



**HAL**  
open science

# Privacy Amplification by Decentralization

Edwige Cyffers, Aurélien Bellet

► **To cite this version:**

| Edwige Cyffers, Aurélien Bellet. Privacy Amplification by Decentralization. 2020. hal-03100005v3

**HAL Id: hal-03100005**

**<https://hal.inria.fr/hal-03100005v3>**

Preprint submitted on 17 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Privacy Amplification by Decentralization

---

Edwige Cyffers  
Inria, France

Aurélien Bellet  
Inria, France

## Abstract

Analyzing data owned by several parties while achieving a good trade-off between utility and privacy is a key challenge in federated learning and analytics. In this work, we introduce a novel relaxation of local differential privacy (LDP) that naturally arises in fully decentralized algorithms, i.e., when participants exchange information by communicating along the edges of a network graph without central coordinator. This relaxation, that we call network DP, captures the fact that users have only a local view of the system. To show the relevance of network DP, we study a decentralized model of computation where a token performs a walk on the network graph and is updated sequentially by the party who receives it. For tasks such as real summation, histogram computation and optimization with gradient descent, we propose simple algorithms on ring and complete topologies. We prove that the privacy-utility trade-offs of our algorithms under network DP significantly improve upon what is achievable under LDP (sometimes even matching the utility of the trusted curator model), showing for the first time that formal privacy gains can be obtained from full decentralization. Our experiments illustrate the improved utility of our approach for decentralized training with stochastic gradient descent.

## 1 INTRODUCTION

With growing public awareness and regulations on data privacy, machine learning and data analytics are starting to transition from the classic centralized approach, where a “curator” is trusted to store and analyze raw data, to more decentralized paradigms. This shift is illustrated by the rise of federated learning (FL) [32], in which each data subject/provider keeps her/his own data and only shares results of local computations. In this work, we are interested specifically in

*fully decentralized* FL algorithms that do not require a central coordinator and instead rely on peer-to-peer exchanges along edges of a network graph, see e.g. [37, 18, 52, 38, 51, 11, 45, 35] for recent work and [32] (Section 2.1 therein) for an overview. Fully decentralized approaches are usually motivated by efficiency and scalability concerns: while a central coordinator can become a bottleneck when dealing with the large number of participants commonly seen in “cross-device” applications [32], in fully decentralized methods each participant can communicate with only a small number of peers at each step [37, 38, 44, 45].

In many applications involving personal or otherwise confidential information, the participants want to keep their raw data private from other parties involved in the FL process. Unfortunately, it is by now well documented that the results of local computations (such as the parameters of a machine learning model) can leak a lot of information about the data [50]. In fact, FL provides an additional attack surface as the participants share intermediate updates [43, 27]. To control the privacy leakage, the prominent approach is based on the standard notion of Differential Privacy (DP) [21]. DP typically requires to randomly perturb the results of computations before sharing them. This leads to a trade-off between privacy and utility which is ruled by the magnitude of the random perturbations.

**Related work.** Several trust models can be considered in FL, leading to different privacy-utility trade-offs. The strongest model is local differential privacy (LDP) [34, 19], where each participant (user) does not trust anyone and aims to protect against an adversary that can observe everything that she/he shares. In LDP, random perturbations are performed locally by each user, making it convenient to design private versions of fully decentralized algorithms in this model (see e.g., [29, 11, 36, 16, 55, 54]). Unfortunately, LDP comes at a great cost in utility: for real summation with  $n$  users, the best possible error under LDP is a factor  $\sqrt{n}$  larger than in the centralized (trusted curator) model of DP [14]. The fundamental limits of machine learning under LDP have been studied in [56, 53].

The limitations of LDP have motivated the study of

intermediate trust models, where LDP is relaxed so as to obtain better utility while still avoiding the need for a trusted curator. A popular approach is to resort to cryptographic primitives to securely aggregate user contributions [20, 49, 12, 15, 30, 9, 47] or to securely shuffle the set of user messages so as to hide their source [17, 24, 7, 6, 28, 25]. At the cost of additional computation/communication overhead, these relaxations can provably lead to significant improvements in the privacy-utility trade-off (sometimes matching the trusted curator model). However, they require all users to interact with each other at each step and/or rely on a central coordinator. These solutions thus appear to be incompatible with full decentralization.

A related line of work has studied mechanisms that “amplify” the DP guarantees of a private algorithm. Beyond privacy amplification by shuffling [24, 7, 25] (based on the shuffling primitive mentioned above), we can cite amplification by subsampling [5] and amplification by iteration [26]. These schemes are generally difficult to leverage in a federated/decentralized setting: the former requires that the identity of subsampled participants remain secret, while the latter assumes that only the final result is revealed.

**Our contributions.** In this work, we propose a *novel relaxation of LDP where users have only a local view of the decentralized system*, which is a natural assumption in fully decentralized settings. This relaxation, called *network differential privacy*, effectively captures the fact that each user only observes information received from her/his neighbors in the network graph. Network DP can also account for potential collusions between users. We initiate the study of algorithms under network DP in a decentralized model of computation where a token containing the current estimate performs a walk on the network graph and is updated sequentially by the user who receives it. This model has been studied in previous work as a way to perform (non-private) decentralized estimation and optimization with less communication and computation overhead than algorithms that require all users to communicate with their neighbors at each step [46, 31, 39, 3].

We start by analyzing the case of a (deterministic) walk over a directed ring for the tasks of computing real summations and discrete histograms. In both cases, we propose simple algorithms which achieve a privacy gain of  $O(1/\sqrt{n})$  compared to LDP, thereby *matching the privacy-utility trade-off of a trusted aggregator* without relying on any costly secure multi-party computation protocol. Noting that the ring topology is not very robust to collusions, we then consider the case of random walks over a complete graph. We provide an algorithm for real summation and prove a privacy amplification result of  $O(1/n^{1/3})$  compared to the same algorithm an-

alyzed under LDP. We also discuss a natural extension for computing discrete histograms. Finally, we turn to the task of optimization with stochastic gradient descent and propose a decentralized SGD algorithm that achieves a privacy amplification of  $O(\ln n/\sqrt{n})$  in some regimes, nearly matching the utility of *centralized private SGD*. Interestingly, the above algorithms can tolerate a constant number of collusions at the cost of some reduction in the privacy amplification effect. At the technical level, our theoretical analysis leverages recent results on privacy amplification by subsampling [5], shuffling [24, 7, 25] and iteration [26] in a novel decentralized context: this is made possible by the restricted view of participants offered by decentralized algorithms and adequately captured by our notion of network DP. At the empirical level, we show through experiments that privacy gains are significant in practice both for simple analytics and for training models in federated learning scenarios.

To the best of our knowledge, our work is the first to show that *formal privacy gains can be naturally obtained from full decentralization* (i.e., from having no central coordinator). Our results imply that the true privacy guarantees of some fully decentralized algorithms have been largely underestimated, providing a new incentive for using such approaches beyond the usual motivation of scalability. We believe that our work opens several promising perspectives, which we outline in the conclusion.

**Paper outline.** The paper is organized as follows. Section 2 introduces our notion of network DP and the decentralized model of computation that we study. Section 3 focuses on the case of a fixed ring topology, while Section 4 considers random walks on a complete graph. We present some numerical results in Section 5 and draw some perspectives for future work in Section 6.

## 2 NETWORK DP AND DECENTRALIZED MODEL

Let  $V = \{1, \dots, n\}$  be a set of  $n$  users (or parties), which are assumed to be honest-but-curious (i.e., they truthfully follow the protocol). Each user  $u$  holds a private dataset  $D_u$ , which we keep abstract at this point. We denote by  $D = D_1 \cup \dots \cup D_n$  the union of all user datasets, and by  $D \sim_u D'$  the fact that datasets  $D$  and  $D'$  of same size differ only on user  $u$ 's data. This defines a *neighboring relation* over datasets which is sometimes referred to as user-level DP [41]. This relation is weaker than the one used in classic DP and will thus provide stronger privacy guarantees. Indeed, it seeks to hide the influence of a *user's whole dataset* rather than a single of its data points.

We consider a fully decentralized setting, in which users are nodes in a network graph  $G = (V, E)$  and an edge  $(u, v) \in E$  indicates that user  $u$  can send messages to user  $v$ . The graph may be directed or undirected, and could in principle change over time although we will restrict our attention to fixed topologies. For the purpose of quantifying privacy guarantees, a decentralized algorithm  $\mathcal{A}$  will be viewed as a (randomized) mapping which takes as input a dataset  $D$  and outputs the transcript of all messages exchanged between users over the network. We denote the (random) output in an abstract manner by  $\mathcal{A}(D) = ((u, m, v) : \text{user } u \text{ sent message with content } m \text{ to user } v)$ .

**Network DP.** The key idea of our new relaxation of LDP is to consider that *a given user does not have access to the full transcript  $\mathcal{A}(D)$  but only to the messages she/he is involved in* (this can be enforced by the use of secure communication channels). We denote the corresponding view of a user  $u$  by

$$\mathcal{O}_u(\mathcal{A}(D)) = ((v, m, v') \in \mathcal{A}(D) : v = u \text{ or } v' = u). \quad (1)$$

**Definition 1** (Network Differential Privacy). *An algorithm  $\mathcal{A}$  satisfies  $(\varepsilon, \delta)$ -network DP if for all pairs of distinct users  $u, v \in V$  and all pairs of neighboring datasets  $D \sim_u D'$ , we have:*

$$\mathbb{P}(\mathcal{O}_v(\mathcal{A}(D))) \leq e^\varepsilon \mathbb{P}(\mathcal{O}_v(\mathcal{A}(D'))) + \delta. \quad (2)$$

Network DP essentially requires that for any two users  $u$  and  $v$ , the information gathered by user  $v$  during the execution of  $\mathcal{A}$  should not depend too much on user  $u$ 's data. Network DP can be thought of as analyzing the composition of the operator  $\mathcal{O}_v$  with the algorithm  $\mathcal{A}$ . The hope is that in some cases  $\mathcal{O}_v \circ \mathcal{A}$  is more private than  $\mathcal{A}$ : in other words, that applying  $\mathcal{O}_v$  *amplifies* the privacy guarantees of  $\mathcal{A}$ . Note that if  $\mathcal{O}_v$  is the identity map (i.e., if each user is able to observe all messages), then Eq. 2 boils down to local DP.

We can naturally extend Definition 1 to account for potential *collusions* between users. As common in the literature, we assume an upper bound  $c$  on the number of users that can possibly collude. The identity of colluders is however unknown to other users. In this setting, we would like to be private with respect to the aggregated information  $\mathcal{O}_{V'} = \cup_{v \in V'} \mathcal{O}_v$  acquired by any possible subset  $V'$  of  $c$  users, as captured by the following generalization of Definition 1.

**Definition 2** (Network DP with collusions). *An algorithm  $\mathcal{A}$  is  $(c, \varepsilon, \delta)$ -network DP if for each user  $u$ , all subsets  $V' \subset V$  such that  $|V'| \leq c$ , and all pairs of neighboring datasets  $D \sim_u D'$ , we have:*

$$\mathbb{P}(\mathcal{O}_{V'}(\mathcal{A}(D))) \leq e^\varepsilon \mathbb{P}(\mathcal{O}_{V'}(\mathcal{A}(D'))) + \delta. \quad (3)$$

**Decentralized computation model.** In this work, we study network DP for decentralized algorithms that perform computations via sequential updates to a *token*  $\tau$  walking through the nodes by following the edges of the graph  $G$ . At each step, the token  $\tau$  resides at some node  $u$  and is updated by

$$\tau \leftarrow \tau + x_u^k, \quad \text{with } x_u^k = g^k(\tau; D_u), \quad (4)$$

where  $x_u^k = g^k(\tau; D_u)$  denotes the contribution of user  $u$ . The notation highlights the fact that this contribution may depend on the current value  $\tau$  of the token as well as on the number of times  $k$  that the token visited  $u$  so far. The token  $\tau$  is then sent to another user  $v$  for which  $(u, v) \in E$ .

Provided that the walk follows some properties (e.g., corresponds to a deterministic cycle or a random walk that is suitably ergodic), this model of computation allows to optimize sums of local cost functions using (stochastic) gradient descent [46, 31, 39, 3] (sometimes referred to as incremental gradient methods) and hence to train machine learning models. In this case, the token  $\tau$  holds the model parameters and  $x_u^k$  is a (stochastic) gradient of the local loss function of user  $u$  evaluated at  $\tau$ . Such decentralized algorithms can also be used to compute summaries of the users' data, for instance any commutative and associative operation like sums/averages and discrete histograms. In these cases, the contributions of a given user may correspond to different values acquired over time, such as power consumption in smart metering or item ratings in collaborative filtering applications.

### 3 WALK ON A RING

In this section, we start by analyzing a simple special case where the graph is a directed ring, i.e.,  $E = \{(u, u+1)\}_{u=1}^{n-1} \cup \{(n, 1)\}$ . The token starts at user 1 and goes through the ring  $K$  times. The ring (i.e., ordering of the nodes) is assumed to be public.

#### 3.1 Real Summation

We first consider the task of estimating the sum  $\bar{x} = \sum_{u=1}^n \sum_{k=1}^K x_u^k$  where the  $x$ 's are bounded real numbers and  $x_u^k$  represents the contribution of user  $u$  at round  $k$ . For this problem, the standard approach in local DP is to add random noise to each single contribution before releasing it. For generality, we consider an abstract mechanism  $\text{Perturb}(x; \sigma)$  which adds centered noise with standard deviation  $\sigma$  to the contribution  $x$  (e.g., the Gaussian or Laplace mechanism). Let  $\sigma_{loc}$  be the standard deviation of the noise required so that  $\text{Perturb}(\cdot; \sigma_{loc})$  satisfies  $(\varepsilon, \delta)$ -LDP.

Consider now the simple decentralized protocol in Algo-

---

**Algorithm 1** Private real summation on the ring.
 

---

```

1:  $\tau \leftarrow 0; a \leftarrow 0$ 
2: for  $k = 1$  to  $K$  do
3:   for  $u = 1$  to  $n$  do
4:     if  $a = 0$  then
5:        $\tau \leftarrow \tau + \text{Perturb}(x_u^k; \sigma_{loc})$ 
6:        $a = n - 2$ 
7:     else
8:        $\tau \leftarrow \tau + x_u^k; a \leftarrow a - 1$ 
9: return  $\tau$ 
    
```

---

rithm 1, where noise with the same standard deviation  $\sigma_{loc}$  is added *only once every  $n - 1$  hops of the token*. By leveraging the fact that the view of each user  $u$  is restricted to the values taken by the token at each of its  $K$  visits to  $u$ , combined with advanced composition [23], we have the following result (see supplementary material for the proof).

**Theorem 1.** *Let  $\varepsilon, \delta > 0$ . Algorithm 1 outputs an unbiased estimate of  $\bar{x}$  with standard deviation  $\sqrt{[Kn/(n-1)]\sigma_{loc}}$ , and is  $(\sqrt{2K \ln(1/\delta')}\varepsilon + K\varepsilon(e^\varepsilon - 1), K\delta + \delta')$ -network DP for any  $\delta' > 0$ .*

To match the same privacy guarantees, LDP incurs a standard deviation of  $\sqrt{Kn}\sigma_{loc}$ . Therefore, Algorithm 1 provides an  $O(1/\sqrt{n})$  reduction in error or, equivalently, an  $O(1/\sqrt{n})$  gain in  $\varepsilon$ . In fact, Algorithm 1 achieves the same privacy-utility trade-off as a *trusted central aggregator* that would iteratively aggregate the raw contributions of all users at each round  $k$  and perturb the result before sending it to the users, as done in federated learning algorithms with a trusted server [32].

**Remark 1.** *We can design variants of Algorithm 1 in which noise addition is distributed across users. Using the Gaussian mechanism, each user can add noise with std. dev.  $\sigma'_{loc} = \sigma_{loc}/\sqrt{n}$ , except for the very first contribution which requires std. dev.  $\sigma_{loc}$  to properly hide the contributions of users in the first cycle. The total added noise has std. dev.  $\sqrt{[Kn/(n-1)] + 1}\sigma_{loc}$ , leading to same utility as Algorithm 1 (up to a constant factor that is negligible when  $K$  is large).*

### 3.2 Discrete Histogram Computation

We now turn to the computation of histograms over a discrete domain  $[L] = \{1, \dots, L\}$ . The goal is to compute  $h \in \mathbb{N}^L$  s.t.  $h_l = \sum_{u=1}^n \sum_{k=1}^K \mathbb{I}[x_u^k = l]$ , where  $x_u^k \in [L]$ . A classic approach in LDP is based on  $L$ -ary randomized response [33], where each user submits its true value with probability  $1 - \gamma$  and a uniformly random value with probability  $\gamma$ . We denote this primitive by  $RR_\gamma : [L] \rightarrow [L]$ .

In our setting with a ring network, we propose Algorithm 2, where each contribution of a user is randomized

---

**Algorithm 2** Private histogram on the ring.
 

---

```

1: Init.  $\tau \in \mathbb{N}^L$  with  $\gamma n$  random elements
2: for  $k = 1$  to  $K$  do
3:   for  $u = 1$  to  $n$  do
4:      $y_u^k \leftarrow RR_\gamma(x_u^k)$ 
5:      $\tau[y_u^k] \leftarrow \tau[y_u^k] + 1$ 
6:   for  $i = 0$  to  $L - 1$  do
7:      $\tau[i] \leftarrow \frac{\tau[i] - \gamma/L}{1 - \gamma}$ 
8: return  $\tau$ 
    
```

---

using  $RR_\gamma$  before being added to the token  $\tau \in \mathbb{N}^L$ . Additionally,  $\tau$  is initialized with enough random elements to hide the first contributions. Note that at each step, the token contains a partial histogram equivalent to a shuffling of the contributions added so far, allowing us to leverage results on *privacy amplification by shuffling* [24, 7, 25]. In particular, we can prove the following utility and privacy guarantees for Algorithm 2 (see supplementary material for the proof).

**Theorem 2.** *Let  $\varepsilon < \frac{1}{2}$ ,  $\delta \in (0, \frac{1}{100})$ , and  $n > 1000$ . Let  $\gamma = L/(\exp(12\varepsilon\sqrt{\log(1/\delta)/n}) + L - 1)$ . Algorithm 2 outputs an unbiased estimate of the histogram with  $\gamma n(K + 1)$  expected random responses. Furthermore, it satisfies  $(\sqrt{2K \ln(1/\delta')}\varepsilon + K\varepsilon(e^\varepsilon - 1), K\delta + \delta')$ -network DP for any  $\delta' > 0$ .*

Achieving the same privacy in LDP would require  $\gamma$  to be constant in  $n$ , hence  $\sqrt{n}$  times more random responses. Equivalently, if we fix utility (i.e.,  $\gamma$ ), Theorem 2 shows that Algorithm 2 again provides a privacy gain of  $\frac{1}{n}\sqrt{n/\ln(1/\delta)} = O(1/\sqrt{n})$  compared to LDP.

**Remark 2.** *For clarity, Theorem 2 relies on the amplification by shuffling result of [24] which has a simple closed form. A tighter and more general result (with milder restrictions on the values of  $n$ ,  $\varepsilon$  and  $\delta$ ) can be readily obtained by using the results of [7, 25].*

**Remark 3.** *Algorithm 1 (real summation) can also be used to perform histogram computation. However, for domains of large cardinality  $L$  (e.g.,  $L \gg n$ ), Algorithm 2 requires fewer random numbers and maintains a sparse (more compact) representation of the histogram.*

### 3.3 Discussion

We have seen that decentralized computation over a ring provides a simple way to achieve utility similar to a trusted aggregator thanks to the sequential communication that hides the contribution of previous users in a summary. We emphasize that this is achieved without relying on a central server (only local communications) or resorting to costly multi-party computation protocols (only two secure communication channels per user are needed). Interestingly, the ring topology is often used in practical deployments and theoretic

cal analysis of (non-private) decentralized algorithms [37, 51, 35, 45, 40], owing to its simplicity and good empirical performance. Finally, we note an interesting connection between the case of network DP over a ring topology and the pan-privacy model for streaming algorithms [22] (see supplementary for details).

Despite the above advantages, the use of a fixed ring topology has some limitations. First, the above algorithms are not robust to collusions: in particular, if two users collude and share their view, Algorithm 1 does not satisfy DP. While this can be mitigated by distributing the noise addition across users (Remark 1), a node placed between two colluding nodes (or with few honest users in-between) would suffer largely degraded privacy guarantees. A similar reasoning holds for Algorithm 2. Second, a fixed ring topology is not well suited to extensions to gradient descent, where we would like to leverage privacy amplification by iteration [26]. In the latter, the privacy guarantee for a given user (data point) grows with the number of gradient steps that come after it. In a fixed ring, the privacy of a user  $u$  with respect to another user  $v$  would thus depend on their relative positions in the ring (e.g., there would be no privacy amplification when  $v$  is the user who comes immediately after  $u$ ). These limitations motivate us to consider random walks on a complete graph.

## 4 WALK ON COMPLETE GRAPH

In this section, we consider the case of a random walk on the complete graph. In other words, at each step, the token is sent to a user chosen uniformly at random among  $V$ . We consider random walks of fixed length  $T > 0$ , hence the number of times a given user contributes is itself random. We assume the token path to be hidden, including the previous sender and the next receiver, so the only knowledge of a user is the content of the messages that she/he receives and sends.

### 4.1 Real Summation

For real summation, we consider the simple and natural protocol shown in Algorithm 3: a user  $u$  receiving the token  $\tau$  for the  $k$ -th time updates it with  $\tau \leftarrow \tau + \text{Perturb}(x_u^k; \sigma_{loc})$ . As in Section 3.1,  $\sigma_{loc}$  is set such that  $\text{Perturb}(\cdot; \sigma_{loc})$  satisfies  $(\varepsilon, \delta)$ -LDP, and thus implicitly depends on  $\varepsilon$  and  $\delta$ . The next theorem gives network DP guarantees, which rely on *the intermediate aggregations of values between two visits of the token to a given user and the secrecy of the path taken by the token*. For clarity, we state below an asymptotic result to give the main order of magnitude, but stress the fact that it is derived from a *non-asymptotic* (albeit more complex) formula given in the supplementary.

**Theorem 3.** *Let  $\varepsilon < 1$  and  $\delta > 0$ . Algorithm 3 outputs*

---

**Algorithm 3** Private summation on a complete graph.

---

```

1:  $\tau \leftarrow 0, k_1 \leftarrow 0, \dots, k_n \leftarrow 0$ 
2: for  $t = 1$  to  $T$  do
3:   Draw  $u \sim \mathcal{U}(1, \dots, n)$ 
4:    $k_u \leftarrow k_u + 1$ 
5:    $\tau \leftarrow \tau + \text{Perturb}(x_u^{k_u}; \sigma_{loc})$ 
6: return  $\tau$ 

```

---

*an unbiased estimate of the sum of  $T$  contributions with standard deviation  $\sqrt{T}\sigma_{loc}$ . Furthermore, for large enough  $n$  and  $T = \Omega(n)$ , it satisfies  $(\varepsilon', N_v\delta + \delta' + \hat{\delta})$ -network DP for all  $\delta', \hat{\delta} > 0$  with*

$$\varepsilon' = O\left(\sqrt{N_u \ln(1/\delta')} \varepsilon / n^{1/3}\right), \quad (5)$$

where  $N_u = \frac{T}{n} + \sqrt{\frac{3}{n}T \ln(1/\hat{\delta})}$ .

*Sketch of proof.* We fix a user  $v$  and quantify how much information about the private data of another user  $u$  is leaked to  $v$  from the visits of the token. The number of contributions from  $u$  follows a binomial law  $\mathcal{B}(T, 1/n)$  that we upper bound by  $N_u$  using Chernoff with prob.  $1 - \hat{\delta}$ . Then, for a contribution of  $u$  at time  $t$ , it is sufficient to consider the cycle formed by the random walk between the two successive passages in  $v$  containing  $t$ . We distinguish whether the contribution is part of a “small cycle” (i.e., aggregated with less than  $c^2 n^{2/3}$  between two visits to  $v$ ), or in a larger cycle. For small cycles, we can use amplification by subsampling [5] thanks to the secrecy of the cycle. For larger cycles, the contribution of  $u$  is aggregated with at least  $c^2 n^{2/3}$  other contributions, hence the privacy loss of this contribution towards  $v$  is bounded by  $O(\varepsilon/n^{1/3})$ . The total privacy loss across the  $N_u$  contributions follows from advanced composition.  $\square$

The same algorithm analyzed under LDP yields  $\varepsilon' = O(\sqrt{N_v \ln(1/\delta')} \varepsilon)$ , which is optimal for averaging  $N_v$  contributions per user in the local model. Theorem 3 thus shows that network DP asymptotically provides a privacy amplification of  $O(1/n^{1/3})$  over LDP. While this theoretical gain is not as strong as the one obtained for the fixed ring topology, we will see in Section 5 that our non-asymptotic formula improves upon local DP as soon as  $n \geq 10$  (Figure 1a). We also show that the gains are significantly stronger in practice than what our theoretical results guarantee (Figure 1b).

### Extension to discrete histogram computation.

We can obtain a similar result for histograms by bounding the privacy loss incurred by larger cycles in the proof of Theorem 3 using amplification by shuffling [24, 7, 25], similar to what we did for the ring topology (Section 3.2). Details are in the supplementary.

---

**Algorithm 4** Private SGD on a complete graph.
 

---

- 1: Initialize  $\tau \in \mathcal{W}$
  - 2: **for**  $t = 1$  to  $T$  **do**
  - 3:   Draw  $u \sim \mathcal{U}(1, \dots, n)$
  - 4:    $Z = [Z_1, \dots, Z_d]$ ,  $Z_i \sim \mathcal{N}(0, \frac{8L^2 \ln(1.25/\delta)}{\varepsilon^2})$
  - 5:    $\tau \leftarrow \Pi_{\mathcal{W}}(\tau - \eta(\nabla_{\tau} f(\tau; D_u) + Z))$
  - 6: **return**  $\tau$
- 

## 4.2 Optimization with SGD

We now turn to the task of private convex optimization with stochastic gradient descent (SGD). Let  $\mathcal{W} \subseteq \mathbb{R}^d$  be a convex set and  $f(\cdot; D_1), \dots, f(\cdot; D_n)$  be a set of convex  $L$ -Lipschitz and  $\beta$ -smooth functions over  $\mathcal{W}$  associated with each user. We denote by  $\Pi_{\mathcal{W}}(w) = \arg \min_{w' \in \mathcal{W}} \|w - w'\|$  the Euclidean projection onto the set  $\mathcal{W}$ . We aim to privately solve the following optimization problem:

$$w^* \in \arg \min_{w \in \mathcal{W}} \{F(w) := \frac{1}{n} \sum_{u=1}^n f(w; D_u)\}. \quad (6)$$

Eq. 6 encompasses many machine learning tasks (e.g., ridge and logistic regression, SVMs, etc).

To privately approximate  $w^*$ , we propose Algorithm 4. Here, the token  $\tau \in \mathcal{W}$  represents the current iterate. At each step, the user  $u$  holding the token performs a projected noisy gradient step and sends the updated token to a random user. We rely on the Gaussian mechanism to ensure that the noisy version of the gradient  $\nabla_{\tau} f(\tau; D_u) + Z$  satisfies  $(\varepsilon, \delta)$ -LDP: the variance  $\sigma^2$  of the noise in line 4 of Algorithm 4 follows from the fact that gradients of  $L$ -Lipschitz functions have sensitivity bounded by  $2L$  [8]. Our network DP guarantee is stated below, again in a simplified asymptotic form.

**Theorem 4.** *Let  $\varepsilon < 1$ ,  $\delta < 1/2$ . Alg. 4 with  $\eta \leq 2/\beta$  achieves  $(\varepsilon', \delta + \hat{\delta})$ -network DP for all  $\hat{\delta} > 0$  with*

$$\varepsilon' = \sqrt{2q \ln(1/\delta)} \varepsilon / \sqrt{\ln(1.25/\delta)}, \quad (7)$$

where  $N_u = \frac{T}{n} + \sqrt{\frac{3}{n} T \ln(1/\hat{\delta})}$  and  $q = \max(\frac{2N_u \ln n}{n}, 2 \ln(1/\delta))$ .

*Sketch of proof.* The proof tracks the evolution of the privacy loss using Rényi Differential Privacy (RDP) [42] and leverages amplification by iteration [26] in a novel decentralized context. We give here a brief sketch (see supplementary for details). Let us fix two users  $u$  and  $v$  and bound the privacy leakage of  $u$  from the point of view of  $v$ . the number of contributions  $N_u$  of user  $u$  as in Theorem 3. We then compute the network RDP guarantee for a fixed contribution of  $u$  at time  $t$ . Crucially, it is sufficient to consider the first time  $v$  receives the token at a step  $t' > t$ . Privacy amplification by iteration tells us that the

larger  $t'$ , the less is learned by  $v$  about the contribution of  $u$ . Note that  $t'$  follows a geometric law of parameter  $1/n$ . Using the weak convexity of the Rényi divergence [26], we can bound the Rényi divergence  $D_{\alpha}(Y_v || Y'_v)$  between two random executions  $Y_v$  and  $Y'_v$  stopping at  $v$  and differing only in the contribution of  $u$  by the expected divergence over the geometric distribution. Combining with amplification by iteration eventually gives us  $D_{\alpha}(Y_v || Y'_v) \leq 4\alpha L^2 \ln n / \sigma^2 n$ . We apply the composition property of RDP over the  $N_u$  contributions of  $u$  and convert the RDP guarantee into  $(\varepsilon, \delta)$ -DP.  $\square$

Algorithm 4 is also a natural approach to private SGD in the local model, and achieves  $\varepsilon' = O(\sqrt{N_u \ln(1/\delta')})$  under LDP. Thus, for  $T = \Omega(n^2 \sqrt{\ln(1/\delta)} / \ln n)$  iterations, Theorem 4 gives a privacy amplification of  $O(\ln n / \sqrt{n})$  compared to LDP. Measuring utility as the amount of noise added to the gradients, the privacy-utility trade-off of Algorithm 4 in network DP is thus nearly the same (up to a log factor) as that of private SGD in the trusted curator model!<sup>1</sup> For smaller  $T$ , the amplification is much stronger than suggested by the simple closed form in Eq. 7: we can numerically find a smaller  $\varepsilon'$  that satisfy the conditions required by our non-asymptotic result, see supplementary for details.

We note that we can easily obtain utility guarantees for Algorithm 4 in terms of optimization error. Indeed, the token performs a random walk on a complete graph so the algorithm performs the same steps as a *centralized* (noisy) SGD algorithm. We can for instance rely on a classic result by [48] (Theorem 2 therein) which shows that SGD-type algorithms applied to a convex function and bounded convex domain converge in  $O(1/\sqrt{T})$  as long as gradients are unbiased with bounded variance.

**Proposition 1.** *Let the diameter of  $\mathcal{W}$  be bounded by  $D$ . Let  $G^2 = L^2 + \frac{8dL^2 \ln(1.25/\delta)}{\varepsilon^2}$ , and  $\tau \in \mathcal{W}$  be the output of Algorithm 4 with  $\eta = D/G\sqrt{t}$ . Then we have:*

$$\mathbb{E}[F(\tau) - F(w^*)] \leq 2DG(2 + \log T) / \sqrt{T}.$$

A consequence of Proposition 1 and Theorem 4 is that for fixed privacy budget and sufficiently large  $T$ , the expected error of Algorithm 4 is  $O(\ln n / \sqrt{n})$  smaller under network DP than under LDP.

## 4.3 Discussion

An advantage of considering a random walk over a complete graph is that our approach is naturally robust to the presence of a (constant) number of colluding users. Indeed, when  $c$  users collude, they can be seen as a unique node in the graph with a transition probability

<sup>1</sup>Incidentally, the analysis of centralized private SGD [8] also sets the number of iterations to be of order  $n^2$ .

of  $\frac{c}{n}$  instead of  $\frac{1}{n}$ . We can then easily adapt the proofs above, as the total number of visits to colluding users follows  $\mathcal{B}(T, c/n)$  and the size of a cycle between two colluding users follows a geometric law of parameter  $1 - c/n$ . Hence, we obtain the same guarantees under Definition 2 as for the case with  $n/c$  non-colluding users under Definition 1. Interestingly, these privacy guarantees hold even if colluding users bias their choice of the next user instead of choosing it uniformly.

The assumption that users do not know the identity of the previous sender and the next receiver may seem quite strong. It is however possible to lift this assumption by bounding the number of times a contribution of a given user  $u$  is directly observed by a given user  $v$  and accounting for the resulting privacy loss separately. When  $T = \Omega(n^2)$ , this term is negligible and the results of Theorem 3-4 still hold. In practical implementations, we can enforce a deterministic bound on the number of times any edge  $(u, v)$  is used, e.g., by contributing only noise along  $(u, v)$  after it has been used too many times. We refer to the supplementary for details.

## 5 EXPERIMENTS

We now present some numerical experiments that illustrate the practical significance of our privacy amplification results in the complete graph setting (Section 4).

### 5.1 Real Summation

**Comparison of analytical bounds.** We numerically evaluate the theoretical (non-asymptotic) bound of Theorem 3 for the task of real summation and compare it to local DP. Recall that the number of contributions of a user is random (with expected value  $T/n$ ). For a fair comparison between network and local DP, we derive an analogue of Theorem 3 for local DP. In addition, to isolate the effect of the number of contributions (which is the same in both settings), we also report the bounds obtained under the assumption that each user contributes exactly  $T/n$  times. Figure 1a plots the value of the bounds for varying  $n$ . We see that our theoretical result improves upon local DP as soon as  $n \geq 10$ , and these gains become more significant as  $n$  increases. We note that the curves obtained under the fixed number of contributions per user also suggest that a better control of  $N_v$  in the analytical bound could make our amplification result significantly tighter.

**Gap with empirical behavior.** Our formal analysis involves controlling the number of contributions of users, as well as the size of cycles using concentration inequalities, which require some approximations. In practical deployments one can instead use the actual values of these quantities to compute the privacy

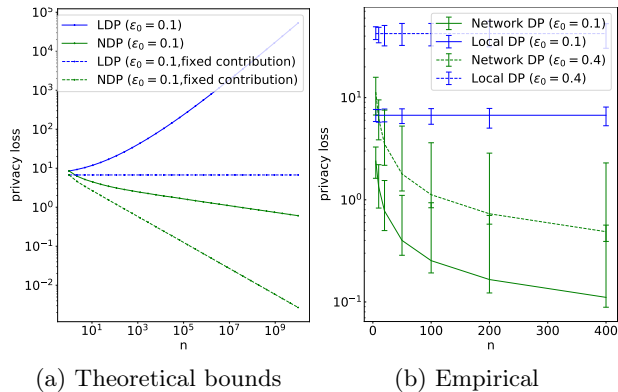


Figure 1: Comparing network and local DP on real summation for  $T = 100n$ .  $\epsilon_0$  rules the amount of local noise added to each contribution (i.e., each single contribution taken in isolation satisfies  $\epsilon_0$ -LDP). For the empirical results of Figure 1b, the curves report the average privacy loss across all pairs of users and all 10 random runs; error bars give best and worst cases.

loss. We thus investigate the gap between our theoretical guarantees and what can be obtained in practice through simulations. Specifically, we sample a random walk of size  $T = 100n$ . Then, for each pair of users, we compute the privacy loss based on the actual walk and the advanced composition mechanism. We repeat this experiment over 10 random walks and we can then report the average, the best and the worst privacy loss observed across all pairs of users and all random runs. Figure 1b reports such empirical results obtained for the case of real summation with the Gaussian mechanism, where the privacy grows with a factor  $\sqrt{m}$  where  $m$  is the number of elements aggregated together (i.e., the setting covered by Theorem 3). We observe that the gains achieved by network DP are significantly stronger in practice than what our theoretical bound guarantees, and are significant even for small  $n$  (see Figure 1a). Our experiments on discrete histogram computation also show significant gains (see supplementary).

### 5.2 Machine Learning with SGD

We now present some experiments on the task of training a logistic regression model in the decentralized setting. Logistic regression corresponds to solving Eq. 6 with  $\mathcal{W} = \mathbb{R}^d$  and the loss functions defined as  $f(w; D_u) = \frac{1}{|D_u|} \sum_{(x,y) \in D_u} \ln(1 + \exp(-yw^\top x))$  where  $x \in \mathbb{R}^d$  and  $y \in \{-1, 1\}$ . We use a binarized version of UCI Housing dataset [1]. We standardize the features and further normalize each data point  $x$  to have unit L2 norm so that the logistic loss is 1-Lipschitz for any  $(x, y)$ . We split the dataset uniformly at random into a training set (80%) and a test set, and further split the training set across  $n = 2000$  users, resulting in each



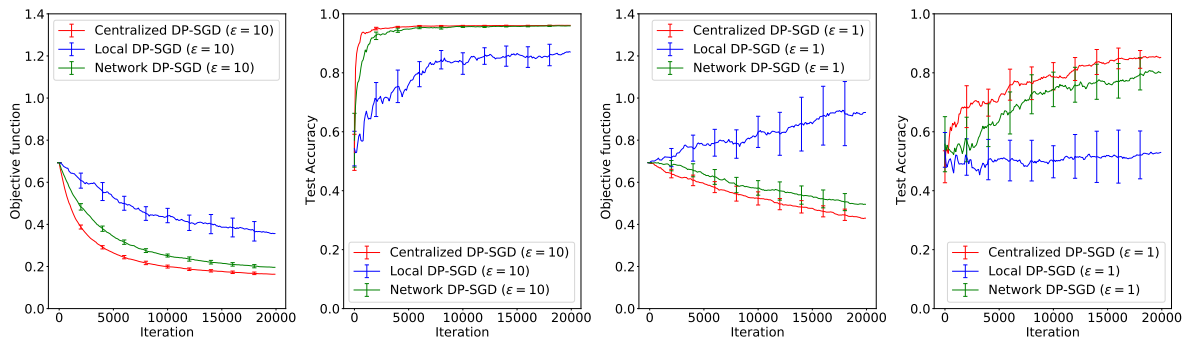


Figure 2: Comparing three settings for SGD with gradient perturbation. Unlike Local and Network DP-SGD, Centralized DP-SGD requires a trusted curator and benefits from amplification by subsampling. Network DP-SGD nearly bridges the gap between Centralized and Local DP-SGD. In all methods,  $\sigma$  is set to ensure  $\epsilon = 10$  (left plots) or  $\epsilon = 1$  (right plots), and  $\delta = 10^{-6}$ . Mean and standard deviations are computed over 20 runs.

user  $u$  having a local dataset  $D_u$  of size 8.

We compare three variants of private SGD based on gradient perturbation with the Gaussian mechanism. *Centralized DP-SGD* is the centralized version of differentially private SGD introduced in [8], which assumes the presence of a trusted curator/aggregator. *Local DP-SGD* corresponds to Algorithm 4 with the noise calibrated for the LDP setting. Finally, *Network DP-SGD* is Algorithm 4 with the noise calibrated according to network DP (see Theorem 4). To make the comparison as fair as possible, all approaches (including Centralized DP-SGD) use the full dataset  $D_u$  of a randomly chosen user  $u$  as the mini-batch at each step.

Given the privacy budget  $(\epsilon, \delta)$  for the whole procedure, each of the three methods leads to a different choice for  $\sigma$  that parametrizes the level of noise added to each gradient. In our experiments, we fix  $\epsilon = 10$  (low privacy) and  $\epsilon = 1$  (stronger privacy) and  $\delta = 10^{-6}$ . We recall that we consider user-level DP ( $X \sim_u X'$  differ in the local database of user  $u$ ). Note that due to composition, more iterations increase the per-iteration level of noise needed to achieve a fixed DP guarantee. As the number of contributions of a given user is random, we upper bound it in advance with a tighter bound than used in our theorems, namely  $cT/n$  where  $c$  is a parameter to tune. If a user is asked to participate more times than budgeted, it simply forwards the token to another user without adding any contribution. In the case of Network DP-SGD, the user still adds noise as the privacy guarantees of others rely on it. Note that the best regime for network DP is when the number of contributions of a user is roughly equal to  $n$ , see Theorem 4. In our experiments, we are not in this regime but the privacy amplification effect is stronger than the closed form of the theorem. In practice, we compute numerically the smallest  $\sigma$  needed to fulfill the conditions of the proof (see supplementary).

Figure 2 shows results for  $T = 20000$ , where the step size  $\eta$  was tuned separately for each approach in  $[10^{-4}, 2]$ . We see that Network DP-SGD nearly matches the privacy-utility trade-off of Centralized DP-SGD for both  $\epsilon = 1$  and  $\epsilon = 10$  without relying on a trusted curator. Network DP-SGD also clearly outperforms Local DP-SGD, which actually diverges for  $\epsilon = 1$ . These empirical results are consistent with our theory and show that Network DP-SGD significantly amplifies privacy compared to local DP-SGD even when the number of iterations  $T$  is much smaller than  $O(n^2/\ln n)$ , a regime which is of much practical importance.

## 6 PERSPECTIVES

We believe that our work opens many interesting perspectives. We would like to consider generalizations of our results to arbitrary graphs by relying on classic graph theoretic notions like the hitting time. Furthermore, we think that time-evolving topologies can help improve robustness to collusions, in particular in rings and other sparse topologies. Network DP can also be used to study other decentralized models of computation. A natural extension of the algorithms studied here is to consider multiple tokens walking on the graph in parallel. We would also like to study randomized gossip algorithms [13], which are popular for decentralized optimization in machine learning [18] and were recently shown to provide DP guarantees in the context of rumor spreading [10]. Finally, we would like to investigate the fundamental limits of network DP and consider further relaxations where users put more trust in their direct neighbors than in more distant users.

## References

- [1] UCI Housing dataset. <https://www.openml.org/d/823>.

- 
- [2] Kareem Amin, Matthew Joseph, and Jieming Mao. Pan-private uniformity testing. *CoRR*, abs/1911.01452, 2019.
- [3] Ghadir Ayache and Salim El Rouayheb. Private Weighted Random Walk Stochastic Gradient Descent. Technical report, arXiv:2009.01790, 2020.
- [4] Victor Balcer, Albert Cheu, Matthew Joseph, and Jieming Mao. Connecting robust shuffle privacy and pan-privacy, 2020.
- [5] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *NeurIPS*, 2018.
- [6] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Differentially Private Summation with Multi-Message Shuffling. Technical report, arxiv:1906.09116, 2019.
- [7] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The Privacy Blanket of the Shuffle Model. In *CRYPTO*, 2019.
- [8] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *FOCS*, 2014.
- [9] James Henry Bell, Kallista A. Bonawitz, Adrià Gascón, Tancrede Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overheads. In *CCS*, 2020.
- [10] Aurélien Bellet, Rachid Guerraoui, and Hadrien Hendrikx. Who started this rumor? Quantifying the natural differential privacy guarantees of gossip protocols. In *DISC*, 2020.
- [11] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. Personalized and Private Peer-to-Peer Machine Learning. In *AISTATS*, 2018.
- [12] 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 *CCS*, 2017.
- [13] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking*, 14(SI):2508–2530, 2006.
- [14] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal Lower Bound for Differentially Private Multi-party Aggregation. In *ESA*, 2012.
- [15] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography*, 2012.
- [16] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Syed Zawad, Feng Yan, Shiyu Li, Hai Helen Li, and Yiran Chen. Towards Decentralized Deep Learning with Differential Privacy. In *CLOUD*, 2019.
- [17] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed Differential Privacy via Shuffling. In *EUROCRYPT*, 2019.
- [18] Igor Colin, Aurélien Bellet, Joseph Salmon, and Stéphan Cléménçon. Gossip Dual Averaging for Decentralized Optimization of Pairwise Functions. In *ICML*, 2016.
- [19] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *FOCS*, 2013.
- [20] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *EUROCRYPT*, 2006.
- [21] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography (TCC)*, 2006.
- [22] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *ICS*, 2010.
- [23] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and Differential Privacy. In *FOCS*, 2010.
- [24] Ulfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, and Kunal Talwar. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *SODA*, 2019.
- [25] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding Among the Clones: A Simple and Nearly Optimal Analysis of Privacy Amplification by Shuffling. Technical report, arXiv:2012.12803, 2020.
- [26] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy Amplification by Iteration. In *FOCS*, 2018.
- [27] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *NeurIPS*, 2020.

- [28] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Pure Differentially Private Summation from Anonymous Messages. Technical report, arXiv:2002.01919, 2020.
- [29] Zhenqi Huang, Sayan Mitra, and Nitin Vaidya. Differentially Private Distributed Optimization. In *ICDCN*, 2015.
- [30] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *NeurIPS*, 2018.
- [31] Björn Johansson, Maben Rabi, and Mikael Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2009.
- [32] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D’Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. Technical report, arXiv:1912.04977, 2019.
- [33] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. In *NIPS*, 2014.
- [34] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What Can We Learn Privately? In *FOCS*, 2008.
- [35] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized sgd with changing topology and local updates. In *ICML*, 2020.
- [36] Chencheng Li, Pan Zhou, Li Xiong, Qian Wang, and Ting Wang. Differentially Private Distributed Online Learning. *IEEE Transactions on Knowledge and Data Engineering*, 2018.
- [37] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In *NIPS*, 2017.
- [38] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous Decentralized Parallel Stochastic Gradient Descent. In *ICML*, 2018.
- [39] Xianghui Mao, Kun Yuan, Yubin Hu, Yuantao Gu, Ali H. Sayed, and Wotao Yin. Walkman: A Communication-Efficient Random-Walk Algorithm for Decentralized Optimization. *IEEE Transactions on Signal Processing*, 68:2513–2528, 2020.
- [40] Othmane Marfoq, Chuan Xu, Giovanni Neglia, and Richard Vidal. Throughput-Optimal Topology Design for Cross-Silo Federated Learning. In *NeurIPS*, 2020.
- [41] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Recurrent Language Models. In *ICLR*, 2018.
- [42] Ilya Mironov. Rényi Differential Privacy. In *CSF*, 2017.
- [43] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, 2019.
- [44] Giovanni Neglia, Gianmarco Calbi, Don Towsley, and Gayane Vardoyan. The role of network topology for distributed machine learning. In *INFOCOM*, 2019.
- [45] Giovanni Neglia, Chuan Xu, Don Towsley, and Gianmarco Calbi. Decentralized gradient methods: does topology matter? In *AISTATS*, 2020.
- [46] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli. Incremental stochastic subgradient algorithms for convex optimization. *SIAM Journal on Optimization*, 20(2):691–717, 2009.
- [47] César Sabater, Aurélien Bellet, and Jan Ramon. Distributed Differentially Private Averaging with Improved Utility and Robustness to Malicious Parties. Technical report, arXiv:2006.07218, 2020.

- [48] Ohad Shamir and Tong Zhang. Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes. In *ICML*, 2013.
- [49] Elaine Shi, T.-H. Hubert Chan, Eleanor G. Rieffel, Richard Chow, and Dawn Song. Privacy-Preserving Aggregation of Time-Series Data. In *NDSS*, 2011.
- [50] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.
- [51] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu.  $D^2$ : Decentralized Training over Decentralized Data. In *ICML*, 2018.
- [52] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized Collaborative Learning of Personalized Models over Networks. In *AISTATS*, 2017.
- [53] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical Risk Minimization in Non-interactive Local Differential Privacy Revisited. In *NeurIPS*, 2018.
- [54] Jie Xu, Wei Zhang, and Fei Wang. A(DP)2SGD: Asynchronous Decentralized Parallel Stochastic Gradient Descent with Differential Privacy. Technical report, arXiv:2008.09246, 2020.
- [55] Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. Improving the Privacy and Accuracy of ADMM-Based Distributed Algorithms. In *ICML*, 2018.
- [56] Kai Zheng, Wenlong Mou, and Liwei Wang. Collect at Once, Use Effectively: Making Non-interactive Locally Private Learning Possible. In *ICML*, 2017.

## SUPPLEMENTARY MATERIAL

### A Proof of Theorem 1 (Real Aggregation on a Ring)

*Proof.* We start by proving the utility claim. Algorithm 1 adds independent noise with standard deviation  $\sigma_{loc}$  to the token every  $n - 1$  contributions. As there are  $Kn$  steps, such noise is added  $\lfloor Kn/(n - 1) \rfloor$  times. By commutativity, the total noise has standard deviation  $\sqrt{\lfloor Kn/(n - 1) \rfloor} \sigma_{loc}$ .

We now turn to the network differential privacy claim. Let us fix two distinct users  $u$  and  $v$  and consider what  $v$  learns about the data of  $u$ . Recall that the structure of the ring is assumed to be public. The view  $\mathcal{O}_v$  of  $v$  (i.e., the information observed by  $v$  during the execution of the protocol as defined in Eq. 1) thus corresponds to the  $K$  values of the token that she receives. We denote these values by  $\tau_1^v, \dots, \tau_K^v$ , each of them corresponding to user contributions aggregated along with random noise. We define the view of  $v$  accordingly as:

$$\mathcal{O}_v(\mathcal{A}(D)) = (\tau_i^v)_{i=1}^K. \quad (8)$$

Let us fix  $i \in \{2, \dots, K\}$ . By construction,  $\tau_i^v - \tau_{i-1}^v$  is equal to the sum of updates between two visits of the token. In particular, we have the guarantee that at least one user different from  $v$  has added noise in  $\tau_i^v - \tau_{i-1}^v$  (as there are  $n > n - 1$  steps), and that  $\tau_i^v - \tau_{i-1}^v$  does not contain more than one contribution made by  $v$ . It follows that the aggregation  $\tau_{i+1}^v - \tau_i^v$  can be rewritten as  $\text{Perturb}(x_u^i; \sigma_{loc}) + z$ , where  $z$  is independent from the contribution of  $u$ . By the  $(\varepsilon, \delta)$ -LDP property of  $\text{Perturb}(\cdot; \sigma_{loc})$  and the post-processing property of differential privacy, we have for any  $x, x'$ :

$$\mathbb{P}(\tau_{i+1}^v - \tau_i^v = \tau | x_u^i = x) \leq e^\varepsilon \mathbb{P}(\tau_{i+1}^v - \tau_i^v = \tau | x_u^i = x') + \delta.$$

For the first token  $\tau_1^v$ , note it also contains noise with standard deviation  $\sigma_{loc}$  added by the first user, so the same guarantee holds.

Finally, we apply the advanced composition theorem [23] to get a differential privacy guarantee for the  $K$  visits of the token, leading to the final privacy guarantee of  $(\sqrt{2K \ln(1/\delta')} \varepsilon + K\varepsilon(e^\varepsilon - 1), K\delta + \delta')$ -network DP.  $\square$

### B Proof of Theorem 2 (Histogram Computation on a Ring)

*Proof.* The proof is similar in spirit to the real summation case (see Appendix A), but leverages privacy amplification by subsampling to be able to quantify how much information is leaked by the value of the token (which is now a histogram).

We start by the utility claim (expected number of contributions). There are  $Kn$  steps with at each step a probability  $\gamma$  of adding a random response, plus the  $\gamma n$  random responses at initialization, leading to a total of  $\gamma n(K + 1)$  random responses in expectation.

We now turn to the differential privacy guarantee. The view of a user  $v$  is the content of the token at each visit of the token as defined in Eq. 8, except that each  $\tau_i^v \in \mathbb{N}^L$  is now a histogram over the domain  $[L]$ . More specifically, for  $i \in \{2, \dots, K\}$ , the difference  $\tau_{i+1}^v - \tau_i^v$  between two consecutive tokens is now a discrete histogram of  $n$  answers obtained by  $RR_\gamma$  (each of them is random with probability  $\gamma$ ). Similarly, in the first round, the token is initialized with  $\gamma n$  random elements. Therefore, we can apply results from amplification by shuffling, because a discrete histogram carries the same more information as a shuffle of the individual values. In particular, we can use Corollary 9 of [24] that we recall below.

**Theorem 5** (Erlingsson). *Let  $n \geq 100, 0 < \varepsilon_0 < \frac{1}{2}$  and  $\delta < \frac{1}{100}$ . For a local randomizer ensuring  $\varepsilon_0$ -LDP, the shuffling mechanism is  $(\varepsilon, \delta)$ -differentially private with*

$$\varepsilon = 12\varepsilon_0 \sqrt{\frac{\log(1/\delta)}{n}}.$$

We can apply this result to the information revealed by the value of the token between two visits to user  $v$ . The required LDP guarantee is ensured by the use of the randomized response mechanism, where we set  $\gamma$  so that  $RR_\gamma$  satisfies  $12\varepsilon \sqrt{\frac{\log(1/\delta)}{n}}$ -LDP, leading to an  $(\varepsilon, \delta)$ -DP guarantee after shuffling. We conclude by the application of advanced composition [23].  $\square$

## C Relation between Network DP on a Ring and Pan-Privacy

In this section, we highlight an interesting connection between the specific case of network DP on a ring topology and the pan-privacy model [22]. In the pan-privacy model, raw data is processed in an online fashion by a central party. This central party is trusted to process raw data but not to store it in perpetuity, and its storage may be subject to breaches (i.e., its internal state may become visible to an adversary). It can thus be seen as an intermediate trust model between the central and local models. Connections between pan-privacy and the shuffle model have been recently studied [4], allowing in some cases to adapt algorithms from one setting to the other. Other recent work has studied the relation between pan-privacy with several breaches and the local model [2].

To formally define pan-privacy, we first need to define what we mean by online algorithms. An online algorithm receives a stream of raw data and sequentially updates an internal state with one data point before deleting it. At the end of the stream, the algorithm publishes a final output based on its last internal state.

**Definition 3** (Online algorithm). *An online algorithm  $\mathcal{A}$  is defined by a sequence of internal algorithms  $\mathcal{A}_1, \dots$  and an output algorithm  $\mathcal{A}_O$ . Given an input stream  $\vec{x}$ , the first internal algorithm  $\mathcal{A}_1 : \mathcal{X} \rightarrow \mathcal{I}$  maps  $x_1$  to a state  $s_1$ , and for  $i \geq 2$ ,  $\mathcal{A}_i : \mathcal{X} \times \mathcal{I} \rightarrow \mathcal{I}$  maps  $x_i$  and the previous state  $s_{i-1}$  to a new state  $s_i$ . At the end of the stream,  $\mathcal{A}$  publishes a final output by executing  $\mathcal{A}_O : \mathcal{I} \rightarrow \mathcal{O}$  on its final internal state. We denote by  $\mathcal{A}_{\mathcal{I}}(\vec{x})$  the internal state of  $\mathcal{A}$  after processing stream  $\vec{x}$ .*

In pan-privacy, the algorithm is trusted to process a raw data stream, but should protect its internal states against potential breaches. The moment of the update where the state is modified by a raw data point is supposed to be atomic. Hence, the observable impact of a data point is restricted to the internal state and the final output. Below, we state the standard definition of pan-privacy with a single breach, i.e., the adversary may observe a single internal state in addition to the final output. Two streams  $\vec{x}, \vec{x}'$  are said to be neighboring if they differ in at most one element.

**Definition 4** (Pan-privacy). *An online algorithm  $\mathcal{A}$  is  $(\varepsilon, \delta)$ -pan private if for every pair of neighboring streams  $\vec{x} \sim \vec{x}'$ , for every time  $t$ , and for every subset  $T \subseteq \mathcal{I} \times \mathcal{O}$ , we have:*

$$\mathbb{P}((\mathcal{A}_{\mathcal{I}}(\vec{x}_{\leq t}), \mathcal{A}_O(\mathcal{A}_{\mathcal{I}}(\vec{x}))) \in T) \leq e^\varepsilon \cdot \mathbb{P}((\mathcal{A}_{\mathcal{I}}(\vec{x}'_{\leq t}), \mathcal{A}_O(\mathcal{A}_{\mathcal{I}}(\vec{x}')) \in T) + \delta,$$

where  $\vec{x}_{\leq t}$  denotes the first  $t$  elements of stream  $\vec{x}$ .

We can now make a connection between the above pan-privacy definition and our simple protocols for network DP on a ring topology introduced in Section 3, in the case where each user contributes only once ( $K = 1$  in our notations). In our network DP setting, the internal state corresponds to the value of the token and the final output is empty (or is equal to the final state of the token, if one performs an additional cycle over the ring during which the token is left unchanged to broadcast it to all users). A breach at time  $t$  (i.e., observation of internal state  $s_t$ ) corresponds to the observation of the token by the  $t$ -th user. Note also that our neighboring relation on the users' datasets is equivalent to that on data streams for the case of  $K = 1$ . Therefore, we can simulate a pan-private algorithm as a network DP algorithm on a ring.

We note that the lack of privacy gains for network DP compared to local DP when considering a ring topology with collusions (see discussion in Section 3.3) is in line with the reduction in [2], which shows that in pure pan-privacy, protection against multiple breaches is equivalent to sequentially interactive local privacy.

While network DP reduces to pan-privacy when the topology is the ring and one considers simple protocols with a single token and a single contribution per user, we emphasize that our model is more general and potentially allows superior privacy-utility trade-offs for more complex protocols and/or topologies. This is illustrated by our results on the complete graph, where breaches cannot follow an arbitrary pattern. Indeed, as a breach corresponds to sending the token to a colluding user, this risk is mitigated by the properties of the random walk: as long as the token is held by non-colluding users, the walk stays unbiased and thus does not return too quickly to colluding users. This additional structure on the potential breaches give us the room for stronger guarantees.

## D Proof of Theorem 3 (Real Summation on the Complete Graph)

*Proof.* We will prove an  $(\varepsilon_f, \delta_f)$ -DP guarantee for Algorithm 3. We note that our proof does not require the global time counter  $t$  to be hidden from users (i.e., the result holds even if users receiving the token know how many users have added contributions to the token since its last visit).

Let us fix two distinct users  $u$  and  $v$ . We aim to quantify how much information about the private data of user  $u$  is leaked to  $v$  from the visits of the token. Recall that we assume the token path to be hidden, including the previous sender and the next receiver. We can thus define the view  $\mathcal{O}_v$  of user  $v$  by:

$$\mathcal{O}_v(\mathcal{A}(D)) = (\tau_{k_i})_{i=1}^{T_v}, \quad (9)$$

where  $k_i$  is the  $i$ -th time that  $v$  receives the token,  $\tau_{k_i}$  the corresponding value of the token, and  $T_v$  the number of times that  $v$  had the token during the whole execution of the protocol.

We aim at bounding the privacy loss that occurs from the contributions of  $u$  from the point of view of  $v$ . We start by bounding the privacy loss for a single contribution of  $u$  occurring at some time  $t$ . This occurs between two passages of the token in  $v$ , so there exists a unique  $i$  such as  $k_i < t < k_{i+1}$ .<sup>2</sup> Note that the token values observed before  $t$  do not depend on the contribution of  $u$  at time  $t$ . Moreover, it is sufficient to bound the privacy loss induced by the observation of the token at  $k_{i+1}$ : indeed, by the post-processing property of DP, no additional privacy loss with respect to  $v$  will occur for observations posterior to  $k_{i+1}$ . Let us distinguish between two cases depending on the size of the cycle from  $v$ :  $k_{i+1} - k_i$  is either smaller or larger than  $c^2 n^{2/3}$ , where  $c$  is a arbitrary positive constant to tune.

**Contributions in large cycles.** If  $k_{i+1} - k_i$  is larger than  $c^2 n^{2/3}$ , then the contribution of  $u$  is mixed with at least  $c^2 n^{2/3}$  other contributions (note that some might also come from  $u$ ). The added Gaussian noise at each step sum up in the aggregation, so we can bound the privacy loss by at most  $\varepsilon/\sqrt{c^2 n^{2/3}} = \varepsilon/cn^{1/3}$ , while the associated  $\delta$  does not change.

**Contributions in small cycles.** If  $k_{i+1} - k_i$  is smaller than  $c^2 n^{2/3}$ , we can bound the privacy loss suffered by  $u$  with respect to  $v$  using amplification by subsampling with replacement [5]. Indeed, the random walk between the time  $k_i$  and  $k_{i+1}$  selects  $k_{i+1} - k_i$  participating users by uniform subsampling with replacement from the whole set of users  $V = \{1, \dots, n\}$ : as the task is real summation, the order of the samples does not matter, and the sampling is uniform as the walk is unbiased. We can thus apply Theorem 10 from [5]: given  $n$  users and  $m$  the size of the cycle, we obtain

$$\varepsilon_{amp} = \log(1 + (1 - (1 - 1/n)^m)(e^\varepsilon - 1)), \quad (10)$$

where  $\varepsilon$  is the level of privacy guaranteed by the local mechanism  $\text{Perturb}(\cdot; \sigma_{loc})$ . This expression is non-decreasing with  $m$ , so we bound it by its value for  $m = c^2 n^{2/3}$ . Denoting  $\gamma_n = 1 - (1 - \frac{1}{n})^{c^2 n^{2/3}}$  and using the inequality  $\log(1 + x) \leq x$ , we have:

$$\varepsilon_{amp} \leq \min(\gamma_n(e^\varepsilon - 1), \varepsilon).$$

In particular, if  $\varepsilon < 1$ , the first term is less than  $2\gamma_n\varepsilon$ .

We also compute the corresponding  $\delta_{amp}$  with the privacy amplification by subsampling result of [5]:

$$\delta_{amp} = \sum_{k=1}^m \binom{m}{k} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{m-k} \delta_{\mathcal{A},k}(\varepsilon),$$

where  $\delta_{\mathcal{A},k}(\varepsilon)$  correspond to the privacy for group of size  $k$  for the algorithm  $\mathcal{A}$ , so we have  $\delta_{\mathcal{A},k}(\varepsilon) \leq k\delta$ . Hence, we can upper bound it by:

$$\delta_{amp} \leq \delta \sum_{k=1}^m \binom{m}{k} k \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{m-k} = \frac{m}{n} \delta.$$

Thus, for  $m < n$ , it stays smaller than the initial  $\delta$ .

Combining the bounds obtained in each case, each contribution of  $u$  has a privacy loss bounded by

$$\varepsilon_c \leq \max(2\gamma_n\varepsilon, \varepsilon/cn^{1/3}),$$

with the initial  $\delta$ .

<sup>2</sup>If the contribution of  $u$  occurs before the first passage of the token at  $v$ , we can take  $k_i = 0$ . As for contributions occurring *after* the last passage of the token at  $v$ , they do not incur any privacy loss.

---

**Algorithm 5** Private histogram computation on a complete graph.
 

---

```

Init.  $\tau \in \mathbb{N}^L$ 
for  $t = 1$  to  $T$  do
    Draw  $u \sim \mathcal{U}(1, \dots, n)$ 
     $y_u^k \leftarrow RR_\gamma(x_u^k)$ 
     $\tau[y_u^k] \leftarrow \tau[y_u^k] + 1$ 
for  $i = 0$  to  $L - 1$  do
     $\tau[i] \leftarrow \frac{\tau[i] - \gamma/L}{1 - \gamma}$ 
return  $\tau$ 
    
```

---

We now bound the number of contributions of  $u$  during the algorithm. As the user receiving the token at a given step is chosen uniformly at random and independently from the other steps, there is a probability of  $1/n$  that the token is at  $u$  at any given step. Thus, the number of visits  $T_u$  to  $u$  follows a binomial law  $\mathcal{B}(T, 1/n)$ . We bound it by  $N_u$  with probability  $1 - \hat{\delta}$  using Chernoff. Recall that the Chernoff bound allows to upper bound (with high probability) the sum of independent random variables  $X_1, \dots, X_T$  of expected value  $p$ , for any real  $\alpha \in [0, 1]$ :

$$\mathbb{P}\left(\sum_{i=1}^T X_i \geq (1 + \alpha)Tp\right) \leq e^{-\alpha^2 pT/3}.$$

In our case, we want to upper bound the probability that the number of contributions  $T_u$  of a given user  $u$  exceeds some threshold  $N_u$  by  $\hat{\delta}$ . Using the previous bound for  $p = 1/n$  and  $\alpha = \sqrt{\frac{3n \log(1/\hat{\delta})}{T}}$ , by considering the random variables equal to 1 if  $v$  has the token and 0 otherwise, we have:

$$\mathbb{P}\left(T_u \geq \underbrace{\frac{T}{n} + \sqrt{\frac{3T}{n} \log(1/\hat{\delta})}}_{N_u}\right) \leq \hat{\delta}.$$

We conclude by using advanced composition over the  $N_u$  contributions of  $u$ . For all  $\delta' > 0, c > 0$ :

$$\varepsilon_f \leq \sqrt{2N_u \log(1/\delta')} \max(2\gamma_n \varepsilon, \varepsilon/cn^{1/3}) + N_u \varepsilon (e^{\max(2\gamma_n \varepsilon, \varepsilon/cn^{1/3})} - 1).$$

The deltas sum up and thus  $\delta_f \leq N_u \delta + \delta' + \hat{\delta}$ .

While the above formula does not have any explicit condition on  $T$  or  $n$ , the dependency is hiding in the Chernoff bound. However, for large enough  $n$  and  $T = \Omega(n)$ , using the Taylor expansion  $\gamma_n \sim c^2/n^{1/3}$ , we obtain the asymptotic privacy amplification of  $O(1/n^{1/3})$  given in the theorem.  $\square$

## E Histogram Computation on the Complete Graph

For discrete histogram computation on the complete graph, we propose Algorithm 5: when receiving the token, each user perturbs his/her contribution with  $L$ -ary randomized response, adds it to the token and forwards the token to another user chosen uniformly at random. As in the case of real summation, we obtain a privacy amplification of  $O(1/n^{1/3})$  compared to the same algorithm analyzed under LDP.

**Theorem 6.** *Let  $\delta > 0, c > 0$  and  $\varepsilon \leq \min\left(1, \ln\left(\frac{c^2 n^{2/3}}{16 \ln(2/\delta)}\right)\right)$ . Algorithm 5 achieves  $(\varepsilon', \frac{T}{n} \delta + \delta' + \hat{\delta})$ -network DP for all  $\delta', \hat{\delta} > 0$  with*

$$\varepsilon' = \sqrt{2N_u \log(1/\delta')} \max(2\gamma_n \varepsilon, \frac{14\sqrt{\ln(4/\delta)}}{cn^{1/3}} \varepsilon) + N_u \varepsilon (e^{\max(2\gamma_n \varepsilon, \frac{14\sqrt{\ln(4/\delta)}}{cn^{1/3}} \varepsilon)} - 1),$$

where  $N_u = \frac{T}{n} + \sqrt{\frac{3}{n} T \ln(1/\hat{\delta})}$  and  $\gamma_n = 1 - (1 - \frac{1}{n})c^2 n^{2/3}$ .

*Proof.* The proof follows the same steps as in the case of real summation (Appendix D), as the properties of the walk stays identical. In the case of small cycles, the same bound applies, so we only need to bound the privacy



loss in case of large cycles. To do this, we leverage privacy amplification by shuffling. Specifically here, we use the tight bound of [25] (Theorem 3.1 therein), that we recall below.

**Theorem 7** (Amplification by shuffling [25]). *For any data domain  $\mathcal{X}$ , let  $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{X} \rightarrow \mathcal{S}^{(i)}$  for  $i \in [n]$  (where  $\mathcal{S}^{(i)}$  is the range space of  $\mathcal{R}^{(i)}$ ) be a sequence of algorithms such that  $\mathcal{R}^{(i)}(z_1, \dots, z_{i-1}, \cdot)$  is an  $\varepsilon_0$ -DP local randomizer for all values of auxiliary inputs  $(z_1, \dots, z_{i-1}) \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$ . Let  $\mathcal{A}_s : \mathcal{X}^n \rightarrow \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$  be the algorithm which takes as input a dataset  $(x_1, \dots, x_n) \in \mathcal{X}^n$ , samples a uniform random permutation  $\pi$  over  $[n]$ , then sequentially computes  $z_i = \mathcal{R}^{(i)}(z_1, \dots, z_{i-1}, x_{\pi(i)})$  for  $i \in [n]$  and outputs  $(z_1, \dots, z_n)$ . Then for any  $\delta \in [0, 1]$  such that  $\varepsilon_0 \leq \log\left(\frac{n}{16 \log(2/\delta)}\right)$ ,  $\mathcal{A}_s$  satisfies  $(\varepsilon, \delta)$ -DP with*

$$\varepsilon \leq \ln \left( 1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1} \left( \frac{8\sqrt{e^{\varepsilon_0} \ln(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_0}}{n} \right) \right).$$

Note that for  $\varepsilon \leq \min\left(1, \ln\left(\frac{c^2 n^{2/3}}{16 \ln(2/\delta)}\right)\right)$ , we can simplify the bound for the sake of clarity. We use the fact that  $\frac{e^x - 1}{e^x + 1} \leq \frac{x}{2}$ , which gives

$$\varepsilon \leq \left( 1 + \frac{\varepsilon_0}{2} \left( \frac{8\sqrt{e^{\varepsilon_0} \ln(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_0}}{n} \right) \right).$$

We then use the hypothesis  $\varepsilon \leq 1$  and the concavity of the logarithm to obtain the following simple bound:

$$\varepsilon \leq \frac{14\sqrt{\ln(4/\delta)}}{\sqrt{n}} \varepsilon_0.$$

We apply this bound to the shuffling of  $c^2 n^{2/3}$  contributions, where the hypothesis of the theorem are fulfilled by our choice of  $\varepsilon$ . We thus have the following bound for the contribution belonging to a large cycle:

$$\varepsilon \leq \frac{14\sqrt{\ln(4/\delta)}}{cn^{1/3}} \varepsilon_0.$$

We conclude the proof by using this bound to control the privacy loss associated with large cycles and following the same steps as in Appendix D. For all  $c > 0, \delta > 0, \delta' > 0$ , we have:

$$\varepsilon_f \leq \sqrt{2N_u \log(1/\delta')} \max(2\gamma_n \varepsilon, \frac{14\sqrt{\ln(4/\delta)}}{cn^{1/3}} \varepsilon) + N_u \varepsilon (e^{\max(2\gamma_n \varepsilon, \frac{14\sqrt{\ln(4/\delta)}}{cn^{1/3}} \varepsilon)} - 1)$$

with  $\delta_f = N_u \delta + \delta' + \hat{\delta}$ ,  $N_u = \frac{T}{n} + \sqrt{\frac{3}{n} T \ln(1/\hat{\delta})}$  and  $\gamma_n = 1 - (1 - \frac{1}{n})^{c^2 n^{2/3}}$ . For large enough  $n$  and  $T = \Omega(n)$ , this gives an asymptotic privacy amplification of  $O(1/n^{1/3})$  as in the case of real summation.  $\square$

**Remark 4** (Choice of amplification by shuffling bound). *We choose here to use the bound of [25] instead of [24] as it is more tight and holds under less restrictive assumptions. Indeed, with [24], we can only obtain some amplification when the number of users is very large ( $n > 10^5$ ).*

## F Proof of Theorem 4 (Stochastic Gradient Descent on a Complete Graph)

*Proof.* The proof tracks privacy loss using Rényi Differential Privacy (RDP) [42] and leverages results on amplification by iteration [26]. We first recall the definition of RDP and the main theorems that we will use. Then, we apply these tools to our setting and conclude by translating the resulting RDP bounds into  $(\varepsilon, \delta)$ -DP.

Rényi Differential Privacy quantifies the privacy loss based on the Rényi divergence between the outputs of the algorithm on neighboring databases.

**Definition 5** (Rényi divergence). *Let  $1 < \alpha < \infty$  and  $\mu, \nu$  be measures such that for all measurable set  $A$ ,  $\mu(A) = 0$  implies  $\nu(A) = 0$ . The Rényi divergence of order  $\alpha$  between  $\mu$  and  $\nu$  is defined as*

$$D_\alpha(\mu\|\nu) = \frac{1}{\alpha - 1} \ln \int \left( \frac{\mu(z)}{\nu(z)} \right)^\alpha \nu(z) dz.$$

In the following, when  $U$  and  $V$  are sampled from  $\mu$  and  $\nu$  respectively, with a slight abuse of notation we will often write  $D_\alpha(U\|V)$  to mean  $D_\alpha(\mu\|\nu)$ .

**Definition 6** (Rényi DP). *For  $1 < \alpha \leq \infty$  and  $\varepsilon \geq 0$ , a randomized algorithm  $\mathcal{A}$  satisfies  $(\alpha, \varepsilon)$ -Rényi differential privacy, or  $(\alpha, \varepsilon)$ -RDP, if for all neighboring data sets  $D$  and  $D'$  we have*

$$D_\alpha(\mathcal{A}(D)\|\mathcal{A}(D')) \leq \varepsilon.$$

We can introduce a notion of *Network-RDP* accordingly.

**Definition 7** (Network Rényi DP). *For  $1 < \alpha \leq \infty$  and  $\varepsilon \geq 0$ , a randomized algorithm  $\mathcal{A}$  satisfies  $(\alpha, \varepsilon)$ -network Rényi differential privacy, or  $(\alpha, \varepsilon)$ -NRDP, if for all pairs of distinct users  $u, v \in V$  and all pairs of neighboring datasets  $D \sim_u D'$ , we have*

$$D_\alpha(\mathcal{O}_v(\mathcal{A}(D))\|\mathcal{O}_v(\mathcal{A}(D'))) \leq \varepsilon.$$

As in classic DP, there exists composition theorems for RDP, see [42]. We will use the following.

**Proposition 2** (Composition of RDP). *If  $\mathcal{A}_1, \dots, \mathcal{A}_k$  are randomized algorithms satisfying  $(\alpha, \varepsilon_1)$ -RDP,  $\dots$ ,  $(\alpha, \varepsilon_k)$ -RDP respectively, then their composition  $(\mathcal{A}_1(S), \dots, \mathcal{A}_k(S))$  satisfies  $(\alpha, \sum_{l=1}^k \varepsilon_l)$ -RDP. Each algorithm can be chosen adaptively, i.e., based on the outputs of algorithms that come before it.*

Finally, we can translate the result of the RDP by using the following result [42].

**Proposition 3** (Conversion from RDP to DP).  *$f$   $\mathcal{A}$  satisfies  $(\alpha, \varepsilon)$ -Rényi differential privacy, then for all  $\delta \in (0, 1)$  it also satisfies  $(\varepsilon + \frac{\ln(1/\delta)}{\alpha-1}, \delta)$  differential privacy.*

Privacy amplification by iteration [26] captures the fact that for algorithms that consist of *iterative contractive updates*, not releasing the intermediate results improve the privacy guarantees for the final result. An important application of this framework is Projected Noisy Stochastic Gradient Descent (PNSGD) in the centralized setting, where the trusted curator only reveals the final model. More precisely, when iteratively updating a model with PNSGD, any given step is hidden by subsequent steps (the more subsequent steps, the better the privacy). The following result from [26] (Theorem 23 therein) formalizes this.

**Theorem 8** (Rényi differential privacy of PNSGD). *Let  $\mathcal{W} \in \mathbb{R}^d$  be a convex set,  $\mathcal{X}$  be an abstract data domain and  $\{f'; x\}_{x \in \mathcal{X}}$  be a family of convex  $L$ -Lipschitz and  $\beta$ -smooth function over  $\mathcal{K}$ . Let  $\text{PNSGD}(D, w_0, \eta, \sigma)$  be the algorithm that returns  $w_n \in \mathcal{W}$  computed recursively from  $w_0 \in \mathcal{W}$  using dataset  $D = \{x_1, \dots, x_n\}$  as:*

$$w_{t+1} = \Pi_{\mathcal{W}}(w_t - \eta(\nabla f(w_t; x_{t+1}) + Z)), \quad \text{where } Z \sim \mathcal{N}(0, \sigma^2 I_d).$$

*Then for any  $\eta \leq 2/\beta, \sigma > 0, \alpha > 1, t \in [n]$ , starting point  $w_0 \in \mathcal{K}$  and  $D \in \mathcal{X}^n$ , PNSGD satisfies  $(\alpha, \frac{\alpha\varepsilon}{n+1-t})$ -RDP for its  $t$ -th input, where  $\varepsilon = \frac{2L^2}{\sigma^2}$ .*

In our context, we aim to leverage this result to capture the privacy amplification provided by the fact that a given user  $v$  will only observe information about the update of another user  $u$  after some steps of the random walk. To account for the fact that this number of steps will itself be random, we will use the so-called weak convexity property of the Rényi divergence [26].

**Proposition 4** (Weak convexity of Rényi divergence). *Let  $\mu_1, \dots, \mu_m$  and  $\nu_1, \dots, \nu_m$  be probability distributions over some domain  $\mathcal{Z}$  such that for all  $i \in [m]$ ,  $D_\alpha(\mu_i\|\nu_i) \leq c/(\alpha - 1)$  for some  $c \in (0, 1]$ . Let  $\rho$  be a probability distribution over  $[m]$  and denote by  $\mu_\rho$  (resp.  $\nu_\rho$ ) the probability distribution over  $\mathcal{Z}$  obtained by sampling  $i$  from  $\rho$  and then outputting a random sample from  $\mu_i$  (resp.  $\nu_i$ ). Then we have:*

$$D_\alpha(\mu_\rho\|\nu_\rho) \leq (1 + c) \cdot \mathbb{E}_{i \sim \rho} [D_\alpha(\mu_i\|\nu_i)].$$

We now have all the technical tools needed to prove our result. Let us denote by  $\sigma^2 = \frac{8L^2 \ln(1.25/\delta)}{\varepsilon^2}$  the variance of the Gaussian noise added at each gradient step in Algorithm 4. Let us fix two distinct users  $u$  and  $v$ . We aim to quantify how much information about the private data of user  $u$  is leaked to  $v$  from the visits of the token. As in the proofs of Theorem 3 (real summation) and Theorem 6 (discrete histogram computation), we reason on the privacy loss induced by each contribution of user  $u$ .

Let us fix a contribution of user  $u$  at some time  $t_1$ . The view  $\mathcal{O}_v$  of user  $v$  on the entire procedure is defined as in the proof of Theorem 3. Note that the token values observed before  $t_1$  do not depend on the contribution of  $u$  at time  $t_1$ . Let  $t_2 > t_1$  be the first time that  $v$  receives the token posterior to  $t_1$ . It is sufficient to bound the privacy loss induced by the observation of the token at  $t_2$ : indeed, by the post-processing property of DP, no additional privacy loss with respect to  $v$  will occur for observations posterior to  $t_2$ .

By definition of the random walk,  $t_2$  follows a geometric law of parameter  $1/n$ , where  $n$  is the number of users. Additionally, if there is no time  $t_2$  (which can be seen as  $t_2 > T$ ), then no privacy loss occurs. Let  $Y_v$  and  $Y'_v$  be the distribution followed by the token when observed by  $v$  at time  $t_2$  for two neighboring datasets  $D \sim_u D'$  which only differ in the dataset of user  $u$ . For any  $t$ , let also  $X_t$  and  $X'_t$  be the distribution followed by the token at time  $t$  for two neighboring datasets  $D \sim_u D'$ . Then, we can apply Proposition 4 to  $D_\alpha(Y_v || Y'_v)$  with  $c = 1$ , which is ensured when  $\sigma \geq L\sqrt{2\alpha(\alpha - 1)}$ , and we have:

$$D_\alpha(Y_v || Y'_v) \leq (1 + 1)\mathbb{E}_{t \sim \mathcal{G}(1/n)} D_\alpha(X_t || X'_t).$$

We can now bound  $D_\alpha(X_t || X'_t)$  for each  $t$  using Theorem 8 and obtain:

$$\begin{aligned} D_\alpha(Y_v || Y'_v) &\leq \sum_{t=1}^{T-t_1} \frac{1}{n} \left(1 - \frac{1}{n}\right)^t \frac{2\alpha L^2}{\sigma^{2t}} \\ &\leq \frac{2\alpha L^2}{\sigma^2 n} \sum_{t=1}^{\infty} \frac{(1-1/n)^t}{t} \\ &\leq \frac{2\alpha L^2 \ln n}{\sigma^2 n}. \end{aligned}$$

To bound the privacy loss over all the  $T_u$  contributions of user  $u$ , we use the composition property of RDP, leading to the following Network RDP guarantee.

**Lemma 1.** *Let  $\alpha > 1$ ,  $\sigma \geq L\sqrt{2\alpha(\alpha - 1)}$  and  $T_u$  be maximum number of contributions of a user. Then Algorithm 3 satisfies  $(\alpha, \frac{4T_u \alpha L^2 \ln n}{\sigma^2 n})$ -Network Rényi DP.*

We can now convert this result into an  $(\varepsilon_c, \delta_c)$ -DP statement using Proposition 3. This proposition calls for minimizing the function  $\alpha \rightarrow \varepsilon_c(\alpha)$  for  $\alpha \in (1, \infty)$ . However, recall that from our use of the weak convexity property we have the additional constraint on  $\alpha$  requiring that  $\sigma \geq L\sqrt{2\alpha(\alpha - 1)}$ . This creates two regimes: for small  $\varepsilon_c$  (i.e, large  $\sigma$  and small  $T_u$ ), the minimum is not reachable, so we take the best possible  $\alpha$  within the interval, whereas we have an optimal regime for larger  $\varepsilon_c$ . This minimization can be done numerically, but for simplicity of exposition we can derive a suboptimal closed form which is the one given in Theorem 4.

To obtain this closed form, we reuse the result of [26] (Theorem 32 therein). In particular, for  $q = \max(\frac{2T_u \ln n}{n}, 2 \ln(1/\delta_c))$ ,  $\alpha = \frac{\sigma \sqrt{\ln(1/\delta_c)}}{L\sqrt{q}}$  and  $\varepsilon_c = \frac{4L\sqrt{q \ln(1/\delta_c)}}{\sigma}$ , the conditions  $\sigma \geq L\sqrt{2\alpha(\alpha - 1)}$  and  $\alpha > 2$  are satisfied. Thus, we have a bound on the privacy loss which holds the two regimes thanks to the definition of  $q$ .

Finally, we bound  $T_u$  by  $N_u = \frac{T}{n} + \sqrt{\frac{3T}{n} \log(1/\hat{\delta})}$  with probability  $1 - \hat{\delta}$  as done in the previous proofs for real summation and discrete histograms. Setting  $\varepsilon' = \varepsilon_c$  and  $\delta' = \delta_c + \hat{\delta}$  concludes the proof.  $\square$

**Remark 5** (Tighter numerical bounds). *As mentioned in the proof, we can compute a tighter bound for small  $\sigma$  when the optimal  $\alpha$  violates the constraints on  $\sigma$ . In this case, we set  $\alpha$  to its limit such that  $\sigma = L\sqrt{2\alpha(\alpha - 1)}$  and deduce a translation into  $(\varepsilon_c, \delta_c)$ -differential privacy. This is useful when  $q \neq \frac{2N_u \ln n}{n}$ , i.e., situations where the number of contributions of a user is smaller than the number of users.*

*In particular, we use this method in our experiments of Section 5.2. In that case, we have a fixed  $(\varepsilon_c, \delta_c)$ -DP constraint and want to find the minimum possible  $\sigma$  that ensures this privacy guarantee. We start with a small candidate for  $\sigma$  and compute the associated privacy loss as explained above. We then increase it iteratively until the resulting  $\varepsilon_c$  is small enough.*

## G Lifting the Assumption of Hidden Sender/Receiver

In the analysis of Section 4, we assumed that a user does not know the identity of the previous (sender) or next (receiver) user in the walk. We discuss here how we can lift this assumption. Our approach is based on separately bounding the privacy loss of contributions that are adjacent to a given user (*spotted contributions*), as these contributions do not benefit from any privacy amplification if the identity of the sender/receiver is known. We first compute the privacy loss resulting from spotted contributions, then discuss in which regimes this term becomes negligible in the total theoretical privacy loss, and finally how to deal with it empirically.

**Definition 8** (Spotted contribution). *For a walk on the complete graph, we define a spotted contribution of  $u$  with respect to  $v$  as a contribution of  $u$  that is directly preceded or followed by a contribution of  $v$ .*

A spotted contribution has a privacy loss bounded by  $\varepsilon$ , as we still have the privacy guarantee given by the local randomizer, but no further amplification of privacy. Thus, we need to bound the number of contributions for a given vertex  $u$  to be spotted by another user  $v$ . As in the proofs of Theorem 3, Theorem 4 and Theorem 6, we bound the number of contributions of  $u$  by  $N_u$  using Chernoff. Now, for each of these contributions, the probability of being spotted is  $2/n$ , so the number of spotted contributions follows a binomial law of parameter  $\mathcal{B}(N_u, 2/n)$ . We then use once again Chernoff to bound the number of spotted contributions with probability  $\tilde{\delta}$ , and use either simple or advanced composition. This leads to the following bound for the privacy loss associated with spotted contributions.

**Lemma 2** (Privacy loss of spotted contributions). *For a random walk with  $N_u$  contributions of user  $u$ , the privacy loss induced by spotted contributions is bounded with probability  $1 - \tilde{\delta}$  by:*

- $\varepsilon_s = \sqrt{\left(\frac{2N_u}{n} + \sqrt{\frac{6N_u}{n} \log(1/\tilde{\delta})}\right) \log(1/\tilde{\delta})} \varepsilon + \left(\frac{2N_u}{n} + \sqrt{\frac{6N_u}{n} \log(1/\tilde{\delta})}\right) \varepsilon(e^\varepsilon - 1)$  with advanced composition,
- $\varepsilon_s = \left(\frac{2N_u}{n} + \sqrt{\frac{6N_u}{n} \log(1/\tilde{\delta})}\right) \varepsilon$  with simple composition.

The above term (along with  $\tilde{\delta}$ ) should be added to the total privacy loss to take into account the knowledge of the previous and next user. The difficulty comes from the fact that the number of spotted contributions has a high variance if the number of contributions per user is small compared to the number of users. We already observed this when bounding the number of contributions per user, where the worst case is far from the expected value (see Figure 1a). Here, the price to pay is higher as the square root dominates the expression in the regime where  $T = o(n^2)$ . However, for  $T = \Omega(n^2)$ , the spotted contribution term becomes negligible and we recover the same order of privacy amplification as in Theorem 3, Theorem 4 and Theorem 6.

The derivations above provide a way to bound the impact of spotted contributions theoretically, but we can also deal with it empirically. In practical implementations, one can also enforce a bound on the number of times that an edge can be used, and dismiss it afterwards, with limited impact on the total privacy loss. Another option to keep the same formal guarantees is to replace real contributions with only noise when an edge has exceeded the bound. These “fake contributions” seldom happen in practice and thus do not harm the convergence.

## H Additional Experiments

Similar to Section 5.1, we run experiments to investigate the empirical behavior of our approach for the task of discrete histogram computation on the complete graph by leveraging results on privacy amplification by shuffling. Here, we have used the numerical approach from [7] to tightly measure the effect of amplification by shuffling based on the code provided by the authors.<sup>3</sup> Figure 3 confirm that the empirical gains from privacy amplification by decentralization are also significant for this task.

<sup>3</sup><https://github.com/BorjaBalle/amplification-by-shuffling>

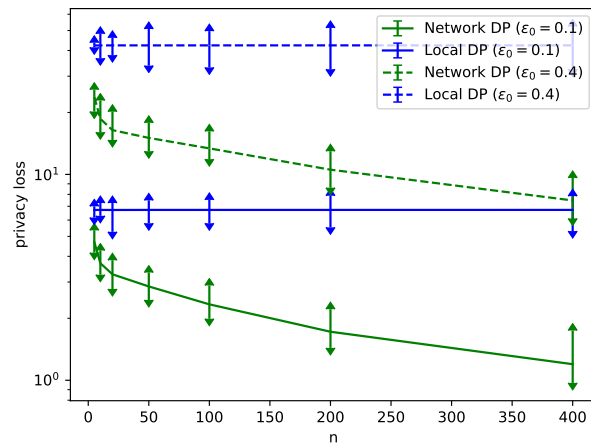


Figure 3: Comparing network and local DP on the task of computing discrete histograms. The results are obtained for  $T = 100n$  (i.e., the expected number of contributions per user is 100). The value of  $\epsilon_0$  rules the amount of local noise added to each contribution (i.e., each single contribution taken in isolation satisfied  $\epsilon_0$ -LDP). Curves report the average privacy loss across all pairs of users and all 10 random runs, while their error bars give the best and worst cases.