Privacy Amplification by Decentralization

Edwige Cyffers, Aurélien Bellet

Privacy Amplification by Decentralization

Edwige Cyffers
ENS Lyon, France
first.last@ens-lyon.fr

Aurélien Bellet
Inria, France
first.last@inria.fr

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 protocols, i.e., when participants exchange information by communicating along the edges of a network graph. This relaxation, that we call network DP, captures the fact that users have only a local view of the decentralized 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 significantly improve upon LDP, and in some cases even match what can be achieved with methods based on trusted/secure aggregation and shuffling. Our experiments illustrate the superior utility of our approach when training a machine learning model with stochastic gradient descent.

1 Introduction

With the 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 popularity of federated learning (Kairouz et al., 2019), in which each data subject (or data provider) keeps her/his own data and only shares results of local computations with a central coordinator. Fully decentralized variants of federated learning remove the need for a central coordinator and instead rely on peer-to-peer communications along edges of a network graph, see e.g., (Lian et al., 2017; Bellet et al., 2018) and (Kairouz et al., 2019, Section 2.1 therein) for an overview. The standard motivation for such fully decentralized approaches is efficiency and scalability: indeed, the central coordinator can represent a bottleneck, especially when the number of participants is large (Lian et al., 2017).

In many applications involving personal or business-related information, the participants want to keep their raw data private. 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 (Shokri et al., 2017). In fact, federated learning provides an additional attack surface as the participants share intermediate updates (Nasr et al., 2019; Geiping et al., 2020). To control the privacy leakage, the prominent approach is based on the standard notion of Differential Privacy (DP) (Dwork et al., 2006b). 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.

1.1 Related Work

Several trust models can be considered in federated learning and lead to different trade-offs between privacy and utility. The strongest model is local differential privacy (LDP) (Kasiviswanathan et al.,

2008; Duchi et al., 2013), where each participant (user) does not trust anyone and assumes that an adversary may observe everything that she/he shares. Unfortunately, this model 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 model of DP (Chan et al., 2012a). The fundamental limits of machine learning under LDP have been studied in (Zheng et al., 2017; Wang et al., 2018).

These limitations have motivated the study of intermediate trust models, where LDP is relaxed to improve utility while still avoiding the need for a trusted curator. A popular approach is to resort to cryptographic primitives to securely aggregate user contributions (Dwork et al., 2006a; Shi et al., 2011; Bonawitz et al., 2017; Chan et al., 2012b; Jayaraman et al., 2018). Recent work has also considered the so-called shuffle model of DP (Cheu et al., 2019; Erlingsson et al., 2019; Balle et al., 2019b,a; Ghazi et al., 2020; Feldman et al., 2020), where users send their contribution to a trusted/secure shuffler which permutes the set of messages so as to hide their source. While these relaxations can provably lead to significant improvements in the privacy-utility trade-off, their practical implementation poses important challenges (especially for large numbers of users). They also do not integrate easily with fully decentralized algorithms that involve exchanges between small subsets of users (i.e., neighbors in the network graph) that are performed in a potentially asynchronous fashion.

Recent work has studied schemes that can "amplify" the DP guarantees of a private algorithm when combined with it. Beyond privacy amplification by shuffling (Erlingsson et al., 2019; Balle et al., 2019b; Feldman et al., 2020), which is based on the shuffling primitive mentioned above, we can mention amplification by subsampling (Balle et al., 2018) and amplification by iteration (Feldman et al., 2018). However, these schemes are generally difficult to apply in a federated/decentralized setting: the former requires that the identity of the subsampled participants remain secret, while the latter assumes that only the final model is revealed.

1.2 Our Contributions

In this work, we propose a *novel relaxation of LDP where users have only a local view of the decentralized system*. More precisely, our notion of *network differential privacy* captures the fact that each user only observes information received from her/his neighbors in the network graph, which is a natural assumption in fully decentralized settings. Network DP can also account for 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 (Ram et al., 2009; Johansson et al., 2009; Mao et al., 2020; Ayache and Rouayheb, 2020). 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 approaches based on secure aggregation and secure shuffling while requiring only on a small number of secure communication channels.

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 $O(1/n^{1/4})$ compared to the same algorithm analyzed 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 (Balle et al., 2018), shuffling (Erlingsson et al., 2019; Balle et al., 2019b; Feldman et al., 2020) and iteration (Feldman et al., 2018) in a novel decentralized context. This is made possible by the restricted view of participants captured by our notion of network DP. At the empirical level, we

show through numerical experiments that privacy gains are significant in practice both for simple analytics and for training machine learning 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 decentralization,* i.e., when users communicate in a peer-to-peer fashion instead of relying on a central (untrusted) aggregator for all communications. 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 many promising perspectives, which we outline in the conclusion.

1.3 Paper Outline

The rest of the paper is organized as follows. Section 2 introduces the problem setting, our notion of network DP, as well as 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 Setting

In this section, we define our main notations and introduce network differential privacy (network DP), our novel relaxation of DP. Then we describe the family of decentralized protocols that we will study under network DP.

2.1 Network Differential Privacy

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 $D = D_1 \cup \cdots \cup D_n$ 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 (McMahan et al., 2018). 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. A decentralized algorithm $\mathcal{A}$ takes as input a dataset D and outputs the transcript of all exchanges between users over the network (i.e., the sender, receiver and content of all messages). Assuming that users communicate over secure channels, *a given user does not have access to the full transcript $\mathcal{A}(D)$ but only to her/his local memory and the messages received*: we denote the corresponding view of user u by $\mathcal{O}_u(\mathcal{A}(D))$. Using these notations, we introduce our notion of network DP.

Definition 1 (Network DP)**.** *An algorithm $\mathcal{A}$ satisfies (ε, δ)-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:*

$$\mathbb{P}(\mathcal{O}_v(\mathcal{A}(D))) \leq e^{\varepsilon}\mathbb{P}(\mathcal{O}_v(\mathcal{A}(D'))) + \delta. \tag{1}$$

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 combination 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. 1 boils down to local DP.

It is possible to 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, ε, δ)-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:*

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

2.2 Decentralized Computation on a Walk

In this work, we study network DP for decentralized protocols in which computation is done via sequential updates to a *token* τ walking through the nodes by following the edges of the graph G. At each step, the token τ resides at some node u and is updated by

$$\tau \leftarrow \tau + x_u^k, \quad \text{with } x_u^k = g^k(\tau; D_u), \tag{3}$$

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 τ of the token as well as on the number of times k that the token visited u so far.

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 (Ram et al., 2009; Johansson et al., 2009; Mao et al., 2020; Ayache and Rouayheb, 2020) and hence to train machine learning models. In this case, the token τ holds the model parameters and x_u^k is a (stochastic) gradient of the local loss function of user u evaluated at τ.

Such decentralized protocols 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 Walking 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 Perturb$(x; \sigma)$ which adds centered noise with standard deviation σ to the contribution x (e.g., the Gaussian or Laplace mechanism). Let σ_{loc} be the standard deviation of the noise required so that Perturb$(\cdot; \sigma_{loc})$ satisfies (ε, δ)-LDP.

Consider now the simple decentralized protocol in Algorithm 1, where noise with the same standard deviation σ_{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 (Dwork et al., 2010), we have the following result (see Appendix A for the proof).

Theorem 1. *Let $\varepsilon, \delta > 0$. Algorithm 1 outputs an unbiased estimate of $\bar{x}$ with standard deviation $\sqrt{\lfloor Kn/(n-1) \rfloor}\sigma_{loc}$. Furthermore, it satisfies $(\sqrt{2K\ln(1/\delta')}\varepsilon + K\varepsilon(e^{\varepsilon} - 1), K\delta + \delta')$-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 ε. In fact, Algorithm 1 achieves the same privacy-utility trade-off as a *trusted* aggregator that would iteratively aggregate the contributions of each round k and perturb the result before sending it back to the users, as done to aggregate iterative user updates in federated learning algorithms with a trusted server (Kairouz et al., 2019).

Algorithm 1 Private real summation on the ring.

$\tau \leftarrow 0$; $a \leftarrow 0$
for $k = 1$ to K **do**
  for $u = 1$ to n **do**
    if $a = 0$ **then**
      $\tau \leftarrow \tau + \text{Perturb}(x_u^k; \sigma_{loc})$
      $a = n - 2$
    else
      $\tau \leftarrow \tau + x_u^k$
      $a \leftarrow a - 1$
return τ

Algorithm 2 Private histogram computation on the ring.

Init. $\tau \in \mathbb{N}^L$ with γn uniformly random elements
for $k = 1$ to K **do**
  for $u = 1$ to n **do**
    $y_u^k \leftarrow RR_\gamma(x_u^k)$
    $\tau[y_u^k] \leftarrow \tau[y_u^k] + 1$
for $i = 0$ to $L - 1$ **do**
  $\tau[i] \leftarrow \frac{\tau[i] - \gamma/L}{1-\gamma}$
return τ

Remark 1. *In Algorithm 1, a single user is responsible for adding the necessary noise in each cycle. Alternatively, it is possible to design variants in which noise addition is distributed across all users. Assuming Gaussian noise for simplicity, each user can for instance add noise with standard deviation $\sigma'_{loc} = \sigma_{loc}/\sqrt{n}$, except for the very first contribution which requires standard deviation σ_{loc} to properly hide the contributions of users in the first cycle. Taking into account this extra noise, the total added noise has standard deviation $\sqrt{\lfloor Kn/(n-1) \rfloor + 1}\sigma_{loc}$. This leads 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 histogram computation over a discrete domain $[L] = \{1, \ldots, L\}$ composed of L elements. The goal is to compute $h \in \mathbb{N}^L$ where $h_l = \sum_{u=1}^{n} \sum_{k=1}^{K} \mathbb{I}[x_u^k = l]$, where each $x_u^k \in [L]$. A classic approach to this problem in LDP is L-ary randomized response (Kairouz et al., 2014), where a user submits its true value with probability $1 - \gamma$ and a uniformly random value with probability γ. We denote this primitive by $RR_\gamma : [L] \to [L]$.

In our setting with a ring network, we propose Algorithm 2, where each contribution of a user is randomized using RR_γ before being added to the token $\tau \in \mathbb{N}^L$. Additionally, τ 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 to leverage results on *privacy amplification by shuffling* (Erlingsson et al., 2019; Balle et al., 2019b; Feldman et al., 2020). In particular, we can prove the following utility and privacy guarantees for Algorithm 2 (see Appendix B for the proof).

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

Theorem 2 shows that for the same amount of noise (fixed utility), 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 simplicity of presentation, Theorem 2 relies on the amplification by shuffling result of (Erlingsson et al., 2019) which has a simple closed-form. A tighter and more general result (with*

Algorithm 3 Private real summation on a complete graph.

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

milder restrictions on the values of n, ε and δ) can be readily obtained by using the results of (Balle et al., 2019b; Feldman et al., 2020).

3.3 Discussion

We have seen that computing over a decentralized ring provides a simple way to achieve utility similar to a trusted/secure aggregator or shuffler thanks to the sequential communication that hides the contribution of the previous users in a summary. We stress the fact that secure aggregation and secure shuffling are non-trivial secure multi-party computation protocols which can pose implementation and scalability challenges (Bonawitz et al., 2017). In contrast, our approach is very simple as it only requires to establish two secure communication channels per user.

Despite these important advantages, the use of a fixed ring topology has some limitations. First, our algorithms are not robust to collusions. The non-collusion assumption is essential in Algorithm 1: if two users collude and share their view, the algorithm does not satisfy DP (when it is the turn of one of the colluding users to add the protecting noise, there is no protection left). While this can be mitigated by distributing the noise addition across all users (Remark 1), a node placed right between two colluded nodes (or with few honest users in-between) would suffer largely degraded privacy guarantees. A similar reasoning holds for the histogram case (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 (Feldman et al., 2018). In this amplification scheme, 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 Walking on a 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.

4.1 Real Summation

For real summation, we propose a simple protocol (Algorithm 3): a user u receiving the token τ for the k-th time updates it with $\tau \leftarrow \tau + \text{Perturb}(x_u^k; \sigma_{loc})$ such that $\text{Perturb}(\cdot; \sigma_{loc})$ satisfies (ε, δ)-LDP. We prove a privacy amplification of $O(1/n^{1/4})$ compared to the same algorithm analyzed under LDP, which relies on *the intermediate aggregations of values between two visits of the token to a given user* and *the secrecy of the path of the token*.

Theorem 3. *Let $\varepsilon, \delta > 0$. Algorithm 3 achieves $(\varepsilon', \frac{T}{n}\delta + \delta' + \hat{\delta})$-network DP for all $\delta', \hat{\delta} > 0$ with*

$$\begin{aligned} \varepsilon' &= \sqrt{2N_v \ln(1/\delta')}\tfrac{\sqrt{2}\varepsilon}{n^{1/4}} + 2\sqrt{2N_v\gamma_n \ln(1/\delta')}\varepsilon \\ &\quad + N_v \tfrac{\sqrt{2}\varepsilon}{n^{1/4}}\left(e^{\frac{\sqrt{2}\varepsilon}{n^{1/4}}} - 1\right) + 4N_v\gamma_n\varepsilon(e^\varepsilon - 1), \end{aligned}$$

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

Sketch of proof. We summarize here the main ingredients of the proof (see Appendix C for details). We fix a user v and quantify how much information about the private data of another user u is leaked

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: $Z = [Z_1, \dots, Z_d]$, with $Z_i \sim \mathcal{N}\big(0, \frac{8L^2 \ln(1.25/\delta)}{\varepsilon^2}\big)$
5: $\tau \leftarrow \Pi_{\mathcal{W}}(\tau - \eta(\nabla_\tau f(\tau; D_u) + Z))$
6: **return** τ

to v from the visits of the token. The number of visits to v follows a binomial law $\mathcal{B}(T, 1/n)$: we can bound it by N_v with probability $1 - \hat{\delta}$ using Chernoff. Between two visits to v, the number of steps in the walk follows a geometric law of parameter $1 - 1/n$. We distinguish between "small" cycles of less of $\sqrt{n}/2$ steps and larger ones. Using Hoeffding's inequality, we bound the number of small cycles by $2N_v(1 - (1 - \frac{1}{n})^{\sqrt{n}/2})$ and assign a privacy loss of at most 2ε to each of these small cycles (this worst case is reached for the sequence $v - u - u - v$). For the larger cycles, the contribution of u is aggregated with at least $\sqrt{n}/2$ others, leading to a privacy loss of at most $\sqrt{2}\varepsilon/n^{1/4}$. Crucially, this holds even though the token may go through u more than once. Indeed, since the cycle is secret to v, it can be seen as subsampling with replacement $\sqrt{n}/2$ users among $n - 1$ choices. We use amplification by subsampling (Balle et al., 2018) to bound the privacy loss by the case where u contributes once. The total privacy loss across the N_v visits to v follows from advanced composition. □

While the privacy gain is not as strong as the one obtained for the fixed ring topology, we will see in Section 5 that the bound in Theorem 3 improves upon local DP as soon as $n > 100$ (see Figure 1a). We will also see, via numerical simulations, that the gains are significantly stronger in practice than what our theoretical bound guarantees (Figure 1b).

4.2 Discrete Histogram Computation

We can obtain a similar privacy amplification result for histogram computation by bounding the privacy loss incurred by larger cycles in the proof of Theorem 3 using amplification by shuffling (Erlingsson et al., 2019; Balle et al., 2019b; Feldman et al., 2020), similar to what we did for the ring (Section 3.2). Details are in Appendix D.

4.3 Optimization with Stochastic Gradient Descent

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 β-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:

$$w^* \in \arg\min_{w \in \mathcal{W}} \Big\{ F(w) := \frac{1}{n} \sum_{u=1}^{n} f(w; D_u) \Big\}. \tag{4}$$

This formulation encompasses many machine learning tasks in the empirical risk minimization framework, e.g., ridge and logistic regression, support vector machines, 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 with 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 (ε, δ)-LDP: the variance σ^2 of the noise in line 4 of Algorithm 4 follows from the fact that gradients of L-Lipschitz functions have sensitivity bounded by $2L$ (Bassily et al., 2014). We now prove a network-DP guarantee for Algorithm 4.

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

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

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

Sketch of proof. The proof tracks the evolution of the privacy loss using Rényi Differential Privacy (RDP) (Mironov, 2017) and leverages amplification by iteration (Feldman et al., 2018) in a novel decentralized context. We give here a brief sketch (see Appendix E for details). Let us fix two users u and v and bound the privacy leakage of u from the point of view of v. We again bound the number of contributions N_u of a user, but unlike in the proof of Theorem 3 we apply this result to the user releasing information (namely u). We then compute the network-RDP guarantee for a fixed contribution of u at time t. Crucially, it is sufficient to take into account the first time that 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 proved in (Feldman et al., 2018, Lemma 25 therein), 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 conclude by applying the composition property of RDP over the N_u contributions of u and converting the RDP guarantee into (ε, δ)-DP. □

Theorem 4 gives a privacy amplification of $O(\ln n/\sqrt{n})$ compared to LDP when the total number of iterations $T = \Omega\big(\frac{n^2\sqrt{\ln(1/\delta)}}{\ln n}\big)$. In other words, measuring utility as the amount of noise added to the gradients, the privacy-utility trade-off of Algorithm 4 is nearly the same (up to a logarithmic factor) as that of private SGD in the trusted curator model![1] For smaller T, the amplification is still much stronger than the closed-form given in Theorem 4: we can numerically find the smallest ε' that satisfy the conditions required in the proof, see Appendix E 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 theorem by Shamir and Zhang (2013, 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. *Assume that the diameter of $\mathcal{W}$ is 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 step size $\eta = D/G\sqrt{t}$. Then:*

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

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

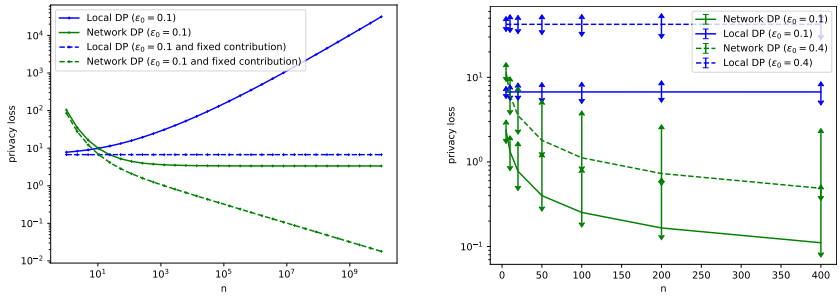
4.4 Robustness to Collusion

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 of $\frac{c}{n}$ instead of $\frac{1}{n}$. We can then easily adapt the proofs above using the fact that the total number of visits to colluding users follows $\mathcal{B}(T, c/n)$ and that 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.

5 Experiments

In this section, we present some numerical experiments that illustrate the significance of privacy amplification by decentralization in the complete graph setting (Section 4).

[1] Incidentally, the analysis of centralized private SGD also sets the number of iterations to be of order n^2, see (Bassily et al., 2014).



(a) Real summation (theoretical bounds) (b) Real summation (empirical)

Figure 1: Comparing network and local DP on the task of real summation. The results are obtained for $T = 100n$ (i.e., the expected number of contributions per user is 100). The value of ε_0 rules the amount of local noise added to each contribution (i.e., each single contribution taken in isolation satisfies ε_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, while their error bars give the best and worst cases.

5.1 Real Summation

Comparison of theoretical bounds. We numerically evaluate the theoretical bound of Theorem 3 for the task of real summation (Algorithm 3) and compare it to local DP. Recall that the number of contributions made by a user is random (with expected value T/n). Therefore, to provide a fair comparison between network DP and local DP, we derive an analogue of Theorem 3 for local DP where we use the same Chernoff bound to control the number contributions of a user as well as advanced composition to measure the total privacy loss. In addition, to isolate the effect of the number of contributions (which is the same in both network DP and local DP), we also report the bounds obtained under the assumption that each user contributes a fixed number of times T/n. Figure 1a plots the value of the bounds for varying n. We see that our theoretical result provides gains compared to local DP as soon as $n > 100$, and these gains become more and 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 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 leads to some approximations. However, in practical deployments one could instead use the actual values of these quantities to compute the privacy loss. We thus investigate the gap between our theoretical privacy guarantees and what can be obtained in practice by running simulations of random walks. Specifically, we sample a random walk of size $T = 100n$. Then, for each pair of users, we compute the privacy loss based on the characteristics of 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 (see Figure 1a). In particular, the gains are significant even for small n. These results again suggest that there is room for improvement in our analysis, for instance by resorting to better concentration bounds.

We have run similar experiments for discrete histogram computation which confirm that the empirical gains from privacy amplification by decentralization are also significant for this task, see Appendix F.

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. 4 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

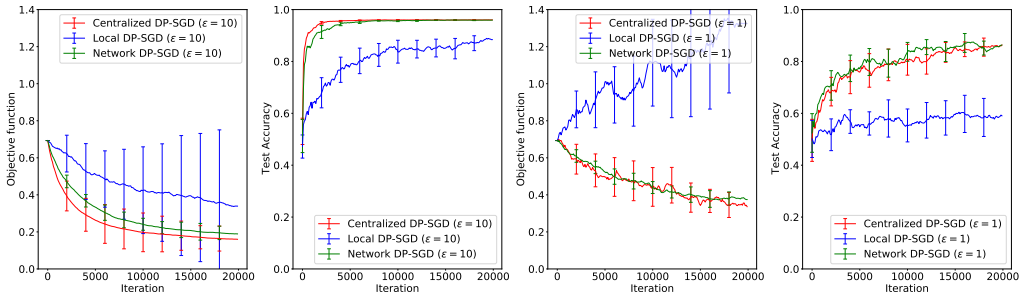


Figure 2: Comparing three settings for SGD with gradient perturbation. Centralized DP-SGD requires a trusted curator and benefits from amplification by subsampling to add less noise. In contrast, local DP-SGD requires a large amount of noise. Network DP nearly bridges the gap between Centralized and Local DP-SGD. The three methods have σ parametrized to ensure $\varepsilon = 10$ for the two left plots (resp. 1 for the right ones) and $\delta = 10^{-6}$. Plots show the mean and standard deviation of the objective function and test accuracy over 20 runs.

use a binarized version of UCI Housing dataset.[2] 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 user u having a local dataset D_u of size 8.

We compare three private SGD approaches which are all based on gradient perturbation with the Gaussian mechanism. *Centralized DP-SGD* is the centralized version of differentially private SGD (*Centralized DP-SGD*) introduced in (Bassily et al., 2014), 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 desired privacy budget (ε, δ) for the whole procedure, each of the three methods however leads to a different choice for σ that parametrizes the level of noise added to each gradient. In our experiments, we fix $\varepsilon = 10$ (low privacy regime) and $\varepsilon = 1$ (stronger privacy regime) and $\delta = 10^{-6}$. We recall that consistently with the rest of the paper we consider the user-level neighborhood relation ($X \sim_u X'$ when they differ in the local database of one user u). Note that due to composition, doing more iterations increase the per-iteration level of noise needed to achieve a fixed DP guarantee. Recall also that the number of contributions of a given user is random. To upper bound this in advance, we use 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 there is nonetheless a privacy amplification effect which is stronger than the closed form of the theorem. In practice, we compute numerically the smallest σ needed to fulfill the conditions of the proof, as explained in Appendix E.

Figure 2 shows results for $T = 20000$, where the step size η was tuned separately for each approach in the interval $[10^{-4}, 2]$. We clearly see that Network DP-SGD nearly matches the privacy-utility trade-off of Centralized DP-SGD for both $\varepsilon = 1$ and $\varepsilon = 10$ without relying on a trusted curator. Network DP-SGD also clearly outperforms Local DP-SGD, which actually diverges for $\varepsilon = 1$. These empirical results are consistent with our theory and show that Network DP-SGD significantly amplifies privacy guarantees 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.

[2]Download link: `https://www.openml.org/d/823`

6 Perspectives

We believe that our work opens many interesting perspectives. Aside from making our bounds for the complete graph setting more tight, 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 ring (and more generally sparse) topologies. Other decentralized models of computation can also be studied under network DP with the goal of proving privacy amplification results. In particular, a natural extension of the protocols studied here is to consider multiple tokens walking on the graph in parallel. We would also like to study randomized gossip algorithms (Boyd et al., 2006), as these protocols are popular for decentralized optimization in machine learning (see Colin et al., 2016, and references therein) and were recently shown to provide differential privacy guarantees in the specific context of rumor spreading (Bellet et al., 2020). 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 in the network than in more distant users.

Acknowledgments and Disclosure of Funding

A.B. was supported by grants ANR-16-CE23-0016-01 and ANR-18-CE23-0018-03, by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No.825081 COMPRISE and by a grant from CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020.

References

Ayache, G. and Rouayheb, S. E. (2020). Private Weighted Random Walk Stochastic Gradient Descent. Technical report, arXiv:2009.01790.

Balle, B., Barthe, G., and Gaboardi, M. (2018). Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *NeurIPS*.

Balle, B., Bell, J., Gascón, A., and Nissim, K. (2019a). Differentially Private Summation with Multi-Message Shuffling. Technical report, arxiv:1906.09116.

Balle, B., Bell, J., Gascón, A., and Nissim, K. (2019b). The Privacy Blanket of the Shuffle Model. In *CRYPTO*.

Bassily, R., Smith, A. D., and Thakurta, A. (2014). Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *FOCS*.

Bellet, A., Guerraoui, R., and Hendrikx, H. (2020). Who started this rumor? Quantifying the natural differential privacy guarantees of gossip protocols. In *DISC*.

Bellet, A., Guerraoui, R., Taziki, M., and Tommasi, M. (2018). Personalized and Private Peer-to-Peer Machine Learning. In *AISTATS*.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *CCS*.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE/ACM Transactions on Networking*, 14(SI):2508–2530.

Chan, T.-H. H., Shi, E., and Song, D. (2012a). Optimal Lower Bound for Differentially Private Multi-party Aggregation. In *ESA*.

Chan, T.-H. H., Shi, E., and Song, D. (2012b). Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography*.

Cheu, A., Smith, A. D., Ullman, J., Zeber, D., and Zhilyaev, M. (2019). Distributed Differential Privacy via Shuffling. In *EUROCRYPT*.

Colin, I., Bellet, A., Salmon, J., and Clémençon, S. (2016). Gossip Dual Averaging for Decentralized Optimization of Pairwise Functions. In *ICML*.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2013). Local privacy and statistical minimax rates. In *FOCS*.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *EUROCRYPT*.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography (TCC)*.

Dwork, C., Rothblum, G. N., and Vadhan, S. (2010). Boosting and Differential Privacy. In *FOCS*.

Erlingsson, U., Feldman, V., Mironov, I., Raghunathan, A., and Talwar, K. (2019). Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *SODA*.

Feldman, V., McMillan, A., and Talwar, K. (2020). Hiding Among the Clones: A Simple and Nearly Optimal Analysis of Privacy Amplification by Shuffling. Technical report, arXiv:2012.12803.

Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. (2018). Privacy Amplification by Iteration. In *FOCS*.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients - how easy is it to break privacy in federated learning? In *NeurIPS*.

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., Pagh, R., and Velingker, A. (2020). Pure Differentially Private Summation from Anonymous Messages. Technical report, arXiv:2002.01919.

Jayaraman, B., Wang, L., Evans, D., and Gu, Q. (2018). Distributed learning without distress: Privacy-preserving empirical risk minimization. In *NeurIPS*.

Johansson, B., Rabi, M., and Johansson, M. (2009). A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2019). Advances and Open Problems in Federated Learning. Technical report, arXiv:1912.04977.

Kairouz, P., Oh, S., and Viswanath, P. (2014). Extremal mechanisms for local differential privacy. In *NIPS*.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. D. (2008). What Can We Learn Privately? In *FOCS*.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. (2017). Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In *NIPS*.

Mao, X., Yuan, K., Hu, Y., Gu, Y., Sayed, A. H., and Yin, W. (2020). Walkman: A Communication-Efficient Random-Walk Algorithm for Decentralized Optimization. *IEEE Transactions on Signal Processing*, 68:2513–2528.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. (2018). Learning Differentially Private Recurrent Language Models. In *ICLR*.

Mironov, I. (2017). Rényi Differential Privacy. In *CSF*.

Nasr, M., Shokri, R., and Houmansadr, A. (2019). 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*.

Ram, S., Nedić, A., and Veeravalli, V. (2009). Incremental stochastic subgradient algorithms for convex optimization. *SIAM Journal on Optimization*, 20(2):691–717.

Shamir, O. and Zhang, T. (2013). Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes. In *ICML*.

Shi, E., Chan, T.-H. H., Rieffel, E. G., Chow, R., and Song, D. (2011). Privacy-Preserving Aggregation of Time-Series Data. In *NDSS*.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*.

Wang, D., Gaboardi, M., and Xu, J. (2018). Empirical Risk Minimization in Non-interactive Local Differential Privacy Revisited. In *NeurIPS*.

Zheng, K., Mou, W., and Wang, L. (2017). Collect at Once, Use Effectively: Making Non-interactive Locally Private Learning Possible. In *ICML*.

Appendix 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 σ_{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 of v (i.e., the information observed by v during the execution of the protocol) corresponds to the K values of the token that she receives. We denote these values by $\tau_1^v, \ldots, \tau_K^v$, each of them corresponding to user contributions aggregated along with random noise. We define the view of v accordingly as:

$$\mathcal{O}_v(\mathcal{A}(D)) = \{\tau_i^v\}_{i=1}^K . \tag{5}$$

Let us fix $i \in \{2, \ldots, K\}$. 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 (ε, δ)-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 τ_1^v, note it also contains noise with standard deviation σ_{loc} added by the first user, so the same guarantee holds.

Finally, we apply the advanced composition theorem (Dwork et al., 2010) 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. □

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

Proof. The proof is similar in spirit to the real summation case (see Appendix 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 γ of adding a random response, plus the γn 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. 5, except that each $\tau_i^v \in \mathbb{N}^L$ is now a histogram over the domain $[L]$. More specifically, for $i \in \{2, \ldots, K\}$, the difference $\tau_{i+1}^v - \tau_i^v$ between two consecutive tokens is now a discrete histogram of n answers obtained by RR_γ (each of them is random with probability γ). Similarly, in the first round, the token is initialized with γn 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 (Erlingsson et al., 2019) 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 ε_0-LDP, the shuffling mechanism is (ε, δ)-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 γ so that RR_γ satisfies $12\varepsilon\sqrt{\frac{\log(1/\delta)}{n}}$-LDP, leading to an (ε, δ)-DP guarantee after shuffling. We conclude by the application of advanced composition (Dwork et al., 2010). □

Appendix C Proof of Theorem 3 (Real Summation on the Complete Graph)

Proof. 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.

Here, contrary to the case of the ring, the path followed by the token is private, so we need to account for what is known to v about this path in the definition of $\mathcal{O}_v$. We consider that a user knows the identity of the users from which she receives the token and the identity of the receiver to which she sends it. We further assume that the global time counter t is known to the users so that a user receiving the token knows how many users have added contributions to the token since its last visit.

We thus define the view $\mathcal{O}_v$ of user v by:

$$\mathcal{O}_v(\mathcal{A}(D)) = \bigcup_{t=1}^{T_v} (u_{k_i-1}, \tau_{k_i}, u_{k_i+1}), \tag{6}$$

where k_i is the i-th time that v receives the token, τ_{k_i} the corresponding value of the token, u_{k_i-1} the user who sent it to v, u_{k_i+1} the user who then received it from v, and T_v the number of times that v had the token during the whole execution of the protocol.

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 v at any given step. Thus, the number of visits T_v to v follows a binomial law $\mathcal{B}(T, 1/n)$. We can bound it by N_v with probability $1 - \hat{\delta}$ using Chernoff, where

$$\mathbb{P}(|T_v - \frac{T}{n}| > x) \leq 2e^{-\frac{2y^2}{T}} = \hat{\delta},$$

with $N_v = \frac{T}{n} + y$.

Between two visits to v, the number of steps follows a geometric law of parameter $1 - 1/n$. We will distinguish between "small" cycles of less of $\sqrt{n}/2$ steps and the larger ones, and bound the privacy loss incurred for each type of cycle.

The size of a fixed cycle follows a geometric law of parameter $\frac{1}{n}$. Let $\mathbf{1}_v^i$ be the indicator variable of the i-th cycle to v being "small". We have:

$$\mathbb{P}(\mathbf{1}_v^i = 0) = (1 - \frac{1}{n})^{\sqrt{n}/2}.$$

We can bound the number of small cycles $\sum_{i=1}^{N_v} \mathbf{1}_v^i$ by using Hoeffding again, as this amounts to upper-bounding a Bernoulli variable of parameter $\mathcal{B}(N_v, \gamma_n)$ with $\gamma_n = 1 - (1 - \frac{1}{n})^{\sqrt{n}/2}$. Therefore:

$$\mathbb{P}\left(\sum_{i=1}^{N_v} \mathbf{1}_v^i \geq 2N_v\gamma_n\right) \leq e^{\frac{-N_v\gamma_n}{2}}. \tag{7}$$

For each of these small cycles, we can bound the privacy loss suffered by u with respect to v by 2ε: this worst case is reached for the cycle $v - u - u - v$. Indeed, except for the two extreme users in the cycle (who are known to v, see Eq.6), the other ones can be seen as the result of subsampling with replacement. Therefore, the amplification by subsampling result given in Theorem 10 of (Balle et al., 2018) applies: for n the number of users and m the size of the cycle, it gives

$$\varepsilon' = \log(1 + (1 - (1 - 1/n)^m)(e^\varepsilon - 1),$$

where ε is the level of privacy guaranteed by the local mechanism Perturb$(\cdot; \sigma_{loc})$.

For the larger cycles, the contribution of u is aggregated with at least $\sqrt{n}/2$ others, leading to a privacy loss of at most $\sqrt{2}\varepsilon/n^{1/4}$, because the random perturbations added by the different users sum up. Crucially, this holds even though the token may go through u more than once due to the fact that $\varepsilon' \leq \varepsilon$ in the amplification by subsampling result of Eq. 7.

Regarding the parameter δ' of DP, the privacy amplification by sampling result of (Balle et al., 2018) gives

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

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 τ

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:

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

As the total of the cycles is upper bounded by T, we can bound the total delta by $\frac{T}{n}\delta$, , where δ is the level provided by the (ε, δ)-LDP guarantee of Perturb$(\cdot; \sigma_{loc})$.

Finally, to bound the privacy loss over the full view (Eq. 6), we use the advanced composition theorem (Dwork et al., 2006b) on each of the two types of cycles, which gives for the final ε':

$$\sqrt{2N_v \log(1/\delta')}\frac{\sqrt{2}\varepsilon}{n^{1/4}} + 2\sqrt{2N_v\gamma_n \log(1/\delta')}\varepsilon + N_v \frac{\sqrt{2}\varepsilon}{n^{1/4}}\left(e^{\frac{\sqrt{2}\varepsilon}{n^{1/4}}} - 1\right) + 4N_v\gamma_n\varepsilon(e^\varepsilon - 1).$$

This leads to the conclusion of the theorem, noting that all pairs of users have the same behavior by the structure of the graph. □

Appendix D 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 $O(1/n^{1/4})$ compared to the same algorithm analyzed under LDP.

Theorem 6. *Let $\delta > 0$ and $\varepsilon \leq \min\left(1, \ln\left(\frac{\sqrt{n}}{32\ln(2/\delta)}\right)\right)$. Algorithm 5 achieves $(\varepsilon', \frac{T}{n}\delta + \delta' + \hat{\delta})$-network DP for all $\delta', \hat{\delta} > 0$ with*

$$\begin{aligned}\varepsilon' \quad = \quad & \sqrt{2N_v \ln(1/\delta')}\frac{28\sqrt{\ln(4/\delta)}\varepsilon}{n^{1/4}} + 2\sqrt{2N_v\gamma_n \ln(1/\delta')}\varepsilon \\ & + N_v \frac{28\sqrt{\ln(4/\delta)}\varepsilon}{n^{1/4}}\left(e^{\frac{28\sqrt{\ln(4/\delta)}\varepsilon}{n^{1/4}}} - 1\right) + 4N_v\gamma_n\varepsilon(e^\varepsilon - 1),\end{aligned}$$

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

Proof. The proof follows the same steps as in the case of real summation (Appendix C), as the properties of the walks stays identical. In the case of small cycles, the same bound of 2ε 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 bound of (Feldman et al., 2020, Theorem 3.1 therein), that we recall below.

Theorem 7 (Amplification by shuffling)**.** *For any data domain $\mathcal{X}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{X} \to \mathcal{S}^{(i)}$ 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 ε_0-DP local randomizer for all values of auxiliary inputs $(z_1, \dots, z_{i-1}) \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}} : \mathcal{X}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm which takes as input a dataset $(x_1, \dots, x_n) \in \mathcal{X}^n$, samples a uniform random permutation π over $[n]$, then*

sequentially computes $z_i = \mathcal{R}^{(i)}\left(z_1, \ldots, z_{i-1}, x_{\pi(i)}\right)$ *for* $i \in [n]$ *and outputs* $(z_1, \ldots, z_n)$*. Then for any* $\delta \in [0,1]$ *such that* $\varepsilon_0 \leq \log\left(\frac{n}{16\log(2/\delta)}\right)$, $\mathcal{A}_{\mathrm{s}}$ *satisfies* (ε, δ)*-DP where*

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

Note that for $\varepsilon \leq \min\left(1, \ln\left(\frac{\sqrt{n}}{32\ln(2/\delta)}\right)\right)$, we can simplify the bound. 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 $\sqrt{n}/2$ contributions, where the hypothesis of the theorem are fulfilled by our choice of ε. We conlude the proof by using this bound to control the privacy loss associated with large cycles and following the same steps as in Appendix C. □

Remark 3 (Choice of amplification by shuffling bound)**.** *We choose here to use the bound of (Feldman et al., 2020) instead of (Erlingsson et al., 2019) as it holds under less restrictive assumptions. Indeed, with (Erlingsson et al., 2019), we can only obtain some amplification when the number of users is very large (*$n > 10^6$*).*

Appendix E Proof of Theorem 4 (Stochastic Gradient Descent on a Complete Graph)

Proof. The proof tracks privacy loss using Rényi Differential Privacy (RDP) (Mironov, 2017) and leverages results on amplification by iteration (Feldman et al., 2018). 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 (ε, δ)-DP.

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

Definition 3 (Rényi divergence)**.** *Let* $1 < \alpha < \infty$ *and* μ, ν *be measures such that for all measurable set* A*,* $\mu(A) = 0$ *implies* $\nu(A) = 0$*. The Rényi divergence of order* α *between* μ *and* ν *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 anf V are sampled from μ and ν respectively, with a slight abuse of notation we will often write $D_\alpha(U||V)$ to mean $D_\alpha(\mu\|\nu)$.

Definition 4 (Rényi DP)**.** *For* $1 < \alpha \leq \infty$ *and* $\varepsilon \geq 0$*, a randomized algorithm* $\mathcal{A}$ *satisfies* (α, ε)*-Rényi differential privacy, or* (α, ε)*-RDP, if for all neighboring data sets* D *and* D' *we have*

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

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

Definition 5 (Network Rényi DP)**.** *For* $1 < \alpha \leq \infty$ *and* $\varepsilon \geq 0$*, a randomized algorithm* $\mathcal{A}$ *satisfies* (α, ε)*-network Rényi differential privacy, or* (α, ε)*-NRDP, if for all pairs of distinct users* $u, v \in V$ *and all pairs of neighboring datasets* $D \sim_u D'$*, we have*

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

As in classic DP, there exists composition theorems for RDP, see (Mironov, 2017). We will use the following.

Proposition 2 (Composition of RDP). *If $\mathcal{A}_1, \ldots, \mathcal{A}_k$ are randomized algorithms satisfying (α, ε_1)-RDP, $\ldots$, (α, ε_k)-RDP respectively, then their composition $(\mathcal{A}_1(S), \ldots, \mathcal{A}_k(S))$ 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 (Mironov, 2017).

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

Privacy amplification by iteration (Feldman et al., 2018) 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 theorem (Feldman et al., 2018, 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 β-smooth function over $\mathcal{K}$. Let PNSGD(D, w_0, η, σ) be the algorithm that returns $w_n \in \mathcal{W}$ computed recursively from $w_0 \in \mathcal{W}$ using dataset $D = \{x_1, \ldots, x_n\}$ 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 (Feldman et al., 2018).

Proposition 4 (Weak convexity of Rényi divergence). *Let $\mu_1, \ldots, \mu_m$ and $\nu_1, \ldots, \nu_m$ 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 ρ be a probability distribution over $[m]$ and denote by μ_ρ (resp. ν_ρ) the probability distribution over $\mathcal{Z}$ obtained by sampling i from ρ and then outputting a random sample from μ_i (resp. ν_i). Then we have:*

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

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. In contrast to the proofs of Theorem 3 (real summation) and Theorem 6 (discrete histogram computation), this time we reason on the privacy loss from the point of view of user u.

Let fix a contribution of user u at some time t_1. As in previous proofs, the view $\mathcal{O}_v$ of user v on the entire procedure is defined as in Eq. 6. 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:

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

To bound the privacy loss over all the T_u contributions of node 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 $\alpha \to \varepsilon_c(\alpha)$ for $\alpha \in (1, \infty)$. However, recall that from our use of the weak convexity property we have the additional constraint on α requiring that $\sigma \geq L\sqrt{2\alpha(\alpha-1)}$. This creates two regimes: for small ε_c (i.e, large σ and small T_u), the minimum is not reachable, so we take the best possible α within the interval, whereas we have an optimal regime for larger ε_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 (Feldman et al., 2018, Theorem 32 therein). In particular, for $q = \max\left(\frac{2T_u \ln n}{n}, 2\ln(1/\delta_c)\right)$, $\alpha = \frac{\sigma\sqrt{\ln(1/\delta_c)}}{L\sqrt{q}}$ 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 $N_u = \frac{T}{n} + \sqrt{\frac{3}{2}T\ln(1/\hat{\delta})}$ 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. □

Remark 4 (Tighter numerical bounds)**.** *As mentioned in the proof, we can compute a tighter bound for small σ when the optimal α violates the constraints on σ. In this case, we set α 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 node is smaller than the number of nodes.*

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 σ that ensures this privacy guarantee. We start with a small candidate for σ and compute the associated privacy loss as explained above. We then increase it iteratively until the resulting ε_c is small enough.

Appendix F 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 by leveraging results on privacy amplification by shuffling. Here, we have used the numerical approach from (Balle et al., 2019b) to tightly measure the effect of amplification by shuffling based on the code provided by the authors.[3] Figure 3 confirm that the empirical gains from privacy amplification by decentralization are also significant for this task.

[3] `https://github.com/BorjaBalle/amplification-by-shuffling`

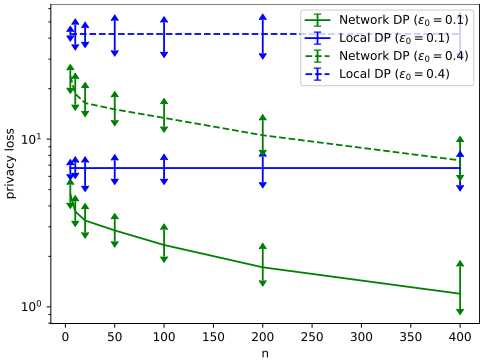


Figure 3: Comparing network and local DP on the task of computing discrete histograms. The results are obtained for $T = 100n$ (i.e., the expected number of contributions per user is 100). The value of ε_0 rules the amount of local noise added to each contribution (i.e., each single contribution taken in isolation satisfied ε_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.