rho}{2k(N-1)} + \frac{\mu\eta^2}{2}\cdot\frac{\Delta f^2T^2c^2N}{\varepsilon^2}\right)\end{aligned} \tag{11}$$

where $\gamma = \mu\tau\eta^2 - 2\tau\eta + 1$.

Proof: See Appendix B. ∎

From Theorem 1, we could find the convergence upper bound between $\mathcal{L}(\omega^T)$ and $\mathcal{L}(\omega^*)$. The result makes it possible to find a suitable learning rate η such that $\gamma < 0$. The above convergence analysis applies equally to the EDP and CDP algorithms.

TABLE II
SERVER AND CLIENT NATURAL STATE ANALYSIS

Server	Client	
	adding noise	no defense
performing attacks	$U_c = A = u_d - u_a - \frac{1}{2}u_r^a$ $U_s = B = -u_p + \frac{1}{2}u_r^a + u_m^a$	$U_c = C = u_d - \frac{1}{2}u_r$ $U_s = D = -u_p + \frac{1}{2}u_r + u_m$
no attacks	$U_c = F = u_d - u_a$ $U_s = G = u_m^a$	$U_c = H = u_d$ $U_s = J = u_m$

IV. Satisfactory Game

In this section, we first analyze the natural state of the server and clients. On this basis, we build the two-stage Stackelberg game to find the optimal strategy for the server and clients. Then, we solve this problem by constructing the utility function.

A. Natural State

Before analyzing the natural state of the rational server and clients, we give some assumptions. Initially, we suppose that the server is *honest-but-curious*. Besides, the clients do not consider the situation of the infection, and it can add Gaussian noise by paying the cost. There are some value variables for the whole system as follows.

1) u_d: The utility of the private data set.
2) u_a: The cost of the client executing the DP noise mechanism.
3) u_r. Privacy loss for clients without Gaussian noise.
4) u_r^a: Privacy loss for clients with Gaussian noise.
5) u_m^a: The utility is derived from the global model trained with Gaussian noise.
6) u_m: The utility is derived from the global model trained without Gaussian noise.
7) u_p: The cost of the server to perform the attack.

Obviously, $u_r > u_r^a, u_m > u_m^a$, because adding noise would make server reconstruction difficult, the privacy loss of the client is lower than without Gaussian noise. Moreover, training with noise will reduce the model's accuracy, reducing the utility value the global model brings.

The server has two actions: 1) performing attacks or no attacks and 2) the client has the other two actions: a) adding noise or b) no defense. We analyze four situations in Table II and construct the game tree in Fig. 2.

1) *Case 1 (Performing Attacks-Adding Noise):* In this case, the server will pay u_p to perform an attack; then, it will receive the utility of the reconstructed information u_r^a

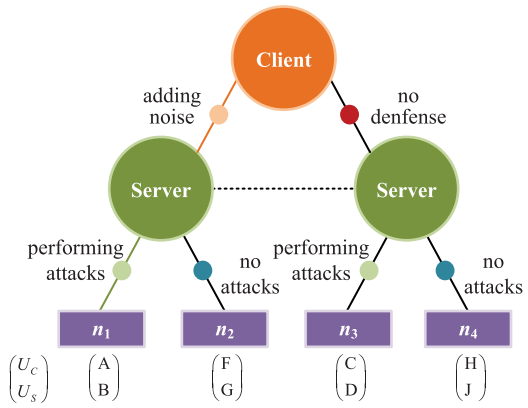

Fig. 2. Game tree.

and training model u_m^a. In the natural state, the server with the probability (1/2) will perform the attack successfully; otherwise, it will lose. Therefore, the benefit of the reconstruction information obtained by the server will be redefined as $(1/2)u_r^a$. The client will execute Gaussian noise to defense attack, which needs cost u_a. Furthermore, if the server successfully executes an attack and accesses private information, the client suffers a privacy loss of $(1/2)u_r^a$.

2) *Case 2 (Performing Attacks-No Defense):* In this case, the server will execute an attack by spending u_p. In the absence of noise added by the clients, the server gains higher utility from the reconstructed privacy records $(1/2)u_r$ and the utility of the global model u_m. However, the client's privacy loss is $(1/2)u_r$ because an adversary attacked the client's private information.
3) *Case 3 (No Attacks-Adding Noise):* In this case, the server will never perform attacks and only receive the utility of the trained model u_m^a. But, because the client will take the action of adding noise, it will pay u_a for that.
4) *Case 4 (No Attacks-No Defense):* In this case, the server and client will take part in model training without other actions. So, the server will receive the utility u_m, which the trained global model brings, and there will be no loss of utility for the client.

We assume that the cost of adding DP is negligible, as in wireless communication networks, the inherent channel noise can be considered as providing free DP [27]. If the adversary successfully executes attacks, the reconstructed information can be leveraged for additional revenue, far surpassing the cost of the attacks themselves. Consequently, we have $u_a + (1/2)u_r^a < (1/2)u_r$ and $u_p < (1/2)u_r^a < (1/2)u_r$.

In the natural state, the rational server and clients in this game converge to n_1, as indicated by colored lines in Fig. 2. At this point, the server executes attacks, while clients strategically add noise to maximize their utility.

B. Problem Formulation

After analyzing the natural state between the malicious server and clients, we consider employing a Stackelberg game framework to balance privacy and utility. As each client completes local training, gradients are generated and transmitted to the server for aggregation. Let us take client i as an example. Client i selects a privacy budget to obscure its data before uploading gradients. The server then receives gradients from participating clients and designs the DLG attack to infer the data set information of client i. We suppose that the protection mechanism is the preliminary knowledge of the adversary. Therefore, a scheme for designing a protection mechanism against attacks is not always optimal. In order to perform an attack efficiently, the server creates an incentive mechanism to encourage clients to increase their privacy budget. The server hopes to acquire more information about the training data set of the client i and receive high-quality gradients with low costs, while client i wishes to get high rewards and leak less private information. Given these considerations, we formulate the conflict as a Stackelberg game, where the server acts as the leader and the client i acts as a follower, seeking to derive the optimal strategy for both parties.

C. Optimal Incentive Strategy

The malicious server's objective is to maximize its utility function, comprising four components: 1) the initial wealth; 2) the cost paid for clients; 3) the rewards of attacked information; and 4) the income derived from model accuracy. After the round ends, the server will pay $R = \{R_1, R_2, \ldots, R_k\}(R_i > 0 \quad \forall i \in \{1, 2, \ldots, k\})$ for each client who uploads gradients according to the degree of contribution. The contribution is proportional to the privacy budget submitted by the client. Besides, the server will perform the DLG attack that rewards can be measured by the function $z = \max_{x^* \in [0,1]^d}(\langle \nabla \mathcal{L}(x, y), \nabla \mathcal{L}(x^*, y) \rangle / [\|\nabla \mathcal{L}(x, y)\| \cdot \|\nabla \mathcal{L}(x^*, y)\|]) - \nu TV(x^*)$. Also, the accuracy of the local trained model is part of the revenue for the server [28] that can be represented by typical logarithmic function as $\alpha \cdot \ln(1 + \sum_{i=1}^{N} \theta_i^t)$, where θ_i^t means the local model accuracy of the client i at round t. In our proposed scheme, we use the relationship between the model precision and privacy budget [19], in which the formula is uniformly expressed as $m \cdot e^{n \cdot \varepsilon} + q$, where $m, n < 0$ and $q > 0$, to define the part of the utility function of the model accuracy $\alpha \cdot \ln[1 + \sum_{i=1}^{k} F(\varepsilon_i)]$. We compute the utility function of server as

$$\mathcal{U}_s = W - R + \beta \cdot \sum_{i=1}^{k} (1 - z) + \alpha \ln\left[1 + \sum_{i=1}^{k} F(\varepsilon_i)\right] \quad (12)$$

where it is assumed that the fixed revenue in the server is W, $F(\varepsilon_i)$ is the local model accuracy with the privacy budget ε_i and $R = R_1 + R_2 + \cdots + R_k = \sum_{i=1}^{k} r_i \cdot \varepsilon_i$ is the total amount of assets paid to the client by the server branch. Both α and β are positive coefficients.

Therefore, we derive the following program that can calculate the optimal strategy s_s^* of the server to incentive clients:

$$\begin{aligned} s_s^* &= \arg\max_R \mathcal{U}_s \\ &= \arg\max_R W - R + \beta \cdot \sum_{i=1}^{k} (1 - z) + \alpha \ln\left[1 + \sum_{i=1}^{k} F(\varepsilon_i)\right]. \end{aligned} \quad (13)$$

D. Optimal Protection Strategy

At the beginning of the training, each client chooses its privacy budget to implement the DP protection mechanism. Taking client i as an example, its objective is to maximize the utility function while protecting privacy. The increased part of the utility function is the reward distributed by the server according to the incentive mechanism. In fact, there is also a loss on the client, which we measure with MI. Specially, we suppose that $\mathcal{O}_l$ is the output of the local model and $\mathcal{O}_a$ for the output of local model with DP noise [29]. They follow the Gaussian function with the variances σ_l^2 and σ_a^2, respectively. After that, we compute the privacy loss between $\mathcal{O}_l$ and $\mathcal{O}_a$ by MI as

$$\mathrm{MI}(\mathcal{O}_l; \mathcal{O}_a) = H(\mathcal{O}_l) + H(\mathcal{O}_a) - H(\mathcal{O}_l; \mathcal{O}_a) \tag{14}$$

where $H(\mathcal{O}_l) = (1/2)[1 + \log(2\pi \cdot \sigma_l^2)]$ represents the entropy of $\mathcal{O}_l$ and $H(\mathcal{O}_a) = (1/2)[1 + \log(2\pi \cdot \sigma_a^2)]$ is the entropy of $\mathcal{O}_a$. The $H(O_l; O_a)$ is joint Shannon entropy of the output $\mathcal{O}_l$ and $\mathcal{O}_a$, we define as

$$H(\mathcal{O}_l; \mathcal{O}_a) = 1 + \log(2\pi) + \frac{1}{2}\log\left[\sigma_l^2 \sigma_a^2 \left(1 - \rho_{l,a}^2\right)\right] \tag{15}$$

where $\rho_{l,a}^2 = (\mathbb{E}[O_l - E[O_l]] \cdot \mathbb{E}[O_a - \mathbb{E}[O_a]] / \sqrt{\sigma_l^2 \cdot \sigma_a^2})$, is the correlation coefficient between $\mathcal{O}_l$ and $\mathcal{O}_a$, and $\mathbb{E}[\cdot]$ denotes the mathematical expectation.

Substituting (15) into (14), we get

$$\mathrm{MI}(\mathcal{O}_l; \mathcal{O}_a) = -\frac{1}{2}\log\left(1 - \rho_{l,a}^2\right). \tag{16}$$

It notices that when the output of the $\mathcal{O}_l$ and $\mathcal{O}_a$ is correlated, the $\rho_{l,a}$ is close to 1. MI is proportional to the degree of correlation between the two models. As the privacy budget increases, the degree of correlation between the two models tends to be higher, and the MI becomes large, which signifies that lots of information may be revealed.

Through the above analysis, the utility function of the client i is defined as

$$\mathcal{U}_i = r_i \cdot \varepsilon_i - \lambda_i \cdot \mathrm{MI}_i \varepsilon_i \tag{17}$$

where $\lambda_i > 0$. Since MI is positively correlated with privacy budgets, we define MI to be a linear function of the privacy budget, expressed by the formula $\mathrm{MI}_i = \psi \cdot \varepsilon_i + b$. Then, we find the optimization strategy for maximizing the utility function of the client i as follows:

$$\begin{aligned} s_i^* &= \arg\max_{\varepsilon_i} U_i \\ &= \arg\max_{\varepsilon_i} r_i \cdot \varepsilon_i - \lambda_i \cdot \mathrm{MI}_i \varepsilon_i \\ &= \arg\max_{\varepsilon_i} r_i \cdot \varepsilon_i - \lambda_i \cdot (\psi \varepsilon_i + b)\varepsilon_i. \end{aligned} \tag{18}$$

In this case, the Stackelberg game requires designing the optimal rewards mechanism for the malicious server while simultaneously performing the DLG attacks to obtain the maximum utility. Then the optimal protection mechanism for the client i is the best response for the server's optimal strategy. The effective protection mechanism is based on the anticipation of predicting attacks [30], which is also in line with our original intention to establish the protection mechanism. Hence, designing the server as the leading player fits the designed target when modeling.

V. Equilibrium Analysis

In this section, we deduce the Nash equilibrium solution of the IMFL. First, we determine the optimal privacy budget for each selected client by maximizing the utility function. Subsequently, we analyze the optimal strategy to find suitable rewards for clients distributed by the server under giving a privacy budget. After that, the game reaches the Nash equilibrium, so the whole system tends to stabilize.

A. Analysis of the Follower Game

In this section, we aim to find the optimal strategy for the follower (e.g., client i) to perform the personalized protection mechanism. In the beginning, we calculate the first-order derivatives of $\mathcal{U}_i$ on the privacy budget ε_i as

$$\frac{\partial \mathcal{U}_i}{\partial \varepsilon_i} = (r_i - \lambda_i b) - 2\lambda_i \psi \varepsilon_i. \tag{19}$$

Later, to facilitate finding the maximum value of the client utility function, we deduce the second-order derivative of $\mathcal{U}_i$ concerning ε_i as

$$\frac{\partial^2 U_i}{\partial {\varepsilon_i}^2} = -2\lambda_i \psi. \tag{20}$$

Based on the above analysis, we could get that $(\partial^2 \mathcal{U}_i / [\partial (\varepsilon_i)^2]) < 0$. Judging that the utility function of the client is concave. To know the optimal solution for client i, we must set $(\partial \mathcal{U}_i / \partial \varepsilon_i) = 0$ to look for the extreme point

$$s_i^* = \frac{r_i - \lambda_i b}{2\lambda_i \psi}. \tag{21}$$

So far, each client can determine its optimal strategy by maximizing the utility function to protect its training data set. After the malicious server acquires the privacy budget of the client, it can use this game to find the optimal strategy to maximize the utility function.

As a follower, when one can anticipate the adversary's attack capabilities, the goal is to maximize revenue within an acceptable range of privacy disclosure. As a leader, the objective is to distribute rewards reasonably based on the client's privacy budget to maximize one's utility function. In this state, no player is willing to alter their strategy so that the system can achieve equilibrium.

B. Analysis of the Leader Game

For the sake of deriving the optimal solution for the server utility function, we use the first-order derivative of $\mathcal{U}_s$ concerning R to deduce the optimal strategy of the client i, and the same method can be applied to figure out the optimal reward that the server dispenses for other clients

$$\frac{\partial \mathcal{U}_s}{\partial R} = -1 + \alpha \cdot \frac{\sum_{i \in k} \frac{\partial F}{\partial (\varepsilon_i)} \cdot \frac{\partial (\varepsilon_i)}{\partial R}}{1 + \sum_{i \in k} F(\varepsilon_i)}. \tag{22}$$

Next, work out the second-order derivative of $\mathcal{U}_s$ with respect to the r_i and use its corresponding property to calculate

$$\frac{\partial^2 \mathcal{U}_s}{\partial (R)^2} = \alpha \cdot \frac{\sum_{i\in k} \frac{\partial^2 F}{\partial (\varepsilon_i)^2} \cdot \left(\frac{\partial \varepsilon_i}{\partial R}\right)^2}{\left(1+\sum_{i\in k} F(\varepsilon_i)\right)} - \alpha \cdot \frac{\left(\sum_{i\in k} \frac{\partial F}{\partial (\varepsilon_i)} \cdot \frac{\partial (\varepsilon_i)}{\partial R}\right)^2}{\left(1+\sum_{i\in k} F(\varepsilon_i)\right)^2}. \tag{23}$$

Particularly, $F(\cdot)$ is a continuous and reversible concave function on R that the first-order derivative $(\partial F/\partial \varepsilon_i) > 0$ and the second-order derivative $[\partial^2 F/\partial(\varepsilon_i)^2] < 0$. Therefore, we deduce the $([[\partial^2 \mathcal{U}_s]/\partial (R)^2) < 0$. From the above discussion, the utility function of the server is strictly concave, and the optimal strategy can calculate at the $(\partial \mathcal{U}_s/\partial R) = 0$.

Let $(\partial \mathcal{U}_s/\partial R) = 0$, we obtain the optimal rewards the server distributed for clients. Then we have

$$R = \sum_{i=1}^{k} R_i \tag{24}$$

where $R_i = r_i \cdot \varepsilon_i$, where r_i is the rewards of the clients.

By analyzing the results, we find that the optimal reward R^* is determined by the privacy budget ε_i and certain coefficients. With the knowledge of the client i's privacy budget, the server can efficiently compute the optimal reward for client i.

C. Analysis of the Nash Equilibrium

This game can be defined as a nonzero-sum Stackelberg game in which the leader and follower both have their optimization objectives. In the game confrontation, the client trains the local model with the private data set, generating gradients as a secret to upload to the server. Before transmitting them, the client chooses the appropriate privacy budget to add noise against the malicious adversary's attacks. In addition, the malicious server performs the DLG attack, back-pushing the client training data set by the gradients to maximize its utility function, and designs an incentive mechanism to stimulate the client to increase the privacy budget, which aims to obtain precise model updates. The optimal strategy for the clients is dependent on the strategy for the server and vice versa. However, we break this dependency loop and find the Nash equilibrium through the Stackelberg game. This game first computes the optimal protection for the client and then calculates the optimal payment for the server. This equilibrium can be expressed as

$$\begin{cases} \mathcal{U}_s\left(s_s^*, s_1^*, \ldots, s_i^*, \ldots, s_k^*\right) \geq \mathcal{U}_s\left(s_s^{-1}, s_1^*, \ldots, s_i^*, \ldots, s_k^*\right) \\ \mathcal{U}_i\left(s_s^*, s_1^*, \ldots, s_i^*, \ldots, s_k^*\right) \geq \mathcal{U}_i\left(s_s^*, s_1^*, \ldots, s_i^{-1}, \ldots, s_k^*\right) \end{cases} \tag{25}$$

where s_s^{-1} means the other strategy for the server except the strategy s_s^* and s_i^{-1} means the other strategy for the client i except the strategy s_i^*.

In this game, both the clients and the server have their own finite strategy spaces. Their utility functions are continuous and concave, satisfying the conditions of the Nash equilibrium. When the strategy combination is (s_s^*, s_i^*), it constitutes a stable solution. At this point, the strategy of each player represents the best response to the other, and neither will change its strategy. Thus, the game reaches equilibrium.

VI. Experiment Evaluation

In this section, we conduct extensive simulation experiments to evaluate the performance of the IMFL. First, we compare the performance of three DP mechanisms. Then, we verify that all three DP mechanisms are effective against DLG. Finally, we provide numerical results to demonstrate the effectiveness of the proposed incentive model.

A. Experimental Setup

1) Data Sets: The MNIST data set is a subset of the National Institute of Standards and Technology (NIST) data set, where the training set contains 60 000 images and labels, and the test set contains 10 000 images and labels, each of which is a grayscale handwritten digital image of 0–9 at 28×28 pixel points. Fashion-MNIST is a clothing image data set containing ten categories; each category has 7000 grayscale images of 28×28 pixels, including 60 000 images in the training set and 10 000 in the test set. CIFAR-10 is a computer vision data set for pervasive object recognition with ten classifications. It consists of 50 000 images from the training set and 10 000 from the test set, all in 32×32 RGB color.

2) Data Distribution: We employ non-IID data processing on the data set to align simulation experiments with reality. This involves initially sorting the training set based on labels, followed by truncating the sorted data sets tailored to the client's specified data set quantity. After that, the clients participate in FL training with the preprocessed data sets.

3) Attack Model: We assume that the central server is *honest-but-curious* and could perform DLG to obtain information about the client data set while participating in regular training. In order to obtain further valid information, the adversary usually chooses to perform the attack in the first round by the attack reconstruction learning algorithm with multiple iterations to correct the dummy reconstruction seed so that the gradient update of the reconstructed seed is infinitely near to the actual gradient update that has been stolen, thus recovering the original information of the data set.

4) Parameter Settings: During the FL training process, the client uploads local model updates, and the server aggregates by the Fed-Avg algorithm. We set the number of clients is selected $k = 250$, the number of communication rounds $T = 100$, the learning rate $\eta = 0.01$, the batchsize $\mathcal{B} = 10$, and the local epoch $E = 10$ to reduce the communication overhead. While employing the Gaussian DP noise addition mechanism, the relaxation term is set to $\delta = 1e^{-5}$. To visually compare the effectiveness of the three noise addition methods against DLG, we set the privacy budget values $\varepsilon = 0.1, 1, 3, 5, 8, 10$. All the parameter settings are presented in Table III.

B. Experimental Results

1) Attack and Protection: We experimentally find that the malicious adversary performing DLG in FL training can enable multiple image reconstructions. Even in the case of

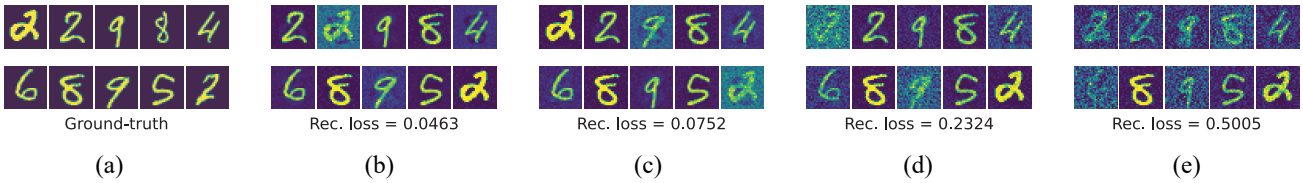

Fig. 3. Reconstruct pictures from the gradients by the adversary performs DLG with prior knowledge (MNIST). (a) Ground truth. (b) Without noise. (c) CDP, $\varepsilon = 10$. (d) EDP, $\varepsilon = 10$. (e) BDP, $\varepsilon = 10$.

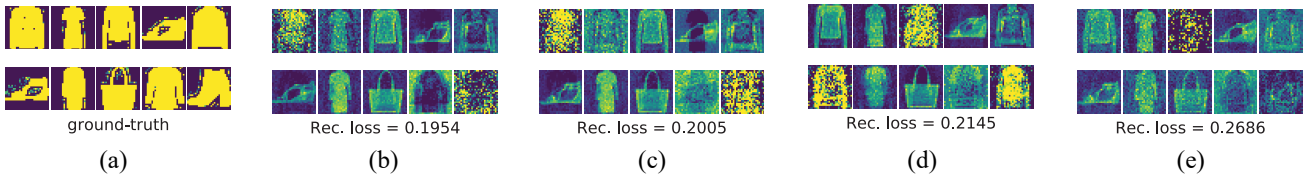

Fig. 4. Reconstruct pictures from the gradients by the adversary performs DLG with prior knowledge (Fashion-MNIST). (a) Ground truth. (b) Without noise. (c) CDP, $\varepsilon = 10$. (d) EDP, $\varepsilon = 10$. (e) BDP, $\varepsilon = 10$.

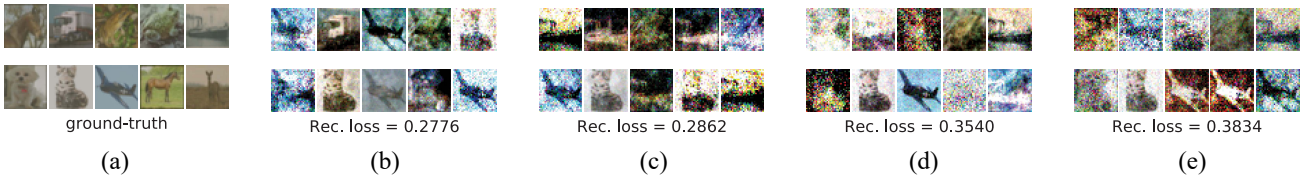

Fig. 5. Reconstruct pictures from the gradients by the adversary performs DLG with prior knowledge (CIFAR-10). (a) Ground truth. (b) Without noise. (c) CDP, $\varepsilon = 10$. (d) EDP, $\varepsilon = 10$. (e) BDP, $\varepsilon = 10$.

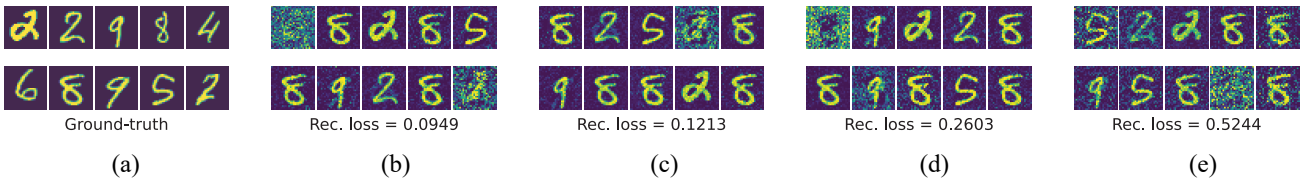

Fig. 6. Reconstruct pictures from the gradients by the adversary performs DLG without prior knowledge (MNIST). (a) Ground truth. (b) Without noise. (c) CDP, $\varepsilon = 10$. (d) EDP, $\varepsilon = 10$. (e) BDP, $\varepsilon = 10$.

TABLE III
VALUE OF EVALUATION SETTINGS

Parameter	Value
number of clients is selected, k	$[100, 500]$ (default is 250)
number of communication rounds, T	$[0, 100]$ (default is 100)
learning rate, η	$(0, 1]$ (default is 0.01)
local epoch, E	$[1, 10]$ (default is 10)
batchsize, $\mathcal{B}$	$[10, 30]$ (default is 10)
privacy budget, ε	$[0, 40]$ (default is 10)
system parameter α	$[160, 240]$ (default is 200)
system parameter λ	$[1, 5]$ (default is 1)
relaxation term, δ	$1e^{-5}$

$\mathcal{B} > 1, E > 1$, the server can still recover some data set information. We consider the two cases of attack, one in which the adversary could obtain auxiliary information, such as getting the data labels in advance, and another in which the adversary could not acquire any knowledge about the data set. Figs. 3–5 illustrates the information obtained by the malicious server performing DLG with prior knowledge in three different data sets. Figs. 3(c)–(e)–5(c)–(e) represent the effect of defense against reconstruction attack from the adversary with three DP protection mechanisms, which all set a privacy budget of 10, respectively, in the FL training process. We find that DP is effective against DLG and that BDP and EDP provide relatively good protection for the same privacy budget settings.

We evaluate the privacy information obtained by the malicious server performing the DLG without prior knowledge in Figs. 6–8, which contains Figs. 6(c)–(e)–8(c)–(e) that show us the attack result after adding the privacy protection mechanisms with CDP, EDP, and BDP all set privacy budget of 10. By comparison, we discover that the adversary with no access to auxiliary information always gets less information about the attacking picture.

2) Three DP Mechanisms: In FL, model training is conducted by uploading local model updates without transferring the local data set, further enabling the protection of the private information of the client. However, multi-image reconstruction techniques could potentially leak client information. Therefore, applying DP perturbation mechanisms to local model updates before uploading them to the server could

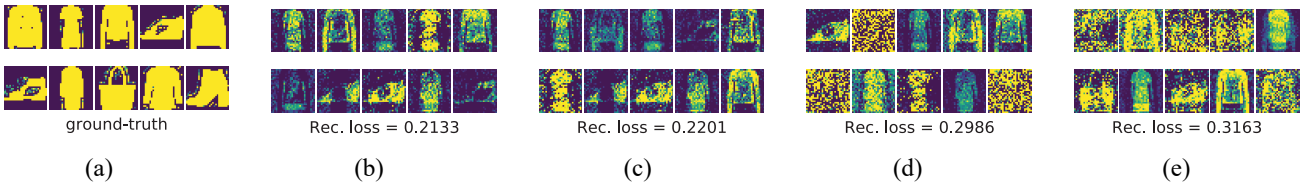

Fig. 7. Reconstruct pictures from the gradients by the adversary performs DLG without prior knowledge (Fashion-MNIST). (a) Ground truth. (b) Without noise. (c) CDP, $\varepsilon = 10$. (d) EDP, $\varepsilon = 10$. (e) BDP, $\varepsilon = 10$.

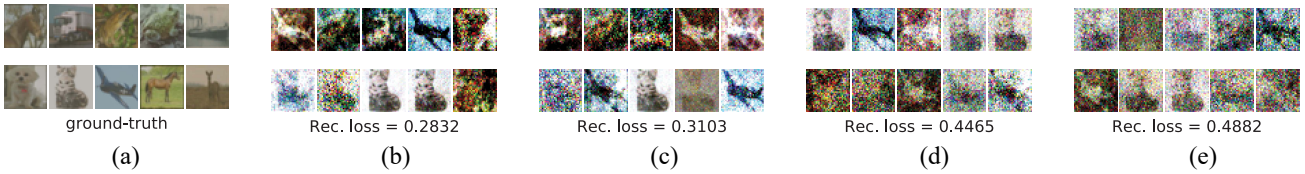

Fig. 8. Reconstruct pictures from the gradients by the adversary performs DLG without prior knowledge (CIFAR-10). (a) Ground truth. (b) Without noise. (c) CDP, $\varepsilon = 10$. (d) EDP, $\varepsilon = 10$. (e) BDP, $\varepsilon = 10$.

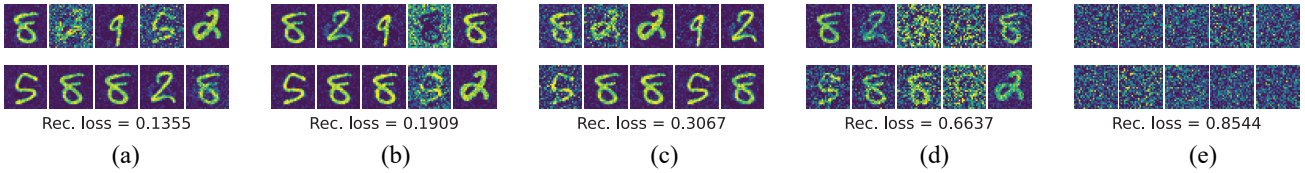

Fig. 9. Influence of privacy budgets on adversary reconstruction of images without prior knowledge in CDP (MNIST). (a) $\varepsilon = 8$. (b) $\varepsilon = 5$. (c) $\varepsilon = 3$. (d) $\varepsilon = 1$. (e) $\varepsilon = 0.1$.

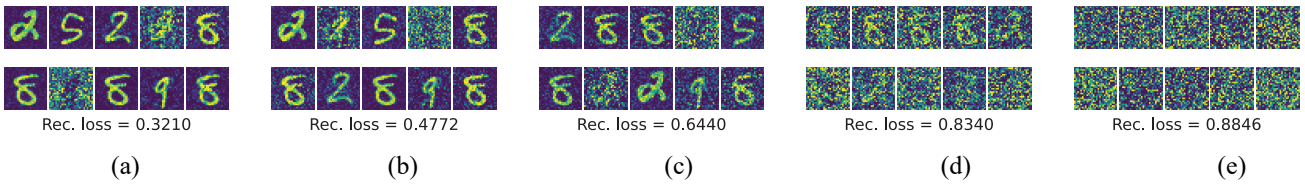

Fig. 10. Influence of privacy budgets on adversary reconstruction of images without prior knowledge in EDP (MNIST). (a) $\varepsilon = 8$. (b) $\varepsilon = 5$. (c) $\varepsilon = 3$. (d) $\varepsilon = 1$. (e) $\varepsilon = 0.1$.

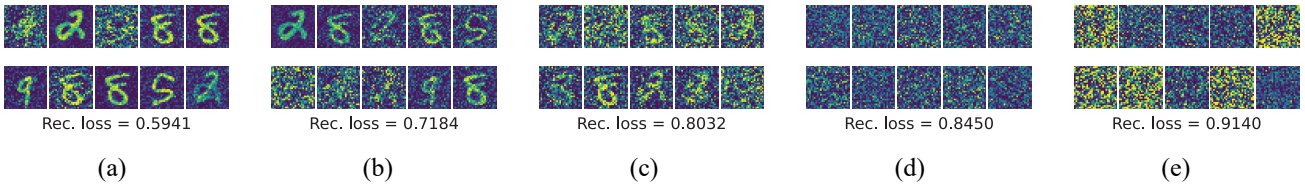

Fig. 11. Influence of privacy budgets on adversary reconstruction of images without prior knowledge in BDP (MNSIT). (a) $\varepsilon = 8$. (b) $\varepsilon = 5$. (c) $\varepsilon = 3$. (d) $\varepsilon = 1$. (e) $\varepsilon = 0.1$.

effectively defend against DLG, as demonstrated in our experiments. During the FL process, it is a significant challenge for the adversary to obtain the auxiliary information of the client, so we conducted simulation experiments for the case where the adversary could not obtain the client-side private data labels. Figs. 9–11 present the performance of defending against DLG when applying the CDP, EDP, and BDP noise addition mechanisms to the FL process, respectively. We observe that, with the same privacy budget, it is relatively easier for an attacker to obtain privacy information by attacking FL based on the CDP protection mechanism compared to FL with the other two DP mechanisms. When the privacy budget value is set to 1, these three DP mechanisms provide different levels of protection, making it difficult for an adversary to access the privacy information.

In Fig. 12, we evaluate the influence of three DP mechanisms on reconstruction loss. As the privacy budget decreases, three noise addition methods make it difficult for the adversary to reconstruct the image. The results of allocating the same privacy budget across test data sets show that BDP and EDP provide consistently better confidentiality than client-level CDP. With respect to BDP and EDP, which also provide record-level protection, the former provides better privacy security.

3) Accuracy: Incorporating the DP mechanism into the gradient descent training of FL is designed to defend against

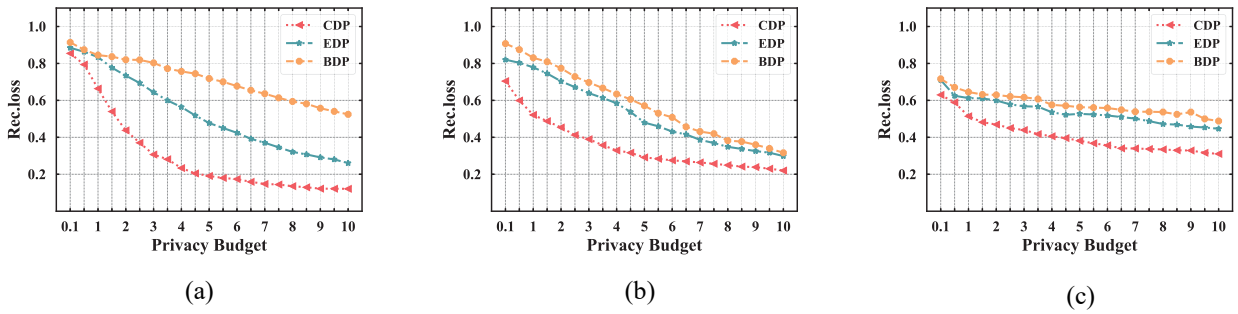

Fig. 12. Influence of the privacy budget on the reconstructed loss function. (a) MNIST. (b) Fashion-MNIST. (c) CIFAR-10.

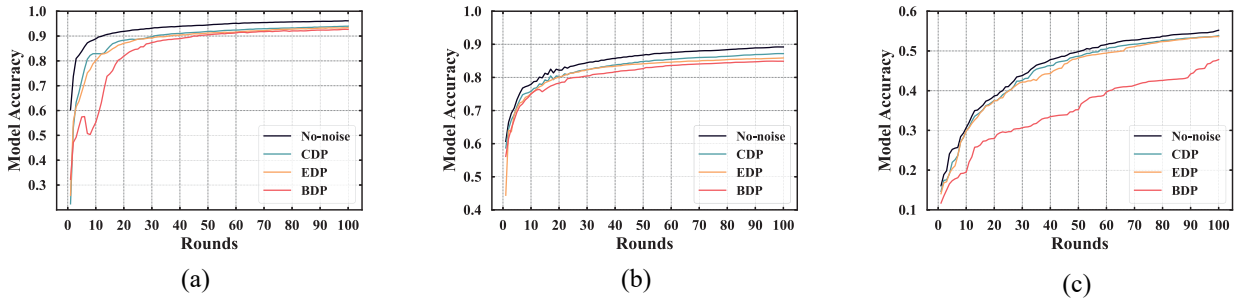

Fig. 13. Influence of the iteration rounds on the global model accuracy. (a) MNIST. (b) Fashion-MNIST. (c) CIFAR-10.

DLG and enhance privacy for local learning updates. In this scenario, even if the adversary obtains the perturbed gradient, it prevents them from deducing the original data. In our proposed model, each of the three DP mechanisms is combined with FL, resulting in different levels of privacy protection. In this model, the server collects and averages the perturbed gradients submitted by users to obtain an average result for updating the parameters of the global model, thereby enhancing privacy protection. We show experimentally that CDP, EDP, and BDP, in conjunction with FL training, could all achieve the desired model accuracy while preserving privacy.

Fig. 13 compares the global model accuracy of the three DP artificial noise mechanisms and the normal no-noise condition versus the number of rounds. The results show that the global model accuracy of FL with the CDP mechanism is closer to the global model accuracy of the original FL under the same privacy budget.

4) Server Utility: We evaluate the effect of the reward on the server utility function, and the result is shown in Fig. 14(a). Since the server utility function is a convex function concerning the reward, we find the optimal strategy for the server, which attains a maximum utility of 5447.268 at a reward value of 460.249. This Nash equilibrium point is the optimal reward value for the clients by the server. In addition to the above, Fig. 14(b) shows the relationship between the reward rate and the privacy budget for different λ parameter settings. As the privacy budget increases, the reward rate continues to get larger. Assuming we fix the privacy budget value, the reward rate grows progressively as λ increases.

5) Clients Utility: As shown in Fig. 15(a), we randomly select five clients, assess their relationship between client utility function and privacy budget under different systems, and derive the optimal strategy for each client. Experimentally,

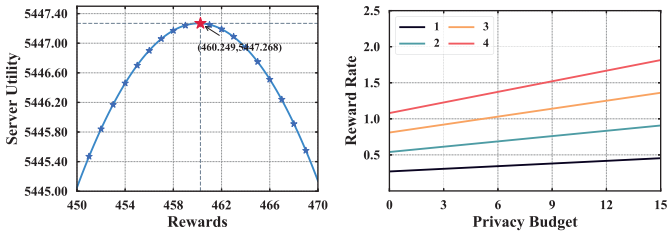

Fig. 14. Influence of the rewards on the server utility function. (a) Server utility versus rewards (existence of unique Nash equilibrium). (b) Reward rate versus privacy budget.

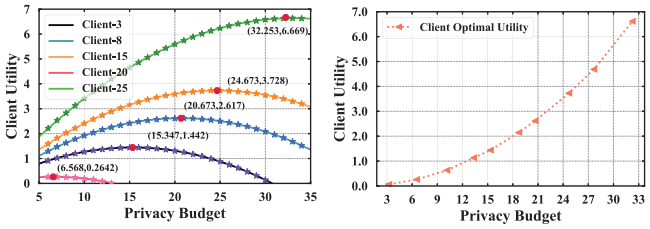

Fig. 15. Influence of the privacy budget on the client utility function. (a) Optimal utility of different devices versus privacy budgets. (b) Relationship between optimal utility functions and privacy budgets.

it is verified that a Nash equilibrium can be found for all clients, i.e., an optimal privacy budget value allows that client to obtain the optimal utility given the available system parameters. We illustrate the variation of the utility function when the client sets the optimal privacy budget at different system configurations in Fig. 15(b). It is found that the optimal utility rises as the privacy budget increases.

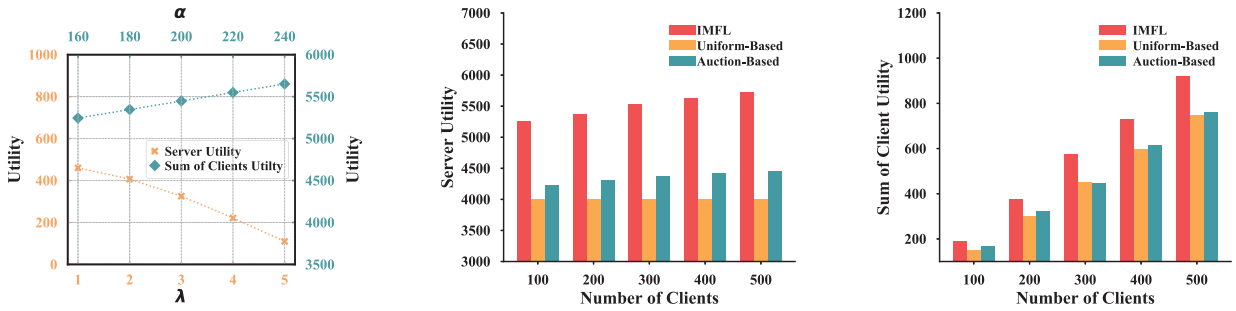

Fig. 16. Influence of the system parameter α, λ, and the number of the clients on the server utility and the sum of the client utility under different schemes. (a) Server utility versus λ and sum of the client utility versus α. (b) Server utility versus number of the clients. (c) Sum of the client utility versus number of the clients.

Particularly, when the client considers adding minimal noise, the local model accuracy significantly contributes to the global model trained by the server, resulting in higher rewards.

6) Influence of System Parameters: To illustrate the influence of the system parameters on the utility function, we evaluated experiments on them. Fig. 16(a) explains the influence of the system parameter λ on the sum of clients' utility and α on the server utility function. Since parameter λ and the privacy budget are negatively correlated in the client, the server expects the client to use more of the privacy budget so that the optimal utility of the client increases as the parameter value λ decreases. The parameter α governs the influence of the client's local model accuracy on the server's utility. As α increases, the impact of the client-side local model accuracy on the server's utility function becomes more pronounced. In the utility function, we model the change in utility caused by the change in accuracy of the local client model partially as a monotonically increasing logarithmic function so that as the number of participating training clients grows, the server optimal utility function becomes larger, but its increasing trend is gradually slow.

In addition, we show the influence of the number of clients in FL on server utility and the sum of the client utility. As illustrated in Fig. 16(b) and (c), the utility of the server and clients based on the IMFL scheme is higher than the uniform cost-based and auction-based schemes with the number of clients changed, which means the incentive mechanism is effective. In IMFL, increasing the number of clients enables the global model to be more accurate and the server utility to rise. However, when the number of clients increases to a certain quantity, the growth of global model accuracy gradually tends to slow down, so the increment of server utility decreases, and the total utility of clients increases. Since the uniform cost-based scheme distributes rewards uniformly to clients, even if the number of clients increases, no client is willing to sacrifice its privacy for model accuracy, but the server has to pay more due to the increase in the number of clients. Therefore, the server utility decreases, and the total client utility increases. In the auction-based scheme, the server selects low-cost clients to participate in the training and cannot guarantee the local model quality and data privacy. Therefore, both server and client utilities are lower than the IMFL scheme.

VII. Related Work

In this section, we review the relevant research work, which mainly focuses on the privacy protection mechanisms and game theory in FL.

A. Privacy Protection Mechanisms in FL

While the collaboration of multiple devices in FL architecture accelerates the model training process and mitigates the risks of attacks on model learning, recent research [31], [32] has demonstrated that the entire training process remains vulnerable to various attacks and threats. Currently, a series of techniques have been proposed to prevent internal and external attacks in the FL process. For example, HE [33] is the common privacy protection mechanism employed in FL systems, which provides multiple cryptographic primitives that could apply to SMC, such as secret sharing [34], zero knowledge proofs [35], and garbled circuits [36]. However, the majority of protocols based on HE only support single-key encryption, which may present a single point of failure risk if the key is compromised. To address this issue, Ma et al. [37] proposed an improved version of the MK-CKKS multikey HE protocol as xMK-CKKS, which uses aggregated public key encryption and is decrypted collaboratively by all participating devices. This scheme could prevent privacy leakage while resisting collusion between participating devices and the server. Even though encryption ensures the security of sensitive private information, it entails considerable communication and computational overhead, hence the creation of a lightweight privacy protection mechanism, DP [38]. On this foundation, Wu et al. [13] designed an FL scheme suitable for multiparty collaborative modeling scenarios combined with the adaptive gradient descent strategy and DP mechanism to resist various background knowledge attacks. Nevertheless, existing DP protection schemes rarely derive an optimal privacy budget that satisfies the personalization and privacy protection requirements of clients. Our work takes this into account by using the DP technique to defend against gradient inference attacks on FL and is able to balance the data privacy

TABLE IV
FUNCTIONALITY COMPARISON

	[37]	[13]	[40]	[22]	[42]	IMFL
Record-level Protection	✘	✘	✘	✘	✘	✔
Client-level Protection	✘	✔	✔	✔	✘	✔
personalized Protection	✘	✘	✘	✘	✘	✔
Incentive Mechanism	✘	✘	✔	✔	✘	✔
Model Validity	✔	✔	✔	✔	✔	✔
Utility Maximization	✘	✘	✘	✘	✘	✔

and utility issues in this technique to provide personalized protection for clients.

B. Game Theory in FL

Game theory [39] could provide a formal and appropriate method for an FL framework to model the interactions between the server and clients. These players must choose the optimal action, considering the impact of other players' strategies. Zhang et al. [40] combined game theory to develop a more robust FL scheme based on joint DP, which aimed to stimulate clients to participate in training and guarantee data privacy. In addition to encouraging the client to participate in the training, the further challenge is how to motivate the client to participate honestly in the training. Chen et al. [22] integrated a dynamic game model based on evolutionary game theory with incentives for reputation and payment to model the game process of users in data sharing, incentivizing them to participate in collaborative tasks of data sharing and maintaining the validity of the model. In order to achieve the goal of obtaining high-accuracy models in the training task, one way is that the proposed FL scheme could encourage clients to participate in the training by an incentive mechanism, and another is to enable the excellent performance of the model by eliminating some clients that are not relevant to the training [41]. Nagalapatti and Narayanam [42] proposed a cooperative game involving the gradients shared by the clients, which combined with the Shapley value-based Federated Averaging (S-FedAvg) algorithm that empowered the server to select relevant clients with high probability to address the problem of federated relevant client selection (FRCS). However, game research projects mentioned above do not consider the case of adversary attacks and the overall utility of the system, which we focused on when designing the IMFL scheme. Our research is dedicated to designing an incentive mechanism with game theory to encourage clients to contribute high-quality local models. In addition, we consider the adversary's attack behavior so that the incentive mechanism remains effective in the presence of the adversary, and we can find the optimal strategies for both the client and the server to maximize the overall utility of the system.

We selected five existing privacy-preserving and incentive FL schemes and compared them with ours from the six aspects. As shown in Table IV, our scheme demonstrates various advantages. IMFL provides personalized instance-level protection to defend against gradient attacks while ensuring the effectiveness of the global model. In addition, IMFL motivates clients to participate in FL, maximizes social utility, and achieves Nash equilibrium.

VIII. CONCLUSION

In FL based on DP to protect the private data sets of participating clients, one of the fundamental challenges lies in striking a balance between utility and privacy. In this article, we proposed an incentive mechanism framework for FL with DP, named IMFL, which maximizes utility while providing a novel approach against inference attacks from gradients of the FL systems. We explore three DP mechanisms that effectively resist DLG with various degrees of privacy-preserving and perform the convergence analysis. Then, we further established the confrontation between the client and server as the two-stage Stackelberg game to derive the optimal strategies of the players. Finally, experiments conducted on real-world data sets indicated that the IMFL is effective. In future work, we aim to explore scenarios involving multiple adversaries and investigate the application of this scheme to more complex ML models.

APPENDIX A
PRIVACY PROOF FOR GAUSSIAN NOISE

Assuming that $\mathbb{R} = \mathbb{R}_1 \cup \mathbb{R}_2$, where $\mathbb{R}_1 = \{\chi \in \mathbb{R} : |\chi| \le (c \cdot \Delta f/\varepsilon)\}$ and $\mathbb{R}_2 = \{\chi \in \mathbb{R} : |\chi| > (c \cdot \Delta f/\varepsilon)\}$. For any output subset $\mathcal{O} \in \mathbb{R}$, we define $\mathcal{O}_1 = \{\nabla\mathcal{L}(\omega_i^t, \mathcal{D}) + \mathcal{N}(0, \sigma^2 I)\}$ and $\mathcal{O}_2 = \{\nabla\mathcal{L}(\omega_i^t, \mathcal{D}) + \mathcal{N}(0, \sigma^2 I)\}$.

Each client adds Gaussian noise to their model gradients during training or before uploading them to the server. At each aggregation, the server receives gradients with noise from clients. Then, we have

$$\begin{aligned}
&\Pr\left[\nabla\mathcal{L}\left(\omega_i^t, \mathcal{D}\right) + \mathcal{N}\left(0, \sigma^2 I\right) \in \mathcal{O}\right] \\
&= \Pr\left[\nabla\mathcal{L}\left(\omega_i^t, \mathcal{D}\right) + \mathcal{N}\left(0, \sigma^2 I\right) \in \mathcal{O}_1\right] \\
&\quad + \Pr\left[\nabla\mathcal{L}\left(\omega_i^t, \mathcal{D}\right) + \mathcal{N}\left(0, \sigma^2 I\right) \in \mathcal{O}_2\right] \\
&\le \Pr\left[\nabla\mathcal{L}\left(\omega_i^t, \mathcal{D}\right) + \mathcal{N}\left(0, \sigma^2 I\right) \in \mathcal{O}_1\right] + \delta \\
&\le e^{\varepsilon}\left(\Pr\left[\nabla\mathcal{L}\left(\omega_i^t, \mathcal{D}'\right) + \mathcal{N}\left(0, \sigma^2 I\right) \in \mathcal{O}_1\right]\right) + \delta. \quad (26)
\end{aligned}$$

Proof of completion.

APPENDIX B
CONVERGENCE PROOF

First, we define the local model update of each participant as follows:

$$\omega_i^{t+1} = \omega^t - \eta\nabla\mathcal{L}\left(\omega_i^t\right). \quad (27)$$

Then, consider the aggregation process performed by the server as

$$\begin{aligned}
\omega^{t+1} &= \sum_{i\in k} p_i\left(\omega^t - \eta\left(\nabla\mathcal{L}\left(\omega_i^t\right) + n_i^t\right)\right) \\
&= \omega^t - \eta\left(\sum_{i\in k} p_i\nabla\mathcal{L}\left(\omega_i^t\right) + n^t\right). \quad (28)
\end{aligned}$$

Hence, we can obtain the difference in the loss function between tth and $(t+1)$th as

$$\begin{aligned}
&\mathcal{L}\left(\omega^{t+1}\right)-\mathcal{L}\left(\omega^{t}\right)\\
&\leq \nabla\mathcal{L}\left(\omega^{t}\right)^{\top}\left(\omega^{t+1}-\omega^{t}\right)+\frac{\mu}{2}\left\|\omega^{t+1}-\omega^{t}\right\|^{2}\\
&\leq \nabla\mathcal{L}\left(\omega^{t}\right)^{\top}\left(-\eta\left(\sum_{i\in k}p_i\nabla\mathcal{L}\left(\omega_i^{t}\right)+n^{t}\right)\right)\\
&\quad+\frac{\mu\eta^{2}}{2}\left\|\sum_{i\in k}p_i\nabla\mathcal{L}\left(\omega_i^{t}\right)+n^{t}\right\|^{2}\\
&=\frac{\mu\eta^{2}}{2}\left\|\sum_{i\in k}p_i\nabla\mathcal{L}\left(\omega_i^{t}\right)\right\|^{2}+\frac{\mu\eta^{2}}{2}\|n^{t}\|^{2}\\
&\quad-\eta\nabla\mathcal{L}\left(\omega^{t}\right)^{\top}\sum_{i\in k}p_i\nabla\mathcal{L}_i\left(\omega_i^{t}\right).
\end{aligned}\tag{29}$$

Considering $p_i=(1/k)$, the expectation of the loss difference between the tth and $(t+1)$th training iterations can be represented as

$$\begin{aligned}
&\mathbb{E}\left\{\mathcal{L}\left(\omega^{t+1}\right)-\mathcal{L}\left(\omega^{t}\right)\right\}\\
&\leq\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\left\|\sum_{i\in k}p_i\nabla\mathcal{L}\left(\omega_i^{t}\right)\right\|^{2}\right\}\\
&\quad+\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\|n^{t}\|^{2}\right\}-\eta\left\|\nabla\mathcal{L}\left(\omega^{t}\right)\right\|^{2}.
\end{aligned}\tag{30}$$

Then, we have

$$\begin{aligned}
&\mathbb{E}\left\{\left\|\sum_{i\in k}p_i\nabla\mathcal{L}\left(\omega_i^{t}\right)\right\|^{2}\right\}\\
&=\frac{1}{N\cdot k}\sum_{i\in N}\left\|\nabla\mathcal{L}\left(\omega_i^{t}\right)\right\|^{2}\\
&\quad+\frac{k-1}{N\cdot k(N-1)}\sum_{i\in N}\sum_{j\in N/i}\left(\nabla\mathcal{L}\left(\omega_i^{t}\right)\right)^{\top}\nabla\mathcal{L}\left(\omega_j^{t}\right)\\
&=\left(\frac{1}{N\cdot k}-\frac{k-1}{N\cdot k(N-1)}\right)\sum_{i\in N}\left\|\nabla\mathcal{L}\left(\omega_i^{t}\right)\right\|^{2}\\
&\quad+\frac{k-1}{N\cdot k(N-1)}\left(\sum_{i\in N}\nabla\mathcal{L}\left(\omega_i^{t}\right)\right)^{2}\\
&=\frac{N-k}{N\cdot k(N-1)}\sum_{i\in N}\left\|\nabla\mathcal{L}\left(\omega_i^{t}\right)-\nabla\mathcal{L}\left(\omega^{t}\right)\right\|^{2}+\left\|\nabla\mathcal{L}\left(\omega^{t}\right)\right\|^{2}.
\end{aligned}\tag{31}$$

Based on the above assumptions, we know that $\mathbb{E}\{\rho_i\}=\rho$. So, (31) can be rewritten as

$$\mathbb{E}\left\{\left\|\sum_{i\in k}p_i\nabla\mathcal{L}\left(\omega_i^{t}\right)\right\|^{2}\right\}\leq\frac{(N-k)\rho}{k(N-1)}+\left\|\nabla\mathcal{L}\left(\omega^{t}\right)\right\|^{2}.\tag{32}$$

Further, we calculate

$$\mathbb{E}\{\|n\|\}=\frac{\Delta fTc}{\varepsilon}\sqrt{\frac{2N}{\pi}}\tag{33}$$

and

$$\mathbb{E}\left\{\|n\|^{2}\right\}=\frac{\Delta f^{2}T^{2}c^{2}N}{\varepsilon^{2}}\tag{34}$$

After that, subtracting (32) into (30), we have

$$\begin{aligned}
&\mathbb{E}\left\{\mathcal{L}\left(\omega^{t+1}\right)-L\left(\omega^{t}\right)\right\}\\
&\leq\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\frac{(N-k)\lambda}{k(N-1)}+\left\|\nabla\mathcal{L}\left(\omega^{t}\right)\right\|^{2}\right\}\\
&\quad+\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\|n^{t}\|^{2}\right\}-\eta\left\|\nabla\mathcal{L}\left(\omega^{t}\right)\right\|^{2}.
\end{aligned}\tag{35}$$

Because Assumption 2, we have

$$\begin{aligned}
&\mathbb{E}\left\{\mathcal{L}\left(\omega^{t+1}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}\\
&\leq\mathbb{E}\left\{\mathcal{L}\left(\omega^{t}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}+\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\frac{(N-k)\rho}{k(N-1)}+\left\|\nabla\mathcal{L}\left(\omega^{t}\right)\right\|^{2}\right\}\\
&\quad+\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\|n^{t}\|^{2}\right\}-\eta\|\nabla\mathcal{L}\left(\omega^{t}\right)\|^{2}\\
&\leq\mathbb{E}\left\{\mathcal{L}\left(\omega^{t}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}+\frac{\mu\eta^{2}(N-k)\rho}{2k(N-1)}\\
&\quad+\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\|n^{t}\|^{2}\right\}+\left(\frac{\mu\eta^{2}}{2}-\eta\right)\|\nabla\mathcal{L}\left(\omega^{t}\right)\|^{2}\\
&\leq\mathbb{E}\left\{\mathcal{L}\left(\omega^{t}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}+\frac{\mu\eta^{2}(N-k)\rho}{2k(N-1)}+\frac{\mu\eta^{2}}{2}\mathbb{E}\left\{\|n^{t}\|^{2}\right\}\\
&\quad+2\tau\left(\frac{\mu\eta^{2}}{2}-\eta\right)\mathbb{E}\left\{\mathcal{L}\left(\omega^{t}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}\\
&=\left(\mu\tau\eta^{2}-2\tau\eta+1\right)E\left\{\mathcal{L}\left(\omega^{t}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}\\
&\quad+\frac{\mu\eta^{2}(N-k)\rho}{2k(N-1)}+\frac{\mu\eta^{2}\Delta f^{2}T^{2}c^{2}N}{2\varepsilon^{2}}.
\end{aligned}\tag{36}$$

Then, we obtain the expectation of the upper bound of convergence as follows:

$$\begin{aligned}
&\mathbb{E}\left\{\mathcal{L}\left(\omega^{T}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}\\
&\leq\gamma^{T}\mathbb{E}\left\{\mathcal{L}\left(\omega^{0}\right)-\mathcal{L}\left(\omega^{*}\right)\right\}+\gamma^{T}\\
&\quad+\sum_{t=0}^{T-1}\gamma^{T-1}\left(\frac{\mu\eta^{2}(N-k)\rho}{2k(N-1)}+\frac{\mu\eta^{2}}{2}\cdot\frac{\Delta f^{2}T^{2}c^{2}N}{\varepsilon^{2}}\right)
\end{aligned}\tag{37}$$

where $\gamma=\mu\tau\eta^{2}-2\tau\eta+1$.

Proof of completion.

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.

[4] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation," in *Proc. Int. MICCAI Brainlesion Workshop*, 2019, pp. 92–104.

[5] N. Rieke et al., "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–7, 2020.

[6] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, Aug. 2020.
[7] W. Yang, W. Xiang, Y. Yang, and P. Cheng, "Optimizing federated learning with deep reinforcement learning for digital twin empowered Industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1884–1893, Feb. 2023.
[8] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
[9] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 739–753.
[10] V. Shejwalkar and A. Houmansadr, "Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Proc. NDSS*, 2021, pp. 1–19.
[11] K. Wei et al., "Personalized federated learning with differential privacy and convergence guarantee," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4488–4503, 2023.
[12] W. Wei, L. Liu, Y. Wu, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2021, pp. 797–807.
[13] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, "An adaptive federated learning scheme with differential privacy preserving," *Future Gener. Comput. Syst.*, vol. 127, pp. 362–372, Feb. 2022.
[14] Z. He, L. Wang, and Z. Cai, "Clustered federated learning with adaptive local differential privacy on heterogeneous IoT data," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 137–146, Jan. 2024.
[15] H. Batool, A. Anjum, A. Khan, S. Izzo, C. Mazzocca, and G. Jeon, "A secure and privacy preserved infrastructure for VANETs based on federated learning with local differential privacy," *Inf. Sci.*, vol. 652, Jan. 2024, Art. no. 119717.
[16] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 2650–2654.
[17] X. Yuan, W. Ni, M. Ding, K. Wei, J. Li, and H. V. Poor, "Amplitude-varying perturbation for balancing privacy and utility in federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1884–1897, 2023.
[18] R. Hu, Y. Guo, and Y. Gong, "Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy," *IEEE Trans. Mobile Comput.*, early access, Dec. 14, 2023, doi: 10.1109/TMC.2023.3343288.
[19] Y. Xu, M. Xiao, H. Tan, A. Liu, G. Gao, and Z. Yan, "Incentive mechanism for differentially private federated learning in Industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 6927–6939, Oct. 2022.
[20] X. Tu, K. Zhu, N. C. Luong, D. Niyato, Y. Zhang, and J. Li, "Incentive mechanisms for federated learning: From economic and game theoretic perspective," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 3, pp. 1566–1593, Sep. 2022.
[21] J. Zhang, Y. Wu, and R. Pan, "Incentive mechanism for horizontal federated learning based on reputation and reverse auction," in *Proc. Web Conf.*, 2021, pp. 947–956.
[22] Y. Chen et al., "DIM-DS: Dynamic incentive model for data sharing in federated learning based on smart contracts and evolutionary game theory," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24572–24584, Dec. 2022.
[23] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.
[24] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2019.
[25] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 16937–16947.
[26] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," 2020, *arXiv:2009.03561*.
[27] D. Liu and O. Simeone, "Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 170–185, Jan. 2021.
[28] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1055–1069, Apr.–Jun. 2020.
[29] M. P. Uddin, Y. Xiang, X. Lu, J. Yearwood, and L. Gao, "Mutual information driven federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1526–1538, Jul. 2020.
[30] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," in *Proc. Privacy Enhanc. Technol.*, 2015, pp. 1–17.
[31] Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-i.i.d. data in AIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1310–1321, Feb. 2022.
[32] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Inf. Fusion*, vol. 90, pp. 148–173, Feb. 2023.
[33] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2020, pp. 1–15.
[34] Y. Dong, X. Chen, L. Shen, and D. Wang, "EaSTFLy: Efficient and secure ternary federated learning," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101824.
[35] Y. Wan, Y. Qu, L. Gao, and Y. Xiang, "Privacy-preserving blockchain-enabled federated learning for B5G-driven edge computing," *Comput. Netw.*, vol. 204, Feb. 2022, Art. no. 108671.
[36] J. Wang, S. Guo, X. Xie, and H. Qi, "Protect privacy from gradient leakage attack in federated learning," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 580–589.
[37] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *Int. J. Intell. Syst.*, vol. 37, no. 9, pp. 5880–5901, 2022.
[38] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Lang., Program.*, 2006, pp. 1–12.
[39] Z. Abou El Houda, B. Brik, A. Ksentini, L. Khoukhi, and M. Guizani, "When federated learning meets game theory: A cooperative framework to secure IIoT applications on edge computing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7988–7997, Nov. 2022.
[40] L. Zhang, T. Zhu, P. Xiong, W. Zhou, and S. Y. Philip, "A robust game-theoretical federated learning framework with joint differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3333–3346, Apr. 2023.
[41] R. Gupta and J. Gupta, "Federated learning using game strategies: State-of-the-art and future trends," *Comput. Netw.*, vol. 225, Apr. 2023, Art. no. 109650.
[42] L. Nagalapatti and R. Narayanam, "Game of gradients: Mitigating irrelevant clients in federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 9046–9054.

Mengqian Li received the B.Eng. degree from the College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang, Hebei, China, in 2019. She is currently pursuing the Ph.D. degree with College of Computer Science and Technology, Guizhou University, Guiyang, China.

Her research interests include federated learning and data security.

Youliang Tian (Senior Member, IEEE) received the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2012.

He is a Professor and the Ph.D. Supervisor with the College of Computer Science and Technology, Guizhou University, Guiyang, China. His current research interests include algorithmic game theory, cryptography and security protocols, big data security and privacy protection, blockchain, and electronic currency.

Junpeng Zhang received the M.S. degree from the College of Computer and Engineering, North China Electric Power University, Beijing, China, in 2006. He is currently pursuing the Ph.D. degree in cyberspace security with Xidian University, Xi'an, Shaanxi, China.

He is currently a Lecturer with the College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang, China. His research has been concerned with privacy preserving, security, and differential privacy.

Zhou Zhou received the Ph.D. degree in software engineering from the College of Computer Science and Technology, Guizhou University, Guiyang, China, in 2023.

She is currently an Experimenter with Guizhou University. Her research interests include federated learning and privacy protection.

Dongmei Zhao received the Ph.D. degree from Xidian University, Xi'an, China, in 2007.

She has been a Professor and the Ph.D. Supervisor from Hebei Normal University, Shijiazhuang, China, where she is currently the Leader of Hebei Key Laboratory of Network and Information Security. Her research interests include network security situation estimation and prediction.

Jianfeng Ma (Member, IEEE) received the Ph.D. degree from Xidian University, Xi'an, China, in 1995.

He has been a Professor and the Ph.D. Supervisor with the Department of Computer Science and Technology, Xidian University since 1998. He was the Special Engaged Professor of the Yangtze River Scholar in China. He is currently the Leader of Shaanxi Key Laboratory of Network and System Security. His research interests include information and network security, wireless and mobile computing systems, and computer networks.

Dr. Ma is the Academic Committee Member of State Key Laboratory of Integrated Services Networks, the Council Member of CCF, and an Editor of *Science China Information Sciences*, *Journal on Communications*, and *Chinese Journal of Computers*.