SELF CHECK-IN: TIGHT PRIVACY AMPLIFICATION FOR PRACTICAL DISTRIBUTED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies of distributed computation with formal privacy guarantees, such as differentially private (DP) federated learning, leverage random sampling of clients in each round (privacy amplification by subsampling) to achieve satisfactory levels of privacy. Achieving this however requires precise and uniform subsampling of clients as well as a highly trusted ochestrating server, strong assumptions which may not hold in practice. In this paper, we explore a more practical protocol, self check-in, to resolve the aforementioned issues. The protocol relies on client making independent and random decision to participate in the computation, freeing the requirement of server-initiated subsampling, and enabling robust modelling of client dropouts. Our protocol has immediate application to employing intermediate trust models, i.e., shuffle and distributed DP models, for realizing distributed learning in practice. To this end, we present a novel analysis based on Rényi differential privacy (RDP) that improves in privacy guarantee over those using approximate DP's strong composition at various parameter regimes for self check-in. We also provide a numerical approach to track the privacy of generic shuffling mechanism including distributed learning with Gaussian mechanism, which can be of independent interest as it is the first evaluation of a generic mechanism as far as we know within the local/shuffle model under the distributed setting in the literature. Empirical studies are given to demonstrate the efficacy of learning as well.

1 INTRODUCTION

Cross-device federated learning (FL) or distributed learning is a scalable and privacy-friendly computational framework of large-scale machine learning. In this framework, clients/users send model updates or gradients to the governing server keeping their data decentralized, while the server aggregates the gradients to update the model, and sends the updated model back to the clients to initiate next round of training (Kairouz et al., 2021c; McMahan et al., 2017a). There have been efforts combining FL with differential privacy (DP) (Dwork et al., 2006a;b) to achieve rigorous privacy guarantees (McMahan et al., 2017b). Typically, a subsampling procedure is taken, where in each round of training, only clients sampled in a uniform and random manner by the server participate in the training. Once all sampled clients finish training and sending out the gradients, the server aggregates the gradients to update the model, assuming that the server does not leak information (such as the subsampled clients) other than the model to adversaries (server is trustworthy). This randomness of subsampling leads to privacy amplification, critical at achieving acceptable levels of utility under meaningful DP guarantees (Abadi et al., 2016; Bassily et al., 2014; Kasiviswanathan et al., 2011; Wang et al., 2019). That there is a trustworthy server collecting raw data and adding noises to the aggregated gradients achieves "central" DP for the system.

Such a system may not be easily realizable in practice however. As noted in Balle et al. (2020b); Kairouz et al. (2021b;c), the server-initiated sampling is "impossible in practice". Clients can fail to follow the server's command or drop out due to issues such as network disconnection or battery outage. Typically, the server also discards *stragglers*, or clients that take a much longer time to complete the training due to reasons such as hardware capabilities, as they affect the time of completing a round of training (Bonawitz et al., 2017). Subsequently, the number of participating in each round becomes a *variable*, an effect not taken into account when quantifying the privacy of the system using the standard approach.

In this paper, we explore a novel distributed protocol, **self check-in**, to tackle the aforementioned issues. First, our protocol lets clients utilize their own randomness to decide (with a certain probability) whether to participate in the server-initiated training, without being specifically indexed by the server. This client "autonomy" consequently reduces the system's reliance on the orchestrating server, additionally guaranteeing each client with (weak) local differential privacy (LDP). As explained later, our protocol also allows for the modelling of drop-out as a random (probabilistic) event.

Self check-in is immediately applicable to distributed learning with intermediate trust models, of which we particularly study two intermediate trust models requiring weaker (compared to central DP) trust assumptions, shuffle model and distributed DP [1] Combining self check-in with these trust models leads us to proposing practical protocols in FL, namely *shuffled check-in* and *distributed check-in*, respectively, relaxing the requirements for a highly trusted entity (hence impractical) as in most previous works. In the following we give a brief description of these models.

The shuffle model is responsible for data anonymization (Cheu et al., 2019; Erlingsson et al., 2019) before revealing the collected data to the untrusted analyzer. Deploying a shuffler in practice is simpler implementation-wise and requires fewer trust assumptions compared to an (trusted) aggregator. For example, the Prochlo (Bittau et al., 2017) implementation leverages the trusted hardware to perform shuffling: raw data are not exposed to the shuffler, in contrast to the aggregator, as its sole responsibility is to mask the data origin (shuffling can be performed on encrypted data by, e.g., removing only metadata or identifiers, such that sensitive contents are not exposed to the shuffler). Other realizations of the shuffle model include the utilization of mix-nets (Chaum, 1981; Cheu et al., 2019) and peer-to-peer protocols (Liew et al., 2022), which also do not expose raw data to other entities.

Distributed DP protocols (Kairouz et al., 2021c) attempt to recover some properties of a centralized, highly trusted aggregator in a distributed setting. Here, clients craft LDP reports based on their private data, encrypt them with a certain cryptographic protocol such that individual reports are secure cryptographically, and only the aggregated results are exposed to the untrusted analyzer. While the individual perturbed report is often not meaningful in terms of local DP guarantees, the aggregated noises provide sufficient DP guarantees under this trust model.

Our contributions are as follows.

- We propose *self check-in*, a general privacy amplification technique to address practical issues of distributed systems leveraging clients' randomness. We further propose *shuffled check-in* and *distributed check-in* protocols as concrete realizations of self check-in with intermediate trust models for practical distributed learning.
- We give a detailed privacy analysis of our proposal, particularly utilizing Rényi differential privacy (RDP) to account for the composition of privacy loss. This leads to a tight result compared to conventional approximate DP approaches. Furthermore, we propose a numerical approach of calculating the RDP bound of shuffled check-in with generic local DP randomizer, which can be of independent interest.
- We evaluate our proposal with machine learning tasks under the distributed setting to demonstrate its efficacy and performance against baselines.

Related work. *Shuffle model:* Girgis et al. (2021c) have considered the same shuffled check-in protocol, but there are several distinct differences in the privacy analysis: there, only an order approximation of the privacy accounting based on strong composition of ϵ_0-LDP randomizer is given. In contrast, we give a precise and analytical result based on RDP that leads to tighter accounting, and is extendable to generic (ϵ_0, δ_0)-LDP randomizers. [2] We also provide a comprehensive empirical evaluation that shows significant improvement in budget saving. *Distributed DP:* Bonawitz et al. (2017) showed that cryptographic primitives such as secure aggregation (SecAgg) enables the server to collect aggregated data (perturbed locally) without revealing individual information. While the

[1] Our protocol achieves local DP (LDP) without any additional trust models (Kasiviswanathan et al., 2011). However, it is well-established that LDP leads to substantial utility loss. We pursue the use of intermediate trust models instead to maintain a certain level of utility without relying on a highly trusted entity.

[2] Although there exist methods of converting a (ϵ_0, δ_0)-LDP randomizer to an ϵ_0-LDP one, it is generally loose as the conversion leads to a term contributing to δ proportional to the number of user. See, e.g., Balle et al. (2020b); Cheu et al. (2019).

SecAgg protocol itself takes drop-outs into account, existing privacy analyses typically ignore such effects by considering only fixed number of subsampled users (Agarwal et al., 2018; 2021; Kairouz et al., 2021a; McMahan et al., 2017b). Let us finally remark that client making independent decision to participate in distributed computation has also been considered in Balle et al. (2020b), but our sampling scheme is simpler and does not require strong trust assumptions.

2 Problem Setup and Preliminaries

Problem setup. We consider the task of learning under the distributed setting, where there are n clients each holding a data record x_i for $i \in [n]$ (*user* DP is guaranteed instead of *sample-level* DP when each user holds more than a data record). The whole decentralized dataset is denoted by $D = (x_1, \ldots, x_n)$. The purpose of the system is to train a model with parameter $\theta \in \Theta$ by minimizing a certain loss function $l : \mathcal{D}^n \times \Theta \to \mathbb{R}_+$ via stochastic gradient descent (SGD), while providing clients with formal privacy guarantees.

Two intermediate trust models are considered:

- Shuffle model: Clients send their perturbed gradients to the trusted *shuffler* in which the gradients are shuffled before being forwarded to an untrusted *aggregator*.
- Distributed DP: Clients encrypt their perturbed gradients via certain cryptographic means (e.g., SecAgg) such that only the aggregated result is available to the untrusted *aggregator*.

It is noted that we treat the the shuffler and distributed DP as black boxes guaranteed to execute the protocols faithfully; there are already a myriad of work on the concrete realizations of these trust models (Bell et al., 2020; Bittau et al., 2017; Bonawitz et al., 2017; Liew et al., 2022) and further discussion would require more specific trust assumptions which are out of scope of this work. This simplification allows us to focus on analyzing the privacy guarantees provided by these trust models.

We next present important definitions and known results of differential privacy.

Definition 1 (Central Differential Privacy (Dwork et al., 2014))**.** Given $\epsilon \geq 0$ and $\delta \geq 0$, a randomization mechanism, $\mathcal{M} : \mathcal{D}^n \to \mathcal{S}$ with domain $\mathcal{D}^n$ and range $\mathcal{S}$ satisfies central (ϵ, δ)-differential privacy (DP) if for any two adjacent databases $D, D' \in \mathcal{D}^n$ with n data instances and for any subset of outputs $S \subseteq \mathcal{S}$, the following holds:

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

We say that an (ϵ, δ)-DP mechanism satisfies *approximate* DP, or is (ϵ, δ)-indistinguishable. (ϵ, δ)-DP is also simply referred to as "DP" when the context is clear. Moreover, we mostly work with the "replacement" version of DP, where adjacent databases have one data instance replaced by another data instance.

When D consists of only a single element, the mechanism, also known as a local randomizer, is said to be satisfying local DP:

Definition 2 (Local Differential Privacy (LDP) (Kasiviswanathan et al., 2011))**.** A randomization mechanism $\mathcal{A} : \mathcal{D} \to \mathcal{S}$ satisfies local (ϵ, δ)-DP if for all pairs $x, x' \in \mathcal{D}$, $\mathcal{A}(x)$ and $\mathcal{A}(x')$ are (ϵ, δ)-indistinguishable.

We often refer to a mechanism satisfying $(\epsilon, 0)$ LDP as an ϵ-LDP randomizer. We next introduce Rényi differential privacy (Bun & Steinke, 2016; Dwork & Rothblum, 2016; Mironov, 2017), the main privacy notion used in this paper.

Definition 3 (Rényi Differential Privacy (RDP) (Mironov, 2017))**.** A randomization mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{S}$ is ϵ-Rényi differential privacy of order $\lambda \in (1, \infty)$ (or (λ, ϵ)-RDP), if for any adjacent databases D, $D' \in \mathcal{D}$, the Rényi divergence of order λ between $\mathcal{M}(D)$ and $\mathcal{M}(D')$ is upper-bounded by ϵ:

$$D_\lambda(\mathcal{M}(D)||\mathcal{M}(D')) = \frac{1}{\lambda - 1} \log \left(\mathbb{E}_{\phi \sim \mathcal{M}(D')} \left[\left(\frac{\mathcal{M}(D)(\phi)}{\mathcal{M}(D')(\phi)} \right)^{\lambda} \right] \right) \leq \epsilon, \tag{2}$$

where $\mathcal{M}(\mathcal{D})(\phi)$ denotes $\mathcal{M}$ taking D as input to output ϕ with certain probability.

We sometimes write ϵ as $\epsilon(\lambda)$ to indicate that it is a function of λ. The main strength of RDP lies in its composition property, which is cleaner than approximate DP. The formal description is as follows.

Lemma 1 (Adaptive composition of RDP (Mironov, 2017)). *Given two mechanisms $\mathcal{M}_1, \mathcal{M}_2$ taking $D \in \mathcal{D}$ as input that are (λ, ϵ_1),(λ, ϵ_2)-RDP respectively, the composition of $\mathcal{M}_1$ and $\mathcal{M}_2$ satisfies $(\lambda, \epsilon_1 + \epsilon_2)$-RDP.*

While the privacy accounting involving composition is preferably done in terms of RDP, we often need to convert the RDP notion back to the approximate DP notion in the final step. This is achieved with the following.

Lemma 2 (RDP-to-DP conversion (Balle et al., 2020a; Canonne et al., 2020)). *If a mechanism $\mathcal{M}$ is $(\lambda, \epsilon(\lambda))$-RDP, $\mathcal{M}$ is also (ϵ, δ)-DP, where $1 < \delta < 0$ is arbitrary and ϵ is given by*

$$\epsilon = \min_{\lambda} \left(\epsilon(\lambda) + \frac{\log(1/\delta) + (\lambda - 1)\log(1 - 1/\lambda) - \log(\lambda)}{\lambda - 1} \right). \tag{3}$$

Given an approximate DP mechanism, we may wish to convert it to the RDP to obtain tighter composition-based privacy accounting. The conversion is performed with the following lemma.

Lemma 3 (DP-to-RDP conversion (Asoodeh et al., 2021)). *If a mechanism $\mathcal{M}$ is (ϵ, δ)-DP, $\mathcal{M}$ is also $(\lambda, \epsilon(\lambda))$-RDP, where $\lambda > 1$ and $\epsilon(\lambda)$ is given by*

$$\epsilon(\lambda) = \min_{r \in (\delta, 1)} \left(r^{\lambda}(r - \delta)^{1-\lambda} + (1 - r)^{\lambda}(e^{\epsilon} - r + \delta)^{1-\lambda} \right). \tag{4}$$

3 Self Check-in

3.1 Model-independent protocol and expression

Let us begin by describing the protocol of self check-in. In each round t, a message is broadcast to all clients to ask for participation in the learning. Each client flips a biased coin to decide whether to participate in the training. The probability of a client participating in the training successfully is modeled by a parameter γ ($0 \leq \gamma \leq 1$), which we call the *check-in rate*. The participating probability follows the Bernoulli distribution, Bern(γ). In Section 3.2, we discuss how to determine γ in practice.

Clients decided to participate download the model θ_t, calculate the gradient, apply local randomizer to it (e.g., clip the gradient norm and add Gaussian noise), and send it (encrypted) to a shuffler or aggregator, depending on the trust model. Then, the aggregator updates the model parameters to θ_{t+1} using the messages processed by the shuffler or SecAgg. The underlying (trust) model-dependent mechanism is denoted by *base mechanism*. Next, we give the model-independent expression of the RDP of self check-in.

Theorem 1 (RDP of self check-in). *Let D, D' be adjacent databases consisting of n clients, and γ the (effective) check-in rate. The RDP of order λ of the self check-in mechanism is bounded by*

$$\epsilon(\lambda) \leq \frac{1}{\lambda - 1} \log \left(\sum_{k=0}^{n} \binom{n}{k} \gamma^k (1 - \gamma)^{n-k} \mathbb{E}_{q_k^{\mathrm{s}}} \left[\left(\frac{p_k^{\mathrm{s}}}{q_k^{\mathrm{s}}} \right)^{\lambda} \right] \right). \tag{5}$$

Here, p_k^{s} is the output probability distribution of the subsampled without replacement (of k data instances) base mechanism. q_k^{s} is the output probability distribution induced by the same mechanism but on D'.

Proof. Let us first give the following observation. Under the shuffle model, k data instances perturbed by a LDP randomizer $\mathcal{A} : \mathcal{D} \to \mathcal{S}$ check in to undergo shuffling for $k \in [n]$; under distributed DP, k data instances check in to be processed by, e.g., SecAgg, to return a perturbed aggregated output without exposing individual information, $\mathrm{agg} : \mathcal{D}^k \to \mathcal{S}$ for $k \in [n]$. Notice that there are two output variables for the self check-in mechanism $\mathcal{M}$: $k \in [n]$, the number of data instances, and $\phi \in \mathcal{S}$, the randomized content of each data instance (shuffler) or the aggregated output (distributed DP).

As clients check in independently with a probability of γ, the total number of clients checking in distributes as a binomial distribution. Hence, following the observation given above on the output variables, the neighboring mechanism may be written as

$$\mathcal{M}(\mathcal{D})(k, \phi) \sim \binom{n}{k} \gamma^k (1 - \gamma)^{n-k} p_k^{\mathrm{s}}, \quad \mathcal{M}(\mathcal{D}')(k, \phi) \sim \binom{n}{k} \gamma^k (1 - \gamma)^{n-k} q_k^{\mathrm{s}}.$$

This gives the expression of Equation 5 following the definition of RDP. To complete the proof, we need to show that p_k^{s} is the probability distribution of a subsampled (without replacement) mechanism. For brevity, we write p_k^{s} and q_k^{s} as p_k and q_k respectively. We note that both p_k and q_k are mixture distributions consisting of distribution with (and without) the differing data instance. More precisely, let the differing data instance be the n-th item of D, D'. Let E be the subset containing the n-th item (E^c the subset with the complementary item). Subsequently,

$$p_k = (1-\gamma)p_k(\cdot|E^c) + \gamma p_k(\cdot|E), \quad q_k = (1-\gamma)q_k(\cdot|E^c) + \gamma q_k(\cdot|E)$$

and $p_k(\cdot|E^c) = q_k(\cdot|E^c)$. This expression is by definition the distribution corresponding to the subsampling without replacement scenario (Wang et al., 2019). Plugging the above expressions to Equation 2, we obtain Equation 5 as desired. ■

Remark 1. Theorem 1 states that the self check-in's RDP is effectively composed of the RDP of a subsampled version of the base mechanism weighted by a binomial distribution. As stated before, one can simply plug the model-dependent base mechanism into this model-independent expression to calculate the specific RDP, as will be demonstrated in the next section.

Remark 2. It is recently shown that performing even tighter privacy accounting is possible via privacy loss distributions (which may be seen as a scaled RDP divergence of order 1) and fast Fourier transforms (Gopi et al., 2021; Koskela et al., 2020). These techniques are however limited to distributions that can be reduced to a one-dimensional problem, and therefore not applicable to self check-in as in Equation 5, a high-dimensional mixture distribution of n components. See Section 5 of Koskela et al. (2021), where a similar problem occurs in the shuffle model. It is henceforth more appropriate to attack the privacy composition problem of self check-in with RDP.

3.2 Practical considerations

In our protocol, each client carries a p-biased coin such that she would decide to participate in the training only when head is returned. Even after a client decides to participate, she may *drop out* due to various reasons as discussed in Introduction. Assuming that clients have a certain constant and independent probability of dropping out, p', the *effective* check-in rate is then $\gamma = p(1-p')$. A practitioner can choose to determine the empirical γ (by monitoring the number of client checking in in each round), or conservatively use p as the check-in rate to yield a valid privacy upper bound (as smaller check-in rate leads to larger amplification). These henceforth allow for robust modelling of dropouts. Moreover, instead of treating p' as a constant, one could further better model the dropout rate as, e.g., a time-dependent function, as in reliability engineering (Bazovsky, 2004), but we leave this system-dependent and slightly orthogonal consideration for future work.

4 Self Check-in with intermediate trust models

Using Theorem 1, we next show how to integrate self check-in with intermediate trust models as a realization of private FL.

4.1 RDP for shuffled check-in

Let shuff be the shuffling mechanism that randomly permutes any number of received messages and outputs them. The shuffled check-in mechanism can then be written formally as

$$\mathcal{M}(D) := \text{shuff}(\{\mathcal{A}(x_i)|\sigma_i = 1, i \in [n]\}), \quad \sigma_i \sim \text{Bern}(\gamma). \tag{6}$$

Discrete ϵ_0-LDP randomizer. We first consider shuffling of discrete-value x_i perturbed by an ϵ_0-LDP randomizer, of which has been applied to distributed learning (Bhowmick et al., 2018; Erlingsson et al., 2020; Girgis et al., 2021b). Roughly speaking, the following procedures are taken in these algorithms: clients clip the l_p-norm ($p \in [1, \infty]$) of the gradient, apply an ϵ_0-LDP mechanism to randomize and represent the clipped gradient with a finite number of bits, and send it to the server. We provide the RDP bounds on these types of mechanism, beginning with the upper bound.

Theorem 2 (Upper Bound)**.** *For any $\epsilon_0 \geq 0$, $n \in \mathbb{N}$, $l \leq n$ such that $n\gamma = l$ ($0 \leq \gamma \leq 1$), and any integer $\lambda \geq 2$, the RDP of the shuffled check-in mechanism is upper-bounded by $\epsilon(\lambda) \leq$*

$\frac{1}{\lambda-1}\log\left(\mathbb{E}\left[\left(\frac{\mathcal{M}(D)}{\mathcal{M}(D')}\right)^{\lambda}\right]\right)$, *where*

$$\mathbb{E}\left[\left(\frac{\mathcal{M}(\mathcal{D})}{\mathcal{M}(\mathcal{D}')}\right)^{\lambda}\right] \leq 1 + 4\binom{\lambda}{2}\gamma^2\left(e^{\epsilon_0}-1\right)^2\left(e^{-\epsilon_0-\Delta^2 n\gamma/2} + e^{-\epsilon_0}/\tilde{l}\right) + \sum_{j=3}^{\lambda}\binom{\lambda}{j}\gamma^j j\Gamma\left(j/2\right)\left(\frac{2\left(e^{2\epsilon_0}-1\right)^2}{e^{2\epsilon_0}}\right)^{j/2}\left(e^{-\Delta^2 n\gamma/2} + \tilde{l}^{-j/2}\right) + \Upsilon_1 e^{-\Delta^2 n\gamma/2} + \Upsilon_{(1-\Delta)n\gamma+1}. \tag{7}$$

Here, $0 \leq \Delta \leq 1$ *is arbitrary,* $\tilde{l} = \lfloor\frac{(1-\Delta)n\gamma}{2\epsilon_0}\rfloor + 1$, $\Gamma\left(z\right) = \int_0^\infty x^{z-1}e^{-x}dx$ *is the Gamma function,* Υ_k *is given by* $\Upsilon_k = \left(\left(1+\gamma\frac{e^{2\epsilon_0}-1}{e^{\epsilon_0}}\right)^{\lambda} - 1 - \lambda\gamma\frac{e^{2\epsilon_0}-1}{e^{\epsilon_0}}\right)e^{-\frac{k-1}{8e^{\epsilon_0}}}$.

Proof sketch. Here, we give an outline of the proof. We notice from Equation 5 that inside the logarithm is a summation over k of the expected moments of a subsampled without replacement (with k data instances), shuffle mechanism. We first obtain a bound on the latter using the results from Girgis et al. (2021a). Then, the summation over k is bounded using the properties of the expected moments and the Chernoff bound (see Lemma 4). The full proof is available in Appendix C. ■

Remark 3. Evaluating Equation 5 for shuffled check-in is cumbersome as it involves a summation over the total population, n. We introduce approximation techniques to speed up the evaluation as in Equation 7, which are based on Lemmas 4 and 5 in Appendix C. Similar techniques are applicable to distributed check-in too, as will be shown in the next subsection.

We next provide the corresponding lower bound.

Theorem 3 (Lower Bound)**.** *For any* $\epsilon_0 \geq 0$, $n \in \mathbb{N}$, $l \leq n$ *such that* $0 \leq \gamma \leq 1$, *and any integer* $\lambda \geq 2$, *the RDP of the shuffled check-in mechanism is lower-bounded by*

$$\epsilon(\lambda) \geq \frac{1}{\lambda-1}\log\left(1 + \left(1-e^{-\Delta^2 n\gamma/(2+\Delta)}\right)\binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{(1+\Delta)n\gamma e^{\epsilon_0}}\right), \tag{8}$$

where $0 \leq \Delta \leq 1$ *is arbitrary.*

Proof sketch. The lower bound is calculated assuming that the underlying ϵ_0-LDP discrete mechanism is a binary randomized response. Again, we first obtain the expression of the subsampled without replacement (with k data instances), shuffle binary randomized response. Using the properties of the expression as well as the Chernoff bound (see Lemma 4), we place a lower bound on the RDP. For the full proof, see Appendix C. ■

Generic (ϵ_0, δ_0)-LDP randomizer. The bounds given above apply only to shuffled check-in with discrete ϵ_0-LDP randomizer. Here, we extend our consideration to generic, (ϵ_0, δ_0)-LDP randomizer not limited to discrete ϵ_0-LDP mechanism. One important application of such a consideration is the FL variant of Differentially Private SGD (DP-SGD) (Abadi et al., 2016; McMahan et al., 2017b), where continuous and isotropic Gaussian noises are added to the clipped gradients.

Given an (ϵ_0, δ_0)-LDP randomizer, our strategy is to first convert it to the corresponding RDP parameters using Lemma 3. The RDP parameters are then plugged into Equation 5 to calculate the resulting $\epsilon(\lambda)$. Note that the subsampling and shuffling mechanisms' approximate DP properties are relatively well studied. By converting them into RDP and utilizing Theorem 1, one can calculate the RDP of shuffled check-in in a rather straightforward way. Note also that Equation 4 of Lemma 3 involves a optimization problem that cannot be written in a closed form. Our procedure is henceforth mainly numerical (in contrast to the analytical bounds for ϵ_0-LDP randomizer). We propose two approaches of tackling the problem, but showing only one in the following due to space constraints (another approach can be found in Appendix B).

Subsampled shuffling conversion. Our approach converts the subsampled shuffle mechanism's DP to RDP using the following procedure:

1. Evaluate subsampled shuffle DP.
2. Convert subsampled shuffle DP to subsampled shuffle RDP.
3. Substitute it into Equation 5 to evaluate the composition of RDP.

The approximate DP of subsampled shuffling can be calculated using the result of Feldman et al. (2022) for shuffling and Theorem 9 of Balle et al. (2018) for subsampling, with both having $O(1)$ time complexity. [3] The second step is a calculation of $O(\lambda)$ in terms of time complexity (Wang et al., 2019), while the third step involves evaluating the summation with respect to n as in Equation 5, which is of $O(n)$. Hence, in terms of time complexity with respect to λ, the overall approach takes $O(n\lambda)$. [4]

4.2 RDP FOR DISTRIBUTED CHECK-IN

Distributed DP is perhaps the most studied method for realizing DP within the distributed setting. While various approaches exist (Agarwal et al., 2018; 2021; Bao et al., 2022; Kairouz et al., 2021a), most of them employ the Gaussian mechanism as the baseline. We henceforth analyze the use of Gaussian mechanism as the underlying randomization mechanism of the distributed check-in protocol for the ease of comparison (other mechanisms may be analyzed analogously). Under distributed DP, we emphasize that only the aggregated values are exposed to the adversary.

Formally, we consider each client randomizing her data instance x_i of dimension d with isotropic Gaussian noise of variance σ^2: $\tilde{x}_i = x_i + \mathcal{N}(0, \sigma^2 I_d)$. Here, w.l.o.g., the global sensitivity of x_i is assumed to be 1 (i.e., the clipping size C is set to 1). The distributed check-in mechanism can be written as

$$\mathcal{M}(D) := \text{agg}(\{\mathcal{A}(x_i)|\sigma_i = 1, i \in [n]\}), \quad \sigma_i \sim \text{Bern}(\gamma). \tag{9}$$

Next, we present the distributed check-in Gaussian RDP defined under replacement DP.

Theorem 4 (Distributed check-in Gaussian RDP (replacement DP))**.** *Consider the mechanism in Equation 9 where agg is a mean operation on collected values. Then, the RDP is*

$$\epsilon(\lambda) \leq \frac{1}{\lambda - 1} \log \left(\sum_{k=1}^{n} \binom{n}{k} \gamma^k (1-\gamma)^{n-k} e^{(\lambda-1)\epsilon^{\text{SG}}_{\gamma,k}} \right) \tag{10}$$

where ϵ^{SG} *is*

$$\epsilon^{\text{SG}}_{\gamma,m}(\lambda) \leq \frac{1}{\lambda - 1} \log \left(1 + \gamma^2 \binom{\lambda}{2} \min \left\{ 4(e^{4/(m\sigma^2)} - 1), 2e^{4/(m\sigma^2)} \right\} + \sum_{j=3}^{\lambda} 2\gamma^j \binom{\lambda}{j} e^{2j(j-1)/(m\sigma^2)} \right)$$

The expression may be further approximated by

$$\epsilon(\lambda) \leq \frac{1}{\lambda - 1} \log \left(e^{(\lambda-1)\epsilon^{\text{SG}}_{\gamma,1} - \Delta^2 n\gamma/2} + e^{(\lambda-1)\epsilon^{\text{SG}}_{\gamma,(1-\Delta)n\gamma+1}} \right) \tag{11}$$

with $\Delta \in [0, 1]$.

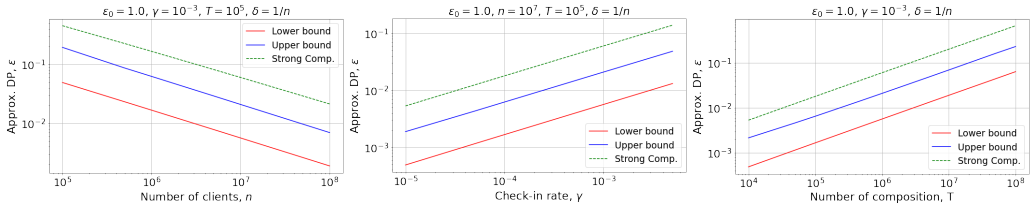
Proof. At the server side, assume k instances are received at certain round. The server aggregates the data and outputs $\frac{1}{k}(\sum_{i=1}^{k} x_i + \mathcal{N}(0, k\sigma^2)) = \frac{1}{k}\sum_{i=1}^{k} x_i + \mathcal{N}(0, \sigma^2/k)$. This mechanism has global sensitivity $2/k$ under replacement DP. Hence, we can consider w.l.o.g. the following two base mechanisms with neighboring databases

$$\mathcal{M}_b \sim \mathcal{N}(0, \sigma^2/k), \;\; \mathcal{M'}_b \sim \mathcal{N}(2/k, \sigma^2/k).$$

The RDP of the base mechanism is therefore $\epsilon = \frac{2\lambda}{k\sigma^2}$ (Mironov, 2017). Applying Lemma 1 and the subsampling lemma of Wang et al. (2019) leads to Equation 10. To obtain Equation 11, one uses monotonicity of the expression and Lemma 4 in Appendix C. ■

[3] The result of privacy amplification by shuffling by Feldman et al. (2022) is valid only when $\epsilon_0 \leq \log(n/16\log(2/\delta))$. For parameters violating this condition, we assume that no amplification occurs, i.e., $\epsilon = \epsilon_0$. This is well corroborated by the numerical experiments performed in Feldman et al. (2022).

[4] Here, we do not include the time complexity of optimizing Equation 4 for convenience as it is irrelevant to the subsequent discussions. We note however that Equation 4 is convex optimization problem which can be solved efficiently.



(a) Approx. DP versus n for $\epsilon_0 = 1, \gamma = 10^{-3}, T = 10^5, \delta = 1/n$

(b) Approx. DP versus γ for $\epsilon_0 = 1, n = 10^7, T = 10^5, \delta = 1/n$

(c) Approx. DP versus T for $n = 10^7, \gamma = 10^{-3}, \delta = 1/n$

Figure 1: Comparison of several bounds on the approximate DP of the shuffled check-in mechanism. We compare the upper (blue) and lower (red) bound of our formulation (RDP) with the bound based on strong composition (Girgis et al., 2021c) (green dashed).

Remark 4. Theorem 4 is derived under replacement DP. Generally speaking, removal DP (where the adjacent database has one data instance added/removed) provides tighter bound. To derive distributed check-in under removal DP however requires an additional trust assumption: the number of check-ins must be hidden from the adversary (otherwise, the adversary could distinguish between the neighboring databases by simply counting the number of check-ins). We provide the result for removal DP in Appendix B.

5 Numerical Results

In this section, we present numerical evaluation results of our proposal to demonstrate the performance of the RDP bounds. We also apply our techniques to training machine learning models under the distributed setting. A link to our code can be found in Appendix A.

5.1 RDP versus approximate DP

Shuffled check-in. In Figure 1, we plot the approximate DP bounds of shuffled check-in in various parameter regimes. The approximate DP is calculated by solving for the optimal λ in Equation 3. We mainly compare with Girgis et al. (2021c), which uses strong composition for privacy accounting. As can be seen in the figure, our upper bound is approximately 3 times tighter than that of accounting using strong compostion, and our lower bound is 10 times tighter.

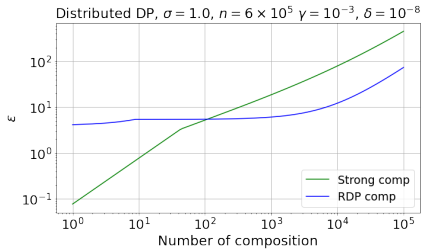
Distributed check-in. To make comparison with RDP-based privacy accounting, we first invent a method of privacy accounting of distributed check-in based on approximate DP and strong composition.

We first find two high probability bounds $1 - \delta_1, 1 - \delta_2$ such that the number of clients checking in is higher than l_1, and lower than l_2, respectively. Then, with probability $1 - \delta_1$, the aggregated value is conservatively estimated to be perturbed with Gaussian noise of variance $l_1\sigma^2$. Moreover, with $1 - \delta_2$, the subsampling rate is l_2/n. Let the resulting DP satisfying the above condition be (ϵ_3, δ_3). Then, the overall DP can be written as $(\epsilon_3, \delta_1 + \delta_2 + \delta_3)$.

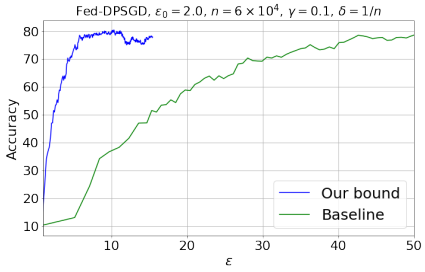
We set $n = 6 \times 10^5$, $\sigma = 1$, $\gamma = 10^{-3}$, $\delta = 10^{-8}$, following Wang et al. (2019). The result as shown in Figure 2a is that, similar to those observed previously (Wang et al., 2019), strong composition leads to tighter bounds initially. However, as the number of composition increases (typical in FL applications), the RDP bounds can be an order of magnitude tighter.

5.2 Private distributed learning

We perform experiments of distributed machine learning tasks to evaluate our proposal's empirical performance. The MNIST handwritten digit dataset (LeCun et al., 1998) is used, assuming that each client is holding one data instance. A convolutional neural network with architecture similar to the one used in Erlingsson et al. (2020) is employed (see Appendix D) as the model to be trained privately. Under the shuffle model, we evaluate two learning algorithms, *LDP-SGD* and *Federated DP-SGD*. Under distributed DP, we perform an evaluation with the vanilla (distributed) Gaussian mechanism.



(a) RDP vs. approximate DP (distributed DP)



(b) Federated DP-SGD

Figure 2: Distributed check-in's ϵ with respect to the number of composition (left), and privacy-utility trade-off of private distributed learning with the Federated DP-SGD algorithms on the MNIST dataset (right). Our approaches (blue) perform better compared to baseline approaches (green).

Due to space constraints, we here present only results of federated learning with Federated DP-SGD (shuffled check-in Gaussian mechanism), relegating other results (including a study of hyperparameter dependence, evaluation with CIFAR10) to Appendix D.

Federated DP-SGD. We adapt DP-SGD (Abadi et al., 2016; Bassily et al., 2014; Song et al., 2013) to distributed learning. Here, the Gaussian mechanism is applied to each client's clipped gradient to achieve (ϵ_0, δ_0)-LDP. Let us remark that, despite its wide usage (under the central DP setting) and simplicity, we do not realize any existing work using the Gaussian mechanism in the LDP/shuffle model literature, possibly due to the difficulty of evaluating an (ϵ_0, δ_0)-LDP randomizer. Hence, we believe that our evaluation is of independent interest as well. One fact worth mentioning for the adaptation to the distributed setting is that as we are working with replacement DP, the sensitivity should be set to twice of those given in Abadi et al. (2016), which is defined in terms of addition/removal DP (Dwork et al., 2006b; Vadhan, 2017).

Baseline method. Before proceeding to present the experiment, we devise a baseline method of privacy accounting without using Theorem 1 to make comparisons with our proposal. To do so, we modify existing techniques of subsampling and shuffling to perform the accounting, similar to the one given in Section 5.1 (see Appendix D for the full description).

Experimental details and results. We evaluate the resulting DP using the approach introduced at the end of Section 4.1 and the above baseline. Two experiments are performed. In the first experiment, the parameters are set to be similar to the one in the evaluation using LDP-SGD: $\epsilon_0 = 2$, $n = 60,000$, $\gamma = 0.1$, $C = 0.05$ and the experiment is run for 5,540 rounds. The result is shown in Figure 2b, showing that our proposed accounting is much tighter. In the second experiment, we set the parameters of the experiment as follows: $\epsilon_0 = 8$, $n = 10^7$ (by bootstrapping from the train dataset), $\gamma = 10^{-4}$, $C = 0.05$. Under this setting, we reach around 90% of test accuracy after running the training for 2,000 rounds. An accuracy-round curve and other details can be found in Appendix D. We obtain the final ϵ as 0.67 with our approach, in contrast with accounting with the baseline method presented above, which yields $\epsilon = 0.80$. This again demonstrates the effectiveness of our approach. As a side note, we find that in general Federated DP-SGD performs better at large ϵ_0.

6 Conclusion

This paper attempts to give a privacy analysis of distributed learning under a more practical setting by taking client "autonomy" into account, as well as utilizing intermediate trust models. While we are making a step towards privacy amplification under a more realistic setting, our protocol has by no means resolved all problems in privacy amplification applied to practical distributed learning. Some of them include the impact of violation of trust assumptions on the shuffler, and malicious clients on distributed learning. We hope that this work can spur the study of privacy amplification with practical distributed protocols in mind within the research community.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016.

Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

Naman Agarwal, Peter Kairouz, and Ziyu Liu. The skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems*, 34:5052–5064, 2021.

Shahab Asoodeh, Jiachun Liao, Flavio P Calmon, Oliver Kosut, and Lalitha Sankar. Three variants of differential privacy: Lossless conversion and applications. *IEEE Journal on Selected Areas in Information Theory*, 2(1):208–222, 2021.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in Neural Information Processing Systems*, 31, 2018.

Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 2496–2506. PMLR, 2020a.

Borja Balle, Peter Kairouz, Brendan McMahan, Om Thakkar, and Abhradeep Guha Thakurta. Privacy amplification via random check-ins. *Advances in Neural Information Processing Systems*, 33: 4623–4634, 2020b.

Ergute Bao, Yizheng Zhu, Xiaokui Xiao, Yin Yang, Beng Chin Ooi, Benjamin Tan, and Khin Mi Mi Aung. Skellam mixture mechanism: a novel approach to federated learning with differential privacy. *Proc. VLDB Endow.*, 15(11):2348–2360, 2022. URL `https://www.vldb.org/pvldb/vol15/p2348-bao.pdf`.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

Igor Bazovsky. *Reliability theory and practice*. Courier Corporation, 2004.

James Henry Bell, Kallista A Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1253–1269, 2020.

Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 441–459, 2017.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658. Springer, 2016.

Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020.

David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 375–403. Springer, 2019.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Eurocrypt*, volume 4004, pp. 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006b.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479. SIAM, 2019.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618*, 2020.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 954–964. IEEE, 2022.

Antonious Girgis, Deepesh Data, and Suhas Diggavi. Renyi differential privacy of the subsampled shuffle model in distributed learning. *Advances in Neural Information Processing Systems*, 34, 2021a.

Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2521–2529. PMLR, 2021b.

Antonious M Girgis, Deepesh Data, and Suhas Diggavi. Differentially private federated learning with shuffling and client self-sampling. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 338–343. IEEE, 2021c.

Antonious M Girgis, Deepesh Data, Suhas Diggavi, Ananda Theertha Suresh, and Peter Kairouz. On the renyi differential privacy of the shuffle model. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2321–2341, 2021d.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pp. 1376–1385. PMLR, 2015.

Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pp. 5201–5212. PMLR, 2021a.

Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pp. 5213–5225. PMLR, 2021b.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021c.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*, pp. 2560–2569. PMLR, 2020.

Antti Koskela, Mikko A Heikkilä, and Antti Honkela. Tight accounting in the shuffle model of differential privacy. *arXiv preprint arXiv:2106.00477*, 2021.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Seng Pei Liew, Tsubasa Takahashi, Shun Takagi, Fumiyuki Kato, Yang Cao, and Masatoshi Yoshikawa. Network shuffling: Privacy amplification via random walks. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, pp. 773–787, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392495. doi: 10.1145/3514221.3526162. URL https://doi.org/10.1145/3514221.3526162.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017a.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017b.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248. IEEE, 2013.

Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pp. 347–450. Springer, 2017.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1226–1235. PMLR, 2019.

Yuqing Zhu and Yu-Xiang Wang. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, pp. 7634–7642. PMLR, 2019.