This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

Differentially Private Federated Learning with an Adaptive Noise Mechanism

Rui Xue, Kaiping Xue, *Senior Member, IEEE*, Bin Zhu, *Graduate Student Member, IEEE*, Xinyi Luo, *Graduate Student Member, IEEE*, Tianwei Zhang, *Member, IEEE*, Qibin Sun, *Fellow, IEEE*, Jun Lu

***Abstract*—Federated Learning (FL) enables multiple distributed clients to collaboratively train a model with owned datasets. To avoid the potential privacy threat in FL, researchers propose the DP-FL strategy, which utilizes differential privacy (DP) to add elaborate noise to the exchanged parameters to hide privacy information. DP-FL guarantees the privacy of FL at the cost of model performance degradation. To balance the trade-off between model accuracy and security, we propose a differentially private federated learning scheme with an adaptive noise mechanism. This is challenging, as the distributed nature of FL makes it difficult to appropriately estimate sensitivity, where sensitivity is a concept in DP that determines the scale of noise. To resolve this, we design a generic method for sensitivity estimates based on local and global historical information. We also provide instances on four commonly used optimizers to verify its effectiveness. The experiments on MNIST, FMNIST and CIFAR-10 convincingly prove that our proposed scheme achieves higher accuracy while keeping high-level privacy protection compared to prior works.**

***Index Terms*—Federated Learning, Differential Privacy, Adaptive Noise.**

I. Introduction

With the development of the digital economy, the datasets critical to artificial intelligence are becoming larger and more dispersed, so there is an urgent need for collaboration among parties to improve the accuracy and generalisation of machine learning models. However, the problem of data privacy protection holds back the data sharing. Aiming at solving such challenge, Federated Learning (FL) [1], [2] is introduced to facilitate cooperation among multiple clients who own datasets to obtain the model with satisfactory accuracy while ensuring that the datasets are not shared. In a typical FL setting, the Parameter Server is responsible for training the global model through aggregating trained local model parameters from clients. Unfortunately, prior studies [3]–[5] have shown that simply restricting the data sharing is not enough, as intermediate model parameters exchanged in FL can still leak clients' private training datasets.

To establish a secure FL environment, researchers have proposed different types of defense solutions. One direction of achieving privacy-preserving FL (PPFL) is to leverage homomorphic encryption (HE) to encrypt clients' local model parameters and allow the Parameter Server to execute FL aggregation over the ciphertext [6]–[8]. The HE-based PPFL protocol does protect the privacy of clients, but one issue is that HE usually introduces the huge computational overhead, which is not practical in FL. Another methodology is the use of data masking [9]–[11]. In these schemes, clients add mask values to uploaded model parameters, where mask values usually include pairwise masks to ensure the elimination from the aggregation results and a self-mask to prevent accidental disclosure of clients' privacy. The problem with such schemes is that they lead to the significant communication overhead.

A more promising and practical solution is to apply Differential Privacy (DP) [12], [13] to protect the privacy of clients' data. This DP-based scheme is generally implemented in two ways: one is that the Parameter Server adds noise to aggregated results (called CDP-FL) [14]–[16], and the other is that clients locally add noise to model parameters to be uploaded (called LDP-FL) [17]–[19]. In the former solution, clients' private data can still be leaked to the Parameter Server, which is not desirable. In the latter one, since each client adds noise, the bias of aggregation results becomes larger, which significantly degrades the performance of the model. To solve this issue, many studies have focused on making trade-offs between model utility and privacy protection [20]–[30]. However, they do not adequately consider the discrepancy in the impact of noise on model parameters with different values. Intuitively, adding too much noise to model parameters with smaller values may violate the original update direction of the model, but not vice versa. That is, there is heterogeneity among different model parameters. Therefore, adding noise that varies with the magnitude of model parameters can promisingly reduce the damage of DP to the model accuracy.

The key to realizing the above mentioned adaptive noise mechanism in LDP-FL is a tight estimation of sensitivity, as sensitivity is a concept in DP that determines the magnitude of noise [13]. The previous works on DP clip the norm by a predetermined threshold for calculating the sensitivity [18], [26]. If this threshold is too small, it will make the update direction of model deviate too much, and if this threshold is too large, it will make the added noise too large. That is, the threshold value that is too small or too large will affect the model performance. There are also some recent works to solve this issue by proposing an adaptive threshold. For example, Fu *et al*. [27] propose to calculate an threshold by the norm of previous gradients to estimate the sensitivity.

R. Xue, K. Xue, B. Zhu, X. Luo, Q. Sun and J. Lu are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China.

T. Zhang is with the School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore

K. Xue and J. Lu are also with the Key Laboratory of Medical Electronics and Digital Health of Zhejiang Province, and Engineering Research Center of Intelligent Human Health Situation Awareness of Zhejiang Province, Jiaxing, 314001, China.

Corresponding Author: K. Xue (kpxue@ustc.edu.cn)

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

Wei *et al*. [29] propose a method to estimate sensitivity by first truncating per-example gradient and then traversing the batch of datasets to find the maximum norm in each layer of model gradients on each batch of data. While when the number of local training for client $K > 1$, what needs to calculate is the sensitivity of local model weights uploaded by clients. And the method proposed in the above works cannot be used to calculate the sensitivity of model weights. What's more, this approach of estimating sensitivity by norm clipping does not take into account the discrepancy among the sensitivity of different components of local model parameters, and it is obvious that heterogeneity also exists among different components. Therefore, in order to effectively mitigate the negative impact of DP on model performance in FL, there needs a more FL-applicable method that allows for more fine-grained estimation of sensitivity.

To deal with the aforementioned issues, we propose a differentially private federated learning scheme with an adaptive noise mechanism. In particular, we design a method for clients to adaptively estimate sensitivity of local model parameters component by component. The detailed process of our scheme is described as follows: 1) clients use global model parameters obtained from the previous FL aggregation and local historical gradients to compute the component-by-component threshold, and use this threshold to truncate uploaded parameters; 2) they can then estimate sensitivity based on its definition; 3) after that, each client adds appropriate noise to local model parameters depending on estimated sensitivity and privacy budget (the concept in DP that measures the level of privacy protection) and upload them; 4) then the Parameter Sever uses collected perturbed model parameters to execute aggregation. We have mainly made the following contributions:

- We propose a general differentially private federated learning scheme, which requires clients to add different scales of noise to different components of local model parameters depending on the estimated sensitivity. It mitigates the degradation of model accuracy due to DP, thus achieving smaller model accuracy loss at higher privacy protection levels. The experiment results show that our proposed scheme outperforms existing works.
- To estimate sensitivity more precisely in the absence of data visibility in FL, we propose a universal method that combines local and global historical information to accurately estimate sensitivity of clients' uploaded model parameters. To verify the effectiveness of our proposed method, we instantiate it on four mainstream optimizers and experimental results convincingly prove that the estimated values can precisely reflect the trend of model parameters. Therefore, using our designed method in sensitivity estimation process in DP-FL can further reduce the impact of DP on FL.
- We conduct extensive experiments under different levels of privacy protection to evaluate the performance of our proposed scheme. The experimental results show that: 1) Our proposed scheme can accelerate the convergence of FL model; 2) The model obtained using our proposed scheme achieves a low accuracy loss, which is much superior than prior works. This indicating that our proposed scheme can maintain satisfactory performance even at a high level of privacy protection.

The reminder of the paper is organized as follows. Section II introduces the related work. Section III states the problem model used in our proposed scheme. The presentation of federated learning and DP is given in Section IV. Section V describes our proposed scheme in detail, provides privacy and utility analysis of our proposed scheme. Section VI demonstrates our extensive experiments. Finally, Section VII concludes this paper.

II. Related Work

Current works on DP-FL can be classified into three categories. The first one is centralized differential privacy (CDP) [14]–[16]. In this setting, clients train a local model and send raw model parameters to the Parameter Server. The Parameter Server aggregates the received parameters to get a new global model and adds noise to it before distribution. In CDP-FL, clients' private data can still be leaked to the Parameter Server, which can cause privacy threats in practice.

To overcome this limitation, local differential privacy (LDP) is proposed, in which clients perturb model parameters locally before uploading. Wei *et al*. [18] rigorously proved that the proposed algorithm satisfied the differential privacy requirement, and formulated a theoretical convergence bound on loss function of LDP-FL. The FL scheme with LDP does not need a trusted Parameter Server. However, the cost of this greater privacy-preserving capability is the degradation of model accuracy. There have been numerous works aiming at reducing the negative impact of the added noise on model performance. For example, since the shuffle model has privacy-enhancing effect, Sun *et al*. [20], Ghazi *et al*. [21] and Girgis *et al*. [22] reduced clients-added noise by incorporating the shuffle model, respectively. But these schemes require a trusted entity to perform shuffling leading to the impracticality. Gratton *et al*. [23] eliminated the need of introducing extra noise by replacing SGD with the alternating direction method of multipliers (called ADMM) and a zero-order distributed algorithm, whose differential privacy property is inherent. There are also some works aimed on optimizing noise mechanism, just as the focus of this paper. Zhang *et al*. [24] proposed to allocate different privacy budgets for different training rounds. Han *et al*. [25] investigated the factors affecting noise and proposed an optimization method to avoid truncation of model parameters by estimating sensitivity through PauTa judgement criterion. Zhou *et al*. [26] proposed that for a given privacy protection level, there existed an optimal number of times to participate in FL aggregation for clients. Fu *et al*. [27] proposed a method to estimate the sensitivity based on the mean value of gradients in the previous round. Zhang *et al*. [28] rigorously provided theoretical analysis of the effect of norm clipping on model performance in LDP-FL. Wei *et al*. [29] proposed an algorithm for optimizing noise mechanism using adaptive sensitivity and time-varying noise level. Yuan *et al*. [30] proposes a

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

novel noise with a geometric progression form of amplitude, and obtained the optimal number of aggregation rounds by theoretical analysis of this noise mechanism. However, none of them took into account the nature of different tolerance to noise for components with different values in client's local model parameters. And no work has been done to propose a method to accurately estimate sensitivity in FL, which would result in a poor mitigation of impact of noise on the model. In the experimental section, we compare our proposed scheme with state-of-the-art related works about optimizing noise mechanism in differential privacy, and the results show that their proposed schemes do not perform as well as our proposed scheme.

In recent years there has also been work [31] that through combining secure aggregation protocols to eliminate reliance on the trusted Parameter Server in CDP and model accuracy as high as that of CDP-FL, but it needs extra communication and computation overheads.

III. Problem Model

A. System Model

We consider a general scenario in FL with the Parameter Sever and multiple clients. We then describe their functions separately.

- **Parameter Server**: The Parameter Server is a central aggregator which is responsible for initialization and update of the global model. It collects and aggregates local model parameters uploaded by clients. Then the Parameter Server updates global model parameters based on aggregation results and distributes them to clients for the next FL aggregation.
- **Client**: Client is an entity that holds a local dataset and wishes to collaborate with others to obtain a model with high accuracy. After receiving global model parameters from the Parameter Server, Client uses owned dataset to train the local model. Then he/she uploads trained local model parameters to the Parameter Server and waits until receiving new global model parameters. Each client only communicates with the Parameter Server.

The designed scheme does not require all clients to participate in the entire FL training process. During the process of FL aggregation, the Parameter Server only needs to collect local model parameters uploaded by clients who have participated in this FL aggregation, and then executes aggregation.

B. Design Goals

In FL, local model parameters uploaded by client may disclose the privacy of dataset he/she owns [3]–[5]. And clients may suspect that the Parameter Server will try to infer privacy information from uploaded model parameters. Therefore, the main goal of our proposed scheme is to design a privacy-preserving federated learning scheme through introducing DP, which ensures that the privacy of single sample in client's owned dataset is not compromised and does not rely on a trusted Parameter Server. What the Parameter Server and external attackers can obtain are only model parameters perturbed by clients. Further, clients want a model with high accuracy while not compromising their privacy. So this paper aims to design a differentially private federated learning scheme that maintains a high level of privacy protection while obtaining a model with satisfactory accuracy.

Apart from the above, it is impractical to require clients to participate in the whole FL training due to problems such as communication delays or unstable connections. So the scheme we design should also support clients to drop out during the training.

C. Security Assumption

In FL, clients want to obtain the optimal model for future analysis or applications, so they are assumed to honestly follow the protocol to train and upload local model parameters, while concerning about the privacy leakage of owned datasets. We assume that the Parameter Server is honest but curious: it will correctly execute FL aggregation, but with the possibility of trying to infer private information about clients from collected local model parameters.

In addition, we consider external attackers who try to capture model parameters uploaded by clients. Attackers who maliciously tamper with communication messages are not considered in this paper. We believe that secure communication channels exist between clients and the Parameter Server.

IV. Preliminaries

This section introduces some knowledge about FL and related concept of DP. The notations we will use throughout this paper are summarized in Table I, where we call a round the period between the end of previous FL aggregation and the end of next FL aggregation.

A. Federated Learning

In this paper, we consider a Feature-Aligned FL system composed of n clients and a Parameter Server, in which datasets of clients have the same feature space and different sample spaces. And clients-owned datasets follow the independent identical distribution (i.i.d). Let $\mathcal{C}_i$ denote the i^{th} client, and $\mathcal{D}_i$ denote the dataset owned by $\mathcal{C}_i$, where $i \in 1, \ldots, n$. Here we assume that $\mathcal{D}_i$ and $\mathcal{D}_j$ have empty intersection for $\forall i \neq j$. The training process for FL is described in Algorithm 1 [32]. The Parameter Server first performs initialization of global model parameters (Line 1 in Algorithm 1). Between two rounds, the participating clients execute local update based on their owned datasets and the current global model (Line 3-11 in Algorithm 1). Then they upload the updated local model parameters to the Parameter Server. The Parameter Server aggregates received results to update the global model (Line 12-14 in Algorithm 1) and distributes it to clients. This process is repeated iteratively until a termination condition is met, such as a specified number of rounds or a satisfactory performance is achieved.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

TABLE I
DESCRIPTIONS OF NOTATIONS IN THIS PAPER

Symbol	Descriptions
$\mathcal{N}(*)$	the normal distribution
$\mathcal{D}, \mathcal{D}'$	two adjacent datasets
$\mathcal{S}$	set of all clients
$\mathcal{C}_i$	i^{th} client
$\mathcal{D}_i$	$\mathcal{C}_i$-owned dataset
$\|*\|$	size of set
Δ_*	sensitivity of function or model parameters
(ϵ, δ^*)	the privacy budget
$\mathcal{Z}_t$	set of participating clients in t^{th} round
K	number of local updates for clients in a round
T	number of FL aggregations
$\boldsymbol{L}$	a vector recording number of times each client has participated in FL aggregation
$\boldsymbol{w}_i$	local model parameters uploaded by $\mathcal{C}_i$
$\mathcal{G}$	optimizer used for local update
$F_i(*)$	local loss function for $\mathcal{C}_i$
$\boldsymbol{g}_k^i$	$\mathcal{C}_i$'s local model gradients obtained from k^{th} local update in a round
$\boldsymbol{w}_k^i$	$\mathcal{C}_i$'s local model parameters obtained from k^{th} local update in a round
$\boldsymbol{w}_0^i$	current global model parameters when performing local update
$*^{(m)}$	m^{th} component of a function or model parameters
M	number of total components of model parameters
σ_0	the noise level
σ_m	noise added to m^{th} component of local model parameters
β	the truncation factor
$\bar{*}$	truncated model parameters
$\tilde{*}$	model parameters after noise
$\boldsymbol{w}_t^*$	global model parameters updated in t^{th} round

Algorithm 1: FedAvg [32]

Data: number of clients n, number of FL aggregations T, number of local updates by client in a round K, batch size $\mathcal{B}$
Result: model parameters $\boldsymbol{w}$

```
The Parameter Server executes the initialization of global model parameters w;
for t ← 1 to T do
    for client C_i ∈ Z_t do
        w_i ← w;
        for epoch k ← 1 to K do
            B ← dividing D_i into batches of size B;
            for each batch b ∈ B do
                w_i ← OptimizerUpdate(w_i, b);
            end
        end
    end
    The Parameter Server performs FL aggregation:
    D ← ∪_{i∈Z_t} D_i;
    w ← Σ_{i∈Z_t} (|D_i|/|D|) w_i ;
end
```

B. Differential Privacy

In LDP-FL, clients locally perturb their updated model parameters to be uploaded. Let $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ be a noise algorithm employed by clients. Then we have the following definition:

Definition 1 ($(\epsilon, \delta)-$DP [33])**.** *A noise algorithm $\mathcal{M}$ satisfies $(\epsilon, \delta)-$DP if and only if for any two adjacent datasets $\mathcal{D}, \mathcal{D}' \subseteq \mathcal{X}$ and output $y \subseteq \mathcal{Y}$, we have*

$$Pr[\mathcal{M}(\mathcal{D}) \in y] \leq e^{\epsilon} Pr[\mathcal{M}(\mathcal{D}') \in y] + \delta,$$

where adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$ differ in only one sample.

Here (ϵ, δ) is the privacy budget. To ensure that the privacy budget is not exceeded throughout FL training, we use Rényi Differential Privacy (RDP) for a better track of the privacy loss, which is a relaxed version of DP. Compared to $(\epsilon, \delta)-$DP, RDP provides an operationally more convenient, quantitatively more accurate way to track the cumulative privacy loss from a composition of multiple mechanisms. RDP is formally defined as follows.

Definition 2 (Rényi Divergence [34])**.** *Given two probability distributions F and G, the Rényi divergence of order α is defined as*

$$D_\alpha(F \| G) = \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim G} \left(\frac{F(x)}{G(x)} \right)^\alpha.$$

Definition 3 ((α, ϵ)-RDP [34])**.** *A noise mechanism $f : \mathcal{X} \to \mathcal{Y}$ satisfies the Rényi differential privacy of order α, i.e. (α, ϵ)-RDP, if for any two adjacent datasets $\mathcal{D}, \mathcal{D}' \subseteq \mathcal{X}$, the following inequality holds:*

$$D_\alpha(f(\mathcal{D}) \| f(\mathcal{D}')) \leq \epsilon.$$

Next, we give the definition of l_2-sensitivity for a function f as follows:

$$\Delta_f = \max_{\mathcal{D}, \mathcal{D}'} \| f(\mathcal{D}) - f(\mathcal{D}') \|_2, \tag{1}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are adjacent datasets, and $\| * \|_2$ is l_2-norm. Then we can introduce theoretical basis of differential privacy guarantee for component adaptive noise.

Lemma 1 (Differential Privacy Assurance for Component Adaptive Noise [35])**.** *Suppose the noise mechanism $\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + Q$ satisfies $(\epsilon, \delta)-$DP, where $\mathcal{D}$ is input dataset, f is a function of M dimensions that meets $\Delta_f \leq 1$ and noise $Q \sim \mathcal{N}(0, \sigma_0^2 I)$, I is an M-dimensional identity matrix. Then for any function f' of M dimension, mechanism $\mathcal{M}'(\mathcal{D}) = f'(\mathcal{D}) + Q'$ still meets $(\epsilon, \delta)-$DP, if $Q' = (Q_1, \ldots, Q_M), Q_i \sim \mathcal{N}(0, \sigma_m^2)$ satisfy $\sum_{m=1}^{M} \frac{\Delta_m^2}{\sigma_m^2} \leq \frac{1}{\sigma_0^2}$, Δ_m is l_2-sensitivity of m^{th} dimension of f', $m \in 1, \ldots, M$.*

V. METHODOLOGY

A. Overview

In FL, participating clients want to cooperate together to obtain the optimal model and do not want their privacy to be compromised. Thus our goal is to design an efficient and practical LDP-FL scheme. However, the introduced noise inevitably causes the degradation of model accuracy, so we want to mitigate the negative effect of noise to the maximum extent possible.

In our proposed scheme, the Parameter Server first executes initialization of global model parameters and distributes them to clients. Then participating clients train the local model

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

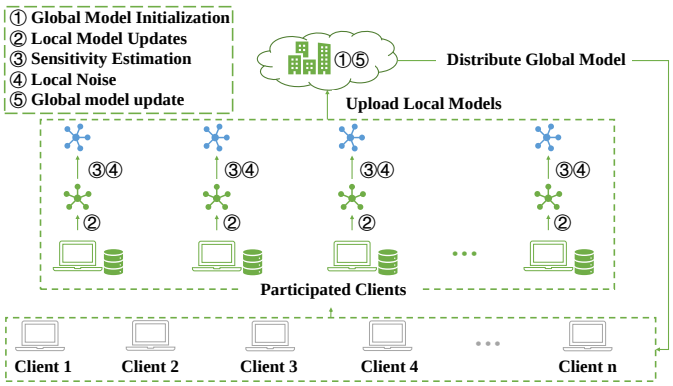


Fig. 1. Overview of our proposed scheme

using current global model parameters and owned datasets. After completing the update of local model, clients perform truncation and then estimate sensitivity of updated local model parameters. After that, clients locally add noise to meet DP requirements according to estimated sensitivity and upload perturbed local model parameters to the Parameter Server. Then, the Parameter Server collects local model parameters and performs FL aggregation to get new global model parameters, which are distributed to clients for the next round.

B. Workflow

Fig.1 shows the workflow of our proposed scheme, which consist of five steps. Algorithm 2 shows overall procedure. Algorithm 3 describes how client updates local model parameters (ClientUpdate). Algorithm 4 describes how cumulative privacy loss is computed (PrivacyCompute). Below we illustrate each step in the proposed scheme.

1) **Global model initialization.** At the beginning, the Parameter Server negotiates the privacy budget (ϵ, δ^*) with clients and initializes global model parameters $\boldsymbol{w}_0^*$ (Line 1-2 in Algorithm 2). After that, the Parameter Server distributes $\boldsymbol{w}_0^*$ to clients.
2) **Local model updates.** In t^{th} round, for the sake of illustration, we consider participating client $\mathcal{C}_i$. He/She first initializes local model parameters $\boldsymbol{w}_0^i$ to the received current global model parameters (Line 1-2 in Algorithm 3) and records number of times L_i he/she has participated in FL aggregation up to now (Line 3 in Algorithm 3). Then $\mathcal{C}_i$ uses owned dataset to perform K local updates to update local model parameters $\boldsymbol{w}_i$ (Line 4-16 in Algorithm 3). It is to be explained that in line 10 of Algorithm 3, we use $\mathcal{G}$ to denote the optimizer algorithm used by client during local training, e.g., if SGD is used, then $\mathcal{G}(\boldsymbol{g}_k^i) = \eta \boldsymbol{g}_k^i$, where η is the learning rate.
3) **Sensitivity estimation.** $\mathcal{C}_i$ firstly obtains updated local model parameters $\boldsymbol{w}_K^i$. For simplicity, we use f_i to denote $\boldsymbol{w}_K^i$. Then based on local historical gradients and received global model parameters from the previous round, $\mathcal{C}_i$ calculates the truncation threshold according to our proposed method and executes truncation of f_i (Line 19-20 in Algorithm 3):

$$\begin{aligned} q^{i(m)} &= \frac{\beta}{2}\left|\boldsymbol{w}_{K-1}^{i(m)} - \mathcal{G}(\boldsymbol{g}_{K-1}^{i(m)})\right|, \\ \bar{f}_i^{(m)} &= \min\{\max\{f_i^{(m)}, \boldsymbol{w}_0^{i(m)} - q^{i(m)}\}, \\ &\qquad \boldsymbol{w}_0^{i(m)} + q^{i(m)}\}, \end{aligned} \tag{2}$$

where β is an adjustable parameter which we call truncation factor. Then sensitivity of truncated model parameters is estimated as follows:

$$\Delta_{f_i}^{(m)} = \beta\left|\boldsymbol{w}_{K-1}^{i(m)} - \mathcal{G}(\boldsymbol{g}_{K-1}^{i(m)})\right|, \tag{3}$$

where superscript (m) denotes the m^{th} component of model parameters.
4) **Local noise.** Based on sensitivity obtained from step 3), client computes the noise to be added and perturbs updated model parameter (Line 21-22 in Algorithm 3):

$$\begin{aligned} \sigma_m &= \sqrt{M}\sigma_0 \Delta_{f_i}^{(m)}, \\ \widetilde{f}_i^{(m)} &= \mathcal{M}(f_i^{(m)}) = \bar{f}_i^{(m)} + \mathcal{N}(0, \sigma_m^2). \end{aligned} \tag{4}$$

Algorithm 2: Our Proposed Scheme

Data: the noise level σ_0, set of all clients $\mathcal{S} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$, sampling ratio q, the privacy budget (ϵ, δ^*), number of local updates by client in a round K, number of FL aggregations T, number of total components of model parameters M, the vector $\boldsymbol{L}$ which records the number of times each client participating in FL aggregation

Result: global model parameters $\boldsymbol{w}_T^*$

1 The Parameter Server performs initialization:
2 $\boldsymbol{w}_0^* = (\boldsymbol{w}_0^{*(1)}, \ldots, \boldsymbol{w}_0^{*(M)})$, $\delta = 0$, $\boldsymbol{L} = (L_1, \ldots, L_n) = (0, \ldots, 0)$;
3 **for** $t \leftarrow 1$ *to* T **do**
4 $R \leftarrow 0$, $\mathcal{Z}_t \leftarrow$ set of participating clients in t^{th} round;
5 **for** *each client* $\mathcal{C}_i \in \mathcal{Z}_t$ *in parallel* **do**
6 $\boldsymbol{w}_i, \delta_i \leftarrow$ ClientUpdate$(\epsilon, \sigma_0, i, \boldsymbol{w}_{t-1}^*, q, L_i, t)$
7 **end**
8 The Parameter Server executes:
9 collect local model parameters $\boldsymbol{w}_i$ and privacy loss δ_i from clients;
10 **for** $i \in Z_t$ **do**
11 $R \leftarrow R + |\mathcal{D}_i|$;
12 **end**
13 $\boldsymbol{w}_t^* \leftarrow \sum_{i \in \mathcal{Z}_t} \frac{|\mathcal{D}_i|}{R} \boldsymbol{w}_i$;
14 calculate current privacy loss: $\delta \leftarrow \max\{\delta_i | i \in \mathcal{Z}_t\}$;
15 **if** $\delta > \delta^*$ **then**
16 **return** $\boldsymbol{w}_t^*$;
17 **break**;
18 **end**
19 **end**

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

Algorithm 3: CLIENTUPDATE in each client

Data: the privacy budget (ϵ, δ^*), the noise level σ_0, the id for client i, current global model parameters $\boldsymbol{w}^*_{t-1}$, sampling ratio q, current number of rounds t

Result: updated local model's parameters $\boldsymbol{w}_i$

1 The initialization of local model parameters:
2 $\boldsymbol{w}^i_0 \leftarrow \boldsymbol{w}^*_{t-1}$;
3 $L_i \leftarrow L_i + 1$;
4 $k \leftarrow 0$;
5 **while** $k \leq K$ **do**
6 $B \leftarrow q|\mathcal{D}_i|$;
7 $\mathcal{B} \leftarrow$ dividing $\mathcal{D}_i$ into batches of size B;
8 **for** *each batch* $b \in \mathcal{B}$ **do**
9 $\boldsymbol{g}^i_k \leftarrow \frac{1}{|b|}\sum_{j\in b} \nabla_{\boldsymbol{w}^i_{k-1}} F_i(\boldsymbol{w}^i_{k-1}, \boldsymbol{x}_j)$;
10 $\boldsymbol{w}^i_k \leftarrow \boldsymbol{w}^i_{k-1} - \mathcal{G}(\boldsymbol{g}^i_k)$;
11 $k \leftarrow k + 1$;
12 **if** $k > K$ **then**
13 **break**;
14 **end**
15 **end**
16 **end**
17 $f_i \leftarrow \boldsymbol{w}^i_K$;
18 **for** $m \leftarrow 1$ *to* M **do**
19 $q^{i(m)} = \frac{\beta}{2}\big|\boldsymbol{w}^{i(m)}_{K-1} - \mathcal{G}(\boldsymbol{g}^{i(m)}_{K-1})\big|$;
20 $\bar{f}^{(m)}_i = \min\{\max\{f^{(m)}_i, \boldsymbol{w}^i_0 - q^{i(m)}\}, \boldsymbol{w}^i_0 + q^{i(m)}\}$;
21 $\Delta^{(m)}_{f_i} = 2q^{i(m)}$, $\sigma_m = \sqrt{M}\sigma_0\Delta^{(m)}_{f_i}$;
22 $\widetilde{f}^{(m)}_i = \bar{f}^{(m)}_i + \mathcal{N}(0, \sigma^2_m)$;
23 **end**
24 $\boldsymbol{w}_i \leftarrow \widetilde{f}_i$;
25 $\alpha^* \leftarrow 1 + \frac{\log(1/\delta^*)}{\epsilon}$;
26 $\delta_i \leftarrow$ PRIVACYCOMPUTE$(\epsilon, \alpha^*, L_i, \sigma_0)$;
27 **Return** $\boldsymbol{w}_i, \delta_i$;

Algorithm 4: PRIVACYCOMPUTE in the Parameter Server

Data: $\epsilon, \alpha^*, L_i, \sigma_0$

Result: δ

1 Define the function $e : \mathbb{Z} \to \mathbb{R}, \alpha \mapsto (\frac{L_i\alpha}{2\sigma^2_0} - \epsilon)(\alpha - 1)$;
2 $x \leftarrow$ the minimum value of e for $\alpha > \alpha^*$;
3 $\delta \leftarrow e^x$;
4 **Return** δ;

In addition, in order to keep client's cumulative privacy loss within the given privacy budget at all times, client needs to calculate current privacy loss before uploading local model parameters. Specifically, client first records number of times he/she has participated in FL aggregation to date, and then uses Algorithm 4 to calculate cumulative privacy loss δ_i (Line 26 in Algorithm 3), which is uploaded to the Parameter Server along with perturbed local model parameters $\boldsymbol{w}_i$.

5) **Global model update.** After collecting participating clients' uploaded local model parameters and current privacy loss, the Parameter Server performs FL aggregation to get new global model parameters from received values (Line 13 in Algorithm 2):

$$\boldsymbol{w}^*_t = \sum_{i\in\mathcal{Z}_t} \frac{|\mathcal{D}_i|}{|\mathcal{D}_t|}\boldsymbol{w}_i,$$

where $\mathcal{Z}_t$ denotes the subset of clients participating in t^{th} FL aggregation and $\mathcal{D}_t = \cup_{i\in\mathcal{Z}_t}\mathcal{D}_i$. Then the Parameter Server calculates the largest privacy loss value among clients as current global privacy loss (Line 14 in Algorithm 2). When this value does not exceed the negotiated threshold (this step is to ensure that privacy loss of all participating clients does not exceed the privacy budget), the Parameter Server distributes updated global model parameters $\boldsymbol{w}^*_t$ to clients for the next round.

As shown in Algorithm 2, the Parameter Server and clients iteratively perform steps 2 to 5 until the number of FL aggregations or privacy loss reaches specified value (Lines 3-19 in Algorithm 2).

C. Sensitivity Estimation

Since this paper proposes a new method for estimating sensitivity, which was not mentioned in previous work, we explain in this section why sensitivity can be estimated as in Eq. (3). For the sake of illustration, we consider sensitivity of local model parameters $\boldsymbol{w}_i$ updated by $\mathcal{C}_i$ in t^{th} round.

In t^{th} round, $\mathcal{C}_i$ performs the following update process:

$$\begin{aligned}\boldsymbol{g}^i_k &\leftarrow \frac{1}{|\mathcal{B}^i_k|}\sum_{j\in\mathcal{B}^k_i} \nabla_{\boldsymbol{w}^i_{k-1}} F_i(\boldsymbol{w}^i_{k-1}, \boldsymbol{x}_j),\\ \boldsymbol{w}^i_k &\leftarrow \boldsymbol{w}^i_{k-1} - \mathcal{G}(\boldsymbol{g}^i_k),\end{aligned} \tag{5}$$

for k from 1 to K, where K is the number of client's local updates in a round and $\mathcal{B}^i_k$ is batch of data to be used by $\mathcal{C}_i$ for k^{th} local update (that is, b in line 8 of Algorithm 3).

According to Eq. (5), we can calculate updated local model parameters $\boldsymbol{w}_i = \boldsymbol{w}^i_K$ by the following formula:

$$\begin{aligned} f_i = \boldsymbol{w}_i &= \boldsymbol{w}^i_K\\ &= \boldsymbol{w}^i_{K-1} - \mathcal{G}(\boldsymbol{g}^i_K)\\ &= \boldsymbol{w}^i_{K-2} - \mathcal{G}(\boldsymbol{g}^i_{K-1}) - \mathcal{G}(\boldsymbol{g}^i_K)\\ &= \dots\\ &= \boldsymbol{w}^i_0 - \sum_{k=1}^{K}\mathcal{G}(\boldsymbol{g}^i_k),\end{aligned} \tag{6}$$

where $\boldsymbol{w}^i_0$ is global model parameters from previous FL aggregation $\boldsymbol{w}^*_{t-1}$. Let $\boldsymbol{h}_{\boldsymbol{w}_i} = \sum_{k=1}^{K}\mathcal{G}(\boldsymbol{g}^i_k)$. Since $\boldsymbol{w}^i_0$ is independent of $\mathcal{C}_i$-owned dataset used in t^{th} round, according to the definition of sensitivity (Eq. (1)), we only need to estimate sensitivity of $\boldsymbol{h}_{\boldsymbol{w}_i}$.

Since historical gradient can be considered as priori information for current gradient [36], historical gradient can be used for estimation. And combined with Eq. (6), we can therefore use $\hat{\boldsymbol{h}}_{\boldsymbol{w}_i}$ to approximate $\boldsymbol{h}_{\boldsymbol{w}_i}$:

$$\begin{aligned}\hat{\boldsymbol{w}}_i &= \boldsymbol{w}^i_{K-1} - \mathcal{G}(\boldsymbol{g}^i_{K-1}),\\ \hat{\boldsymbol{h}}_{\boldsymbol{w}_i} &= \boldsymbol{w}^i_0 - \hat{\boldsymbol{w}}_i.\end{aligned}$$

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

TABLE II
SENSITIVITY ESTIMATION OF MODEL PARAMETERS USING DIFFERENT OPTIMIZERS

Optimizer	Sensitivity
SGD [37]	$\Delta_{f_i} = \beta\lvert \boldsymbol{w}^i_{K-1} - \eta \boldsymbol{g}^i_{K-1}\rvert$
Momentum [38]	$\boldsymbol{v}^i_{K-1} = \gamma \boldsymbol{v}^i_{K-2} + \eta \boldsymbol{g}^i_{K-1}$ $\Delta^i_{sum} = \gamma \boldsymbol{v}^i_{K-1} + \eta \boldsymbol{g}^i_{K-1}$ $\Delta_{f_i} = \beta\lvert \boldsymbol{w}^i_{K-1} - \Delta^i_{sum}\rvert$
Adam [39]	$\boldsymbol{m}_{K-1} = \gamma_1 \boldsymbol{m}_{K-2} + (1-\gamma_1)\boldsymbol{g}_{K-1}$ $\boldsymbol{v}_{K-1} = \gamma_2 \boldsymbol{v}_{K-2} + (1-\gamma_2)\boldsymbol{g}^2_{K-1}$ $\widetilde{\boldsymbol{m}}_K = \frac{\gamma_1 \boldsymbol{m}_{K-1} + (1-\gamma_1)\boldsymbol{g}_{K-1}}{1-\gamma_1^K}$ $\widetilde{\boldsymbol{v}}_K = \frac{\gamma_2 \boldsymbol{v}_{K-1} + (1-\gamma_2)(\boldsymbol{g}^i_{K-1})^2}{1-\gamma_2^K}$ $\Delta_{f_i} = \beta\lvert \boldsymbol{w}^i_{K-1} - \frac{\eta \widetilde{\boldsymbol{m}}_K}{\sqrt{\widetilde{\boldsymbol{v}}_K + \epsilon}}\rvert$
RMSPROP	$\mathbb{E}[\boldsymbol{g}^2]^i_{K-1} = \gamma \mathbb{E}[\boldsymbol{g}^2]^i_{K-2} + (1-\gamma)(\boldsymbol{g}^i_{K-1})^2$ $\Delta^i_{sum} = \eta \frac{\boldsymbol{g}^i_{K-1}}{\sqrt{\gamma \mathbb{E}[\boldsymbol{g}^2]^i_{K-1} + (1-\gamma)(\boldsymbol{g}^i_{K-1})^2 + \epsilon}}$ $\Delta_{f_i} = \beta\lvert \boldsymbol{w}^i_{K-1} - \Delta^i_{sum}\rvert$

Let f_i be $\boldsymbol{w}^i_K$, then from the definition of sensitivity (Eq. (1)) and the truncation process of parameters (Eq. (2)), we can obtain that,

$$\begin{aligned}
\Delta^{(m)}_{f_i} &= \max_{\mathcal{D},\mathcal{D}'} \|\boldsymbol{w}^i_K(\mathcal{D}) - \boldsymbol{w}^i_K(\mathcal{D}')\| \\
&= \max_{\mathcal{D},\mathcal{D}'} \|(\boldsymbol{w}^i_0 - \sum_{k=1}^{K} \mathcal{G}(\boldsymbol{g}^i_k))(\mathcal{D}) \\
&\qquad - (\boldsymbol{w}^i_0 - \sum_{k=1}^{K} \mathcal{G}(\boldsymbol{g}^i_k))(\mathcal{D}')\| \\
&= \max_{\mathcal{D},\mathcal{D}'} \Big\| \boldsymbol{h}_{\boldsymbol{w}_i}(\mathcal{D}) - \boldsymbol{h}_{\boldsymbol{w}_i}(\mathcal{D}') \Big\| \\
&\leq \beta \big| \boldsymbol{w}^{i(m)}_{K-1} - \mathcal{G}(\boldsymbol{g}^{i(m)}_{K-1}) \big|,
\end{aligned}$$

where β is the truncation factor. So far we obtain the estimation for sensitivity of model parameters.

Our proposed method for estimating sensitivity is general and can be applied to different optimizers. Now let's describe how to instantiate it on four common optimizers (SGD, Momentum, Adam, PMSPROP).

First let us show what $\mathcal{G}$ is in four optimizers (Eq. (5)). We consider the s^{th} update of model. For SGD, $\mathcal{G}(\boldsymbol{g}_s) = \eta \boldsymbol{g}_s$, where η is the learning rate. For Momentum, we have $\mathcal{G}(\boldsymbol{g}_s) = \boldsymbol{v}_s$, where $\boldsymbol{v}_s = \gamma \boldsymbol{v}_{s-1} + \eta \boldsymbol{g}_s$, γ, η are parameters in Momentum. For Adam, $\mathcal{G}(\boldsymbol{g}_s) = \frac{\eta \boldsymbol{m}_s}{\epsilon + \sqrt{\boldsymbol{v}_s}}$, where $\boldsymbol{m}_s = \frac{\gamma_1 \boldsymbol{m}_{s-1} + (1-\gamma_1)\boldsymbol{g}_s}{1-\gamma_1^s}$, $\boldsymbol{v}_s = \frac{\gamma_2 \boldsymbol{v}_{s-1} + (1-\gamma_2)\boldsymbol{g}_s^2}{1-\gamma_2^s}$, $\gamma_1, \gamma_2, \epsilon, \eta$ are parameters in Adam. For RMSPROP, we have $\mathcal{G}(\boldsymbol{g}_s) = \eta \frac{\boldsymbol{g}_s}{\sqrt{\mathbb{E}[\boldsymbol{g}^2]_s + \epsilon}}$, where $\mathbb{E}[\boldsymbol{g}^2]_s = \gamma \mathbb{E}[\boldsymbol{g}^2]_{s-1} + (1-\gamma)(\boldsymbol{g}_s)^2$, η, γ, ϵ are parameters in RMSPROP.

Then according to Eq. (3), we can then obtain the sensitivity of client $\mathcal{C}_i$'s local model parameters to be uploaded. TABLE II shows the estimated sensitivity of model parameters using four optimizers, respectively.

D. Privacy Analysis

To proof that our proposed scheme satisfies DP requirement, we first give the following lemmas:

Lemma 2 (RDP of Gaussian Mechanism [34])**.** *If sensitivity of a function $f : \mathcal{X} \to \mathcal{Y}$ meets $\Delta_f = 1$, then the Gaussian mechanism $G_\sigma(f) = f(\mathcal{D}) + \mathcal{N}(0, \sigma^2)$ satisfies $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP, where $\mathcal{D} \subseteq \mathcal{X}$.*

To analyze the cumulative privacy budget of our proposed scheme, we introduce the composition theorem of RDP.

Lemma 3 (Sequential Composition Theorem of RDP [34])**.** *Let $f : \mathcal{X} \to \mathcal{Y}_1$ satisfy (α, ϵ_1)-RDP and $g : \mathcal{Y}_1 \times \mathcal{X} \to \mathcal{Y}_2$ satisfy (α, ϵ_2)-RDP. We define $X \leftarrow f(\mathcal{D})$ and $Y \leftarrow g(X, \mathcal{D})$, where $\mathcal{D} \subseteq \mathcal{X}$. Then the mechanism (X, Y) satisfies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.*

Lemma 4 (Parallel Composition Theorem of RDP [34])**.** *Given a set of mutually disjoint data sets $\mathcal{D}_1, \ldots, \mathcal{D}_n$ satisfying $\mathcal{D}_i \subseteq \mathcal{X}$ for any $i = 1, \ldots, n$. Let $f : \mathcal{X} \to \mathcal{Y}$ satisfy (α, ϵ)-RDP, then the combination of these mechanisms $(f(\mathcal{D}_1), \ldots, f(\mathcal{D}_n))$ satisfy $(\alpha, \epsilon)-$RDP.*

Lemma 5 (From RDP to DP [34])**.** *If the noise mechanism $\mathcal{M}$ satisfies (α, ϵ)-RDP, then it also satisfies $(\epsilon + \frac{\log(1/\delta)}{\alpha - 1}, \delta)$-DP, where δ is any number within the range of 0 to 1.*

Lemma 5 is the theoretical guarantee that RDP and DP are translatable to each other.

Theorem 1. *Given a target privacy budget (ϵ, δ), the number of times client $\mathcal{C}_i$ uploads local model parameters L_i and any integer α which satisfies $\alpha > 1 + \frac{\log(1/\delta)}{\epsilon}$, if σ_0 satisfies*

$$\frac{L_i \alpha}{2\sigma_0^2} + \frac{\log(1/\delta)}{\alpha - 1} \leq \epsilon, \tag{7}$$

where M is the number of components of model parameters. Then for client $\mathcal{C}_i$, our proposed scheme satisfies $(\epsilon, \delta)-$DP.

Proof. Let us first show that in order to ensure that our proposed scheme satisfies $(\epsilon, \delta)-$DP, client $\mathcal{C}_i$ needs to add noise to local model parameters in a single upload that satisfies $(\alpha, \frac{\epsilon'}{L_i})-$RDP, where ϵ' satisfies

$$\epsilon' + \frac{\log(1/\delta)}{\alpha - 1} \leq \epsilon. \tag{8}$$

Assuming that $\mathcal{C}_i$ uploads his/her local model parameters L_i times throughout FL. According to sequential composition theorem of RDP (Lemma 3), when client $\mathcal{C}_i$ adds noise to local model parameters in a single upload that satisfy $(\alpha, \frac{\epsilon'}{L_i})-$RDP, then after the entire FL, the noise mechanism for client $\mathcal{C}_i$ satisfies $(\alpha, \epsilon')-$RDP. Further from Lemma 4, we know that the noise mechanism for our propose scheme satisfies $(\alpha, \epsilon')-$RDP. By the conversion between RDP and DP (Lemma 5), if ϵ' satisfies $\epsilon' + \frac{\log(1/\delta)}{\alpha - 1} \leq \epsilon$, then the proposed scheme satisfies (ϵ, δ)-DP.

Then, according to the definition of sensitivity (Eq. (1)), the truncation of local model parameters (Eq. (2)) and Eq. (3), we can obtain that the sensitivity of local model parameters Δ_f satisfies

$$\Delta_f \leq \sqrt{\sum_m (\Delta^{(m)}_{f_i})^2}.$$

Thus, we only need to select the appropriate β to satisfy the requirement for sensitivity in Lemma 2. Then, according to

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

Lemma 2, in $\mathcal{C}_i$'s single upload of local model parameters, to ensure that the noise mechanism for $\mathcal{C}_i$ satisfies $(\alpha, \frac{\epsilon'}{L_i})-$RDP, the added noise scale σ_0 needs to satisfy

$$\sigma_0^2 \geq \frac{L_i \alpha}{2\epsilon'}. \tag{9}$$

Combining Eq. (8) and Eq. (9), we prove that, if α satisfies $\alpha > 1 + \frac{\log(1/\delta)}{\epsilon}$ and σ_0 satisfies

$$\sigma_0^2 \geq \frac{L_i \alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha - 1})}, \tag{10}$$

our proposed scheme satisfies $(\epsilon, \delta)-$DP. Then we can write Eq. (10) as

$$\frac{L_i \alpha}{2\sigma_0^2} + \frac{\log(1/\delta)}{\alpha - 1} \leq \epsilon.$$

Finally, let us calculate the noise σ_m added to each component of local model parameters to be uploaded by $\mathcal{C}_i$, $m = 1, \ldots, M$. Since RDP and DP can be converted to each other (Lemma 5), Lemma 1 applies to RDP as well. Given the estimated sensitivity of each component of local model parameters $\Delta_{f_i}^{(m)}$ (Line 21 in Algorithm 3), then according to Lemma 1 (it has been shown above that selecting the appropriate β can ensure that the sensitivity of local model parameters satisfies the condition in Lemma 1), σ_m only needs to satisfy $\sum_{m=1}^{M} (\frac{\Delta_{f_i}^{(m)}}{\sigma_m})^2 \leq \frac{1}{\sigma_0^2}$. The σ_m taken in Line 21 of Algorithm 3 clearly satisfies this condition.

In summary, therefore, when σ_0 satisfies

$$\frac{L_i \alpha}{2\sigma_0^2} + \frac{\log(1/\delta)}{\alpha - 1} \leq \epsilon,$$

our proposed scheme is $(\epsilon, \delta)-$DP for all participating clients. Thus our proposed scheme satisfies $(\epsilon, \delta)-$DP. The proof is completed.

□

Now let us explain how Algorithm 4 tracks the cumulative privacy loss for $\mathcal{C}_i$.

In the initialization phase of FL (the input **Data** in Algorithm 2), we predetermine the privacy budget (ϵ, δ^*) and the noise level σ_0, initialize L_i to 0 for each client (Line 2 in Algorithm 2). Once $\mathcal{C}_i$ uploads local model parameters (i.e., participates in FL aggregation), we let $L_i \leftarrow L_i + 1$ (Line 3 in Algorithm 3). When tracking the privacy loss, since ϵ has been predetermined, we only need to calculate current δ_i for each client.

The process of calculating δ_i is described as follows (Algorithm 4). First we transform Eq. (7) in Theorem 1 into

$$\delta \geq \exp\{(\alpha - 1)(\frac{L_i \alpha}{2\sigma_0^2} - \epsilon)\}. \tag{11}$$

Then according to Eq. (11), the current privacy loss for $\mathcal{C}_i$ is (ϵ, δ_i), where

$$\delta_i = \min_{\alpha}\{\exp\{(\alpha - 1)(\frac{L_i \alpha}{2\sigma_0^2} - \epsilon)\}\}, \tag{12}$$

α is any integer that satisfies $\alpha > 1 + \frac{\log(1/\delta^*)}{\epsilon}$, L_i is the current number of times $\mathcal{C}_i$ has participated in FL aggregation. According to the above equation, we iterate over the range of values of α to find the minimum of $\exp\{(\alpha - 1)(\frac{L_i \alpha}{2\sigma_0^2} - \epsilon)\}$ (Line 2-3 in Algorithm 4), which is current δ_i for $\mathcal{C}_i$.

Finally, to ensure that Eq. (11) is satisfied for each client, we take current global privacy loss as $\delta = \max_i\{\delta_i | i = 1, \ldots, n\}$ (Line 14 in Algorithm 2).

Our proposed scheme needs to support for dropouts, thus we can not determine L_i for client in the initialization phase of FL. Let us explain how to predetermine σ_0 (Eq. (7)) in our experiments.

From Eq. (12), we can see that δ_i increases monotonically with L_i. Since $L_i \leq T$, in order to guarantee that the final privacy loss δ_i does not exceed δ^*, we just need to choose the noise level σ_0 to satisfy

$$\min_{\alpha}\{\exp\{(\alpha - 1)(\frac{T\alpha}{2\sigma_0^2} - \epsilon)\}\} \leq \delta^*, \tag{13}$$

where α satisfies $\alpha > 1 + \frac{\log(1/\delta^*)}{\epsilon}$.

E. Convergence Analysis

In this section, we analyze the unbiasedness of our proposed scheme and its convergence.

Lemma 6. *Given any real number x, the noise mechanism $\mathcal{M}$ which is given in Eq. (4) satisfies $\mathbb{E}(\mathcal{M}(x)) = x$.*

Proof. Since $\mathcal{N}(0, \sigma_m^2)$ in Eq. (4) follows a normal distribution, then

$$\mathbb{E}(\mathcal{M}(x)) = \mathbb{E}(x + \mathcal{N}(0, \sigma_m^2)) = x.$$

The proof is completed. □

Lemma 7. *Let $\mathcal{M}$ be the noise machanism proposed in Eq. (4). Given any real number x, then the variance of $\mathcal{M}$ satisfies $Var[\mathcal{M}(x)] \leq \frac{ML_i\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}$, wnere α is an arbitrary integer satisfying that $\frac{\log(1/\delta)}{\alpha-1} \leq \epsilon$, M is the number of components of model parameters.*

Proof. Since $\mathcal{N}(0, \sigma_m^2)$ in Eq. (4) follows a normal distribution, then

$$\begin{aligned} Var[\mathcal{M}(x)] &= \mathbb{E}[(\mathcal{M}(x) - \mathbb{E}(\mathcal{M}(x)))^2] \\ &= \mathbb{E}[(\mathcal{N}(0, \sigma_m^2))^2] \\ &= \sigma_m^2. \end{aligned}$$

According to Eq. (7), we can obtain that:

$$2\sigma_0^2(\epsilon - \frac{\log(1/\delta)}{\alpha - 1}) \geq L_i \alpha,$$

where L_i is the number of times $\mathcal{C}_i$ uploads local model parameters. Then when α satisfies $\alpha \geq \frac{\log(1/\delta)}{\epsilon} + 1$, we have

$$\sigma_0^2 \geq \frac{L_i \alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}.$$

To minimize the accuracy loss, we take $\sigma_0^2 = \frac{L_i\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}$. Substituting $\sigma_m = \sqrt{M}\sigma_0\Delta_{f_i}^{(m)}$ (Line 21 in Algorithm 3) into the above equation, and choose β to satisfy $\Delta_{f_i}^{(m)} \leq 1$, then we can know that

$$\sigma_m^2 = \frac{ML_i\alpha(\Delta_{f_i}^{(m)})^2}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})} \leq \frac{ML_i\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}.$$

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

Then we have proved that

$$Var[\mathcal{M}(x)] = \sigma_m^2 \leq \frac{ML_i\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}.$$

The proof is completed. □

Theorem 2 (Unbiasedness)**.** *our proposed scheme proposed in this paper introduces no bias, i.e.,* $\mathbb{E}(\mathcal{M}(\boldsymbol{w}^*)) = \boldsymbol{w}^*$.

Proof. According to Lemma 6, we can know that for all components of local model parameters $\boldsymbol{w}_i$, the equation

$$\mathbb{E}(\mathcal{M}(\boldsymbol{w}_i^{(m)})) = \boldsymbol{w}_i^{(m)}, m = 1, \ldots, M$$

holds. Then based on line 13 in Algorithm 2, in t^{th} FL aggregation, we can conclude that:

$$\begin{aligned}\mathbb{E}(\mathcal{M}(\boldsymbol{w}_t^*)) &= \mathbb{E}\big[\sum_{i\in\mathcal{Z}_t}\frac{|\mathcal{D}_i|}{R}\mathcal{M}(\boldsymbol{w}_i)\big] = \sum_{i\in\mathcal{Z}_t}\frac{|\mathcal{D}_i|}{R}\mathbb{E}[\mathcal{M}(\boldsymbol{w}_i)] \\ &= \sum_{i\in\mathcal{Z}_t}\frac{|\mathcal{D}_i|}{R}\boldsymbol{w}_i = \boldsymbol{w}_t^*,\end{aligned}$$

where $\boldsymbol{w}_i$ represents the uploaded local model parameters by $\mathcal{C}_i$, $\boldsymbol{w}_t^*$ is the aggregated result of uploaded local model parameters, R is the total size of datasets owned by clients participating in t^{th} FL aggregation. And since the above holds for all t, thus we can prove that $\mathbb{E}(\mathcal{M}(\boldsymbol{w}^*)) = \boldsymbol{w}^*$. □

Lemma 8. *Let global model parameters obtained after FL aggregation be* $\boldsymbol{w}^*$*. Then* $Var[\mathcal{M}(\boldsymbol{w}^*)] \preceq \frac{MT\alpha}{2n(\epsilon-\frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M$*, where* n *is the total number of clients,* M *is the number of components of model parameters,* T *is the total number of rounds,* $\preceq$ *represents the element-by-element comparison.*

Proof. Since Lemma 7 shows that the equation

$$Var[\mathcal{M}(\boldsymbol{w}_i^{(m)})] \leq \frac{ML_i\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}$$

holds for all m, combined with Theorem 2 we can know that

$$Var[\mathcal{M}(\boldsymbol{w}_i)] \preceq \frac{ML_i\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M, i = 1, \ldots, n.$$

Then for the sake of analysis, we assume that each client holds data set of the same size, and consider the following cases in turn:

- Only one client participates in a round, then the expectation of the number of times client uploads local model parameters is $\mathbb{E}[L_i] = \frac{T}{n}$. So it follows from line 13 in Algorithm 2 that the variance from the noise mechanism $\mathcal{M}$ introduced by our proposed scheme is

$$Var[\mathcal{M}(\boldsymbol{w}^*)] \preceq \frac{MT\alpha}{2n(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M.$$

- Two clients participate in a round, then the expectation of the number of times each client uploads parameters is $\mathbb{E}[L_i] = \binom{n-1}{1}T/\binom{n}{2}$. So it follows from line 13 in Algorithm 2 that the variance from the noise mechanism $\mathcal{M}$ introduced by our proposed scheme is

$$\begin{aligned}Var[\mathcal{M}(\boldsymbol{w}^*)] &\preceq \frac{n-1}{2}\frac{2MT\alpha}{2n(n-1)(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M \\ &= \frac{MT\alpha}{2n(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M.\end{aligned}$$

- Three clients participate in a round, then the expectation of the number of times each client uploads parameters is $\mathbb{E}[L_i] = \binom{n-1}{2}T/\binom{n}{3}$. So it follows from line 13 in Algorithm 2 that the variance from the noise mechanism $\mathcal{M}$ introduced by our proposed scheme is

$$\begin{aligned}Var[\mathcal{M}(\boldsymbol{w}^*)] &\preceq \frac{3T(n-1)(n-2)}{6n(n-1)(n-2)}\frac{M\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M \\ &= \frac{MT\alpha}{4n(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M \\ &\preceq \frac{MT\alpha}{2n(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M.\end{aligned}$$

$\vdots$

- The whole clients participate in a round, then the expectation of the number of times each client uploads parameters is $\mathbb{E}[L_i] = T$. So it follows from line 13 in Algorithm 2 that the variance from the noise mechanism $\mathcal{M}$ introduced by our proposed scheme is

$$Var[\mathcal{M}(\boldsymbol{w}^*)] \preceq \frac{1}{n}\frac{MT\alpha}{2(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M.$$

Taken together, the above analysis shows that, regardless of the number of clients involved in each round, we have $\text{Var}[\mathcal{M}(\boldsymbol{w}^*)] \preceq \frac{MT\alpha}{2n(\epsilon-\frac{\log(1/\delta)}{\alpha-1})}\mathbf{I}_M$. □

Theorem 3 (Convergence Analysis)**.** *Let global model parameters obtained using our proposed scheme be* $\widetilde{\boldsymbol{w}}^*$ *and global model parameters without noise be* $\boldsymbol{w}^*$*, then* $|\widetilde{\boldsymbol{w}}^* - \boldsymbol{w}^*| \prec \mathcal{O}\left(\frac{\sigma_0 M\sqrt{\log(1/\delta)}}{n\epsilon}\right)$*, where* n *is the number of clients,* M *is the number of components of model parameters,* $\prec$ *represents the element-by-element comparison.*

Proof. According to Theorem 2, Lemma 8 and 3-sigma rule [40], we have

$$Pr\left[|\mathcal{M}(\boldsymbol{w}^{*(m)}) - \boldsymbol{w}^{*(m)}| \leq \sqrt{3\frac{MT\alpha}{2n(\epsilon - \frac{\log(1/\delta)}{\alpha-1})}}\right] \to 1,$$

for $m = 1, \ldots, M$, $\mathcal{M}(\boldsymbol{w}^*)$ is global model parameters obtained using our proposed scheme $\widetilde{\boldsymbol{w}}^*$. Thus the equation

$$|\widetilde{\boldsymbol{w}}^* - \boldsymbol{w}^*| \preceq \sqrt{3\frac{MT\alpha(\alpha-1)}{2n(\epsilon(\alpha-1) - \log(1/\delta))}}$$

has a great probability of holding. Then according to Eq. (13), we can take α as $\alpha = \frac{T+2\epsilon\sigma_0^2}{2T}$ in the above equation.

Finally, we can derive that

$$|\widetilde{\boldsymbol{w}}^* - \boldsymbol{w}^*| \prec \mathcal{O}\left(\frac{\sigma_0 M\sqrt{\log(1/\delta)}}{n\epsilon}\right),$$

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

where $\mathcal{M}$ is the number of components of model parameters, n is the number of clients. □

F. Support for dropouts

In FL, It is common for clients to drop out during the FL training. In our proposed framework, when a client drops out, here we assume to be $\mathcal{C}_i$, the number of exposures of $\mathcal{C}_i$'s local parameters L_i does not increase, and his/her cumulative privacy loss does not grow. Thus the global privacy loss calculated by the Parameter Server is not smaller than the privacy loss of dropped clients. Parameter Server only collects local model parameters uploaded by participating clients in this round. And when a client joins in the middle, he/she just needs to initialize local model parameters to the current global model parameters before performing local training for the participation in next round and calculate the cumulative privacy loss by adding one to L_i. In summary, our proposed scheme does not require clients to participate in the whole FL aggregation process and thus is user-friendly for dropouts.

VI. Simulation Experiments and Analysis

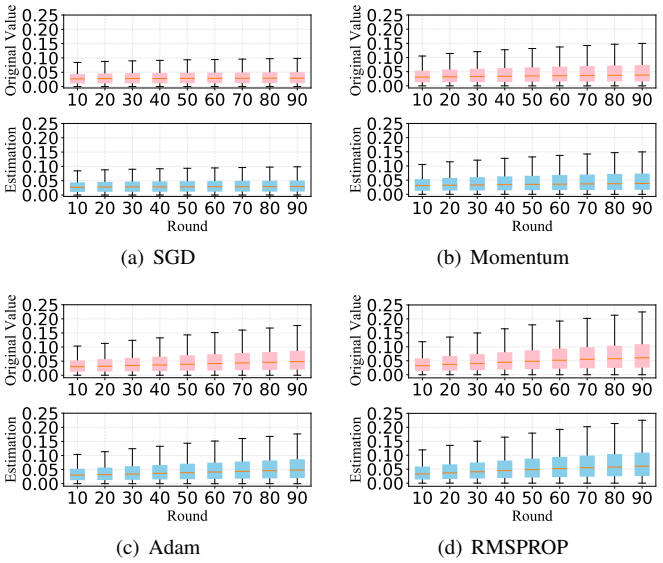
A. Experimental Settings

We evaluate our proposed scheme on three commonly-used datasets: MNIST, FMNIST and CIFAR-10. For MNIST and FMNIST, we use a two-layer convolutional neural network, which consists of two 2D convolutional layers and two fully connected layer, where we use ReLU function as activation function. For CIFAR-10, we use a network consisting of two consecutive two-layer convolutional layers and two max pooling layers, where we use ReLU function as activation function and softmax function as activation function of the last layer.

In FL, if not specifically pointed out, we set the number of clients $n = 50$, the number of participating clients in a round $|\mathcal{Z}_t| = n$, the truncation factor $\beta = 1.1$. For each optimizer in Table II, we set $\eta = 0.01$ for SGD, $\gamma = 0.9, \eta = 0.01$ for Momentum, $\gamma_1 = 0.9, \gamma_2 = 0.999, \epsilon = 10e-8, \eta = 0.001$ for Adam and $\gamma = 0.9, \epsilon = 10e-8, \eta = 0.001$ for RMSPROP, respectively. Moreover, we assume that the total size of datasets owned by all clients is fixed, which indicates that the more clients, the smaller dataset owned by each client. In the following experiments, we pre-determine the privacy budget ϵ and $\delta^* = 10^{-5}$. To eliminate the effect of noise randomness on resuts, we perform ten experiments for each configuration and take the average value as final result.

B. Performance Evaluation

It should be noted that the proposed scheme is based on the idea that components of model parameters with different values differ in their tolerance to noise, which in turn depends on sensitivity. Therefore, in order to evaluate the validity of our proposed scheme, it is first necessary to verify the accuracy and adaptiveness of our designed method for sensitivity estimates. Since our sensitivity is obtained by truncating client's local model parameters based on the computed adaptive threshold (Eq. (2) and Eq. (3)),



(a) SGD (b) Momentum

(c) Adam (d) RMSPROP

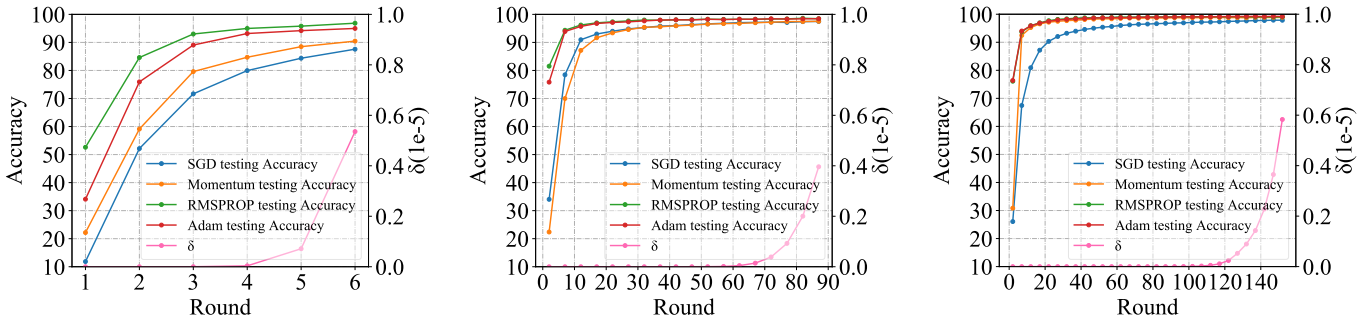
Fig. 2. Experiments to verify effectiveness of method for sensitivity estimates with different optimizers on MNIST dataset

we can verify the effectiveness of our proposed method by observing whether the truncated local model parameters can vary exactly with the raw local model parameters. We take four optimizers (SGD, Momentum, Adam, and RMSPROP), $\epsilon = 2.0$ (according to Eq. (2) and Eq. (3), the value of ϵ does not affect the estimation of sensitivity, so it is sufficient to experiment with only one value) and MNIST dataset. At every interval of ten rounds, we record raw local model parameters for a particular client (assumed to be $\mathcal{C}_i$) and truncated local model parameters calculated by Eq. (2). Then we plot their distributions as box line graphs, which are shown in Fig. 2, where "Original Value" means $\boldsymbol{w}_K^i$ (Line 17 in Algorithm 3) and "Estimation" means $\bar{f}_i$ (Line 20 in Algorithm 3). We observe that the truncated results by our proposed method can indeed accurately reflect the trend of original model parameters.

We then conduct experiments under different privacy budgets with four optimizers. We vary ϵ as 0.5, 2.0, and 8.0, set the number of local updates K to 16 and the noise level as $\sigma_0 = 26.0$ for $\epsilon = 0.5, 2.0$ and $\sigma_0 = 10.0$ for $\epsilon = 8.0$, respectively. Fig. 3(a)-3(c) show the model accuracy after each round. In each figure, the pink line (corresponding to the right vertical axis) represents the cumulative privacy budget, expressed in terms of δ given a fixed ϵ. Note that δ for the first few rounds is not 0 (rounds 1-4 in Fig. 3(a), rounds 1-60 in Fig. 3(b) rounds 1-100 in Fig. 3(c)), but the value is so small that it appears to be 0. We observe that when ϵ equals 0.5 (Fig. 3(a)), the model accuracy using four optimizers is 87.58% (SGD), 90.45% (Momentum), 94.98% (Adam) and 96.85% (RMSPROP), respectively. This indicates that our proposed scheme can achieve satisfactory model accuracy at a high privacy protection level (i.e. $\epsilon = 0.5$). When ϵ equals 2.0 (Fig. 3(b)), the model accuracy curves of four optimizers converge after 20^{th} round. When ϵ equals 8.0 (Figure 3(c)), the accuracy of obtained model is around 97%. These results demonstrate that our proposed adaptive noise mechanism is

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944



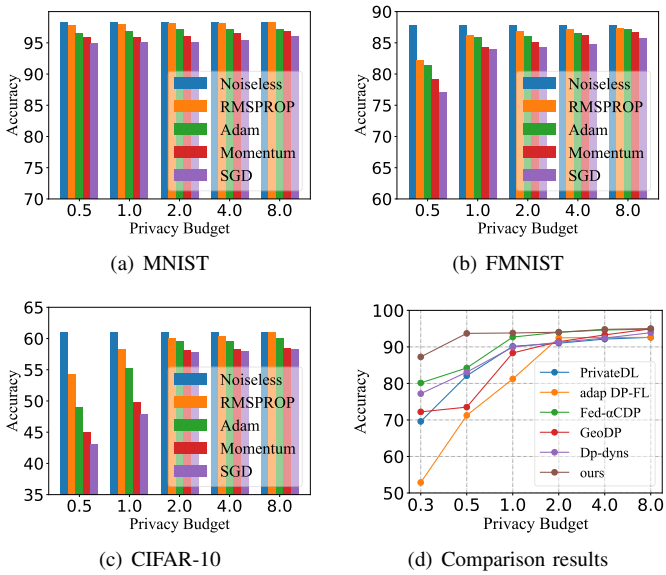
(a) Low privacy budget (b) Medium privacy budget (c) High privacy budget

Fig. 3. Experimental results for different optimizers under different privacy budgets on the MNIST dataset

indeed effective in mitigating the damage caused by noise added to the model.

We also compare our proposed scheme with the vanilla FL system without DP protection. In this experiment, we set K to 32 and choose MNIST, FMNIST and CIFAR-10 datasets. We vary the value of ϵ, and experimental results are shown in Fig. 4. For MNIST, when ϵ equals 0.5 (Fig. 4(a)), compared to the model without adding any noise, the accuracy loss of obtained model is 3.29% for SGD, 2.38% for Momentum, 1.71% for Adam and 0.39% for RMSPROP, respectively. For FMNIST, when ϵ equals 1.0 (Fig. 4(b)), the accuracy loss of model is 3.8% for SGD, 3.46% for Momentum, 1.97% for Adam and 1.67% for RMSPROP, respectively. For CIFAR-10, when ϵ equals 2.0 (Fig. 4(c)), the accuracy loss of model is 3.17% for SGD, 2.96% for Momentum, 1.51% for Adam and 1.03% for RMSPROP, respectively. These results demonstrate that our proposed scheme can achieve high model accuracy at high level of privacy protection (i.e., $\epsilon = 0.5$ for MNIST, $\epsilon = 1.0$ for FMNIST and $\epsilon = 2.0$ for CIFAR-10), which in turn indicates high utility of our proposed scheme in terms of making trade-off between privacy protection and model accuracy.

What is more, in order to demonstrate the superiority of optimization we have made for noise mechanism, we compare our proposed scheme with several state-of-the-art works that make optimization of noise mechanism in DP. We adapt the noise mechanism proposed in [41] to fit FL. Then we apply noise mechanism proposed in [25] (PrivateDL), [27] (adap DP-FL), [29] (Fed-αCDP), [30] (GeoDP), [41] (DP-dyns) and ours to the same FL scenario, respectively (i.e., parameters settings not related to DP, including the same optimizer, local dataset size, number of clients, etc.). We measure model accuracy in a given (ϵ, δ^*), and experimental results are shown in Fig. 4(d). As can be seen from the figure, when ϵ is equal to 0.3, the accuracy obtained with six methods is 52.87% (adap DP-FL), 69.59% (PrivateDL), 72.21% (GeoDP), 77.20% (DP-dyns), 80.14% (Fed-αCDP), 87.26% (ours), respectively. When ϵ is equal to 0.5, the accuracy is 71.25% (adap DP-FL), 73.51% (GeoDP), 82.11% (PrivateDL), 83.08% (DP-dyns), 84.23% (Fed-αCDP), 93.70% (ours), respectively. Experimental results confirm that our proposed scheme performs best on a very high level of privacy protection (i.e. $\epsilon = 0.3, 0.5$). Thus, our proposed scheme does mitigate the impact of noise on model



(a) MNIST (b) FMNIST

(c) CIFAR-10 (d) Comparison results

Fig. 4. Accuracy results for different privacy budgets

accuracy while satisfying high level of privacy protection.

In the following, we give model accuracy analysis in the variation of several parameters, which are used in our proposed scheme. It is worth noting that in order to briefly and clearly demonstrate differences in the effects of different parameters on model accuracy, we use MNIST dataset, set number of local updates K to 64 for SGD and K to 16 for the remaining three optimizers.

C. Impact of the Number of Clients n

To verify the scalability of our proposed scheme, i.e., whether it can support a large number of clients, we do experiments on the effect of the number of clients n on the model accuracy. As pointed out earlier, a larger number of participating clients means that each client has a smaller dataset. Thus, according to Line 6-8 in Algorithm 3, we know that $|b|$ decreases with increasing n, which leads to more noise to be added in order to meet DP requirements. We change the number of clients, and the evaluation results are shown in Fig. 5(a). We can see that model accuracy decreases very slightly as the number of clients increases. The model accuracy difference between 50 and 700 participating clients is only

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

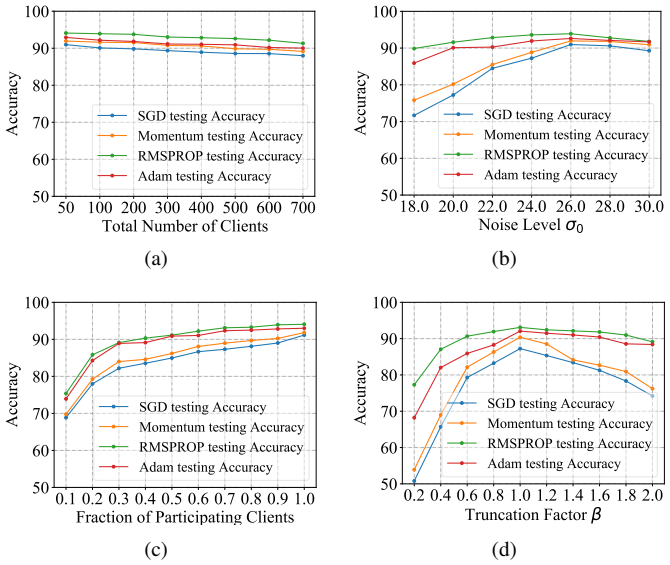


Fig. 5. Accuracy Analysis under different parameters on MNIST dataset

2.98% for SGD, 2.81% for Momentum, 2.90% for Adam, and 2.78% for RMSPROP. We conclude that even if the number of clients grows within a certain range, our proposed scheme can effectively mitigate the impact of noise on model accuracy.

D. Impact of the Noise Level σ_0

To verify that our proposed scheme is highly tolerant of the noise level σ_0, where σ_0 determines the magnitude of the added noise. The smaller σ_0 is, the less noise is added to model parameters (Line 21 in Algorithm 3), but the fewer rounds are allowed due to the privacy budget constraints (Line 1-2 in Algorithm 4). However, excessive noise will significantly damage model accuracy. Fig. 5(b) reports the model accuracy with different noise levels. The model accuracy with all four optimizers is highest at the noise level of 26. The model still performs well when σ_0 is equal to 30: the model accuracy drop is 1.7% for SGD, 1.04% for Momentum, 0.94% for Adam, and 2.09% for RMSPROP compared to that when $\sigma_0 = 26$. This indicates that the noise level tolerated by our proposed scheme is high, thus demonstrating the efficiency of ours.

E. Impact of the Number of Participating Clients in a Round

When the clients drop out during FL, the ratio of the number of clients participating in a round to the total number of clients $u = \frac{|\mathcal{Z}_t|}{n}$ is less than 1. So to verify that our scheme does effectively support clients drops, we investigate the impact of u on model performance. This parameter u affects not only the aggregation results (Line 13 in Algorithm 2), but also the number of rounds (Section V-D). When u is small, fewer training data samples are utilized in each round. But the exposure times of local parameters for each client grow slowly, and δ in Algorithm 2 also grows slowly correspondingly (Algorithm 4). Thus allowing for more rounds to improve model accuracy. Fig. 5(c) shows the impact of the proportion of participating clients in each round. We observe that model accuracy increases as the proportion of participating clients increases. When u is greater than 0.6, the model accuracy has reached a relatively satisfactory level (86.67% for SGD, 88.08% for Momentum, 91.05% for Adam, 92.21% for RMSPROP). Therefore, experimental results show that our scheme works well even when there are client dropouts.

F. Impact of the Truncation Factor β

Finally, let us explore what value of β equals when our proposed scheme works best, where β is defined as the truncation factor (Section V-B). According to the noise mechanism executed by client (Line 19-22 in Algorithm 3), it is obvious that β controls the truncation ratio of model parameters, and it also affects the magnitude of added noise. The smaller β is, the more model parameters are truncated, and thus the negative impact on model accuracy will be greater. The larger β is, the less part of model parameters is discarded, but added noise will be larger to meet the DP requirements (Line 19-21 in Algorithm 3). Fig. 5(d) shows the results with the variation of β. We observe that model accuracy achieves the highest when β is between 1.0 and 1.2.

VII. Conclusion

In this paper, we proposed a differentially private federated learning with an adaptive noise mechanism. Intuitively, adding more noise to components of model parameters with smaller values leads to a deviation in the direction of model update, but not vice versa. Therefore, we considered to introduce the noise that varies with values of model parameters. The implementation of this idea requires an accurate estimate of sensitivity, where sensitivity is a concept in DP that determines the magnitude of noise. However, the fact that datasets are not shared among clients in FL makes it difficult to precisely estimate sensitivity. To address this challenge, we proposed a method for clients to estimate sensitivity locally using global and local historical model parameters, which is proven to be effective by theoretical and experimental analysis. Then, based on the estimated sensitivity, we design a novel adaptive noise mechanism for clients that satisfies the DP requirements. Finally, We experimentally demonstrated that, compared with other existing work, our proposed scheme is more effective in mitigating the impact of DP on FL model performance.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grants No. 62372425, No. 61972371 and No. U19B2023, Key Laboratory of Medical Electronics and Digital Health of Zhejiang Province under Grant No. MEDH202201, and Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. Y202093.

References

[1] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proceedings of the 2019 IEEE Conference on Computer Communications (Infocom)*, 2019, pp. 2512–2520.

[4] 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 *Proceedings of the 2019 IEEE Symposium on Security and Privacy (S&P)*, 2019, pp. 739–753.

[5] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proceedings of the 2019 Annual Conference on Neural Information Processing Systems (NeurIPS)*, vol. 32. Curran Associates, Inc., 2019.

[6] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[7] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference (ATC)*. USENIX Association, 2020.

[8] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10 782–10 793, 2020.

[9] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, 2017, pp. 1175–1191.

[10] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure single-server aggregation with (Poly)logarithmic overhead," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2020, pp. 1253–1269.

[11] P. Xu, M. Hu, T. Chen, W. Wang, and H. Jin, "LaF: Lattice-based and communication-efficient federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2483–2496, 2022.

[12] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2016, pp. 308–318.

[14] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2018, arXiv preprint. [Online]. Available: https://arxiv.org/abs/1712.07557

[15] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," 2018, arXiv preprint. [Online]. Available: https://arxiv.org/abs/1710.06963

[16] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[17] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *Proceedings of the 35th IEEE International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 638–649.

[18] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[19] M. Seif, R. Tandon, and M. Li, "Wireless federated learning with local differential privacy," in *Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2604–2609.

[20] L. Sun, J. Qian, and X. Chen, "LDP-FL: Practical private aggregation in federated learning with local differential privacy," in *Proceedings of the 2021 International Joint Conference on Artificial Intelligence, IJCAI*. International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 1571–1578.

[21] B. Ghazi, R. Kumar, P. Manurangsi, R. Pagh, and A. Sinha, "Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message," in *Proceedings of the 2021 International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 3692–3701.

[22] A. Girgis, D. Data, and S. Diggavi, "Renyi differential privacy of the subsampled shuffle model in distributed learning," in *Proceedings of the 2021 Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021, pp. 29 181–29 192.

[23] C. Gratton, N. K. Venkategowda, R. Arablouei, and S. Werner, "Privacy-preserved distributed learning with zeroth-order optimization," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 265–279, 2021.

[24] T. Zhang, A. Song, X. Dong, Y. Shen, and J. Ma, "Privacy-preserving asynchronous grouped federated learning for IoT," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5511–5523, 2021.

[25] R. Han, D. Li, J. Ouyang, C. H. Liu, G. Wang, D. Wu, and L. Y. Chen, "Accurate differentially private deep learning on the edge," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 9, pp. 2231–2247, 2021.

[26] H. Zhou, G. Yang, H. Dai, and G. Liu, "PFLF: Privacy-preserving federated learning framework for edge computing," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1905–1918, 2022.

[27] J. Fu, Z. Chen, and X. Han, "Adap DP-FL: Differentially private federated learning with adaptive noise," 2022, arXiv preprint. [Online]. Available: https://arxiv.org/abs/2211.15893

[28] X. Zhang, X. Chen, M. Hong, Z. S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," in *Proceedings of the 2022 International Conference on Machine Learning (ICML)*, 2022.

[29] W. Wei, L. Liu, J. Zhou, K.-H. Chow, and Y. Wu, "Securing distributed SGD against gradient leakage threats," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 7, pp. 2040–2054, 2023.

[30] 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 Transactions on Information Forensics and Security*, vol. 18, pp. 1884–1897, 2023.

[31] T. Stevens, C. Skalka, C. Vincent, J. Ring, S. Clark, and J. Near, "Efficient differentially private secure aggregation for federated learning via hardness of learning with errors," in *Proceedings of the 31st USENIX Security Symposium (USENIX Security)*. Boston, MA: USENIX Association, 2022, pp. 1379–1395.

[32] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 54. PMLR, 2017, pp. 1273–1282.

[33] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Third Theory of Cryptography Conference (TCC)*. Springer, 2006, pp. 265–284.

[34] I. Mironov, "Rényi differential privacy," in *Proceedings of the 30th IEEE Computer Security Foundations Symposium (CSF)*, 2017, pp. 263–275.

[35] Z. Xu, S. Shi, A. X. Liu, J. Zhao, and L. Chen, "An adaptive and fast convergent approach to differentially private deep learning," in *Proceedings of the 2020 IEEE Conference on Computer Communications (Infocom)*, 2020, pp. 1867–1876.

[36] S. Ruder, "An overview of gradient descent optimization algorithms," 2017, arXiv preprint. [Online]. Available: https://arxiv.org/abs/1609.04747

[37] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, 2018.

[38] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, arXiv preprint. [Online]. Available: https://arxiv.org/abs/1412.6980

[40] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[41] W. Wei and L. Liu, "Gradient leakage attack resilient deep learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 303–316, 2021.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3318944

Rui Xue received her bachelor's degree in mathematics and applied mathematics from the School of Mathematical Science, University of Science and Technology of China (USTC) in 2021. She is currently a graduated student in Information Security from the School of Cyber Science and Technology, USTC. Her research interests include privacy-preserving machine learning and applied cryptography.

Kaiping Xue (Senior Member, IEEE) received his bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his doctor's degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a postdoctoral researcher with the Department of Electrical and Computer Engineering, University of Florida. Currently, he is a Professor in the School of Cyber Science and Technology, USTC. He is also a director of Network and Information Center, USTC. His research interests include next-generation Internet architecture design, transmission optimization and network security. His work won best paper awards in IEEE MSN 2017 and IEEE HotICN 2019, the Best Paper Honorable Mention in ACM CCS 2022, the Best Paper Runner-Up Award in IEEE MASS 2018, and the best track paper in MSN 2020. He serves on the Editorial Board of several journals, including the IEEE Transactions on Dependable and Secure Computing (TDSC), the IEEE Transactions on Wireless Communications (TWC), and the IEEE Transactions on Network and Service Management (TNSM). He has also served as a (Lead) Guest Editor for many reputed journals/magazines, including IEEE Journal on Selected Areas in Communications (JSAC), IEEE Communications Magazine, and IEEE Network. He is an IET Fellow and an IEEE Senior Member.

Bin Zhu (Graduate Student Member, IEEE) received his bachelor's degree in Information Security from the School of Cyber Science and Technology, University of Science and Technology of China (USTC) in 2019. He is currently working toward the Ph.D degree from the School of Cyber Science and Technology, USTC. His research interests include Network security and applied cryptography.

Xinyi Luo (Graduate Student Member, IEEE) received her bachelor's degree in Information Security from School of the Gifted Young, University of Science and Technology of China (USTC) in July, 2020. She is currently working toward the Ph.D degree from the School of Cyber Science and Technology, USTC. Her research interests include blockchain technology, network security and applied cryptography.

Tianwei Zhang (Member, IEEE) received the bachelor's degree from Peking University in 2011 and the Ph.D. degree from Princeton University in 2017. He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University. his research focuses on computer system security. He is particularly interested in security threats and defenses in machine learning systems, autonomous systems, computer architecture and distributed systems.

Qibin Sun (Fellow, IEEE) received the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1997. He is currently a professor in the School of Cyber Science and Technology, USTC. His research interests include multimedia security, network intelligence and security, and so on. He has published more than 120 papers in international journals and conferences. He is a fellow of IEEE.

Jun Lu received his bachelor's degree from southeast university in 1985 and his master's degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1988. Currently, he is a professor in the School of Cyber Science and Technology and the Department of EEIS, USTC. He is also the president of Jiaxing University. His research interests include theoretical research and system development in the field of integrated electronic information systems, network and information security. He is an Academician of the Chinese Academy of Engineering (CAE).

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on October 06,2023 at 19:40:43 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.