Differentially Private Federated Learning with Shuffling and Client Self-Sampling

Antonious M. Girgis, Deepesh Data, and Suhas Diggavi
Email: amgirgis@g.ucla.edu, deepesh.data@gmail.com, suhas@ee.ucla.edu.

***Abstract*—This paper studies a distributed optimization problem in the federated learning (FL) framework under differential privacy constraints, whereby a set of clients having local samples are connected to an untrusted server, who wants to learn a global model while preserving the privacy of clients' local datasets. We propose a new client sampling called *self-sampling* that reflects the random availability of clients in the learning process in FL. We analyze the differential privacy of the SGD with client self-sampling by composing amplification by sub-sampling along with amplification by shuffling. Furthermore, we analyze the convergence of the proposed SGD algorithm showing that we can get a reasonable learning performance while preserving the privacy of clients' data even with client self-sampling.**

I. Introduction

In this paper we consider a federated learning (FL) framework [2]–[4], where the data is generated across multiple clients. The server wants to learn a machine learning model that minimizes a convex objective function using the local datasets, without collecting the data at the central server due to privacy considerations. In order to generate a learning model, the commonly used mechanism is Stochastic Gradient Descent (SGD) [5]. FL introduces several unique challenges to this traditional model that cause tension with the objective: (i) We need to provide privacy guarantees on the local datasets at each client against any adversary that can observe the global model; (ii) work with a dynamic client population in each round of communication between the server and the clients. This happens due to scale (*e.g.,* tens of millions of devices) and only a small fraction of clients are sampled at each iteration depending on their availability.

Since we need to give privacy to the local data residing at the clients, the traditional framework to give guarantees is through the notion of local differential privacy (*e.g.,* see [6]–[9]), where the server is itself untrusted. The challenge is that the traditional privacy approach to the learning problem uses local differential privacy (LDP) [6]–[8], [10], [11], which is known to give poor learning performance [7], [11], [12]. In recent works, a new privacy framework using anonymization has been proposed in the so-called *shuffling model* [13]–[21]. This model enables significantly better privacy-utility performance by amplifying privacy through this anonymization. Another mechanism to amplify privacy is through randomized sub-sampling [11], [22], [23]. This naturally arises in the considered SGD framework, since clients do mini-batch stochastic sampling of local data to compute gradients, and also there is sampling of clients themselves in each iteration, as in the FL framework [2]–[4]. There are several works that studied federated learning under privacy constraints (*e.g.*, [24]–[29] and references therein). These works consider client sampling techniques, such as choosing uniformly at random a *fixed* number of clients at each iteration [25]. Choosing a fixed number of clients at each iteration requires a selection by the shuffler.

In this paper, we extend our work in [25] to explore a distributed self-sampling approach initiated by the clients that does not need a selection by the shuffler. Self-sampling is desirable from a system-level perspective where coordination is not needed in order to randomly sub-sample which clients will participate in each iteration of SGD. At each iteration of the training process, clients independently toss a biased coin. If the biased coin of a client turns a head, that client participates in the current iteration and share its model privately with the untrusted server. One of the main challenges in our self-sampling scheme is that the number of participated clients at each iteration is unknown a priori as it is random varying from iteration to iteration. We analyze the privacy of our self-sampling scheme by composing amplification by sub-sampling along with amplification by shuffling. Furthermore, we analyze the convergence rate of the SGD with client self-sampling and shuffling.

In [29], the authors have proposed a novel sampling scheme called *random check-in*, in which each client

All authors are with the University of California, Los Angeles, USA.
For a full version of this paper, see [1].

independently chooses which time slot to participate in the training process. However, their sampling scheme is different from ours in the following sense: (i) We consider multiple data samples at each client, whereas, in their work they assume that each client has a single sample. This provides an additional layer of sampling the local datasets at clients that amplifies the central privacy of the SGD. Furthermore, this creates non-uniform sampling of data points, because clients either do not participate or they participate with a mini-batch gradient of a certain size. (ii) Our self-sampling scheme allows flexibility to the clients to participate in more than one iteration. In contrast, in [29] each client participates only in one time slot of the training process. These differences also lead to distinct technical approaches to proving privacy and the trade-offs.

II. Preliminaries and Problem Formulation

Preliminaries: We formally define local differential privacy (LDP) and (central) differential privacy (DP).

Definition 1 (Local Differential Privacy - LDP [11])**.** For $\epsilon_0 \geq 0$, a randomized mechanism $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ is said to be ϵ_0-local differentially private (in short, ϵ_0-LDP), if for every pair of inputs $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and $\mathcal{S} \subseteq \mathcal{Y}$, we have

$$\Pr[\mathcal{R}(\boldsymbol{x}) \in \mathcal{S}] \leq \exp(\epsilon_0) \Pr[\mathcal{R}(\boldsymbol{x}') \in \mathcal{S}]. \quad (1)$$

Here, ϵ_0 captures the privacy level, lower the ϵ_0, higher the privacy. Let $\mathcal{D} = \{d_1, \ldots, d_n\}$ denote a dataset comprising n points from $\mathfrak{S}$. We say that two datasets $\mathcal{D} = \{d_1, \ldots, d_n\}$ and $\mathcal{D}' = \{d'_1, \ldots, d'_n\}$ are neighboring if they differ in one data point.

Definition 2 (Central Differential Privacy - DP [30], [31])**.** For $\epsilon, \delta \geq 0$, a randomized mechanism $\mathcal{M} : \mathfrak{S}^n \to \mathcal{Y}$ is said to be (ϵ, δ)-differentially private (in short, (ϵ, δ)-DP), if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathfrak{S}^n$ and every subset $\mathcal{E} \subseteq \mathcal{Y}$, we have

$$\Pr\left[\mathcal{M}\left(\mathcal{D}\right) \in \mathcal{E}\right] \leq \exp(\epsilon) \Pr\left[\mathcal{M}\left(\mathcal{D}'\right) \in \mathcal{E}\right] + \delta. \quad (2)$$

Typically, we are interested in a strong privacy regime in which ϵ is small and $\delta \ll 1/n$.

Problem Formulation: We consider a federated learning (FL) framework [2]–[4] as depicted in Figure 1, where there are m clients, and client i has a local dataset $\mathcal{D}_i = \{d_{i1}, \ldots, d_{ir}\}$ consisting of r data points drawn from a universe $\mathfrak{S}$. Let $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}_i$ denote the entire dataset and $n = mr$ denote the total number of data points in the system. The clients are connected to an untrusted server in order to solve the following empirical risk minimization (ERM) problem

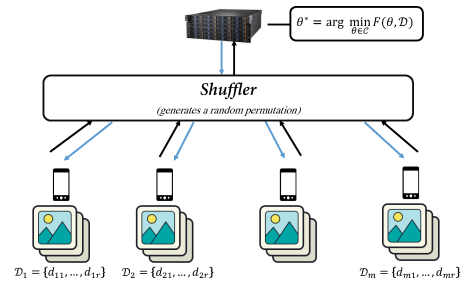

Fig. 1: We have m clients, each having a dataset $\mathcal{D}_i$ of r samples. The clients are connected to a central server, who wants to learn a global model θ^* that minimizes (3).

$$\min_{\theta \in \mathcal{C}} \left(F(\theta, \mathcal{D}) := \frac{1}{m} \sum_{i=1}^{m} F_i(\theta, \mathcal{D}_i) \right). \quad (3)$$

Here, $\mathcal{C} \subset \mathbb{R}^d$ is a closed convex set and $F_i(\theta, \mathcal{D}_i) = \frac{1}{r} \sum_{j=1}^{r} f\left(\theta, d_{ij}\right)$ is a local loss function dependent on the local dataset $\mathcal{D}_i$ at client i evaluated at the model parameters $\theta \in \mathcal{C}$.

Solving the ERM problem (3) in the FL framework introduces several unique challenges, such as the locally residing data $\{\mathcal{D}_i\}$ at all clients need to be kept private and only a small fraction of clients participate in each round of communication. Our goal is to solve (3) while preserving privacy on the training dataset $\mathcal{D}$ and dealing with a dynamic client population in each iteration.

Our Algorithm: In order to solve (3) in the presence of the above-mentioned challenges in the FL setting, we propose distributed self-sampling SGD (dss-SGD), a differentially-private SGD algorithm that works with private updates and dynamic client population. The procedure is described in Algorithm 1.

In any time slot $t \in [T]$ of dss-SGD, each client independently and identically tosses a biased coin with probability q. If the biased coin of the i'th client returns a head (one), then the ith client participates in the current time slot and share its model privately with the untrusted server with the help of the trusted shuffler. Otherwise, the i'th client does not participate in the current time slot. Let $\mathcal{U}_t$ denote the set of participating clients at time $t \in [T]$. Each client $i \in \mathcal{U}_t$ computes the gradient $\nabla_{\theta_t} f\left(\theta_t; d_{ij_i}\right)$ for a randomly chosen sample d_{ij_i} from its local dataset $\mathcal{D}_i$. The i'th client clips the ℓ_p-norm of the gradient $\nabla_{\theta_t} f\left(\theta_t; d_{ij_i}\right)$ and applies the LDP-compression mechanism $\mathcal{R}_p$, where $\mathcal{R}_p : \mathcal{B}_p^d \to \{0,1\}^b$ is an ϵ_0-LDP mechanism when inputs come from an ℓ_p-norm ball. After

that, each client i sends the private gradient $\mathcal{R}_p\left(\mathbf{g}_t\left(d_{ij_i}\right)\right)$ to the secure shuffler that sends a random permutation of the received gradients to the server. Finally, the server takes the average of the received gradients and updates the global model.

Our dss-SGD is different from the CLDP-SGD algorithm proposed in [25] in the client sampling scheme. In [25], a fixed number of clients are chosen uniformly at random in each iteration that requires a selection by the shuffler. While, in dss-SGD, each individual client decides to participate in each iteration depending on independent randomness generated at the client-side. Hence, the proposed self-sampling does not need the coordination with the shuffler that reflects the random availability of the clients in practical FL. This modification in the client sampling raises challenges in analyzing the central privacy of the algorithm as well as analyzing the convergence of the SGD, since the number of clients participating at each iteration is random.

III. Main Results

In this section, we state the privacy guarantees, the communication cost per client, and the privacy-convergence trade-off for the dss-SGD Algorithm. Observe that the probability that an arbitrary data point $d_{ij} \in \mathcal{D}$ is chosen at time $t \in [T]$ is given by $\bar{q} = \frac{q}{r}$. Furthermore, in any time slot $t \in [T]$, since clients participate independently with probability q, the number of clients participating in any time slot $t \in [T]$ is a binomial random variable K_t with mean $\mathbb{E}[K_t] = qm$. Note that $K_t = |\mathcal{U}_t|$.

Let $\mathcal{B}_p(L)$ be the ℓ_p norm ball with radius L, i.e., $\mathcal{B}_p(L) \triangleq \left\{\mathbf{x} \in \mathbb{R}^d \left(\sum_{i=1}^d |x_i|^p\right)^{1/p} \leq L\right\}$. Our dss-SGD algorithm and the result of Theorem 1 (stated below) are given for a general local randomizer $\mathcal{R}_p$ that satisfies the following conditions: (i) The randomized mechanism $\mathcal{R}_p$ is an ϵ_0-LDP mechanism. (ii) $\mathcal{R}_p$ is unbiased, i.e., $\mathbb{E}[\mathcal{R}_p(\mathbf{x})|\mathbf{x}] = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{B}_p(L)$. (iii) The output of $\mathcal{R}_p$ can be represented using $b \in \mathbb{N}^+$ bits. (iv) $\mathcal{R}_p$ has a bounded variance: $\sup_{\mathbf{x}\in\mathcal{B}_p(L)} \mathbb{E}\|\mathcal{R}_p(\mathbf{x}) - \mathbf{x}\|_2^2 \leq cL^2 \max\{d^{2-\frac{2}{p}}, d\}$, where c is a constant.

In [25], we proposed unbiased ϵ_0-LDP mechanisms $\mathcal{R}_p$ for several values of the norm $p \in [1, \infty]$ that require $b = \mathcal{O}(\log(d))$ bits of communication and satisfy the above conditions.

Theorem 1. *Let the set $\mathcal{C}$ be convex with diameter D[1] and the function $f(\theta; .) : \mathcal{C} \to \mathbb{R}$ be convex and L-Lipschitz continuous with respect to the ℓ_g-norm, which is the dual of the ℓ_p-norm[2]. Let $\theta^* = \arg\min_{\theta\in\mathcal{C}} F(\theta)$ denote the minimizer of the problem (3). Let $n = mr$ denote the total number of data points in the dataset $\mathcal{D}$. For participation probability $0 < q \leq 1$, let $\bar{q} = \frac{q}{r}$. If we run Algorithm $\mathcal{A}_{dss}$ for T iterations, then we have*

1) ***Privacy:*** *For $\epsilon_0 = \min\left\{\mathcal{O}(1), \mathcal{O}\left(\sqrt{\frac{n\log(1/\delta')}{\bar{q}T\log(\bar{q}T/\delta')}}\right)\right\}$, where $\delta' > 0$ is an arbitrary, $\mathcal{A}_{dss}$ is (ϵ, δ)-DP, where*

$$\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{\bar{q}T\log(\bar{q}T/\delta')\log(1/\delta')}{n}}\right),$$
$$\delta = 2\delta' + Te^{-c'qm},$$

where $c' \in (0, 1)$ is a constant.

2) ***Communication:*** *Our algorithm $\mathcal{A}_{dss}$ requires $q \times b$ bits of communication in expectation[3] per client per iteration, where expectation is taken with respect to the sampling of clients.*

3) ***Convergence:*** *If we run $\mathcal{A}_{dss}$ with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, where $G^2 = L^2\max\{d^{1-\frac{2}{p}}, 1\}\left(1 + \frac{cd}{\bar{q}n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\right)$, then*

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD\log(T)}{\sqrt{T}}\right.$$

Algorithm 1 $\mathcal{A}_{\text{dss}}$: dss-SGD

Initialize: $\theta_0 \in \mathcal{C}$
for $t \in [T]$ **do**
  Start with an empty set of client $\mathcal{U}_t = \phi$.
  for clients $i \in [m]$ **do**
    if a q-biased coin returns head **then**
      Update $\mathcal{U}_t \leftarrow \mathcal{U}_t \bigcup \{i\}$
      $j_i \xleftarrow{\text{u.a.r}} [r]$.
      $\mathbf{g}_t\left(d_{ij_i}\right) \leftarrow \nabla_{\theta_t} f\left(\theta_t; d_{ij_i}\right)$
      $\tilde{\mathbf{g}}_t\left(d_{ij_i}\right) \leftarrow \frac{\mathbf{g}_t\left(d_{ij_i}\right)}{\max\left\{1, \frac{\|\mathbf{g}_t\left(d_{ij_i}\right)\|_p}{C}\right\}}$
      $\mathbf{q}_t\left(d_{ij_i}\right) \leftarrow \mathcal{R}_p\left(\tilde{\mathbf{g}}_t\left(d_{ij_i}\right)\right)$
      Client i sends $\mathbf{q}_t\left(d_{ij_i}\right)$ to the shuffler.
  The shuffler randomly shuffles the elements in $\{\mathbf{q}_t(d_{ij_i}) : i \in \mathcal{U}_t\}$ and sends them to the server.
  $\overline{\mathbf{g}}_t \leftarrow \frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t} \boldsymbol{q}_t\left(d_{ij_i}\right)$
  $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}\left(\theta_t - \eta_t\overline{\mathbf{g}}_t\right)$, where $\prod_{\mathcal{C}}$ denotes the projection operator onto the set $\mathcal{C}$.
Output: The model θ_T.

[1]Diameter of a bounded set $\mathcal{C} \subset \mathbb{R}^d$ is defined as $\sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}} \|\boldsymbol{x}-\boldsymbol{y}\|$.

[2]For any data point $d \in \mathfrak{S}$, the function $f : \mathcal{C} \to \mathbb{R}$ is L-Lipschitz continuous w.r.t. ℓ_g-norm if for every $\theta_1, \theta_2 \in \mathcal{C}$, we have $|f(\theta_1; d) - f(\theta_2; d)| \leq L\|\theta_1 - \theta_2\|_g$.

[3]A client communicates in an iteration only when its q-biased coin returns a head in that iteration.

$$\max\{d^{\frac{1}{2}-\frac{1}{p}},1\}\sqrt{\frac{cd}{\bar{q}n}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)+e^{-c'qm}\Bigg), \quad (4)$$

where $c=4$ if $p\in\{1,\infty\}$ and $c=14$ otherwise.

A proof of Theorem 1 is presented in Section IV.

Remark 1 (Impact of self-sampling of clients)**.** In our algorithm dss-SGD, the number of clients participating in any time slot $t\in[T]$ is a binomial random variable $K_t=|\mathcal{U}_t|$. The impact of such client sampling appears in the privacy parameter δ that has an additive term $Te^{-c'qm}$. This term does not appear if we choose uniformly at random a fixed number of clients at each time slot (See [25, Theorem 1]). However, in cross-device federated learning [4], the number of participating clients at each time slot is typically in thousands, i.e., qm is equal to a few thousands. Thus, the terms $Te^{-c'qm}\ll 1/n$ and $e^{-c'qm}$ are negligible.

Remark 2 (Optimality of dss-SGD for ℓ_2-norm case)**.** Suppose that our target is to achieve $\epsilon=\mathcal{O}(1)$ and $\delta\ll 1/n$. Substituting $\epsilon_0=\epsilon\sqrt{\frac{n}{\bar{q}T\log(2qT/\delta')\log(2/\delta')}}$, $T=n/\bar{q}$, and $p=2$ in (4), we recover the optimal excess risk of central differential privacy presented in [32], except an additive term $Te^{-c'qm}$ in δ.

IV. Proof of Theorem 1

Privacy: Hereafter, we denote $\mathcal{R}_p$ by $\mathcal{R}$, for simplicity, which is an ϵ_0-LDP mechanism. This implies that the mechanism $\mathcal{A}_{dss}$ guarantees local differential privacy ϵ_0 for each sample d_{ij} per iteration. Thus, it remains to analyze the central DP guarantee of the mechanism $\mathcal{A}_{dss}$ in each iteration and also for the entire execution.

Fix a time slot $t\in[T]$. Let $\mathcal{M}_t(\theta_t,\mathcal{D})$ denote the private mechanism at time t that takes the dataset $\mathcal{D}$ and an auxiliary input θ_t and generates the parameter θ_{t+1} as an output. Let $K_t=|\mathcal{U}_t|$ denote the random variable corresponding to the number of participating clients at the t'th time slot. Thus, the mechanism $\mathcal{M}_t$ on input dataset $\mathcal{D}=\bigcup_{i=1}^m\mathcal{D}_i$ when $K_t>0$ can be defined as:

$$\mathcal{M}_t(\theta_t;\mathcal{D})=\mathcal{H}_{K_t}\circ\mathrm{samp}^{\mathrm{iid}}_{m,q}\left(\mathcal{G}_1,\ldots,\mathcal{G}_m\right), \quad (5)$$

where $\mathcal{G}_i=\mathrm{samp}^{\mathrm{fix}}_{r,1}\left(\mathcal{R}(\boldsymbol{x}^t_{i1}),\ldots,\mathcal{R}(\boldsymbol{x}^t_{ir})\right)$ and $\boldsymbol{x}^t_{ij}=\nabla_{\theta_t}f(\theta_t;d_{ij}),\forall i\in[m],j\in[r]$. Here, $\mathrm{samp}^{\mathrm{iid}}_{m,q}$ denotes the sampling operation for choosing each of the m elements independently with probability q, $\mathrm{samp}^{\mathrm{fix}}_{r,1}$ denotes the sampling operation for choosing uniformly at random a single element from a set of r elements, and $\mathcal{H}_{K_t}$ denotes the shuffling operation on K_t elements, which outputs a random permutation of the K_t input elements. For convenience, in the rest of the proof, we suppress the auxiliary input θ_t and simply denote $\mathcal{M}_t(\theta_t;\mathcal{D})$ by $\mathcal{M}_t(\mathcal{D})$. We can do this because θ_t only affects the gradients, and the analysis in this part is for an arbitrary set of gradients.

In the following lemma, we state the privacy guarantee of the mechanism $\mathcal{M}_t$ for each $t\in[T]$.

Lemma 1. *Fix an arbitrary iteration $t\in[T]$. Let $\overline{q}=\frac{q}{r}$. Suppose $\mathcal{R}$ is an ϵ_0-LDP mechanism with $\epsilon_0=\mathcal{O}(1)$. Then, for any $\tilde{\delta}>0$, the mechanism $\mathcal{M}_t$ is $(\overline{\epsilon},\overline{\delta})$-DP, where $\overline{\epsilon}=\mathcal{O}\left(\epsilon_0\sqrt{\frac{\overline{q}\log(1/\tilde{\delta})}{n}}\right)$ and $\overline{\delta}=\overline{q}\tilde{\delta}+e^{-c'qm}$ for some constant $c'\in(0,1)$.*

We provide a proof sketch of Lemma 1 in Section IV-A, while the detailed proof is presented in Appendix A in the full version [1]. Note that the Algorithm $\mathcal{A}_{dss}$ is a sequence of T adaptive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_T$, where $\mathcal{M}_t$ for each $t\in[T]$ satisfies the privacy guarantee stated in Lemma 1. Now, we invoke the strong composition theorem [31, Theorem 3.20] to obtain the privacy guarantee of the algorithm $\mathcal{A}_{dss}$ and conclude that for any $\delta',\tilde{\delta}>0$, $\mathcal{A}_{dss}$ is (ϵ,δ)-DP for

$$\epsilon=\sqrt{2T\log\left(1/\delta'\right)}\overline{\epsilon}+T\overline{\epsilon}\left(e^{\overline{\epsilon}}-1\right),$$
$$\delta=\overline{q}T\tilde{\delta}+\delta'+Te^{-c'qm},$$

where $\overline{\epsilon}$ is from Lemma 1. Note that when $\overline{\epsilon}=\mathcal{O}\left(\sqrt{\frac{\log(1/\delta')}{T}}\right)$, we have $\epsilon=\mathcal{O}\left(\sqrt{T\log(1/\delta')}\overline{\epsilon}\right)$. When $\epsilon_0=\mathcal{O}(1)$, it follows from Lemma 1 that this condition on $\overline{\epsilon}$ is satisfied when $\epsilon_0=\mathcal{O}\left(\sqrt{\frac{n\log(1/\delta')}{\overline{q}T\log(1/\tilde{\delta})}}\right)$. Together, these conditions imply that when $\epsilon_0=\min\left\{\mathcal{O}(1),\mathcal{O}\left(\sqrt{\frac{n\log(1/\delta')}{\overline{q}T\log(1/\tilde{\delta})}}\right)\right\}$, then (by substituting the bound on $\overline{\epsilon}=\mathcal{O}\left(\epsilon_0\sqrt{\frac{\overline{q}\log\left(1/\tilde{\delta}\right)}{n}}\right)$ from Lemma 1 into the bound on $\epsilon=\mathcal{O}\left(\sqrt{T\log(1/\delta')}\overline{\epsilon}\right)$ above), we get $\epsilon=\mathcal{O}\left(\epsilon_0\sqrt{\frac{\overline{q}T\log\left(1/\tilde{\delta}\right)\log(1/\delta')}{n}}\right)$. By setting $\tilde{\delta}=\frac{\delta'}{\overline{q}T}$, we get $\epsilon_0=\min\left\{\mathcal{O}(1),\mathcal{O}\left(\sqrt{\frac{n\log(1/\delta')}{\overline{q}T\log(\overline{q}T/\delta')}}\right)\right\}$, $\epsilon=\mathcal{O}\left(\epsilon_0\sqrt{\frac{\overline{q}T\log(\overline{q}T/\delta')\log(1/\delta')}{n}}\right)$, and $\delta=2\delta'+Te^{-c'qm}$, where $\delta'>0$ is an arbitrary constant.

Communication: Suppose that the randomized mechanism $\mathcal{R}$ is ϵ_0-LDP having output alphabet $\mathcal{Y}=\{1,2,\ldots,B=2^b\}$. Therefore, the expected number of bits per client in Algorithm $\mathcal{A}_{dss}$ is given by $q\times b$ bits per iteration, where expectation is taken over the client sampling.

Convergence: Note that the number of clients participating $|\mathcal{U}_t|$ at any time slot $t\in[T]$ is a binomial

random variable K_t. At iteration $t \in [T]$ of Algorithm 1 when $K_t > 0$, the server averages the K_t received compressed and privatized gradients and obtains $\overline{\mathbf{g}}_t = \frac{1}{K_t} \sum_{i \in \mathcal{U}_t} \mathbf{q}_t(d_{ij_i})$. Now we show that the average gradient $\overline{\mathbf{g}}_t$ is unbiased:

Claim 1. We have $\mathbb{E}[\overline{\mathbf{g}}_t] = \nabla F(\theta_t)$, where expectation is taken with respect to the random participation of clients, the sampling of data points, and the randomness of the mechanism $\mathcal{R}_p$.

We prove Claim 1 in [1, Appendix B-A]. Now we show that $\overline{\mathbf{g}}_t$ has bounded second moment.

Lemma 2. *For any $d \in \mathfrak{S}$, if the function $f(\theta; .) : \mathcal{C} \to \mathbb{R}$ is convex and L-Lipschitz continuous w.r.t. the ℓ_g-norm, which is the dual of the ℓ_p-norm, then we have*

$$\mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{iid}_{m,q}, \mathcal{R}_p, \\ j_i \in [r], i \in \mathcal{U}_t}} \|\overline{\mathbf{g}}_t\|_2^2 \leq L^2 \max\{d^{1-\frac{2}{p}}, 1\} \left(1 + \frac{2cd}{\bar{q}n} \left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\right) + e^{-c'qm}, \quad (6)$$

where $c' \in (0,1)$ is a constant, and $c = 4$ if $p \in \{1, \infty\}$ and $c = 14$ if $p \in (1, \infty)$. Note that $\bar{q} = \frac{q}{r} mr = qm$.

The proof of Lemma 2 is presented in [1, Appendix B-B]. Although the number of participating clients K_t at each iteration $t \in [T]$ is varying from iteration to iteration, Lemma 2 shows that for $\epsilon_0 = \mathcal{O}(1)$, the second moment of the descent direction $\overline{\mathbf{g}}_t$ decreases with order $\mathcal{O}\left(\frac{d}{qm\epsilon_0^2}\right)$, where $qm = \mathbb{E}[K_t]$. Now, we can use standard SGD convergence results for convex functions. In particular, we use the result from [33], which is stated in Lemma 5 in [1, Appendix B-C]. Using that result (and ignoring the exponentially small term $e^{-c'\bar{q}m}$), we have that the output θ_T of Algorithm 1 satisfies

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD\log(T)\max\{d^{\frac{1}{2}-\frac{1}{p}}, 1\}}{\sqrt{T}} \left(1 + \sqrt{\frac{2cd}{\bar{q}n}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right)\right), \quad (7)$$

where we used the inequality $\sqrt{1 + \frac{2cd}{\bar{q}n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2} \leq \left(1 + \sqrt{\frac{2cd}{\bar{q}n}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right)$. Note that if $\sqrt{\frac{cd}{\bar{q}n}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) \leq \mathcal{O}(1)$, then we recover the convergence rate of vanilla SGD without privacy. So, the interesting case is when $\sqrt{\frac{cd}{\bar{q}n}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) \geq \Omega(1)$, which gives $\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD\log(T)\max\{d^{\frac{1}{2}-\frac{1}{p}},1\}}{\sqrt{T}}\sqrt{\frac{cd}{\bar{q}n}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right)$. This completes the proof of Theorem 1.

A. Proof-Sketch of Lemma 1

In Lemma 1, we are amplifying the privacy by using the subsampling as well as shuffling ideas. Consider two neighboring datasets $\mathcal{D} = \bigcup_{i=1}^m \mathcal{D}_i$, $\mathcal{D}' = \mathcal{D}_1' \bigcup (\bigcup_{i=2}^m \mathcal{D}_i)$ that are different only in the first data point at the first client d_{11}. The first step in the proof is to show that for arbitrary $\tilde{\delta} > 0$,

$$\Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] = \sum_{k=1}^m \Pr[K_t = k] e^{\ln\left(1+\frac{k}{mr}\tilde{\epsilon}(k)\right)} \times \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S} | K_t = k] + \bar{q}\tilde{\delta}, \quad (8)$$

where $\tilde{\epsilon}(k) = \mathcal{O}\left(\epsilon_0\sqrt{\log\left(1/\tilde{\delta}\right)/k}\right)$. The main idea of the proof of (8) is to split the probability distribution of the output of the mechanism $\mathcal{M}_t$ into a summation of three conditional probabilities depending on the event whether the first client is available or not and whether the first client chooses the first data point or not. We use bipartite graphs to get relations between these events, where each vertex corresponds to one of the possible outputs of the sampling procedure, and each edge connects two neighboring vertices. See [1, Appendix A] for more details. Observe that $\tilde{\epsilon}(k)$ is a decreasing function of k. Let $p_k = \Pr[K_t = k]$ and $\mu_k = \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S} | K_t = k]$. Thus, for any $\varepsilon \in (0,1)$, we can bound the first term on the RHS of (8) as follows.

$$\sum_{k=1}^m p_k e^{\ln\left(1+\frac{k}{mr}\tilde{\epsilon}(k)\right)}\mu_k \leq 2e^{\tilde{\epsilon}(1)} \sum_{\substack{k<(1-\varepsilon)qm \\ k>(1+\varepsilon)qm}} p_k + \sum_{k=(1-\varepsilon)qm}^{(1+\varepsilon)qm} p_k\mu_k. \quad (9)$$

Since K_t is a binomial random variable, we use the Chernoff bound to bound the first term of (9) by $e^{-c'qm}$ (by setting $\varepsilon = 0.5$), for some constant $c' \in (0,1)$. Furthermore, the second term in (9) can be bounded by $e^{\bar{\epsilon}} \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}]$, where $\bar{\epsilon} = \mathcal{O}\left(\epsilon_0\sqrt{\frac{\bar{q}}{n}\log(1/\tilde{\delta})}\right)$. Subtituting in (8), we get the following

$$\Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] \leq e^{\bar{\epsilon}} \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] + \bar{q}\tilde{\delta}e^{-c'qm},$$

which completes the proof of Lemma 1.

A detailed proof can be found in [1, Appendix A].

ACKNOWLEDGEMENTS

This work was supported in part by NSF grants #1740047, #2007714, and UC-NL grant LFR-18-548554.

REFERENCES

[1] A. M. Girgis, D. Data, and S. Diggavi, "Differentially private federated learning with shuffling and client self-sampling," 2021, available online on arXiv.

[2] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[5] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[6] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[7] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2013, pp. 429–438.

[8] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," *Information Systems*, vol. 29, no. 4, pp. 343–364, 2004.

[9] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," *arXiv preprint arXiv:1812.00984*, 2018.

[10] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: Simultaneously solving how and what," in *Annual International Cryptology Conference*. Springer, 2008, pp. 451–468.

[11] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[12] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete distribution estimation under local privacy," in *ICML*, 2016, pp. 2436–2444.

[13] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *SODA*. SIAM, 2019, pp. 2468–2479.

[14] B. Ghazi, N. Golowich, R. Kumar, R. Pagh, and A. Velingker, "On the power of multiple anonymous messages." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 1382, 2019.

[15] B. Balle, J. Bell, A. Gascón, and K. Nissim, "Improved summation from shuffling," *arXiv preprint arXiv:1909.11225*, 2019.

[16] B. Ghazi, R. Pagh, and A. Velingker, "Scalable and differentially private distributed aggregation in the shuffled model," *arXiv preprint arXiv:1906.08320*, 2019.

[17] B. Balle, J. Bell, A. Gascon, and K. Nissim, "Differentially private summation with multi-message shuffling," *arXiv preprint arXiv:1906.09116*, 2019.

[18] B. Ghazi, R. Kumar, P. Manurangsi, and R. Pagh, "Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead," in *ICML*, 2020.

[19] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2019, pp. 375–403.

[20] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," in *Annual International Cryptology Conference*. Springer, 2019, pp. 638–667.

[21] B. Balle, J. Bell, A. Gascon, and K. Nissim, "Private summation in the multi-message shuffle model," *arXiv preprint arXiv:2002.00817*, 2020.

[22] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," in *Theory of Cryptography Conference*. Springer, 2010, pp. 437–454.

[23] J. Ullman, "Cs7880. rigorous approaches to data privacy," 2017. [Online]. Available: http://www.ccs.neu.edu/home/jullman/cs7880s17/HW1sol.pdf

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

[25] A. M. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of federated learning: Privacy, communication and accuracy trade-offs," *arXiv preprint arXiv:2008.07180*, 2020.

[26] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, S. Song, K. Talwar, and A. Thakurta, "Encode, shuffle, analyze privacy revisited: formalizations and empirical evaluation," *arXiv preprint arXiv:2001.03618*, 2020.

[27] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpsgd: Communication-efficient and differentially-private distributed sgd," in *Advances in Neural Information Processing Systems*, 2018, pp. 7564–7575.

[28] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of ACM CCS*, 2016, pp. 308–318.

[29] B. Balle, P. Kairouz, H. B. McMahan, O. Thakkar, and A. Thakurta, "Privacy amplification via random check-ins," *arXiv preprint arXiv:2007.06605*, 2020.

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

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

[32] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 2014, pp. 464–473.

[33] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *International conference on machine learning*, 2013, pp. 71–79.

[34] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.