

---

# 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. 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 gradient descent, we propose simple algorithms and prove privacy amplification results on ring and complete topologies. The resulting privacy-utility trade-off significantly improves upon LDP, and in some cases even matches what can be achieved with approaches based on secure aggregation and secure shuffling. Our experiments confirm the practical significance of the gains compared to LDP.

## 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 [23], 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 [23, 26, 6]. The classic motivation for fully decentralized approach is efficiency and scalability: indeed, the central coordinator can represent a bottleneck, especially when the number of participants is large [26].

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 [32]. In fact, federated learning provides an additional attack surface as the participants share intermediate updates [29, 19]. To control the privacy leakage, the prominent approach is based on the standard notion of Differential Privacy (DP) [15]. 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.

In this context, several trust models can be considered and lead to different trade-offs between privacy and utility. The strongest model is local differential privacy (LDP) [25, 13], where each participant (user) does not trust anyone and assumes that an adversary may observe everything that she/he shares with anyone. 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 [9] (see also [34, 33] for the limits of machine learning under LDP). This motivates 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 [14, 31, 7, 10, 21]. Recent work has also considered the so-called shuffle model of DP [11, 17, 4, 3, 20], 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.

In this work, we initiate the study of a *novel relaxation of LDP where users have only a local view of the decentralized system*. More specifically, we consider that each user only observes information received from her/his neighbors in the network graph, which is a natural assumption in fully decentralized settings. We introduce *network DP*, a suitable adaptation of DP which captures this setting and also accounts for potential collusions between users. Using our notion of network DP, we study 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 considered in previous work as a way to perform (non-private) decentralized estimation and optimization with less communication and computation than algorithms that require all users to communicate with their neighbors at each step, see [30, 22, 27, 1].

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 amplification 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 relying 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 in this setting and prove a privacy amplification result of  $O(1/n^{1/4})$ . Interestingly, we show that we can tolerate a constant number of collusions at the cost of some degradation in the privacy amplification effect. We also discuss natural extensions to discrete histograms and gradient descent. Our theoretical analysis leverages recent results on privacy amplification by subsampling [2], shuffling [17, 11, 4] and iteration [18], drawing interesting connections with these schemes. We illustrate the relevance of our approach through numerical experiments, showing that privacy gains provided by decentralization are significant in practice, and suggesting that our analysis can be tightened.

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 without relying on a central (untrusted) aggregator to handle all communications. Our results show that the true privacy guarantees of some fully decentralized protocols have been 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.

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 concluding remarks in Section 6.

## 2 Setting

In this section, we define our main notations and introduce network differential privacy (network DP), a novel relaxation of differential privacy which is natural in fully decentralized settings. Then, we describe the simple 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  $D = D_1 \cup \dots \cup D_n$  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*

between datasets, which is sometimes referred to as user-level DP [28]. 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 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  $\mathcal{A}$  is  $(\varepsilon, \delta)$ -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:*

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

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. Note that if  $\mathcal{O}_v$  is the identity map in Eq. 1 (i.e., if each user is able to observe all messages), then we recover 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  $f$  on the number of users that can possibly collude. The identity of colluders is however unknown to others. We would thus like to remain private with respect to the aggregated information  $\mathcal{O}_{V'} = \cup_{v \in V'} \mathcal{O}_v$  acquired by any possible subset  $V'$  of  $f$  users, as captured by the following generalization of Definition 1.

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

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

## 2.2 Decentralized Computation on a Walk

We study network DP for decentralized protocols in which computation is done 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

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

where  $x_u^k = g^k(D_u; \tau)$  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.

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 [30, 22, 27, 1] 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  $\tau$ .

It is easy to see that such decentralized protocols can also be used to compute summaries of the users' data, such as any commutative and associative operation like sums/averages, and discrete histograms. In these cases, the contributions of a given user made at each visit of the token 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.

```

 $\tau \leftarrow 0; a \leftarrow 0;$ 
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  $\tau$ 

```

**Algorithm 1:** Real summation.

```

Init.  $\tau \in \mathbb{N}^L$  with  $\gamma n$  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  $\tau$ 

```

**Algorithm 2:** Discrete histogram.

Figure 1: Private decentralized algorithms for computations over a directed ring.

### 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 differential privacy 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  $x$  (this includes in particular the Gaussian and Laplace mechanisms). Let  $\sigma_{loc}$  be the standard deviation of the noise required so that  $\text{Perturb}(\cdot; \sigma_{loc})$  satisfies  $(\epsilon, \delta)$ -LDP.

Consider now the simple decentralized protocol in Algorithm 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 [16], we can show the following result (see Appendix A for the proof).

**Theorem 1.** *Let  $\epsilon, \delta > 0$ . Algorithm 1 outputs an unbiased estimate of  $\bar{x}$  with standard deviation  $\sqrt{\lceil Kn/(n-1) \rceil} \sigma_{loc}$ . Furthermore, it satisfies  $(\sqrt{2K \ln(1/\delta')} \epsilon + K \epsilon (e^\epsilon - 1), K \delta + \delta')$ -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  $O(1/\sqrt{n})$  amplification in  $\epsilon$ . 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 typically done for instance to aggregate iterative user updates in federated learning algorithms[23].

**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 the noise addition is distributed across all users. For instance, assuming Gaussian noise for simplicity, each user can add noise with standard deviation  $\sigma'_{loc} = \sigma_{loc}/\sqrt{n}$ , except for the very first contribution which requires standard deviation  $\sigma_{loc}$  to properly hide the contributions of users in the first cycle. Taking into account this extra noise at the first step of the algorithm, the total added noise has standard deviation  $\sqrt{\lceil Kn/(n-1) \rceil + 1} \sigma_{loc}$ . This leads to same utility as Algorithm 1, up to a constant factor that is negligible when  $K$  is large.*

### 3.2 Discrete Histograms

We now turn to histogram computation over a discrete domain  $[L] = \{1, \dots, L\}$  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  $x_u^k \in [L]$ .

For this problem, a classic approach in LDP is  $L$ -ary randomized response [24], where a 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 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 to leverage results on *privacy amplification by shuffling* [17, 4]. 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  $\gamma = L/(\exp(12\epsilon\sqrt{\frac{\log(1/\delta)}{n}}) + L - 1)$ . 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  $(\sqrt{2K \ln(1/\delta')}\epsilon + K\epsilon(e^\epsilon - 1), K\delta + \delta')$ -network DP for any  $\delta' > 0$ .*

Theorem 2 shows that for the same amount of noise (fixed utility), Algorithm 2 again provides a privacy amplification of  $\frac{1}{n}\sqrt{n/\log(1/\delta)} = O(1/\sqrt{n})$  compared to LDP.

**Remark 2.** *For clarity of presentation, Theorem 2 relies on the amplification by shuffling result of [17] which has a simple closed-form. A tighter and more general result (without restriction on the values of  $n$ ,  $\epsilon$  and  $\delta$ ) can be readily obtained by using the results of [4].*

### 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 [7]. In contrast, our approach is much simpler 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. In particular, the non-collusion assumption is essential in Algorithm 1: if two users collude and share their view, the algorithm does not satisfy differential (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 (see Remark 1), a node placed right between two colluded nodes (or with very few honest users in-between) would suffer largely degraded differential 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 [18]. In this amplification scheme, the privacy guarantee for a given user (data point) grows with the number of data points that come after it. In a fixed ring, the level of privacy for a user  $u$  with respect to the view of another user  $v$  would thus depend on their relative positions in the ring. For instance, there would be no privacy amplification effect 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

We propose to study a very simple algorithm (Algorithm 3): a user  $u$  receiving the token  $\tau$  for the  $k$ -th time updates it with  $\tau \leftarrow \tau + \text{Perturb}(x_u^k; \sigma_{loc})$  such that  $\text{Perturb}(\cdot; \sigma_{loc})$  satisfies  $(\epsilon, \delta)$ -LDP. The following result shows an amplification of  $O(1/n^{1/4})$  for the privacy guarantees compared to LDP, which is obtained thanks to both *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*.

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

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

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

```

 $\tau \leftarrow 0, k_1 \leftarrow 0, \dots, k_n = n \leftarrow 0;$ 
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  $\tau$ 

```

**Algorithm 3:** Private decentralized algorithm for real aggregation on a complete graph.

*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 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  $2\varepsilon$  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 [2] to bound the privacy loss by the case where  $u$  contributes once. Finally, the total privacy loss across the  $N_v$  visits to  $v$  is obtained using advanced composition.  $\square$

While the privacy amplification effect 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 2a). We will also see, via numerical simulations, that the gains are significantly stronger in practice than what our bound guarantees.

**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  $f$  users collude, they can be seen as a unique node in the graph with a transition probability of  $\frac{f}{n}$  instead of  $\frac{1}{n}$ . We can then analyze this case as above except that the size of a cycle between two colluding users follows a geometric law of parameter  $1 - f/n$  and the total number of visits to colluding users follows  $\mathcal{B}(T, f/n)$ . Hence, we can obtain the same guarantees under Definition 2 as for the case with  $n/f$  non-colluding nodes under Definition 1.

## 4.2 Extensions via Amplification by Shuffling and Amplification by Iteration

The privacy amplification result of Theorem 3 can be extended to other problems than real summation by leveraging recent privacy amplification schemes. In particular, we can tackle histogram computation by bounding the privacy loss incurred by larger cycles in the proof of Theorem 3 using amplification by shuffling [17, 11, 4], similar to what we did for the ring topology in Section 3. Another extension very relevant to machine learning is to perform iterative (stochastic) gradient descent. Indeed, there exists a significant literature on decentralized algorithms for optimizing sums of local functions based on (deterministic or random) walks [30, 22, 27, 1]. Intuitively, the privacy of a step in a cycle increases with the subsequent steps as it becomes hard to trace back in which direction the gradient of the sample was pointing when one observes only the result of the last step. To quantify this effect, we can leverage recent work on privacy amplification by iteration [18].

The detailed formal analysis of these extensions is left for future work, but we provide numerical results based on privacy amplification by shuffling and by iteration in Section 5.

## 5 Numerical Results

In this section, we present some numerical results to illustrate the significance of privacy amplification by decentralization in the complete graph setting (Section 4), both theoretically and empirically.

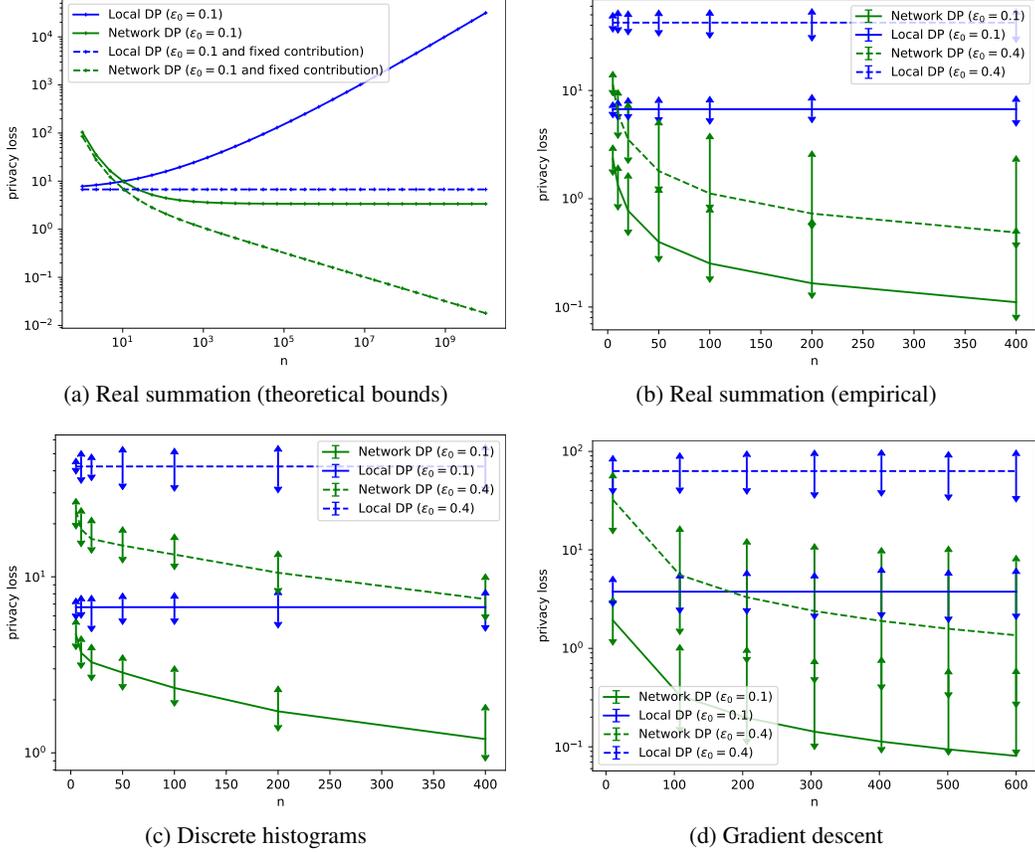


Figure 2: Numerical results for  $T = 100n$  (i.e., the expected number of contributions per user is 100) for different tasks. The value of  $\varepsilon_0$  rules the amount of local noise added to each contribution (i.e., each single contribution taken in isolation satisfied  $\varepsilon_0$ -LDP). For empirical results (Figures 2b to 2d), 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 Comparison of Theoretical Bounds for Network and Local DP

First, we numerically evaluate our theoretical bound of Theorem 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 2a plots the value of the bounds for varying  $n$ . We can see that our theoretical result provides gains in the total privacy loss compared to local DP as soon as  $n > 100$ , and these gains become more and more significant as  $n$  increases. The curves obtained under the fixed number of contributions per user also suggest that a better control of this quantity has the potential to make our amplification result significantly tighter.

## 5.2 Gap between Theoretical Bounds and Empirical Behavior

As discussed above, 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. We now 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, the underlying task (real aggregation, histogram computation or gradient descent) 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 2b 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). The results show that the gains achieved by network DP are significantly stronger in practice than what our theoretical bounds guarantee (see Figure 2a). In particular, the gains are significant even for small  $n$ . These results again suggest that there is some room for improvement in our analysis, for instance by resorting to better concentration bounds.

Finally, we run similar experiments to investigate the extensions of our approach to discrete histograms and gradient descent by leveraging results on privacy amplification by shuffling and by iteration. Figures 2c and 2d confirm that the empirical gains from privacy amplification by decentralization are also significant for these two tasks.

## 6 Perspectives

We believe that our work opens many interesting perspectives. Aside from making our analysis of the complete graph setting more tight, we plan to generalize Theorem 3 to arbitrary graphs using classic graph theoretic notions like the hitting time. Furthermore, we think that considering time-evolving topologies can help improve robustness to collusions, for instance in ring 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 we studied is to consider multiple tokens walking on the graph in parallel. We would also like to study randomized gossip algorithms [8]: these protocols are popular for decentralized optimization in machine learning (see [12] and references therein) and were recently shown to provide differential privacy guarantees in the specific context of rumor spreading [5]. 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 distant users.

## References

- [1] Ghadir Ayache and Salim El Rouayheb. Private Weighted Random Walk Stochastic Gradient Descent. Technical report, arXiv:2009.01790, 2020.
- [2] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *NeurIPS*, 2018.
- [3] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Differentially Private Summation with Multi-Message Shuffling. Technical report, arxiv:1906.09116, 2019.
- [4] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The Privacy Blanket of the Shuffle Model. In *CRYPTO*, 2019.
- [5] Aurélien Bellet, Rachid Guerraoui, and Hadrien Hendrikkx. Who started this rumor? Quantifying the natural differential privacy guarantees of gossip protocols. In *DISC*, 2020.
- [6] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. Personalized and Private Peer-to-Peer Machine Learning. In *AISTATS*, 2018.
- [7] 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.
- [8] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking*, 14(SI):2508–2530, 2006.
- [9] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal Lower Bound for Differentially Private Multi-party Aggregation. In *ESA*, 2012.
- [10] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography*, 2012.

- [11] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed Differential Privacy via Shuffling. In *EUROCRYPT*, 2019.
- [12] Igor Colin, Aurélien Bellet, Joseph Salmon, and Stéphan Cléménçon. Gossip Dual Averaging for Decentralized Optimization of Pairwise Functions. In *ICML*, 2016.
- [13] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *FOCS*, 2013.
- [14] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *EUROCRYPT*, 2006.
- [15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography (TCC)*, 2006.
- [16] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and Differential Privacy. In *FOCS*, 2010.
- [17] 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.
- [18] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy Amplification by Iteration. In *FOCS*, 2018.
- [19] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *NeurIPS*, 2020.
- [20] 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.
- [21] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *NeurIPS*, 2018.
- [22] 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.
- [23] 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.
- [24] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. In *NIPS*, 2014.
- [25] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What Can We Learn Privately? In *FOCS*, 2008.
- [26] 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.
- [27] 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.
- [28] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Recurrent Language Models. In *ICLR*, 2018.
- [29] 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.

- [30] 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.
- [31] Elaine Shi, T.-H. Hubert Chan, Eleanor G. Rieffel, Richard Chow, and Dawn Song. Privacy-Preserving Aggregation of Time-Series Data. In *NDSS*, 2011.
- [32] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.
- [33] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical Risk Minimization in Non-interactive Local Differential Privacy Revisited. In *NeurIPS*, 2018.
- [34] Kai Zheng, Wenlong Mou, and Liwei Wang. Collect at Once, Use Effectively: Making Non-interactive Locally Private Learning Possible. In *ICML*, 2017.

## 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 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  $\tau_1^v, \dots, \tau_K^v$ , each of them corresponding to user contributions aggregated along with random noise. We define the view of  $v$  accordingly as:

$$\mathcal{O}_v(\mathcal{A}(D)) = \{\tau_i^v\}_{i=1}^K. \quad (4)$$

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 node, so the same guarantee holds.

Finally, we apply the advanced composition theorem [16] 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 aggregation 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. 4, 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 [17] that we recall below.

**Theorem 4** (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 [16].  $\square$

### C Proof of Theorem 3 (Real Aggregation 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:

$$\mathcal{O}_v(\mathcal{A}(D)) = \bigcup_{t=1}^{T_v} (u_{k_{i-1}}, \tau_{k_i}, u_{k_{i+1}}), \quad (5)$$

where  $k_i$  is the  $i$ -th time that  $v$  receives the token,  $\tau_{k_i}$  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:

$$\mathbb{P}\left(\sum_{i=1}^{N_v} \mathbf{1}_v^i \geq 2N_v\gamma_n\right) \leq e^{-\frac{N_v\gamma_n}{2}}. \quad (6)$$

For each of these small cycles, we can bound the privacy loss suffered by  $u$  with respect to  $v$  by  $2\varepsilon$ : 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.5), the other ones can be seen as the result of subsampling with replacement. Therefore, the amplification by subsampling result given in Theorem 10 of [2] 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  $\varepsilon$  is the level of privacy guaranteed by the local mechanism  $\text{Perturb}(\cdot; \sigma_{loc})$ .

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. 6.

Regarding the parameter  $\delta'$  of DP, the privacy amplification by sampling result of [2] gives

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

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  $\delta$  is the level provided by the  $(\varepsilon, \delta)$ -LDP guarantee of  $\text{Perturb}(\cdot; \sigma_{loc})$ .

Finally, to bound the privacy loss over the full view (Eq. 5), we use the advanced composition theorem [15] on each of the two types of cycles, which gives for the final  $\varepsilon'$ :

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

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