IEEE Open Journal of the Computer Society

Received 11 June 2021; accepted 17 July 2021. Date of publication 26 July 2021;
date of current version 24 August 2021. The review of this paper was arranged by Guest Editors Sahil Garg, Alagan Anpalagan, Shree Krishna Sharma, Li-Chun Wang, Muhammad Zeeshan Shakir, Ning Zhang, and Linglong Dai.

Digital Object Identifier 10.1109/OJCS.2021.3099108

Concentrated Differentially Private Federated Learning With Performance Analysis

RUI HU [1] (Student Member, IEEE), YUANXIONG GUO [2] (Senior Member, IEEE), AND YANMIN GONG [1] (Senior Member, IEEE)

[1] Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249 USA
[2] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA

CORRESPONDING AUTHOR: Yanmin Gong (e-mail: gongyanmin@gmail.com)

The work of Rui Hu and Yanmin Gong were supported by the U.S. National Science Foundation under Grants CNS-2029685, CNS-1850523, and CNS-2106761. The work of Yuanxiong Guo was supported by the U.S. National Science Foundation under Grants CNS-2029685 and CNS-2106761.

ABSTRACT Federated learning engages a set of edge devices to collaboratively train a common model without sharing their local data and has advantage in user privacy over traditional cloud-based learning approaches. However, recent model inversion attacks and membership inference attacks have demonstrated that shared model updates during the interactive training process could still leak sensitive user information. Thus, it is desirable to provide rigorous differential privacy (DP) guarantee in federated learning. The main challenge to providing DP is to maintain high utility of federated learning model with repeatedly introduced randomness of DP mechanisms, especially when the server is not fully trusted. In this paper, we investigate how to provide DP to the most widely adopted federated learning scheme, federated averaging. Our approach combines local gradient perturbation, secure aggregation, and zero-concentrated differential privacy (zCDP) for better utility and privacy protection without a trusted server. We jointly consider the performance impacts of randomnesses introduced by the DP mechanism, client sampling and data subsampling in our approach, and theoretically analyze the convergence rate and end-to-end DP guarantee with non-convex loss functions. We also demonstrate that our proposed method has good utility-privacy trade-off through extensive numerical experiments on the real-world dataset.

INDEX TERMS Federated learning, security and privacy, convergence analysis, zero-concentrated differential privacy.

I. INTRODUCTION

With the development of Internet-of-Things (IoT) technologies, smart devices with built-in sensors, Internet connectivity, and programmable computation capability have proliferated and generated huge volumes of data at the network edge over the past few years. These data can be collected and analyzed to build machine learning models that enable a wide range of intelligent services, such as personal fitness tracking [1], traffic monitoring [2], smart home security [3], and renewable energy integration [4]. However, data are often sensitive in many services and can leak a lot of personal information about the users. Due to the privacy concern, users could be reluctant to share their data, prohibiting the deployment of these intelligent services.

Federated Learning is a novel machine learning paradigm where a group of edge devices collaboratively learn a shared model under the orchestration of a central server without sharing their local data. It mitigates many of the privacy risks resulting from the traditional, centralized machine learning paradigm, and has received significant attention recently [5]. At each communication round of federated learning, edge devices download the shared model from the server and compute updates to it using their own datasets, and then these updates will be gathered by the server to update the shared model. Although only model updates are transmitted between edge devices and the server instead of the raw data, such updates could contain hundreds of millions of parameters in modern

machine learning models such as deep neural networks, resulting in high bandwidth usage per round. Moreover, many learning tasks require a large number of communication rounds to achieve a high model utility, and hence the communication of the whole training process is expensive. Since most edge devices are resource-constrained, the bandwidth between the server and edge devices is rather limited, especially in uplink transmissions. Therefore, in the state-of-the-art federated learning algorithms, each edge device would perform multiple local iterations in each round to obtain a more accurate model update, so that the total number of communication rounds to achieve a desired model utility will be reduced.

Besides communication overhead, federated learning faces several additional challenges, among which privacy leakage is a major one [5]. Although in federated learning edge devices keep their data locally and only exchange ephemeral model updates which contain less information than raw data, this is not sufficient to guarantee data privacy. For example, by observing the model updates from an edge device, it is possible for the adversary to recover the private dataset in that device using reconstruction attack [6] or infer whether a sample is in the dataset of that device using membership inference attack [7]. Especially, if the server is not fully trusted, it can easily infer the private information of edge devices from the received model updates during the training by employing existing attack methods. Therefore, how to protect against those advanced privacy attacks and provide rigorous privacy guarantee for each device in federated learning without a fully trusted server is challenging and needs to be addressed.

In order to motivate and retain edge devices in federated learning, it is desirable to provide rigorous differential privacy (DP) guarantee for devices. While there have been multiple works focusing on the integration of DP and federated learning [8]–[17], most of the work demonstrate the performance of proposed approaches by experiments, whose results heavily rely on hyper-parameter tuning. The main focus of this paper is to *provide a differentially-private federated learning approach with convergence performance bound*. The closest work to ours is [18], which also provide performance analysis for federated learning with record-level DP. However, they do not discuss client sampling in their analysis, which is a core design factor of federated learning for communication-efficiency and scalability [19]. In addition, their performance analysis is based on convexity assumption of the loss function, which is not true for many federated learning tasks. Our performance analysis is more general and fits both convex and non-convex loss functions.

In this paper, we aim to bridge the gap by providing performance analysis and DP guarantee for the state-of-the-art federated averaging scheme with client sampling [19]. In order to protect the shared model updates, we ask each device to perturb its gradient in each local iteration so that the shared model updates are differentially private before aggregation. When combining with periodic averaging and client sampling directly, gradient perturbation results in too much noise to the model updates and leads to low model utility. Thus we also integrate a secure aggregation protocol with low communication overhead to reduce the added noise magnitude. Furthermore, we utilize the zero-concentrated differential privacy (zCDP) to tightly capture the end-to-end privacy loss, so that less noise will be added under the same DP guarantee. Our performance analysis works for both convex and non-convex loss functions and thus is more general than prior work. The proposed differentially private federated learning scheme only assumes an "honest-but-curious" server, which is a more practical assumption than a fully trusted server.

In summary, the main contributions of this paper are as follows.

- We propose a differentially-private federated learning scheme with periodic averaging and device sampling without a fully trusted server. Our approach can rigorously protect the data privacy of each device with only marginal degradation of the model utility by integrating secure aggregation and gradient perturbation.
- We tightly compute the end-to-end privacy loss of our approach using zCDP, taking into account the privacy amplification effects of data subsampling, partial device participation and secure aggregation. Compared with using traditional (ϵ, δ)-DP and its simple composition property to count the privacy loss, our approach enables devices to add less noise and hence improves the model utility under the same privacy guarantee.
- We rigorously analyze the convergence rate of state-of-the-art federated averaging schemes with differentially-private noises, client sampling and non-convex loss functions. Our approach obtains the same asymptotic convergence rate as the classic non-private federated learning algorithm.
- We conduct extensive evaluations on a real-world dataset and the experimental results show that our approach has nice convergence property and good utility-privacy trade-off.

The rest of the paper is organized as follows. Preliminaries on privacy notations used in this paper are described in Section II. Section III introduces the system setting and problem formulation, and Section IV presents our private federated learning scheme. The privacy guarantee and convergence properties of our approach are rigorously analyzed in Section V and Section VI, respectively. Section VII shows the evaluation results based on the real-world dataset. Finally, Section VIII describes the related works, and Section IX concludes the paper.

II. PRELIMINARIES

In what follows, we briefly describe the basics of DP and their properties. DP is a rigorous notion of privacy and has become the de-facto standard for measuring privacy risk. (ϵ, δ)-DP [20] is the classic DP notion with the following definition:

Definition 1 ((ϵ, δ)-DP): A randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private if for any two adjacent datasets $D, D' \subseteq \mathcal{D}$ that have the same size but differ in at most one data sample and any subset of outputs $\mathcal{S} \subseteq range(\mathcal{M})$, it

satisfies that:

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta. \quad (1)$$

The above definition reduces to ϵ-DP when $\delta = 0$. Here the parameter ϵ is also called the privacy budget. Given any function f that maps a dataset $D \in \mathcal{D}$ into a scalar $o \in \mathbb{R}$, we can achieve (ϵ, δ)-DP by adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the output scalar o, where the noise magnitude σ is proportional to the sensitivity of f, given as $\Delta_2(f) := \|f(D) - f(D')\|_2$.

The notion ρ-zCDP [21] is a relaxed version of (ϵ, δ)-DP. zCDP has a tight composition bound and is more suitable to analyze the end-to-end privacy loss of iterative algorithms. To define zCDP, we first define the privacy loss. Given any subset of outputs $\mathcal{S} \in range(\mathcal{M})$, the privacy loss Z of the mechanism $\mathcal{M}$ is a random variable defined as:

$$Z := \log \frac{\Pr[\mathcal{M}(D) = \mathcal{S}]}{\Pr[\mathcal{M}(D') = \mathcal{S}]}. \quad (2)$$

zCDP imposes a bound on the moment generating function of the privacy loss Z. Formally, a randomized mechanism $\mathcal{M}$ satisfies ρ-zCDP if for any two adjacent datasets $D, D' \subseteq \mathcal{D}$, it holds that for all $\alpha \in (1, \infty)$,

$$\mathbb{E}[e^{(\alpha-1)Z}] \leq e^{(\alpha-1)\rho}. \quad (3)$$

Here, (3) requires the privacy loss Z to be concentrated around zero, and hence it is unlikely to distinguish D from D' given their outputs. zCDP has the following properties [21], [22]:

Lemma 1: Let $f : \mathcal{D} \to \mathbb{R}$ be any real-valued function with sensitivity $\Delta_2(f)$, then the Gaussian mechanism, which returns $f(D) + \mathcal{N}(0, \sigma^2)$, satisfies $\Delta_2(f)^2/(2\sigma^2)$-zCDP.

Lemma 2: Suppose two mechanisms satisfy ρ_1-zCDP and ρ_2-zCDP, then their composition satisfies $\rho_1 + \rho_2$-zCDP.

Lemma 3: Suppose that a mechanism $\mathcal{M}$ consists of a sequence of E adaptive mechanisms, $\mathcal{M}_1, \ldots, \mathcal{M}_E$, where each $\mathcal{M}_j$ satisfies ρ_j-zCDP $(1 \leq j \leq E)$. Let $D_1, \ldots, D_E$ be the result of a randomized partitioning of the dataset D. The mechanism $\mathcal{M}(D) = (\mathcal{M}_1(D_1), \ldots, \mathcal{M}_E(D_E))$ satisfies $\max_j \rho_j$-zCDP.

After we use zCDP to quantify the total privacy loss of the iterative algorithms, we could easily convert the privacy loss in zCDP back to DP with the following lemma [21]:

Lemma 4: If $\mathcal{M}$ is a mechanism that provides ρ-zCDP, then $\mathcal{M}$ is $(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta)$-DP for any $\delta > 0$.

III. SYSTEM MODELING AND PROBLEM FORMULATION

A. FEDERATED LEARNING SYSTEM

Consider a federated learning setting that consists of a central server and n devices which are able to communicate with the server. Each device $i \in [n]$ has collected a local dataset $D_i = \{\xi_1^i, \ldots, \xi_m^i\}$ of m datapoints. The devices want to collaboratively learn a shared model $\boldsymbol{\theta} \in \mathbb{R}^d$ under the orchestration of the central server. Due to the privacy concern and high latency of uploading all local datapoints to the server, federated learning allows devices to train the model while keeping their data locally. Specifically, the shared model $\boldsymbol{\theta}$ is learned by minimizing the overall empirical risk on the union of all local datasets, that is,

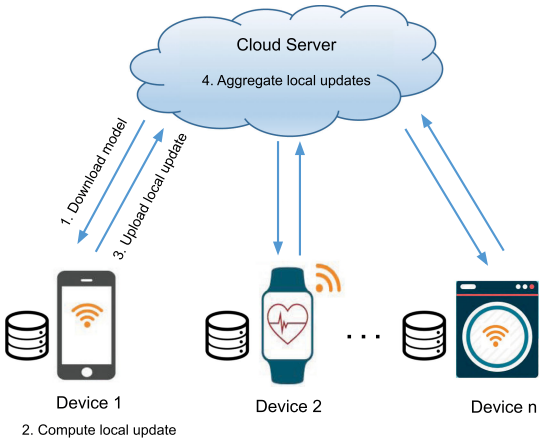

FIG. 1. System architecture of federated learning.

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}) \text{ with } f_i(\boldsymbol{\theta}) := \frac{1}{m} \sum_{\xi_i \in D_i} l(\boldsymbol{\theta}, \xi_i). \quad (4)$$

Here, f_i represents the local objective function of device i, $l(\boldsymbol{\theta}; \xi_i)$ is the loss of the model $\boldsymbol{\theta}$ at a datapoint ξ_i sampled from local dataset D_i.

In federated learning, the central server is responsible for coordinating the training process across all devices and maintaining the shared model $\boldsymbol{\theta}$. The system architecture of federated learning is shown in Fig. 1. At the beginning of each iteration, devices download the shared model $\boldsymbol{\theta}$ from the server and compute local updates on $\boldsymbol{\theta}$ using their local datasets. Then, each device uploads its computed result to the server, where the received local results are aggregated to update the shared model $\boldsymbol{\theta}$. This procedure repeats until certain convergence criteria are satisfied.

The classic approach to solve Problem (4) is the federated averaging (FedAvg) algorithm [19]. In FedAvg, the server first selects a subset of devices uniformly at random and then lets the selected devices perform multiple iterations of SGD to minimize the local objectives before sending their local computation results to the server. Let τ represent the local iteration period and $t \in [0, \ldots, T-1]$ represent the index of communication round. Specifically, at round t, a set of r devices Ω_t are selected to download the current shared model $\boldsymbol{\theta}^t$ from the server and perform τ local iterations on $\boldsymbol{\theta}^t$. Let $\boldsymbol{\theta}_i^{t,s}$ denote the local model of device $i \in \Omega_t$ at s-th local iteration of the t-th round. At each local iteration $s = 0, \ldots, \tau - 1$, device i updates its model by

$$\boldsymbol{\theta}_i^{t,s+1} = \boldsymbol{\theta}_i^{t,s} - \eta g(\boldsymbol{\theta}_i^{t,s}), \quad (5)$$

where η is the learning rate, $g(\boldsymbol{\theta}_i^{t,s}) := (1/\gamma) \sum_{\xi_i \in X_i} \nabla l(\boldsymbol{\theta}_i^{t,s}, \xi_i)$ represents the stochastic gradient computed based on a mini-batch X_i of γ datapoints sampled from the local dataset D_i. Note that when $s = 0$, the local model $\boldsymbol{\theta}_i^{t,s} = \boldsymbol{\theta}^t$ for all devices in Ω_t. After τ local iterations, the selected devices upload their local

models to the server where the shared model is updated by $\boldsymbol{\theta}^{t+1} = (1/r)\sum_{i\in\Omega_t} \boldsymbol{\theta}_i^{t,\tau}$. Therefore, each device is selected to participate with probability r/n in each round and only needs to periodically communicate for rT/n rounds in expectation.

B. THREAT MODEL

The adversary considered here can be the "honest-but-curious" central server or devices in the system. The central server and devices participating in federated learning will honestly follow the designed training protocol, and will not actively inject false messages into the training process. However, they are curious about a target device's private data and may infer it from the shared information during the training process. The adversary can also be a passive outsider attacker who eavesdrops all shared messages in the execution of the training protocol. In federated learning, raw user data are kept locally, which provides some level of privacy protection. However, local model updates are shared in each iteration, which are being trained over private user data. Access to shared model updates allows adversaries to launch model inversion attacks to reconstruct the raw training data [6], [23], or use membership inference attacks [7] to infer if a data record was in the raw training database. Thus, keeping data locally and only sharing model updates do not provide enough protection for user privacy, and we need to provide rigorous privacy guarantee to defend against the aforementioned attacks.

IV. OUR PRIVATE FEDERATED LEARNING SCHEME

In this section, we propose our method that enables multiple devices to jointly learn an accurate model for a given machine learning task in a private manner, without sacrificing much accuracy of the trained model. We first discuss how to preserve the data privacy of each device in the system with DP techniques. Then, we improve the accuracy of our method with secure aggregation and finally summarize the overall algorithm.

A. PREVENTING PRIVACY LEAKAGE WITH DIFFERENTIAL PRIVACY

The aforementioned FedAvg method is able to prevent the direct information leakage of devices via keeping the raw data locally, however, it could not prevent more advanced attacks that infer private information of local training data by observing the messages communicated between devices and the server [6], [7]. According to our threat model described in Section III-B, devices and the server in the system are "honest-but-curious," and attackers outside the system can eavesdrop the transmitted messages. These attackers are able to obtain the latest shared model $\boldsymbol{\theta}^t$ sent from the server to devices and the local models $\{\boldsymbol{\theta}_i^{t,\tau}\}_{i\in\Omega_t}$ sent from devices to the server, both of which contain the private information of devices' training data. Our goal is to prevent the privacy leakage from these two types of messages with DP techniques.

Towards that goal, we leverage the gradient perturbation with Gaussian noise [24] to achieve DP so that the attacker is not able to learn much about an individual sample in D_i from the shared massages. Specifically, at s-th local iteration of t-th round, device $i \in \Omega_t$ updates its local model by

$$\boldsymbol{\theta}_i^{t,s+1} = \boldsymbol{\theta}_i^{t,s} - \eta\left(g(\boldsymbol{\theta}_i^{t,s}) + \mathbf{b}_i^{t,s}\right), \tag{6}$$

where $\mathbf{b}_i^{t,s}$ is the Gaussian noise sampled from the distribution $\mathcal{N}(0, \sigma^2\mathbf{I}_d)$. After τ local iterations, the uploaded local model $\boldsymbol{\theta}_i^{t,\tau}$ will preserve a certain level of DP guarantee for device i, which is proportional to the size of noise σ. Due to the post-processing property of DP [20], the sum of local models, i.e., the updated shared model $\boldsymbol{\theta}^{t+1}$, will also preserve the same level of DP guarantee for device i.

B. IMPROVING MODEL UTILITY WITH SECURE AGGREGATION

Although DP can be achieved using the above mechanism, the accuracy of the learned model may be low due to the large noise magnitude. At each round of the training, all uploaded local models are exposed to the attacker, leading to a large amount of information leakage. However, we observe that the server only needs to know the average values of the local models. Intuitively, one can reduce the privacy loss of devices by hiding the individual local models and restricting the server to receive only the sum of local models without disturbing the learning process. This can be achieved via a secure aggregation protocol so that the server can only decrypt the sum of the encrypted local models of selected devices without knowing each device's local model. In the following, we provide a customized secure aggregation protocol similar as [25], which is efficient in terms of the amortized computation and communication overhead across all communication rounds. Note that one of the main contributions in this paper is to analyze the benefits of secure aggregation in reducing privacy loss as elaborated in Section V rather than optimizing the design of the secure aggregation protocol.

In our setting, a secure aggregation protocol should be able to 1) hide individual messages for devices, 2) recover the sum of individual messages of a random set of devices at each round, and 3) incur low communication cost for participating devices. Denote by p_i^t the plaintext message of device i that needs to be uploaded to the server. Note that the secure aggregation protocol only works for integers, hence the local model parameter $\boldsymbol{\theta}_i^{t,\tau}$ should be converted optimally to integers to obtain the plaintext p_i^t [25]. Our proposed protocol consists of the following two main steps:

- *Encryption uploading:* Devices in Ω_t upload their own encrypted local models $\{c_i^t\}_{i\in\Omega_t}$ to the server.
- *Decryption:* The server decrypts the sum of the messages received from devices in Ω_t.

The basic idea of the protocol is to protect the message p_i^t of device i by hiding it with a random number r_i^t in the plaintext space, i.e., $c_i^t = p_i^t + r_i^t$. However, the challenge here is how to remove the random number r_i^t from the received ciphertext at the server part. To this end, we require that $\sum_{i\in\Omega_t} r_i^t = 0$, which prevents the attacker from recovering each individual

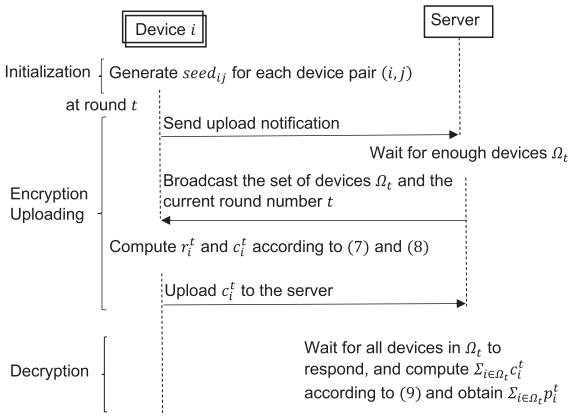

FIG. 2. **Basic protocol for efficient secure aggregation in our approach.**

message p_i^t but enables the server to recover $\sum_{i\in\Omega_t} p_i^t$. However, this requires the devices to communicate with each other in order to generate such secrets $\{r_i^t\}_{i\in\Omega_t}$, which is inefficient in terms of communication overhead.

To save the communication overhead, we introduce a pseudorandom function (PRF) G here. The PRF G takes a random seed $seed_{i,j}$ that both device i and j agree on during initialization and the round number t, and outputs a different pseudorandom number $G(seed_{i,j}, t)$ at each round. Device i could calculate the shared secret r_{ij}^t without interacting with device j at each round as long as they both use the same seed and round number, and thus each device could calculate r_i^t without interactions. This procedure greatly reduces the amortized communication overhead of our protocol over multiple rounds.

The detailed protocol is depicted in Fig. 2. All devices need to go through an initialization step upon enrollment which involves pairwise communications with all other devices (which can be facilitated by the server) to generate a random seed $seed_{ij}$. After this initialization step, all enrolled devices could upload their messages through the encryption uploading step. At each round, only a subset of selected devices would upload their messages. Devices send a notification signal to the server once they are ready to upload their local models, and the server waits until receiving notifications from enough devices. The server then broadcasts the information Ω_t to all devices in Ω_t. Device $i \in \Omega_t$ would first compute its secret at the current round as follows:

$$r_i^t = \sum_{j\in\Omega_t\setminus\{i\}} \left(r_{ij}^t - r_{ji}^t\right), \tag{7}$$

where $r_{ij}^t = G(seed_{i,j}, t)$ is a secret known by both device i and j. Device i could then generate the ciphertext for p_i^t by

$$c_i^t = p_i^t + r_i^t. \tag{8}$$

In the decryption step, the server receives $\{c_i^t\}_{i\in\Omega_t}$ from all selected devices. The server could then recover the sum of

Algorithm 1: Private Federated Learning Algorithm.

Input: number of rounds T, local iteration period τ, number of selected devices per round r, learning rate η

for $t = 0$ to $T-1$ **do**
  Server uniformly selects a set Ω_t of r devices
  Server broadcasts $\boldsymbol{\theta}^t$ to all devices in Ω_t
  for all devices in Ω_t in parallel **do**
    $\boldsymbol{\theta}_i^{t,0} \leftarrow \boldsymbol{\theta}^t$
    for $s = 0$ to $\tau - 1$ **do**
      Sample a mini-batch X_i and compute gradient $g(\boldsymbol{\theta}_i^{t,s}) \leftarrow (1/\gamma)\sum_{\xi_i\in X_i} \nabla l(\boldsymbol{\theta}_i^{t,s}, \xi_i)$
      Sample DP noise $\boldsymbol{b}_i^{t,s} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$
      $\boldsymbol{\theta}_i^{t,s+1} \leftarrow \boldsymbol{\theta}_i^{t,s} - \eta(g(\boldsymbol{\theta}_i^{t,s}) + \boldsymbol{b}_i^{t,s})$
    end for
    Generate encrypted local model c_i^t using the secure aggregation protocol and send it to the server
  end for
  Server decrypts the average of the received local models $(1/r)\sum_{i\in\Omega_t} c_i^t$ to get the new global model $\boldsymbol{\theta}^{t+1}$
end for

plaintext messages from devices in Ω_t as follows:

$$\begin{aligned}\sum_{i\in\Omega_t} c_i^t &= \sum_{i\in\Omega_t} p_i^t + \sum_{i\in\Omega_t}\sum_{j\in\Omega_t\setminus\{i\}} (r_{ij}^t - r_{ji}^t) \\ &= \sum_{i\in\Omega_t} p_i^t.\end{aligned} \tag{9}$$

Note that in the above protocol, we assume all devices in Ω_t have stable connections to the server. The entire process of our approach, integrating gradient perturbation, secure aggregation, and multiple steps of local SGD, is summarized in Algorithm 1.

V. PRIVACY ANALYSIS

As mentioned before, our goal of using DP techniques is to prevent the outside attacker or the "honest-but-curious" server and devices from learning sensitive information about the local data of a device. Using the secure aggregation protocol, the local model is encrypted and the attacker can only obtain the sum of local models. Thus, as long as the sum of local models is differentially private, we can prevent the attacks launched by the attacker.

Instead of using the traditional (ϵ, δ)-DP notion, we use zCDP to tightly account the end-to-end privacy loss of our approach across multiple iterations and then convert it to an (ϵ, δ)-DP guarantee. In the following, we first compute the sensitivity of the gradient $g(\boldsymbol{\theta}_i^{t,s})$ (as given in Corollary 1) based on Assumption 1 to show that each iteration of Algorithm 1 achieves zCDP. Note that Assumption 1 is a common assumption for differentially private learning algorithms and can be achieved by gradient clipping techniques [24]. Then,

we compute the sensitivity of the uploaded local model $\boldsymbol{\theta}_i^{t,\tau}$ (as given in Lemma 5) to further capture the zCDP guarantee of each communication round. Finally, we show that Algorithm 1 satisfies (ϵ_i, δ)-DP for device i after T communication rounds in Theorem 1.

Assumption 1 (Bounded gradients): The L_2-norm of the stochastic gradient $\nabla l(\mathbf{x}, \xi)$ is bounded, i.e., for any $\mathbf{x} \in \mathbb{R}^d$ and $\xi \in \bigcup_{i\in[n]} D_i$, $\|\nabla l(\mathbf{x}, \xi)\|_2 \leq G$.

Corollary 1: The sensitivity of the stochastic gradient $g(\boldsymbol{\theta}_i^{t,s})$ of device i at each local iteration is bounded by $2G/\gamma$.

Proof: For device i, given any two neighboring datasets X_i and X_i' of size γ that differ only in the j-th data sample, the sensitivity of the stochastic gradient computed at each local iteration in Algorithm 1 can be computed as

$$\begin{aligned}&\|g(\boldsymbol{\theta}_i^{t,s}; X_i) - g(\boldsymbol{\theta}_i^{t,s}; X_i')\|_2\\ &= \frac{1}{\gamma}\|\nabla l(\boldsymbol{\theta}_i^{t,s}; \xi_j) - \nabla l(\boldsymbol{\theta}_i^{t,s}; \xi_j')\|_2.\end{aligned}$$

By Assumption 1, the sensitivity of $g(\boldsymbol{\theta}_i^{t,s})$ can be estimated as $\Delta_2(g(\boldsymbol{\theta}_i^{t,s})) \leq 2G/\gamma$. ■

By Lemma 1 and Corollary 1, each iteration of Algorithm 1 achieves $2G^2/\gamma^2\sigma^2$-zCDP for every active device at this iteration. During the local computation, the local dataset will be randomly shuffled and partitioned into m/γ mini-batches, each containing γ datapoints. Assume τ is divided evenly by m/γ, then the whole local dataset will be accessed for $\tau\gamma/m$ times at each round. Therefore, at round t, the uploaded local model $\boldsymbol{\theta}_i^{t,\tau}$ satisfies $2\tau G^2/m\gamma\sigma^2$-zCDP by using Lemma 3 and Lemma 2.

Given the zCDP guarantee of $\boldsymbol{\theta}_i^{t,\tau}$ and its sensitivity given in Lemma 5, we can observe that the variance of Gaussian noise added to $\boldsymbol{\theta}_i^{t,\tau}$ is equivalent to $m\tau\eta^2\sigma^2/\gamma$. Therefore, we can obtain that the variance of Gaussian noise added to the sum of uploaded local models is $rm\tau\eta^2\sigma^2/\gamma$, due to the independence of Gaussian noise. By Lemma 1, we can obtain the zCDP guarantee of the sum of uploaded local models if we can measure the sensitivity of $\sum_{i\in\Omega_t}\boldsymbol{\theta}_i^{t,\tau}$. It is easy to verify that, for device $i \in \Omega_t$, the sensitivity of the sum of uploaded local models $\sum_{i\in\Omega_t}\boldsymbol{\theta}_i^{t,\tau}$ is equivalent to the sensitivity of $\boldsymbol{\theta}_i^{t,\tau}$. Finally, we obtain that $\sum_{i\in\Omega_t}\boldsymbol{\theta}_i^{t,\tau}$ satisfies $2\tau G^2/rm\gamma\sigma^2$-zCDP, which means round t of Algorithm 1 achieves $2\tau G^2/rm\gamma\sigma^2$-zCDP for each device in Ω_t. Finally, we compute the overall privacy guarantee for a device after T communication rounds and give the (ϵ, δ)-DP guarantee in Theorem 1.

Lemma 5: The sensitivity of the uploaded local model $\boldsymbol{\theta}_i^{t,\tau}$ at round t is bounded by $2\eta\tau G/\gamma$.

Proof: Without adding noise, the local model of device $i \in \Omega_t$ after τ local iterations at round t can be written as

$$\boldsymbol{\theta}_i^{t,\tau} = \boldsymbol{\theta}_i^{t,0} - \eta g(\boldsymbol{\theta}_i^{t,0}) - \cdots - \eta g(\boldsymbol{\theta}_i^{t,\tau-1}).$$

According to the sensitivity of $g(\boldsymbol{\theta}_i^{t,s})$ given in Corollary 1, we have that

$$\begin{aligned}\Delta_2(\boldsymbol{\theta}_i^{t,\tau}) &= \eta\Big\|g(\boldsymbol{\theta}_i^{t,0}; X_i^{t,0}) - g(\boldsymbol{\theta}_i^{t,0}; X_i^{t,0'}) + \ldots\\ &\quad + g(\boldsymbol{\theta}_i^{t,\tau-1}; X_i^{t,\tau-1}) - g(\boldsymbol{\theta}_i^{t,\tau-1}; X_i^{t,\tau-1'})\Big\|\\ &\leq \frac{2\eta\tau G}{\gamma}.\end{aligned}$$

■

Theorem 1: In Algorithm 1, let the mini-batch X_i be randomly sampled without replacement from D_i every m/γ local iterations and the Gaussian noise $\mathbf{b}_i^{t,s}$ be sampled from $\mathcal{N}(0, \sigma^2\mathbf{I}_d)$. Assume τ can be divided evenly by m/γ, and let C_i represent the number of rounds device i gets selected for out of T communication rounds, then Algorithm 1 achieves (ϵ_i, δ)-DP for device i in the system, where

$$\epsilon_i = \frac{2C_i\tau G^2}{rm\gamma\sigma^2} + 2\sqrt{\frac{2C_i\tau G^2}{rm\gamma\sigma^2}\log\frac{1}{\delta}}. \tag{10}$$

Proof: It is proved that each round of Algorithm 1 achieves $2\tau G^2/rm\gamma\sigma^2$-zCDP for the device in Ω_t. Due to the device selection, not all devices will upload their models to the server at round t. If their models are not sent out, they do not lose their privacy at that round. Let C_i represent the number of communication rounds device i participated during the whole training process. By Lemma 2, the overall zCDP guarantee of device i in the system after T rounds of communication is $2\tau C_i G^2/rm\gamma\sigma^2$. Theorem 1 then follows by Lemma 4. Note that, each device in the system participates the communication with probability r/n at each round, hence C_i is equivalent to Tr/n in expectation. ■

VI. CONVERGENCE ANALYSIS

In this section, we present the main theoretical results on the convergence properties of our approach. Before stating our results, we give some assumptions and summarize the update rule of our approach as follows. Assumption 2 implies that the objective function f are L-smooth. Assumption 3 ensures that the divergence between local stochastic gradients is bounded. These two assumptions are standard in literature [26]–[28].

Assumption 2 (Smoothness): The local objective function f_i is L-smooth, i.e., for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $i \in [n]$, we have $f_i(\mathbf{y}) \leq f_i(\mathbf{x}) + \langle\nabla f_i(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + (L/2)\|\mathbf{y} - \mathbf{x}\|^2$.

Assumption 3 (Bounded divergence): Let ξ_i be randomly sampled from the local dataset D_i. The stochastic gradient of each device is unbiased and will not diverge a lot from the exact gradient, i.e., for any $\mathbf{x} \in \mathbb{R}^d$ and $i \in [n]$, $\mathbb{E}[\nabla l(\mathbf{x}, \xi_i)] = \nabla f_i(\mathbf{x})$ and $\mathbb{E}\|\nabla l(\mathbf{x}, \xi_i) - \nabla f_i(\mathbf{x})\|^2 \leq \beta^2$.

To prove the convergence of our approach, we first represent the update rule of our approach in a general manner. In Algorithm 1, the total number of iterations is K, i.e., $K = T\tau$. At iteration k where $k = t\tau + s$, each device i evaluates the stochastic gradient $g(\boldsymbol{\theta}_i^k)$ based on its local dataset and updates current model $\boldsymbol{\theta}_i^k$. Thus, n devices have different versions $\boldsymbol{\theta}_1^k, \ldots, \boldsymbol{\theta}_n^k$ of the model. After τ local iterations, devices upload their encrypted local models to the server to generate the

new shared model, i.e., $(1/r)\sum_{i\in\Omega_k}\boldsymbol{\theta}_i^k$ with $(k \mod \tau = 0)$, where $\Omega_k = \Omega_t, \forall k \in [t\tau, t\tau+1, \ldots, t\tau+\tau-1]$.

Now, we can present a virtual update rule that captures Algorithm 1. Define matrices $\boldsymbol{\Theta}^k, \mathbf{G}^k, \mathbf{B}^k \in \mathbb{R}^{d\times n}$ for $k = 0, \ldots, K-1$ that concatenate all local models, gradients and noises:

$$\boldsymbol{\Theta}^k := \left[\boldsymbol{\theta}_1^k, \boldsymbol{\theta}_2^k, \ldots, \boldsymbol{\theta}_n^k\right],$$
$$\mathbf{G}^k := \left[g(\boldsymbol{\theta}_1^k), g(\boldsymbol{\theta}_2^k), \ldots, g(\boldsymbol{\theta}_n^k)\right],$$
$$\mathbf{B}^k := \left[\mathbf{b}_1^k, \mathbf{b}_2^k, \ldots, \mathbf{b}_n^k\right].$$

If device i is not selected to upload its model at iteration k, $\boldsymbol{\theta}_i^k = g(\boldsymbol{\theta}_i^k) = \mathbf{b}_i^k = \mathbf{0}_d$. Besides, define matrix $\mathbf{J}^{\Omega_k} \in \mathbb{R}^{n\times n}$ with element $\mathbf{J}_{i,j}^{\Omega_k} = 1/r$ if $i \in \Omega_k$ and $\mathbf{J}_{i,j}^{\Omega_k} = 0$ otherwise. Unless otherwise stated, $\mathbf{1}^k \in \mathbb{R}^n$ is a column vector of size n with element $\mathbf{1}_i^k = 1$ if $i \in \Omega_k$ and $\mathbf{1}_i^k = 0$ otherwise. To capture periodic averaging, we define $\mathbf{J}^k$ as

$$\mathbf{J}^k := \begin{cases} \mathbf{J}^{\Omega_k}, k \bmod \tau = 0 \\ \mathbf{I}_n, \text{ otherwise.} \end{cases}$$

where $\mathbf{I}_n$ is a $n \times n$ identity matrix. Then a general update rule of our approach can be expressed as follows:

$$\boldsymbol{\Theta}^{k+1} = \left(\boldsymbol{\Theta}^k - \eta(\mathbf{G}^k + \mathbf{B}^k)\right)\mathbf{J}^k. \tag{11}$$

Note that the secure aggregation does not change the sum of local models. Multiplying $\mathbf{1}^k/r$ on both sides of (11), we have

$$\frac{\boldsymbol{\Theta}^{k+1}\mathbf{1}^k}{r} = \frac{\boldsymbol{\Theta}^k\mathbf{1}^k}{r} - \eta\left(\frac{\mathbf{G}^k\mathbf{1}^k}{r} + \frac{\mathbf{B}^k\mathbf{1}^k}{r}\right). \tag{12}$$

Then define the averaged model at iteration k as

$$\hat{\boldsymbol{\theta}}^k := \frac{\boldsymbol{\Theta}^k\mathbf{1}^k}{r} = \frac{1}{r}\sum_{i\in\Omega_k}\boldsymbol{\theta}_i^k.$$

After rewriting (12), one yields

$$\hat{\boldsymbol{\theta}}^{k+1} = \hat{\boldsymbol{\theta}}^k - \eta\left(\frac{1}{r}\sum_{i\in\Omega_k} g(\boldsymbol{\theta}_i^k) + \mathbf{b}_i^k\right). \tag{13}$$

Since devices are selected at random to perform updating in each round, and $g(\boldsymbol{\theta}_i^k)$ is the stochastic gradient computed on a subset of data samples $X_i \in D_i$, the randomness in our federated learning system comes from the device selection, stochastic gradient, and Gaussian noise. In the following, we bound the expectation of several intermediate random variables, which we denote by $\mathbb{E}_{\{\Omega_k, X_i, \mathbf{b}_i^k | i\in[n]\}}[\cdot]$. To simplify the notation, we use $\mathbb{E}[\cdot]$ instead of $\mathbb{E}_{\{\Omega_k, X_i, \mathbf{b}_i^k | i\in[n]\}}[\cdot]$ in the rest of the paper, unless otherwise stated.

As given in Lemma 6 and Lemma 7, we first analyze the expectation of the perturbed stochastic gradients and the network error that captures the divergence between local models and the averaged model at each local iteration. Based on these, we derive the convergence results of the expected gradient norm of the objective function after T communication rounds, as given in Theorem 2.

Lemma 6: The expectation and variance of the averaged perturbed stochastic gradients at iteration k are

$$\mathbb{E}\left[\frac{1}{r}\sum_{i\in\Omega_k}\left(g(\boldsymbol{\theta}_i^k) + \mathbf{b}_i^k\right)\right] = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_i^k), \tag{14}$$

and

$$\mathbb{E}\left[\left\|\frac{1}{r}\sum_{i\in\Omega_k}\left(g(\boldsymbol{\theta}_i^k) + \mathbf{b}_i^k\right) - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2\right] \le \frac{d\sigma^2}{r} + \frac{\beta^2}{\gamma} + \frac{4(n-r)^2}{n^3}\sum_{i=1}^n \left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2. \tag{15}$$

Proof: To simplify the notation, we set $\mathcal{G}^k := (1/r)\sum_{i\in\Omega_k}(g(\boldsymbol{\theta}_i^k) + \mathbf{b}_i^k)$. Given Assumption 1, we have

$$\mathbb{E}\left[\mathcal{G}^k\right] = \sum_{\substack{\Omega\in[n],\\ |\Omega|=r}} P_r(\Omega_k = \Omega)\left(\frac{1}{r}\sum_{i\in\Omega_k}\mathbb{E}\left[g(\boldsymbol{\theta}_i^k) + \mathbf{b}_i^k\right]\right)$$
$$= \frac{1}{r}\frac{1}{\binom{n}{r}}\binom{n-1}{r-1}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_i^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_i^k).$$

Here, $\mathbb{E}[\mathbf{b}_i^k] = \mathbf{0}_d$ since $\mathbf{b}_i^k \sim \mathcal{N}(0, \sigma^2\mathbf{I}_d)$. let $\overline{\mathcal{G}^k} := \mathbb{E}[\mathcal{G}^k]$, we have

$$\mathbb{E}\left[\left\|\mathcal{G}^k - \overline{\mathcal{G}^k}\right\|^2\right]$$
$$= \mathbb{E}\left[\left\|\frac{1}{r}\sum_{i\in\Omega_k} g(\boldsymbol{\theta}_i^k) - \nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2 + \left\|\frac{1}{r}\sum_{i\in\Omega_k}\mathbf{b}_i^k\right\|^2\right]$$
$$+ \mathbb{E}\left[\left\|\frac{1}{r}\sum_{i\in\Omega_k}\nabla f_i(\boldsymbol{\theta}_i^k) - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2\right]$$
$$\le \frac{d\sigma^2}{r} + \sum_{\substack{\Omega\in[n],\\ |\Omega|=r}} P_r(\Omega_k = \Omega)\frac{1}{r}\sum_{i\in\Omega_k}\mathbb{E}\left[\left\|g(\boldsymbol{\theta}_i^k) - \nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2\right]$$
$$+ 2\sum_{\substack{\Omega\in[n],\\ |\Omega|=r}} P_r(\Omega_k = \Omega)\left(\frac{1}{r} - \frac{1}{n}\right)^2 r\sum_{i\in\Omega_k}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2$$
$$+ 2\sum_{\substack{\Omega\in[n],\\ |\Omega|=r}} P_r(\Omega_k = \Omega)\frac{1}{n^2}(n-r)\sum_{i\notin\Omega_k}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2$$
$$\le \frac{d\sigma^2}{r} + \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left\|g(\boldsymbol{\theta}_i^k) - \nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2\right]$$
$$+ \frac{2(n-r)}{n^2}\frac{1}{\binom{n}{r}}\left(\binom{n}{r} - \binom{n-1}{r-1}\right)\sum_{i=1}^n \left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2$$

$$+\frac{2(n-r)^2}{rn^2}\frac{1}{\binom{n}{r}}\binom{n-1}{r-1}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2$$
$$\leq \frac{d\sigma^2}{r}+\frac{\beta^2}{\gamma}+\frac{4(n-r)^2}{n^3}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2,$$

where we use the independence of Gaussian noise and Assumption 3. Note that based on Assumption 3, we have

$$\begin{aligned}\mathbb{E}\left\|g(\boldsymbol{\theta}_i^k)-\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2 &= \frac{1}{\gamma^2}\left\|\sum_{\xi_i\in X_i}\nabla f_i(\boldsymbol{\theta}_i^k,\xi_i)-\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2\\ &= \frac{1}{\gamma^2}\sum_{\xi_i\in X_i}\left\|\nabla f_i(\boldsymbol{\theta}_i^k,\xi_i)-\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2\\ &\leq \frac{\beta^2}{\gamma},\end{aligned}$$

due to the independence of random variable ξ_i. ■

Lemma 7: Assume $k=t\tau+s$, the expected network error at iteration k is bounded as follows:

$$\mathbb{E}\left[\frac{1}{r}\sum_{i\in\Omega_k}\left\|\hat{\boldsymbol{\theta}}^k-\boldsymbol{\theta}_i^k\right\|^2\right]\leq 2s^2\eta^2\left(\frac{d\sigma^2(r+1)}{r}+\frac{2\beta^2}{\gamma}\right)$$
$$+4s\eta^2\frac{2(n-r)^2+n^2}{n^3}\sum_{h=0}^{s-1}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^{t\tau+h})\right\|^2. \tag{16}$$

Proof: Since $k=t\tau+s$ and all devices in Ω_k start from the same model received from the server $\boldsymbol{\theta}^{t\tau}$ to update, i.e., $\hat{\boldsymbol{\theta}}^{t\tau}=\boldsymbol{\theta}_i^{t\tau}=\boldsymbol{\theta}^{t\tau}, \forall i\in\Omega_k$. For device $i\in\Omega_k$, we have

$$\boldsymbol{\theta}_i^k=\boldsymbol{\theta}_i^{t\tau}-\eta\sum_{h=0}^{s}g(\boldsymbol{\theta}_i^{t\tau+h})+\mathbf{b}_i^{t\tau+h}. \tag{17}$$

Given that $\hat{\boldsymbol{\theta}}^k=(1/r)\sum_{i\in\Omega_k}\boldsymbol{\theta}_i^k$, one yields $\forall j\in\Omega_k$,

$$\begin{aligned}\left\|\hat{\boldsymbol{\theta}}^k-\boldsymbol{\theta}_j^k\right\|^2 &\leq 2\eta^2\left\|\frac{1}{r}\sum_{i\in\Omega_k}\sum_{h=0}^{s-1}g(\boldsymbol{\theta}_i^{t\tau+h})+\mathbf{b}_i^{t\tau+h}\right\|^2\\ &\quad+2\eta^2\left\|\sum_{h=0}^{s-1}g(\boldsymbol{\theta}_j^{t\tau+h})+\mathbf{b}_j^{t\tau+h}\right\|^2\\ &\leq 2s\eta^2\sum_{h=0}^{s-1}\left\|\frac{1}{r}\sum_{i\in\Omega_k}g(\boldsymbol{\theta}_i^{t\tau+h})+\mathbf{b}_i^{t\tau+h}\right\|^2\\ &\quad+2s\eta^2\sum_{h=0}^{s-1}\left\|g(\boldsymbol{\theta}_j^{t\tau+h})+\mathbf{b}_j^{t\tau+h}\right\|^2,\end{aligned}$$

where we use the inequality $\|\sum_{i=1}^n\mathbf{a}_i\|^2\leq n\sum_{i=1}^n\|\mathbf{a}_i\|^2$. By Lemma 6 and the fact that $\mathbb{E}[(\mathbf{X}-\mathbb{E}[\mathbf{X}])^2]=\mathbb{E}[\mathbf{X}^2]-\mathbb{E}[\mathbf{X}]^2$, we have that

$$\mathbb{E}\left[\left\|\frac{1}{r}\sum_{i\in\Omega_k}g(\boldsymbol{\theta}_i^{t\tau+h})+\mathbf{b}_i^{t\tau+h}\right\|^2\right]$$
$$\leq\frac{d\sigma^2}{r}+\frac{\beta^2}{\gamma}+\frac{4(n-r)^2+n^2}{n^3}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^{t\tau+h})\right\|^2,$$

which is not related to the index of device j. In addition,

$$\left\|g(\boldsymbol{\theta}_j^{t\tau+h})+\mathbf{b}_j^{t\tau+h}\right\|^2\leq d\sigma^2+\frac{\beta^2}{\gamma}+\left\|\nabla f_j(\boldsymbol{\theta}_j^{t\tau+h})\right\|^2,$$

thus, the expected network error at iteration k is

$$\mathbb{E}\left[\frac{1}{r}\sum_{j\in\Omega_k}\left\|\hat{\boldsymbol{\theta}}^k-\boldsymbol{\theta}_j^k\right\|^2\right]\leq 2s\eta^2\left(\frac{sd\sigma^2(r+1)}{r}+\frac{2s\beta^2}{\gamma}\right.$$
$$\left.+\frac{4(n-r)^2+2n^2}{n^3}\sum_{h=0}^{s-1}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^{t\tau+h})\right\|^2\right).$$

Lemma 7 is finally obtained by relaxing the constant of the second term. ■

Theorem 2 (Convergence Result of Our Approach): For Algorithm 1, suppose the total number of iterations $K=T\tau$ where T is the number of communication rounds and τ is the local iteration period. Under Assumptions 2-3, if the learning rate satisfies $5\eta L+3\tau^2\eta^2L^2\leq 1$, and all devices are initialized at the same point $\boldsymbol{\theta}^0\in\mathbb{R}^d$, then after K iterations the expected gradient norm is bounded as follows

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla f(\hat{\boldsymbol{\theta}}^k)\|^2\right]\leq\frac{2(f(\boldsymbol{\theta}^0)-f^*)}{\eta K}+\frac{\eta L\beta^2}{2\gamma}+\frac{\eta Ld\sigma^2}{2r}$$
$$+(\tau-1)(2\tau-1)\left(\frac{2\eta^2L^2\beta^2}{3\gamma}+\frac{\eta^2L^2d\sigma^2(r+1)}{3r}\right). \tag{18}$$

Here, σ^2 is the variance of Gaussian noise, β^2 is the upper bound of the variance of local stochastic gradients, L is the Lipschitz constant of the gradient, n is the total number of devices, and r is the number of selected devices at each round.

Proof: According to Assumption 2, the global loss function f is L-smooth. Let $\mathcal{G}^k:=(1/r)\sum_{r\in\Omega_k}(g(\boldsymbol{\theta}_i^k)+\mathbf{b}_i^k)$, we have the expectation of the objective gap between two iterations, i.e.,

$$\mathbb{E}\left[f(\hat{\boldsymbol{\theta}}^{k+1})-f(\hat{\boldsymbol{\theta}}^k)\right]$$
$$\leq\frac{\eta^2L}{2}\mathbb{E}\left[\left\|\mathcal{G}^k\right\|^2\right]-\eta\mathbb{E}\left[\frac{1}{r}\sum_{i\in\Omega_k}\langle\nabla f(\hat{\boldsymbol{\theta}}^k),\mathbb{E}\left[g(\boldsymbol{\theta}_i^k)+\mathbf{b}_i^k\right]\rangle\right]$$
$$=\frac{\eta^2L}{2}\mathbb{E}\left[\left\|\mathcal{G}^k\right\|^2\right]-\eta\mathbb{E}\left[\frac{1}{r}\sum_{i\in\Omega_k}\langle\nabla f(\hat{\boldsymbol{\theta}}^k),\nabla f_i(\boldsymbol{\theta}_i^k)\rangle\right]$$

$$\leq \frac{\eta^2 L}{2}\mathbb{E}\left[\left\|\mathcal{G}^k\right\|^2\right] - \frac{\eta}{2}\|\nabla f(\hat{\boldsymbol{\theta}}^k)\|^2 - \frac{\eta}{2n}\sum_{i=1}^{n}\|\nabla f_i(\boldsymbol{\theta}_i^k)\|^2 + \frac{\eta}{2}\mathbb{E}\left[\frac{1}{r}\sum_{i\in\Omega_k}\left\|\nabla f_i(\hat{\boldsymbol{\theta}}^k) - \nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2\right], \tag{19}$$

where we use the inequality $-2\langle \mathbf{a}, \mathbf{b}\rangle = \|\mathbf{a}-\mathbf{b}\|^2 - \|\mathbf{a}\|^2 - \|\mathbf{b}\|^2$ for any two vectors $\mathbf{a}, \mathbf{b}$. After minor rearranging, it is easy to show

$$\mathbb{E}\left[\|\nabla f(\hat{\boldsymbol{\theta}}^k)\|^2\right] \leq \frac{2}{\eta}\mathbb{E}\left[f(\hat{\boldsymbol{\theta}}^k) - f(\hat{\boldsymbol{\theta}}^{k+1})\right] + L\eta\mathbb{E}\left[\left\|\mathcal{G}^k\right\|^2\right] - \frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\boldsymbol{\theta}_i^k)\|^2 + L^2\mathbb{E}\left[\frac{1}{r}\sum_{i\in\Omega_k}\left\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}_i^k\right\|^2\right]. \tag{20}$$

By Lemma 6 and Lemma 7, we have that the last three terms of (20) is bounded by B_k, which is

$$B_k \geq \frac{4\eta L(n-r)^2 + (\eta L - 1)n^2}{n^3}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2 + \frac{s\eta^2L^2(2(n-r)^2+n^2)}{n^3}\sum_{h=0}^{s-1}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^{t\tau+h})\right\|^2 + 2\eta^2L^2s^2\left(d\sigma^2\left(1+\frac{1}{r}\right)+\frac{2\beta^2}{\gamma}\right)+\eta L\left(\frac{d\sigma^2}{r}+\frac{\beta^2}{\gamma}\right).$$

Then, taking the total expectation and averaging of (20) over all iterations, we have

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla f(\hat{\boldsymbol{\theta}}^k)\|^2\right] \leq \frac{2(f(\boldsymbol{\theta}^0)-f^*)}{\eta K} + \frac{1}{K}\sum_{k=0}^{K-1}B_k, \tag{21}$$

where we use the fact that $f(\boldsymbol{\theta}^K) \geq f^*$. Next, our goal is to find the upper bound of $(1/K)\sum_{k=0}^{K-1}B_k$. Note that

$$\frac{1}{K}\sum_{k=0}^{K-1}B_k \leq \frac{4\eta L(n-r)^2 + (\eta L-1)n^2}{Kn^3}\sum_{k=0}^{K-1}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2 + \tau^2\eta^2L^2\frac{n^2+2(n-r)^2}{Kn^3}\sum_{k=0}^{K-1}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2 + \frac{\eta^2L^2}{3}(\tau-1)(2\tau-1)\left(d\sigma^2\left(1+\frac{1}{r}\right)+\frac{2\beta^2}{\gamma}\right) + \frac{\eta L}{2}\left(\frac{d\sigma^2}{r}+\frac{\beta^2}{\gamma}\right), \tag{22}$$

based on the fact that $1^2+\cdots+n^2 = n(n+1)(2n+1)/6$. Since we have

$$\tau^2\eta^2L^2\frac{n^2+2(n-r)^2}{Kn^3} + \frac{4\eta L(n-r)^2+(\eta L-1)n^2}{Kn^3} \leq \frac{(3\tau^2\eta^2L^2+5\eta L-1)n^2}{Kn^3},$$

then if the learning rate η satisfies that $5\eta L + 3\tau^2\eta^2L^2 \leq 1$, we can finally obtain a constant bound for $(1/K)\sum_{k=1}^{K}B_k$, i.e.,

$$\frac{1}{K}\sum_{k=0}^{K-1}B_k \leq \frac{\eta L d\sigma^2}{2r}\left(\frac{2\eta L(r+1)}{3}(\tau-1)(2\tau-1)+1\right) + \frac{\eta L\beta^2}{2\gamma}\left(\frac{4\eta L}{3}(\tau-1)(2\tau-1)+1\right).$$

Substituting the expression of $(1/K)\sum_{k=0}^{K-1}B_k$ back to (21), we finally obtain Theorem 2. ■

By setting the learning rate $\eta = \mathcal{O}(\sqrt{n/K})$, Algorithm 1 achieves the asymptotic convergence rate of $\mathcal{O}(1/\sqrt{nK}) + \mathcal{O}(\tau^2\sigma^2/K)$, when K is sufficiently large. If we further assume that the objective function is strongly convex, i.e., the following Assumption 4 holds, Algorithm 1 achieves the non-asymptotic convergence result stated in Corollary 2.

Assumption 4: The objective function f is λ-strongly convex if for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ we have $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \geq \lambda\|\mathbf{x}-\mathbf{y}\|$ for some constant $\lambda > 0$.

Corollary 2 (Convergence Result for Convex Loss): For Algorithm 1, suppose the total number of iterations $K = T\tau$ where T is the number of communication rounds and τ is the local iteration period. Under Assumptions 2-4, if the learning rate satisfies $5\eta L + 3\tau^2\eta^2L^2 \leq 1$, and all devices are initialized at the same point $\boldsymbol{\theta}^0 \in \mathbb{R}^d$. Then after K iterations, the expected optimality gap is bounded as follows

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}f(\hat{\boldsymbol{\theta}}^k) - f^*\right] \leq \frac{(1-\eta\lambda)}{K\eta\lambda}\left(f(\boldsymbol{\theta}^0)-f^*\right) + \frac{\eta L\beta^2}{4\lambda\gamma} + \frac{\eta L d\sigma^2}{4r\lambda} + \eta^2L^2(\tau-1)(2\tau-1)\left(\frac{\beta^2}{3\lambda\gamma}+\frac{d\sigma^2(r+1)}{6r\lambda}\right). \tag{23}$$

Proof: According to Assumption 4, the inequality (19) can be written as

$$\mathbb{E}\left[f(\hat{\boldsymbol{\theta}}^{k+1})\right] \leq \eta\lambda f^* + (1-\eta\lambda)f(\hat{\boldsymbol{\theta}}^k) + \frac{\eta^2L}{2}\mathbb{E}\left[\left\|\mathcal{G}^k\right\|^2\right] + \frac{\eta L^2}{2}\mathbb{E}\left[\frac{1}{r}\sum_{i\in\Omega_k}\left\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}_i^k\right\|^2\right] - \frac{\eta}{2n}\sum_{i=1}^{n}\|\nabla f_i(\boldsymbol{\theta}_i^k)\|^2. \tag{24}$$

Let the last three terms in (24) be bounded by B_k', which is equivalent to $\eta B_k/2$. Taking the total expectation and averaging over K iterations of both sides of (24), one can obtain

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}f(\hat{\boldsymbol{\theta}}^{k+1}) - f^*\right] \leq \frac{1-\eta\lambda}{K\eta\lambda}\left(f(\boldsymbol{\theta}^0)-f^*\right) + \frac{1}{K\eta\lambda}\sum_{k=0}^{K-1}B_k'. \tag{25}$$

Next, our goal is to find the upper bound of $(1/K)\sum_{k=0}^{K-1} B'_k$. Using the inequality (22), we have

$$\frac{1}{K}\sum_{k=0}^{K-1} B'_k \leq \Bigg[\frac{4\eta^2 L(n-r)^2 + \eta(\eta L-1)n^2}{2Kn^3} + \tau^2\eta^3 L^2 \frac{n^2+2(n-r)^2}{2Kn^3}\Bigg]\sum_{k=0}^{K-1}\sum_{i=1}^{n}\left\|\nabla f_i(\boldsymbol{\theta}_i^k)\right\|^2 + \frac{\eta^3 L^2}{6}(\tau-1)(2\tau-1)(d\sigma^2(1+1/r)+2\beta^2/\gamma) + \frac{\eta^2 L}{4}(d\sigma^2/r+\beta^2/\gamma).$$

If the learning rate η satisfies that $5\eta L + 3\tau^2\eta^2 L^2 \leq 1$, we obtain

$$\frac{1}{K}\sum_{k=0}^{K-1} B'_k \leq \frac{\eta^2 L d\sigma^2}{4r}\left(\frac{2\eta L(r+1)}{3}(\tau-1)(2\tau-1)+1\right) + \frac{\eta^2 L\beta^2}{4\gamma}\left(\frac{4\eta L}{3}(\tau-1)(2\tau-1)+1\right).$$

Theorem 2 follows by substituting the expression of $(1/K)\sum_{k=0}^{K-1} B'_k$ back to (25). ■

VII. EXPERIMENTS

In this section, we evaluate the performance of our proposed scheme. We first describe our experimental setup and then show the convergence properties of our approach. Next, we demonstrate the effectiveness of our approach by comparing it with a baseline approach. Finally, we show the trade-off between privacy and model utility in our approach and how our secure aggregation protocol improves the accuracy of the learned model.

A. EXPERIMENTAL SETUP

Datasets and Learning Tasks. We explore the benchmark dataset *Adult* [29] using both logistic regression and neural network models in our experiments. The Adult dataset contains 48 842 samples with 14 numerical and categorical features, with each sample corresponding to a person. The task is to predict if the person's income exceeds \$50000 based on the 14 attributes, namely, *age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, and native-country*. To simulate a distributed setting based on the Adult dataset, we evenly assign the original Adult data to 16 devices such that each device contains 3052 data samples. We train a logistic regression classifier and a 3-layer neural network classifier (with ReLU activation function) and use the softmax cross-entropy as the loss function.

Baseline. We use a distributed version of the state-of-the-art differentially private learning scheme in [24] as a baseline to evaluate the efficiency of our proposed scheme, called DP-DSGD (Differentially Private Distributed SGD). In DP-DSGD, only one step of SGD is performed to update the local model on a device during each communication period, and Gaussian noise is added to each model update before sending it out.

Hyperparameters. We take 80% of the data on each device for training, 10% for testing and 10% for validation. We tune the hyperparameters on the validation set and report the average accuracy on the testing sets of all devices. The gradient norm G is enforced by clipping, which is widely used in differentially private learning. For all experiments, we set the privacy failure probability $\delta = 10^{-4}$ and the number of selected devices per round $r = 10$. Note that due to the randomized nature of differentially private mechanisms, we repeat all the experiments for 5 times and report the average results.

B. CONVERGENCE PROPERTIES OF OUR APPROACH

In this subsection, we show the algorithmic convergence properties of our approach under several settings of noise magnitude σ and local iteration period τ. Specifically, for the logistic regression, we show the testing accuracy and the expected training loss with respect to the number of communication rounds T when $\sigma \in \{10^{-5}, 10^{-4}, 5\times10^{-4}, 10^{-3}\}$ and $\tau \in \{1, 5, 10, 40\}$. The results for the logistic regression are depicted in Fig. 3. Similarly, for the neural network, we show the testing accuracy and expected gradient norm with respect to the number of communication rounds T when $\sigma \in \{10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}\}$ and $\tau \in \{1, 5, 10, 20\}$. The results for the neural network are finally shown in Fig. 4.

For the logistic regression, the testing accuracy and expected loss generally decrease sharply and then slowly afterwards. As the noise magnitude σ increases, the expected training loss of the logistic regression converges to a higher bound and the testing accuracy decreases, which is consistent with the convergence properties of our approach where a larger σ implies a larger convergence error. For all settings of noise, with a larger local iteration period, the expected loss drops more sharply at the beginning and arrives at a higher stationary point, which is consistent with our approach's convergence properties where a larger τ implies a larger convergence error. When $\sigma = 10^{-3}$ and $\tau = 40$, we can see that after the expected loss decreases to 7 using about 40 rounds of communication, it increases as more computations and communications are involved. The reason is that after the loss arrived at a stationary point, keeping training brings additional noise into the well-trained model and hence the model performance drops. Similar trends have been observed for the neural network classifier. When $\sigma = 10^{-2}$, the testing accuracy drops from the initialized value quickly as the noise is added into the system, and then it increases as more computations and communications are involved.

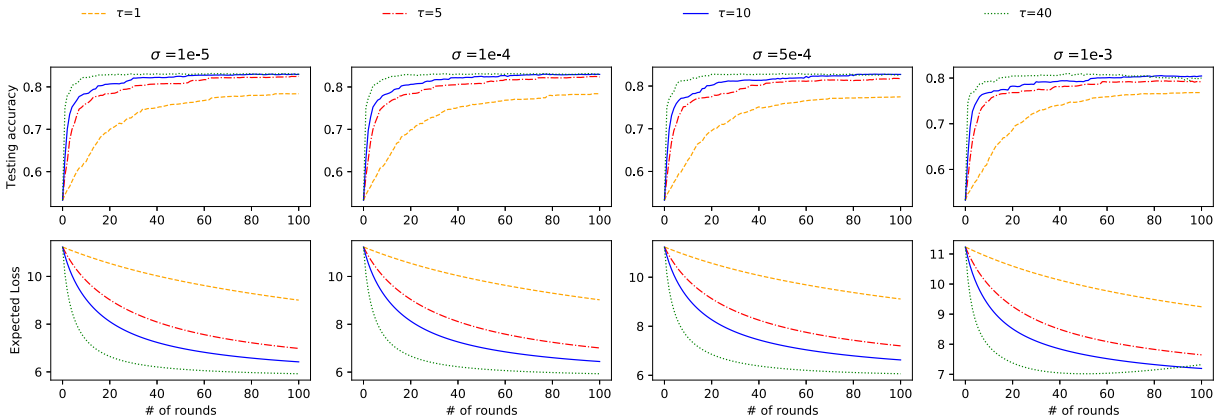

FIG. 3. **Convergence of the expected loss (logistic regression). Here, we show the convergence of our approach in the first 100 communication rounds.**

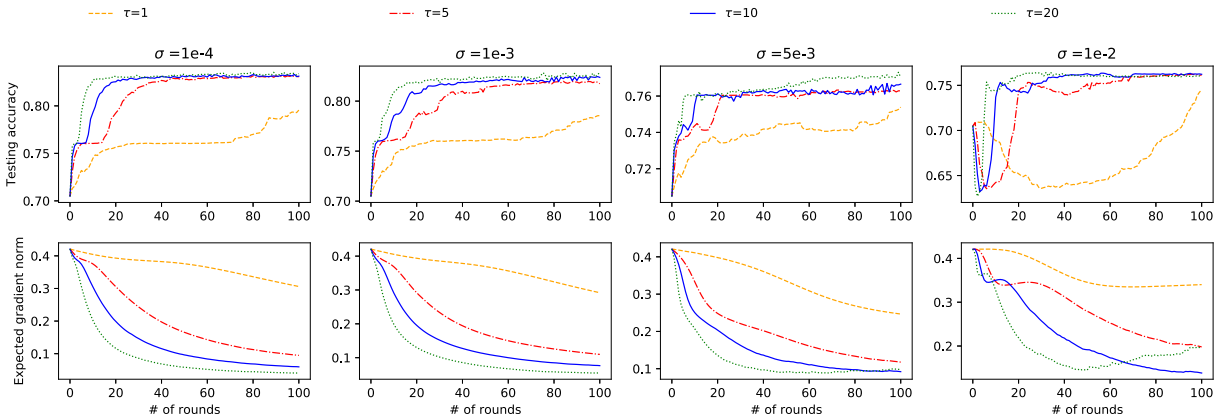

FIG. 4. **Convergence of the expected gradient norm (neural network). Here, we show the convergence of our approach in the first 100 communication rounds.**

C. MODEL UTILITY OF OUR APPROACH

In this subsection, we show the model utility of our approach compared with DP-DSGD. Specifically, for the logistic regression, we set the number of communication rounds $T = 20$ for both approaches and $\tau = 10$ for our approach. Both approaches preserve $(10, 10^{-4})$-DP after 20 rounds of communication. For our approach, we compute the noise magnitude σ by Theorem 1. Note that we randomly sample the active devices for each round and make sure each device participates the same number of communication rounds beforehand, and then we use the sampling result for both approaches. For DP-DSGD, it achieves $(2C_iG^2/n\gamma^2\sigma^2 + 2\sqrt{2\log(1/\delta)C_iG^2/n\gamma^2\sigma^2}, \delta)$-DP for device i, which can be used to compute the noise magnitude σ given ϵ and δ. The testing accuracy and expected loss with respect to the number of communicate rounds are shown in Fig. 5. For the neural network, we set the number of communication rounds $T = 50$, the overall privacy budget $\epsilon = 10$ for both approaches, and $\tau = 5$ for our approach. The testing accuracy and expected gradient norm with respect to the number of communication rounds are shown in Fig. 6.

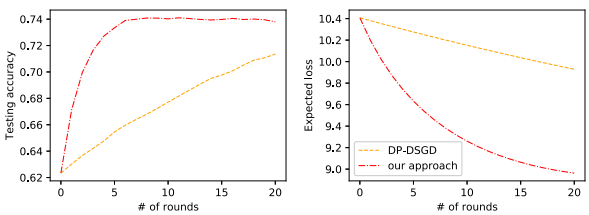

FIG. 5. **Testing accuracy and expected training loss of our approach in comparison with DP-DSGD (logistic regression). Here, we set $T = 20$ and $\epsilon = 10$ for both approaches, and set $\tau = 10$ for our approach.**

For logistic regression, we observe that our approach exhibits faster convergence than DP-DSGD at the beginning, and finally achieves a higher accuracy and a lower expected loss than DP-DSGD within 20 rounds of communication. For neural network, we observe the similar trend. Our approach converges faster than DP-DSGD and achieves a higher accuracy and a lower expected gradient norm than DP-DSGD. Therefore, our approach achieves higher model utilities than DP-DSGD under the same privacy guarantee.

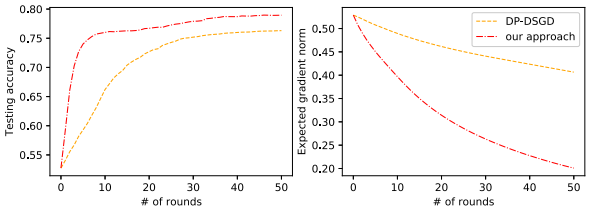

FIG. 6. Testing accuracy and expected gradient norm of our approach in comparison with DP-DSGD (neural network). Here, we set $T = 50$ and $\epsilon = 10$ for both approaches, and set $\tau = 5$ for our approach.

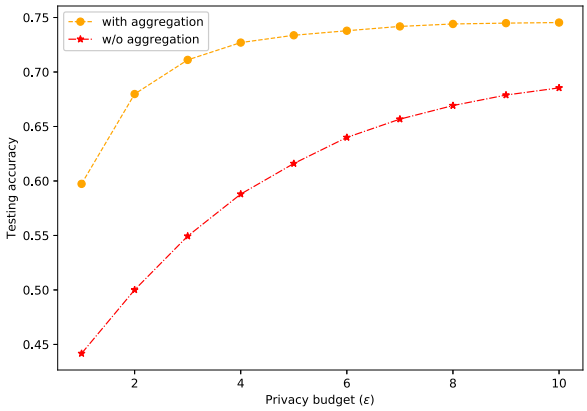

FIG. 7. Trade-off between privacy and accuracy (logistic regression). Here, we set $\tau = 2$ and $T = 20$.

D. TRADE-OFF BETWEEN PRIVACY AND UTILITY

To observe the privacy-utility tradeoff, we evaluate the effects of different values of privacy budgets ϵ on the testing accuracy of trained classifiers. In addition, we compare our approach with the approach without secure aggregation (i.e., same as our approach but without secure aggregation) to show how secure aggregation improves the accuracy. For logistic regression, we set the local iteration period $\tau = 2$ and the number of communication rounds $T = 20$. For neural network, we set the local iteration period $\tau = 5$ and the number of communication rounds $T = 50$. We show the testing accuracy with respect to different values of privacy budget ϵ of logistic regression and neural network in Fig. 7 and Fig. 8, respectively. As expected, a larger value of ϵ results in a higher accuracy while providing a lower DP guarantee. Moreover, our approach with secure aggregation always outperforms the approach without secure aggregation because less noise is added in each iteration.

VIII. RELATED WORK

Privacy issue has received significant attention recently in distributed learning scenarios handling user-generated data. Among distributed learning schemes that preserve privacy, many of them rely on secure multi-party computation or homomorphic encryption, which involve both high computation and communication overhead and are only applicable to

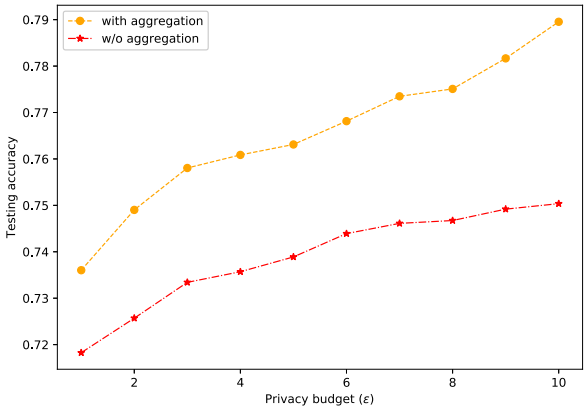

FIG. 8. Trade-off between privacy and accuracy (neural network). Here, we set $T = 50$ and $\tau = 5$.

simple learning tasks such as linear regression [30] and logistic regression [31]. Furthermore, these privacy-preserving solutions could not prevent the information leakage from the final learned model. Model inversion attacks and membership inference attacks have already shown that sensitive information about the training data could be extracted even when the adversaries only have access to the final learned model [6], [7], [23]. DP provides rigorous protections against such attacks and has become the de-facto standard for privacy, and it is being increasingly adopted in private data analysis [20].

A wide range of differentially private distributed learning algorithms (see [24], [32]–[37] and references therein) have been proposed based on different optimization methods (e.g., alternating direction method of multipliers, gradient descent, and distributed consensus) and noise addition mechanisms (e.g., output perturbation, objective perturbation, and gradient perturbation). These schemes share a common goal to preserve utility of learned model with the presence of DP noises. Among them, multiple works focus on the integration of DP and federated learning. However, most of them (e.g., [8]–[17]) demonstrate the performance of proposed approaches merely by experiments, with no theoretical analysis on the convergence. However, experimental observations are not always reliable since the performance of machine learning algorithms heavily rely on hyper-parameter tuning, and an algorithm that is observed to perform better than other algorithms may just be the algorithm that is better tuned. Moreover, algorithms without rigorous convergence may have low accuracies under special cases that are not observed in experiments. Hence it is meaningful to have performance analysis that could be used to guide the algorithm design and hyper-parameter tuning. In this paper, we provide the rigorous performance analysis and tight end-to-end DP bound for our scheme.

The works closest to ours are [9], [18], [35]–[38], which also provide performance analysis for federated learning. These work, however, focus on some special cases of federated learning schemes or have different privacy assumptions. Specifically, the work in [36], [37] focus on FedSGD,

where only one step of stochastic gradient descent is performed in each communication round. Multi-step local update in each round is considered in [18], [38], however, they do not consider client sampling, which could greatly impact the performance of the algorithms and corresponding privacy analysis. It is worth noting that performing multiple steps of local updates in each communication round as well as adopting client sampling are two core design factors that make federated learning communication-efficient and practical in large scale [19]. We consider both factors when designing our differentially private federated learning scheme in this paper, where the previous two settings could be considered as special cases of our approach. The threat model considered in [35], [38] also differs from our work. In addition, a lot of federated learning tasks have non-convex loss functions, but the performance analysis in prior works [18], [35]–[38] rely on a common convexity assumption of the loss functions. We remove this assumption in our paper and thus our analysis is much more general than previous approaches.

There are also some related work that are orthogonal to our approach [25], [39], [40]. Agarwal *et al.* [39] proposed a modified distributed SGD scheme based on gradient quantization and binomial mechanism to make the scheme both private and communication-efficient. Li *et al.* [40] developed a method that compresses the transmitted messages via sketches to simultaneously achieve communication efficiency and DP in distributed learning. Our work is orthogonal to theirs by focusing on reducing the number of communication rounds via more local computation per round instead of the size of messages transmitted per round. Besides, [25] proposed a secure aggregation method to protect model updates during the training of federated learning, but its focus is to present suitable cryptographic techniques that ensure secure aggregation in federated learning under unreliable mobile environments, without a detailed study of the integration of DP, secure aggregation, and learning algorithms. It is also worth noting that many prior works assume a trusted central server [9], which is a stronger assumption than our "honest-but-curious" threat model.

IX. CONCLUSION

In this paper, we have proposed a differentially private federated learning approach based on the state-of-the-art and mostly adopted federated learning scheme, *federated averaging with client sampling*. We have provided a rigorous convergence analysis of our proposed approach, which is valid for both convex and non-convex loss functions. We have also tightly accounted the privacy loss over the interactive learning process using zCDP and provided an end-to-end analysis on its DP guarantee. We have conducted extensive experiments on the real-world dataset, and the experimental results have validated the effectiveness of the proposed scheme with both good model utility and strong privacy protection.

REFERENCES

[1] A. Pothitos, "IoT and wearables: Fitness tracking," 2017, [Online]. Available: http://www.mobileindustryreview.com/2017/03/iot-wearables-fitness-tracking.html

[2] P. Goldstein, "Smart cities gain efficiencies from IoT traffic sensors and data," 2018, [Online]. Available: https://statetechmagazine.com/article/2018/12/smart-cities-gain-efficiencies-iot-traffic-sensors-and-data-perfcon

[3] A. Weinreic, "The future of the smart home: Smart homes and IoT: A century in the making," 2018, [Online]. Available: https://statetechmagazine.com/article/2018/12/smart-cities-gain-efficiencies-iot-traffic-sensors-and-data-perfcon

[4] E. Folk, "How IoT is transforming the energy industry," 2019, [Online]. Available: https://www.renewableenergymagazine.com/emily-folk/how-iot-is-transforming-the-energy-industry-20190418

[5] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[6] M. Al-Rubaie and J. M. Chang, "Reconstruction attacks against mobile-based continuous authentication systems in the cloud," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 12, pp. 2648–2663, Dec. 2016.

[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 3–18.

[8] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557*.

[9] B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–14. [Online]. Available: https://openreview.net/pdf?id=BJ0hF1Z0b

[10] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "LDP-fed: Federated learning with local differential privacy," in *Proc. 3rd ACM Int. Workshop Edge Syst., Analytics Netw.*, 2020, pp. 61–66.

[11] Z. Liang, B. Wang, Q. Gu, S. Osher, and Y. Yao, "Exploring private federated learning with laplacian smoothing," 2020, *arXiv:2005.00218*.

[12] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, 2019, pp. 1–11.

[13] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2587–2596.

[14] Y. Zhao *et al.*, "Local differential privacy based federated learning for internet of things," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8836–8853, Jun. 2021.

[15] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, 2019, pp. 13–23.

[16] V. Mugunthan, A. Peraire-Bueno, and L. Kagal, "Privacyfl: A simulator for privacy-preserving and secure federated learning," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 3085–3092.

[17] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10782–10793, Nov. 2020.

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

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

[20] C. Dwork *et al.*, "The algorithmic foundations of differential privacy," *Foundations Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[21] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory Cryptography Conf.*, 2016, pp. 635–658.

[22] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 332–349.

[23] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1322–1333.

[24] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[25] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.
[26] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–14.
[27] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," in *Proc. ICML Workshop on Coding Theory for Mach. Learn.*, 2019.
[28] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–12.
[29] C. L. Blake, "UCI Repository of Machine Learning Databases, Irvine, University of California," 1998, [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository
[30] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proc. IEEE Symp. Secur. Privacy*, 2013, pp. 334–348.
[31] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 19–38.
[32] Y. Guo and Y. Gong, "Practical collaborative learning for crowdsensing in the internet of things with differential privacy," in *Proc. IEEE Conf. Commun. Netw. Secur.*, 2018, pp. 1–9.
[33] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1002–1012, 2019.
[34] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proc. ACM Workshop Privacy Electron. Soc.*, 2012, pp. 81–90.
[35] K. Wei *et al.*, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Trans. Mobile Comput.*, early access, Feb. 4, 2021, doi: 10.1109/TMC.2021.3056991.
[36] M. Seif, R. Tandon, and M. Li, "Wireless federated learning with local differential privacy," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020, pp. 2604–2609.
[37] A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2521–2529.
[38] Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-iid data in AIoT," *IEEE Trans. Ind. Informat.*, early access, Apr. 19, 2021, doi: 10.1109/TII.2021.3073925.
[39] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7564–7575.
[40] T. Li, Z. Liu, V. Sekar, and V. Smith, "Privacy for free: Communication-efficient learning with differential privacy using sketches," 2019, *arXiv:1911.00972*.

RUI HU (Student Member, IEEE) received the B.Eng. degree in electrical engineering from Jinan University, Guangzhou, China, in 2017. She is currently working toward the Ph.D. degree in electrical and computer engineering with the University of Texas at San Antonio, San Antonio, TX, USA. Her research interests include security and privacy in machine learning, social networks, and Internet of Things. She was the recipient of the Dorrough Distinguished Graduate Fellowship and Graduate Student Professional Development Awards in 2018 and 2019, respectively.

YUANXIONG GUO (Senior Member, IEEE) received the B.Eng. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2009, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2012 and 2014, respectively. Since 2019, he has been an Assistant Professor with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX, USA. His current research interests include data analytics, security, and privacy with applications to Internet of Things and edge computing. He is on the Editorial Board of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is the Track Co-Chair for IEEE VTC 2021-Fall. He was the recipient of the Best Paper Award in the IEEE Global Communications Conference 2011.

YANMIN GONG (Senior Member, IEEE) received the B.Eng. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2009, the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2016. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX, USA. Her research interests include security and privacy for machine learning, machine learning in wireless networks, wireless security, and Internet of things. She was the recipient of the Best Paper Award of IEEE Global Communications Conference 2017, the NSF CRII Award 2019, and the NSF CAREER Award 2021. She is the Technical Program Committee Member of the IEEE INFOCOM and IEEE CNS. She is an Associate Editor for the IEEE WIRELESS COMMUNICATION.