Differentially Private Decentralized Learning with Random Walks

Edwige Cyffers [1] **Aurélien Bellet** [2] **Jalaj Upadhyay** [3]

Abstract

The popularity of federated learning comes from the possibility of better scalability and the ability for participants to keep control of their data, improving data security and sovereignty. Unfortunately, sharing model updates also creates a new privacy attack surface. In this work, we characterize the privacy guarantees of decentralized learning with random walk algorithms, where a model is updated by traveling from one node to another along the edges of a communication graph. Using a recent variant of differential privacy tailored to the study of decentralized algorithms, namely Pairwise Network Differential Privacy, we derive closed-form expressions for the privacy loss between each pair of nodes where the impact of the communication topology is captured by graph theoretic quantities. Our results further reveal that random walk algorithms yield better privacy guarantees than gossip algorithms for nodes close to each other. We supplement our theoretical results with empirical evaluation of synthetic and real-world graphs and datasets.

1. Introduction

Federated learning allows multiple data owners to collaboratively train a model without sharing their data (Kairouz et al., 2021). Some federated algorithms rely on a central server to orchestrate the process and aggregate model updates (McMahan et al., 2017), with the downsides of creating a single point of failure and limiting the scalability in the number of participants (Lian et al., 2017). In this work, we focus on fully decentralized algorithms that replace the central server by peer-to-peer communications between participants, viewed as nodes in a sparse network graph (Koloskova et al., 2020; 2021; Lian et al., 2017; Le Bars et al., 2023; Nedic et al., 2018; Tang et al., 2018). In addition to their scalability, these algorithms can exploit the natural graph structure in some applications, such as social networks where users are linked to their friends, or the geographical position of devices that induces faster communications with the closest users.

Keeping data decentralized can reduce communication costs and latency, and it is also welcomed from a privacy perspective when the data contains personal information or represents a crucial asset for businesses. However, sharing model parameters can indirectly leak sensitive information and allow data reconstruction attacks (see e.g. Geiping et al., 2020; Nasr et al., 2018; Shokri et al., 2017; Zhu et al., 2019). To mitigate this problem, *differential privacy* (DP) (Dwork et al., 2006) has become the *de facto* standard in machine learning to provide robust guarantees of privacy. In a nutshell, DP compares two learning scenarios that only differ from the data of a single user and ensures that the output distribution of the algorithm remains similar. This, in particular, means that no attacker can learn too much about the private data of a user by inspecting the outputs, even if they have access to arbitrary auxiliary information.

Decentralized learning algorithms can be made differentially private by having each node add noise to their model updates before sharing them with their neighbors in the graph (Bellet et al., 2018; Cheng et al., 2019; Huang et al., 2015; Xu et al., 2021; Zhang et al., 2018). An important challenge is then to bound as tightly as possible the privacy leakage based on the level of noise and the threat model considered to achieve the best possible privacy-utility trade-off. The baseline approach relies on *local DP*, which assumes that all information a node sends is observed by all other nodes. This leads to overly pessimistic privacy guarantees for decentralized algorithms because nodes only observe the messages sent by their direct neighbors.

Recent work has shown that it is possible to leverage the graph topology of decentralized algorithms to develop more tailored privacy guarantees specific to the relation between the different nodes based on the notion of *pairwise network differential privacy* (PNDP) (Cyffers & Bellet, 2022; Cyffers et al., 2022). PNDP takes into account the fact that each node only has a local view of the communications, and allows to reason on the privacy leakage between each pair of

[1]Université de Lille, Inria, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France [2]Inria, Univ Montpellier, Montpellier, France [3]Rutgers University. Correspondence to: Edwige Cyffers <edwige.cyffers@inria.fr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

nodes based on these local views. Intuitively, if two nodes are farther apart, the privacy leakage should depend on their relative position in the graph, which matches the natural setting where edges come with some trust level, such as in social networks where edges indicate relationships. Cyffers et al. (2022) showed that this intuition is correct for *gossip algorithms*, where all nodes update their current model and mix it with their neighbors at every step. However, a major drawback of gossip algorithms is that, even in their more asynchronous versions, they generate redundant communications and require all nodes to be largely available (since any node can be updated at any time). Redundant communication and availability has been touted as a major obstacle in distributed private learning (Smith et al., 2017).

Our Contributions. In this work, we study the privacy guarantees of random walk algorithms (Lopes & Sayed, 2007; Johansson et al., 2010; Mao et al., 2020; Cyffers & Bellet, 2022; Even, 2023), a popular alternative to gossip in the fully decentralized setting. In these algorithms, a *token* holding the model's current state is updated by a node at each time step and then forwarded to a random neighbor. Random walk algorithms do not require global synchronization as the token is sent as soon as the current node finishes its update. They can also easily cope with temporary unavailability, and are known to be fast in practice.

We first introduce a private version of decentralized *stochastic gradient descent* (SGD) based on random walks on arbitrary graphs: in a nutshell, the node holding the model at a given step updates it with a local SGD step, adds Gaussian noise and forwards it to one of its neighbor chosen with appropriate probability. Focusing on the strongly convex setting, we then establish a convergence rate for our algorithm by building upon recent results on SGD under Markovian sampling (Even, 2023), and show that the result compares favorably to its gossip SGD counterpart. Our main contribution lies in precisely characterizing the privacy loss between all pairs of nodes using a PNDP analysis. We obtain elegant closed-form expressions that hold for arbitrary graphs, capturing the effect of a particular choice of graph through graph-theoretic quantities. We also show how our general closed-form expression yields explicit and interpretable results for specific graphs. Finally, we use synthetic and real graphs and datasets to illustrate our theoretical results and show their practical relevance compared to the gossip algorithms analyzed in previous work.

In summary, our contributions are as follows:

1. We propose a private version of random walk stochastic gradient descent for arbitrary graphs (Algorithm 1);
2. We establish its convergence rate for strongly convex loss functions (Theorem 2);
3. We derive closed-form expressions for the privacy loss between each pair of nodes that capture the effect of the topology by graph-theoretic quantities (Theorem 3);
4. We theoretically and experimentally compare our guarantees to those of gossip algorithms, highlighting the superiority of our approach in several regimes.

2. Related Work

Random walks for decentralized optimization. Optimizing the sum of local objective functions by having a token walk on the graph has a long history in the optimization community (Johansson et al., 2010; Lopes & Sayed, 2007; Mao et al., 2020). The main difficulty is to handle the bias introduced by the sampling of the nodes, as a random walk locally forces a structure that differs from the stationary distribution (Sun et al., 2018). One way to avoid this bias is to restrict the underlying graph structure to be the complete graph (Cyffers & Bellet, 2022; Cyffers et al., 2023) or to perform an update only after several steps on the walk (Hendrikx, 2022), but this comes at a high communication cost. In this work, we rely on a recent proof that casts Markov chain updates as a specific case of stochastic gradient descent with delays in order to get rid of the dependency of the Markov sampling by waiting a sufficient number of steps for the analysis (Even, 2023).

Private decentralized optimization. A classic line of work to improve privacy in a decentralized setting aims to prove that nodes cannot access enough information to reconstruct exactly the contribution of a given node (Gupta et al., 2018; Mo & Murray, 2017). However, these approaches do not provide robust guarantees against approximate reconstruction attacks or adversaries with auxiliary knowledge. Another direction is to rely on local DP (Bellet et al., 2018; Cheng et al., 2019; Huang et al., 2015; Xu et al., 2021; Zhang et al., 2018), where each node assumes that everything they share is public, but this comes at a high cost for utility (Chan et al., 2012; Wang et al., 2018; Zheng et al., 2017). While it is possible to mitigate this drawback by using other schemes such as shuffling or secure aggregation (Bonawitz et al., 2017; Cheu et al., 2019; Feldman et al., 2020; Liew et al., 2022), it requires additional computation, communication as well as an overhaul of the system architecture, making it very difficult to deploy in practice. These limitations have motivated the development of intermediary trust models specific to fully decentralized settings.

Network differential privacy. Bellet et al. (2020) were the first to suggest that decentralization can amplify privacy. This was made precise by Cyffers & Bellet (2022), who introduced Network Differential Privacy (NDP) and proved better privacy guarantees for the complete graph and the ring graph. For the complete graph, Cyffers et al. (2023) further used NDP to analyze the privacy guarantees of decentralized

ADMM. The case of the ring graph was further studied by Yakimenka et al. (2022b), taking into account the additional complexity of dealing with straggler nodes that slow down the computation. In this work, we rely on the pairwise NDP variant introduced in Cyffers et al. (2022), where the authors study private gossip algorithms on arbitrary graphs. We provide both theoretical and empirical comparisons to these prior results, showing significant advantages in favor of our random walk algorithm.

3. Preliminaries

Definition 1 (Irreducibility) A Markov chain defined by transition matrix $W \in \mathbb{R}^{n\times n}$ is called *irreducible* if for any two states i, j, there exists an integer t such that

$$Pr[X_t = j|X_0 = i] > 0.$$

Definition 2 (Aperiodicity) The period of a state i is defined as the greatest common divisor of the set of natural numbers, $\{t : Pr[X_t = i|X_0 = i] > 0\}$. A state i is called *aperiodic* if its period equals 1. A Markov chain is *aperiodic* if all its states are aperiodic.

Irreducibility and aperiodicity ensure that, after enough steps, the probability of transit from any state to any other state is positive.

Definition 3 (Stationary distribution) For a Markov chain defined by the transition matrix $W \in \mathbb{R}^{n\times n}$, a probability distribution π is a stationary distribution if $P\pi = \pi$.

4. Problem Setting and Background

We consider a set $V = \{1, \dots, n\}$ of users (nodes), each user v holding a local dataset $\mathcal{D}_v$. The nodes aim to privately optimize a separable function over the joint data $\mathcal{D} = \cup_{v\in V}\mathcal{D}_v$:

$$f(x) = \sum_{v=1}^{n} \pi_v f_v(x), \tag{1}$$

where $x \in \mathbb{R}^d$ represents the parameters of the model and the local function f_v depends only on the local dataset of node v, and $\pi_v \geq 0$ is the weight given to f_v (in practice, the vector π will correspond to the stationary distribution of the random walk as defined below). Below, we introduce the notions and assumptions related to random walks and precisely define the privacy threat model we consider.

4.1. Random Walks

We consider that the underlying network is represented by a connected graph $G = (V, E)$. Two nodes u and v are neighbors when there is an edge $(u, v) \in E$, which indicates that u and v can communicate. Our random walk algorithm will involve a token following a Markov chain on this graph, where the probability of taking each edge is given by an $n \times n$ transition matrix W: if the token is in u at step t, v receives the token at time $t + 1$ with probability W_{uv}.

Definition 4 (Transition matrix) A *transition matrix* W on graph $G = (V, E)$ is a stochastic matrix, i.e., $\forall u \in V$, $\sum_{v\in V} W_{uv} = 1$, which satisfies $(u, v) \notin E \Rightarrow W_{uv} = 0$.

In particular, the probability for the token to go from node u to node v in k steps is given by the k-th power of the transition matrix W^k_{uv}. To derive convergence in optimization, we need standard assumptions on this transition matrix, which ensure that the Markov chain behaves similarly to a fixed distribution after enough iterations.

Assumption 1 The transition matrix is *aperiodic* and *irreducible*, that is, there exists a time t_0 such that for all $t \geq t_0$ and any pair of vertices u and v, $W^t_{uv} > 0$, i.e., the token can go from u to v in t steps.

Under this assumption, the Markov chain has a *stationary distribution* π (belonging to the n-dimensional simplex) such that $\pi = \pi W$, and the convergence speed is governed by the mixing time (Levin & Peres, 2017).

Definition 5 (Mixing time) The mixing time $\tau_{\text{mix}}(\iota)$ of a Markov chain of transition matrix W is the time needed for the walk to be close to a factor ι of its asymptotic behavior:

$$\tau_{\text{mix}}(\iota) = \min\left(t : \max_v \|(W^t)v - \pi\|_{TV} \leq \iota\right), \tag{2}$$

where $\|P - Q\|_{TV}$ is the *total variation distance* between two probability measures P and Q defined over the same measurable space $(\Omega, \mathcal{F})$: $\|P - Q\|_{TV} = \sup_{A\in\mathcal{F}} |P(A) - Q(A)|$.

We sometimes omit ι and write $\tau_{\text{mix}} := \tau_{\text{mix}}(1/4)$. From this quantity, it is easy to derive the mixing time for an arbitrary ι from the following equation: $\tau_{\text{mix}}(\iota) \leq \lceil \log_2(1/\iota)\rceil \tau_{\text{mix}}$ (Levin & Peres, 2017). The mixing time depends on the spectral gap of the graph, which is the difference between the largest eigenvalue of the transition matrix W (which is always equal to one and associated with π) and the absolute value of the second-largest eigenvalue. We denote this quantity by λ_W, from which we get the following bound on the mixing time: $\tau_{\text{mix}} \leq \lambda_W^{-1} \ln(1/\min_v \pi_v)$.

A natural choice of the transition matrix is to give the same importance to every node and to assume symmetry in the weight of the communications.

Assumption 2 The transition matrix W is bi-stochastic and symmetric.

Under this assumption, we have $\pi = \mathbb{1}/n$, where $\mathbb{1}$ is an all-one vector. As W is symmetric, it can be decomposed using the spectral theorem, and all the eigenvalues are real. For any connected graph, it is possible to construct a transition matrix that satisfies this assumption, for instance by using the Hamilton weighting on the graph where transitions are chosen uniformly among the neighbors. Let d_u denote the degree of node u. Then

$$W_{uv} = \mathbb{P}(X^{t+1} = i | X^t = j) = \frac{1}{\max\{d_u, d_v\}},$$
$$W_{uu} = 1 - \sum_{v \neq u} W_{uv}.$$

Remark 1 Our optimization results of Section 5 will only require Assumption 1 and thus cover (1) with non-uniform weights. Conversely, our privacy results of Section 6 will only require Assumption 2.

4.2. Privacy Threat Model

In this paper, we aim to quantify how much information each node $u \in V$ leaks about its local dataset $\mathcal{D}_u$ to any other node v during the execution of a decentralized learning algorithm. We assume nodes to be *honest-but-curious* (i.e., they faithfully follow the protocol) and non-colluding (see Appendix D for the possibility of collusion and how it can be seen as a modification of the graph).

We consider the graph G and the transition matrix W to be known by all nodes. This scenario is quite common; for instance, in the healthcare domain (where nodes represent hospitals and collaboration between hospitals is public knowledge) or in social networks. We will measure the privacy leakage using Differential Privacy (DP), and more precisely, a variant known as the Pairwise Network Differential Privacy (PNDP) (Cyffers & Bellet, 2022; Cyffers et al., 2022) tailored to the analysis of decentralized algorithms. In the rest of this section, we introduce the relevant definitions and tools and formally define what is observed by each node during the execution of a random walk algorithm.

Differential Privacy. DP (Dwork & Roth, 2014) quantifies the privacy loss incurred by a randomized algorithm $\mathcal{A}$ by comparing its output distribution on two *adjacent* datasets $\mathcal{D}$ and $\mathcal{D}'$. The guarantee depends thus on the granularity chosen for the adjacency relation. In this work, we adopt user-level DP, where $\mathcal{D} = \cup_{v \in V} \mathcal{D}_v$ and $\mathcal{D}' = \cup_{v \in V} \mathcal{D}'_v$ are adjacent datasets, denoted by $\mathcal{D} \sim \mathcal{D}'$, if there exists at most one user $v \in V$ such that $\mathcal{D}_v \neq \mathcal{D}'_v$. We further denote $\mathcal{D} \sim_v \mathcal{D}'$ if $\mathcal{D}$ and $\mathcal{D}'$ differ only in the local dataset of user v.

We use Rényi Differential Privacy (RDP) to measure the privacy loss, due to its theoretical convenience and better composition properties than the classical (ϵ, δ)-DP. We recall that any (α, ε)-RDP algorithm is also $(\varepsilon + \ln(1/\delta)/(\alpha - 1), \delta)$-DP for any $0 < \delta < 1$ (Mironov, 2017).

Definition 6 (Rényi Differential Privacy) An algorithm $\mathcal{A}$ satisfies (α, ε)-Rényi Differential Privacy (RDP) for $\alpha > 1$ and $\varepsilon > 0$ if for all pairs of neighboring datasets $\mathcal{D} \sim \mathcal{D}'$:

$$D_\alpha \left(A(\mathcal{D}) || A(\mathcal{D}') \right) \leq \varepsilon \,, \tag{3}$$

where $D_\alpha(X||Y)$ is the *Rényi divergence* between the random variables X and Y:

$$D_\alpha(X||Y) = \frac{1}{\alpha - 1} \ln \int \left(\frac{\mu_X(z)}{\mu_Y(z)} \right)^\alpha \mu_Y(z) dz \,.$$

with μ_X and μ_Y the respective densities of X and Y.

The Gaussian mechanism ensures RDP by adding Gaussian noise to the output of a non-private function g, i.e., $\mathcal{A}(\mathcal{D}) = g(\mathcal{D}) + \eta$ with $\eta \sim \mathcal{N}(0, \sigma^2)$ satisfies $(\alpha, \alpha \Delta_g^2 / 2\sigma^2)$-RDP for any $\alpha > 1$, where $\Delta_g = \sup_{\mathcal{D} \sim \mathcal{D}'} \|g(\mathcal{D}) - g(\mathcal{D}')\|_2$ is the sensitivity of g (Mironov, 2017).

This baseline privacy guarantee can be amplified when the result is not directly observed but instead used for subsequent computations. In particular, when considering the consecutive applications of *non-expansive* (i.e. 1-Lipschitz) operators, we can rely on the so-called *privacy amplification by iteration* effect that we will leverage in our analysis.

Theorem 1 (Privacy amplification by iteration, Feldman et al., 2018) Let $T^1, \ldots, T^K, T'^1, \ldots, T'^K$ be non-expansive operators, an initial random state $x^0 \in U$, and $(\zeta^k)_{k=1}^K$ a sequence of noise distributions. Consider the noisy iterations $x^{k+1} = T^{k+1}(x^k) + \eta^{k+1}$ and $\bar{x}^{k+1} = T^{k+1}(\bar{x}^k) + \bar{\eta}^{k+1}$ where η^k and $\bar{\eta}^k$ are drawn independently from distribution ζ^{k+1}. Let $s_k = \sup_{x \in U} \|T^k(x) - \bar{T}^k(x)\|$. Let $(a_k)_{k=1}^K$ be a sequence of real numbers such that

$$\forall k \leq K, \sum_{i \leq k} s_i \geq \sum_{i \leq k} a_i, \text{ and } \sum_{i \leq K} s_i = \sum_{i \leq K} a_i \,. \tag{4}$$

Then,

$$D_\alpha(x^K || \bar{x}^K) \leq \sum_{k=1}^K \sup_{x : \|x\| \leq a_k} D_\alpha(\zeta_k * \mathbf{x} \| \zeta_k) \,, \tag{5}$$

where $*$ is the convolution of probability distributions and $\mathbf{x}$ denotes the distribution of the random variable that is always equal to x.

Pairwise Network Differential Privacy. PNDP allows us to capture the limited view that nodes have in decentralized algorithms and to model privacy guarantees specific to each pair of nodes (Cyffers & Bellet, 2022; Cyffers et al., 2022). Below, the view of a user v is denoted by $\mathcal{O}_v\big(\mathcal{A}(\mathcal{D})\big)$.

Definition 7 (Pairwise Network DP) For $b : V \times V \to \mathbb{R}^+$, an algorithm $\mathcal{A}$ satisfies (α, b)-Pairwise Network DP

(PNDP) if for all pairs of distinct users $u, v \in V$ and neighboring datasets $\mathcal{D} \sim_u \mathcal{D}'$:

$$D_\alpha(\mathcal{O}_v(\mathcal{A}(\mathcal{D}))||\mathcal{O}_v(\mathcal{A}(\mathcal{D}'))) \leq b(u,v)\,. \tag{6}$$

We denote by $\varepsilon_{u\to v} = b(u,v)$ the privacy loss from u to v and say that u is $(\alpha, \varepsilon_{u\to v})$-PNDP with respect to v when inequality (6) holds for $b(u,v) = \varepsilon_{u\to v}$.

For the random walk algorithms we will consider, the complete output $\mathcal{A}(\mathcal{D})$ consists of the trajectory of the token and its successive values during training. At a given step, the token of the random walk shares its current value only with its current location, but the other nodes cannot see this state. Thus, we define the view of a node v as

$$\mathcal{O}_v\big(\mathcal{A}(\mathcal{D})\big) = \{(t, x_t, w) : \text{ the token } x_t \text{ was in } v \text{ at time } t \text{ and then sent to } w\}\,. \tag{7}$$

In this definition, nodes know to whom they send the token, but not from whom they receive it. Ensuring the anonymity of the sender can be achieved by using mix networks (Sampigethaya & Poovendran, 2006) or anonymous routing (Dingledine et al., 2004). However, our results directly extend to the case where the sender's anonymity cannot be ensured, see Remark 3 in Section 6.

5. Private SGD with Random Walks

In this section, we introduce a decentralized *stochastic gradient descent* (SGD) random walk algorithm to privately approximate the minimizer of (1), and analyze its convergence in the strongly convex case. This algorithm, presented in Algorithm 1, generalizes the private random walk algorithm on the complete graph, introduced and analyzed by Cyffers & Bellet (2022), to arbitrary graphs. Differential privacy is achieved by adding Gaussian noise to the local gradient at each step. The step size is constant over time as commonly done in (centralized) differentially private stochastic gradient descent (DP-SGD) (Bassily et al., 2014).

The non-private version of this algorithm converges in various settings under Assumption 1. In this work, we adapt a recent proof for the non-private version (Even, 2023). For simplicity, we focus on strongly convex and smooth objectives with bounded gradients at the global optimum.

Assumption 3 (Bounded gradient and strong convexity) We assume that f is μ-strongly convex and L-smooth. Let x^* be its minimizer. We assume that, for $\zeta_* \geq 0$, $\forall v \in V, \|\nabla f_v(x^*)\|^2 \leq \zeta_*^2$.

In the case of stochastic gradient descent, stochasticity also comes from the fact that we sample from the local dataset. To handle both cases, we define g_t as an unbiased estimator of $\nabla f_{v_t}(x_t)$. We thus require to bound the variance of this estimator.

Algorithm 1: PRIVATE RANDOM WALK GRADIENT DESCENT (RW DP-SGD)

Input: transition matrix W on a graph G, number of iterations T, noise variance σ^2, starting node v_0, initial token value x_0, step size γ, gradient sensitivity Δ, local loss function f_v

for $t = 0$ *to* $T-1$ **do**
- Draw $\eta \sim \mathcal{N}(0, \Delta^2\sigma^2)$
- Compute g_t s.t. $\mathbb{E}[g_t] = \nabla f_{v_t}(x_t)$
- $x_{t+1} \leftarrow x_t - \gamma(g_t + \eta)$
- Draw $u \sim W_{v_t}$ in the set of neighbors of v_t
- Send token to u
- $v_{t+1} \leftarrow u$

end

Assumption 4 (Bounded local noise) We assume that the stochastic gradients respect the following condition: $\mathbb{E}\left[\|g_t - \nabla f_{v_t}(x_t)\|^2 \mid x_t, v_t\right] \leqslant \sigma_{sgd}^2$.

Theorem 2 Under Assumptions 1, 3, and 4, for step size $\gamma = \min\left(\frac{1}{L}, \frac{1}{T\mu}\ln\left(T\frac{\|x_0 - x^\star\|^2}{\frac{39L}{\mu^2}\tau_{\text{mix}}\zeta_\star^2}\right)\right)$ the iterates verify:

$$\mathbb{E}(\|x_T - x^*\|^2) \leq 2e^{-\frac{T\mu}{L}}\|x_0 - x^*\|^2 + \left(\frac{39\tau_{\text{mix}}\zeta_*^2 L}{\mu^3 T} + \frac{(d\sigma^2\Delta^2 + \sigma_{sgd}^2)L}{\mu^2 T}\right)\ln\frac{T\mu^2\|x_0 - x^*\|^2}{39L\tau_{\text{mix}}\zeta_*^2}\,.$$

Proof. We adapt the proof from Even (2023) that keeps track of the shift due to the Markov sampling of the random walk thanks to a comparison with delayed gradient. Following the same bounding steps and taking expectation over the noise distribution we obtain a similar inequality on the iterates than the non-private version up to an additional term for the noise. Setting the step size to balance the terms leads to the final equality. See Appendix A for a full proof. □

The convergence rate in Theorem 2 has three terms: the exponential convergence towards the minimizer parameterized by the condition number L/μ of f, the impact of the stochasticity of the random walk in $\tilde{\mathcal{O}}(\tau_{\text{mix}}\zeta_*^2 L/\mu^3 T)$ and the additional term due to the noise injection in $\tilde{\mathcal{O}}(\sigma^2\Delta^2 L/\mu^2 T)$. A similar term would appear when adapting the non-private convergence proof for other settings such as under the Polyak-Lojasiewicz condition.

Comparison with private gossip (Cyffers et al., 2022). In our random walk algorithm, each step involves computing the gradient of a single node and a single message, whereas the private gossip SGD algorithm of Cyffers et al. (2022) alternates between the computation of local gradients at all nodes in parallel and a multi-step gossip communication phase until (approximate) consensus.

Rephrasing the result of Cyffers et al. (2022) in our notation, each communication phase in Cyffers et al. (2022) requires $\tilde{\mathcal{O}}(\tau_{\text{mix}} \cdot \ln(\zeta_*^2/\sigma^2))$ steps where all the nodes send updates synchronously. For the optimization part, the first term is the same in $\mathcal{O}(e^{-\frac{T\mu}{L}}\|x_0 - x^*\|^2)$, but there is only one other term in $\mathcal{O}(\sigma^2 L/n\mu T)$. Hence, we lose a factor of n, but the reduced communication compensates for this. Our analysis is tighter in the sense that we can separate the privacy noise that is independent of the Markov chain, thereby improving the rate of the spectral gap factor compared to a naive analysis. In contrast, the private gossip analysis casts this noise as gradient heterogeneity, because it re-uses the non-private convergence analysis of Koloskova et al. (2020).

Special case of averaging. One can use the above algorithm to privately compute the average of values at each node. In this case, we assume that each node has a private value y_v (a float or a vector) and define the local objective function as:

$$f_v(x) = \|x - y_v\|^2 \,. \tag{8}$$

Note that, in this case, we have $L = \mu = 2$. A natural approach is to compute the running average of the visited values with noise injection at each step, which corresponds to Algorithm 1 with a decreasing step-size $\gamma_t = 1/t$. The drawback of this approach is that damping the first terms of the sum (when the Markov chain is not yet well mixed) requires a lot of steps and results in slow convergence (at least τ_{mix} steps). Adopting a constant step-size instead, as in Algorithm 1, does not modify the convergence leading terms that stay in $\mathcal{O}(1/T)$ for an adequate step-size.

One way to completely remove the influence of the first terms is to have a *burn-in* phase, when the token walks without performing any update, to come closer to the stationary distribution. Then, after $\tau_{\text{mix}}(\iota/2)$ steps, a running average of $4\delta^2/(\iota^3\lambda_W)$ is obtained, as proven in Theorem 12.21 of Levin & Peres (2017).

6. Privacy Analysis

In this section, we derive the privacy guarantees of Algorithm 1 for arbitrary graphs and show how this leads to improved trade-offs for specific graphs widely used in decentralized learning. Our main result is a closed-form expression for the privacy loss between each pair of nodes, which holds for arbitrary graphs.

Theorem 3 Consider a graph G with transition matrix W satisfying Assumption 2. After T iterations, for a level of noise $\sigma^2 \geq 2\alpha(\alpha-1)$, the privacy loss of Algorithm 1 from node u to v is bounded by:

$$\varepsilon_{u\to v} \leq \mathcal{O}\left(\frac{\alpha T \ln(T)}{\sigma^2 n^2} - \frac{\alpha T}{\sigma^2 n} \ln\left(I - W + \frac{1}{n}\mathbb{1}\mathbb{1}^\top\right)_{uv}\right) .$$

We recall that the logarithm of a matrix corresponds to the matrix whose eigenvalues are the log of the original eigenvalues and the eigenvectors remain identical. In particular, if $\lambda_1, \cdots, \lambda_n$ are n-eigenvalues (counting multiplicity) of a bistochastic symmetric matrix M and $x_1, \cdots, x_n$ are the corresponding eigenvectors, then $\log(M) = \sum_{i=1}^n x_i x_i^\top \log(\lambda_i)$. Note that Assumption 2 ensures this matrix is well-defined.

Sketch of proof. We give a high-level overview of the proof here and refer to Appendix B for details. We fix the two vertices u and v and see how a token visit to u will leak information to v. By the post-processing property of RDP, it is sufficient to compute the privacy loss that occurs when the token reaches for the first time v after the visit in u. For computing this loss, we use the weak convexity of the Rényi divergence (Van Erven & Harremos, 2014) to condition over the number of steps before reaching v. The length of the walk is parameterized by the power of the transition matrix. For a given length, we bound the privacy loss by using privacy amplification by iteration (Theorem 1). Refactoring the sum leads to the logarithm of the matrix. We finish by using composition over the $\mathcal{O}(T/n)$ times the token visits u during the walk. □

Remark 2 For the complete graph, the second term is equal to zero as the transition matrix is exactly $W = \frac{1}{n}\mathbb{1}\mathbb{1}^\top$. Thus, we recover the bound of Cyffers & Bellet (2022). In other words, Theorem 3 is a generalization of this previous result to arbitrary graphs.

Remark 3 Theorem 3 holds for the definition of the view of the node given in (7), where the sender is kept anonymous. We provide in Appendix B.1 a similar theorem if the senders are known. In this case, although the formula is more complex, the asymptotic is the same as it roughly shifts the privacy guarantees from one hop.

6.1. Interpretation of the Formula via Communicability

The privacy loss of Theorem 3 has two terms that we can interpret as follows. The first term is the same as in Cyffers & Bellet (2022) for the complete graph: we have an $O(1/n^2)$ privacy amplification factor compared to local DP, matching what would be obtained in central DP with n users. Our analysis reveals that this term also appears for arbitrary graphs: we can interpret it as a baseline privacy loss that occurs from the collaboration of all agents.

However, in graphs differing from the complete graph, this baseline privacy loss is corrected by the second term which depends on the specific pair (u, v) of the nodes considered. Note that this second term can be negative for some pairs as eigenvector components can be of arbitrary signs. This quantity can be seen as a variant of known graph centrality metrics and, more precisely, communicability.

Definition 8 (Communicability, Estrada & Hatano, 2008; Estrada & Knight, 2015) For a transition matrix W and c_i a non-increasing positive series ensuring convergence, the communicability between two vertices u, v is defined by:

$$G_{uv} = \sum_{i=1}^{\infty} c_i W^i_{uv} \,.$$

For $c_i = 1/i!$, this corresponds to the original notion of communicability as presented in (Estrada & Hatano, 2008), while for $c_i = \alpha^i$ we recover the Katz centrality (Katz, 1953). Our formula corresponds to the case $c_i = \frac{\alpha}{\sigma^2 i}$ as proven in Appendix B, and the convergence of the infinite sum is ensured by the fact that we remove the graph component associated with the eigenvalue 1.

Communicability is used to detect local structures in complex networks, with applications to community detection and graph clustering, for instance, in physical applications (Estrada & Knight, 2015). Good communicability is supposed to capture how well connected are the two nodes in the networks, i.e. how close they are. Hence, having the second term of the privacy loss proportional to the communicability of the nodes shows that our formula matches the intuition that nodes leak more information to closer nodes than to more distant ones.

6.2. Application to Specific Graphs

We now give closed-form expressions of the privacy loss (3) for specific graphs. To show the power of our bound, we note that the results for two network graphs studied in Cyffers & Bellet (2022), namely the complete graph and the ring graph, follow as a corollary of our general result. We illustrate below the possibility to derive closed formulas for other classes of graphs. Proofs are given in Appendix E.

Star graph. We consider a star graph with a central node in the first position linked to the $n-1$ other nodes. We choose the transition matrix such that the probability of self-loop $\kappa > 0$ is an arbitrarily small constant, and the distribution over the non-central nodes is uniform. Then we have the following privacy guarantees.

Theorem 4 Let $u, v \in V$ be two distinct nodes of the star graph and $\kappa > 0$ be an arbitrarily small constant. For a single contribution of node u in Algorithm 1 on the star graph, the privacy loss to node v is bounded by:

$$\varepsilon_{u\to v} \leq \begin{cases} -\frac{\alpha(1-\kappa)}{\sigma^2(n-1)} \ln\left(1 - \frac{1}{n-1}\right) & u \neq 1 \text{ and } v \neq 1 \\ \frac{\alpha(1-\kappa)}{2\sigma^2\sqrt{n-1}} \ln\left(\frac{\sqrt{n-1}+1}{\sqrt{n-1}-1}\right) & u = 1 \text{ or } v = 1 \end{cases} .$$

Sketch of proof. At a high level, as κ is an arbitrarily small constant, we can have an upper bound on the entries of all the powers of the adjacency matrix of the star graph. □

In particular, composing over the $\mathcal{O}(T/n)$ contributions, we see that extremal nodes enjoy a privacy amplification factor of order $\mathcal{O}(n^2)$ and the central node of order $\mathcal{O}(n)$.

Ring graph. We consider a symmetric ring where nodes are enumerated from 1 to n, which thus slightly differs from the case studied in Cyffers & Bellet (2022); Yakimenka et al. (2022a), where the ring is directed and thus deterministic (up to the possibility of skipping in Yakimenka et al., 2022a). For this graph, our results shows that the amplification is parameterized by the distance between the nodes in the ring.

Theorem 5 Let $u, v \in V$ be two distinct nodes of the ring with $a = (u + v - 2) \mod n$ and $\alpha' = \alpha \mathbf{1}_{|u-v|=1}$. For a single contribution of node u in Algorithm 1 on the ring graph, the privacy loss to node v, $\varepsilon_{u\to v}$, is bounded by:

$$\frac{\alpha' \log(T) \cos\left(\frac{2\pi a}{n}\right)}{n\sigma^2} + \frac{2\alpha}{n\sigma^2} \sum_{k=1}^{n-1} \cos\frac{\pi a k}{n} \ln \frac{3\csc^2(\pi k/n)}{4},$$

in the case of equal probability between going left, right, or self-looping. Furthermore, if the probability of self-looping is set to $\kappa > 0$, then $\varepsilon_{u\to v}$ is bounded by

$$\frac{\alpha'}{n\sigma^2} + \frac{\alpha(1-\kappa)}{n\sigma^2} \sum_{t=2}^{T} \sum_{k=1}^{n} \cos^{t-1}\frac{2\pi k}{n} \cos\frac{2\pi(a+1)k}{n}.$$

Sketch of proof. At a high level, we use the fact that the adjacency matrix for a ring graph is a circulant matrix, so its eigenvectors are the Fourier modes. Therefore, its full spectral decomposition can be easily computed. □

To get an intuition regarding Theorem 5, consider two nodes that are close to each other. For simplicity of calculation, consider nodes 1 and 2 and the statement of the theorem. Then $a = 0$ and $\cos^{t-1}(2\pi k/n)\cos(2\pi(a+1)k/n) = \cos^t(2\pi k/n)$. Therefore, $\varepsilon_{u\to v} \leq \frac{\alpha}{\sigma^2 n} + \frac{\alpha(1-\kappa)}{\sigma^2 n} \sum_{k=1}^{n} \frac{\cos^2(2\pi k/n)}{1-\cos(2\pi k/n)}$.

7. Experiments

In this section, we illustrate our results numerically on synthetic and real graphs and datasets and show that our random walk approach achieves superior privacy-utility trade-offs compared to gossip as long as the mixing time of the graph is good enough. The code is available at `https://github.com/totilas/DPrandomwalk`

Privacy losses and comparison with the gossip counterpart. We generate synthetic graphs with $n = 2048$ nodes and report the privacy loss averaged over 5 runs for every pair of nodes of the graphs as a function of the length of their shortest path in Figure 1. The transition matrix is computed using the *Hamilton weighting*. To compare with the private gossip algorithm (Cyffers et al., 2022), we consider

Figure 1. Comparison of privacy loss for random walks in bold lines and gossip in dashed lines for the same synthetic graphs with $n = 2048$. Random walks allow privacy amplification even for very close nodes, but the decay is slower than for gossip.

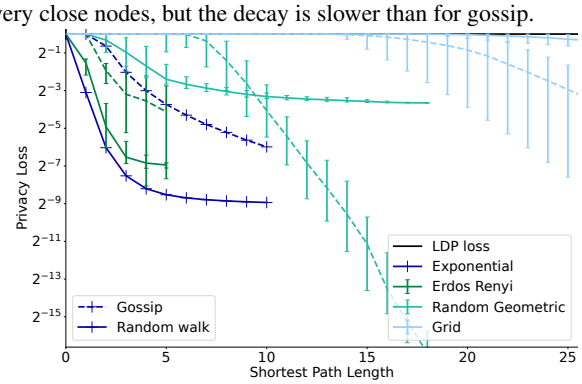
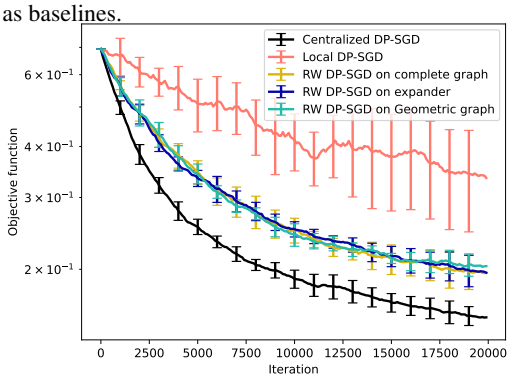


Figure 2. Private logistic regression on the Houses dataset where we compare our RW DP-SGD to with Local and Centralized DP-SGD as baselines.



the task of averaging (thus with $L = \mu = 2$) and consider the same precision level and the same graphs: an exponential graph, an Erdös-Rényi graph with $q = c\log(n)/n$ for $c > 1$, a grid and a geometric random graph. Our private random walk approach incurs a smaller privacy loss for close enough nodes than the private gossip algorithm. Remarkably, our approach improves upon the baseline local DP loss even for very close nodes. Conversely, the privacy loss of our approach is generally higher for more distant nodes. Nevertheless, random walks offer uniformly better privacy guarantees than gossip algorithms for graphs with good connectivity such as Erdös-Rényi graphs or expanders.

Logistic regression on synthetic graphs. We train a logistic regression model on a binarized version of the UCI Housing dataset.[1] The objective function corresponds to $f_v(x) = \frac{1}{|\mathcal{D}_v|}\sum_{(d,y)\in\mathcal{D}_v} \ln(1 + \exp(-yx^\top d))$ where $d \in \mathbb{R}^d$ and $y \in \{-1, 1\}$. As in Cyffers & Bellet (2022); Cyffers et al. (2022) and Yakimenka et al. (2022b), we standardize the features, normalize each data point, and split the dataset uniformly at random into a training set (80%) and a test set (20%). We further split the training set across 2048 users, resulting in local datasets of 8 samples each.

In a first experiment, we compare centralized DP-SGD, local DP-SGD, and our random walk-based DP-SGD. For all algorithms, we follow common practice and clip the updates to control the sensitivity tightly. We set $\varepsilon = 1$ and $\delta = 10^{-6}$. Following Cyffers et al. (2022), we use the mean privacy loss over all pairs of nodes (computed by applying Theorem 3) to set the noise level needed for our random walk-based DP-SGD. Figure 2 reports the objective function through iterations for complete, hypercube, and random geometric graphs. Experimentally, the behavior is the same for all the graphs, meaning that the token walk is diverse enough in every case to have a behavior similar to a uniformly random choice of nodes. The improvement in the privacy-utility trade-off compared to the local DP is significant.

We then compare our random walk algorithm to its gossip counterpart (Cyffers et al., 2022) on the same logistic regression task. For both algorithms, we fix the mean privacy loss $\bar{\varepsilon}$ across all pairs of nodes to three different levels ($\bar{\varepsilon} \in \{0.5, 1, 2\}$) and we report the accuracy reached by each algorithm on 4 graphs: complete, exponential, geometric and grid. As shown in Table 1, our random walk algorithm outperforms gossip in all cases, which can be explained by a combination of two factors. First, as seen previously in Figure 1, our algorithm yields a lower mean privacy loss for graphs with good expansion property, especially when the degree of the nodes is high (because the privacy guarantees of gossip degrade linearly with the degree, while random walk is insensitive to it). This gain directly leads to less noise injection when fixing the mean privacy loss, and thus better utility. The second factor that explains why the gain in utility is so pronounced (even for the grid) comes from differences in the SGD version of gossip and random walk. In both methods, the privacy guarantee degrades as a function of the number of participations of nodes (linearly in Rényi DP). In gossip, allowing each node to participate 10 times means that the model will essentially be learned by applying 10 "global" gradient updates (i.e., aggregated over the n nodes), because all local gradients are gossiped until convergence between each gradient computation (see Algorithm 3 in Cyffers et al., 2022). In our random walk algorithm, the model is learned by applying $10 \times n$ "local" gradient updates. Even if this represents the same amout of information about the data, better progress is made with many noisy steps than with a small number of less noisy steps, in the same way as mini-batch SGD tends to progress faster than GD in practice.

Privacy loss on real graphs and communicability. We consider two real-world graph datasets well-suited for com-

[1] https://www.openml.org/d/823/

Figure 3. Link between graph structure and privacy loss. Left: (a) example of Facebook Ego graph communicability and privacy loss, logarithmic scale. Middle: (b) same on the Southern women graph. Right: (c) the corresponding mean privacy loss and Katz centrality.

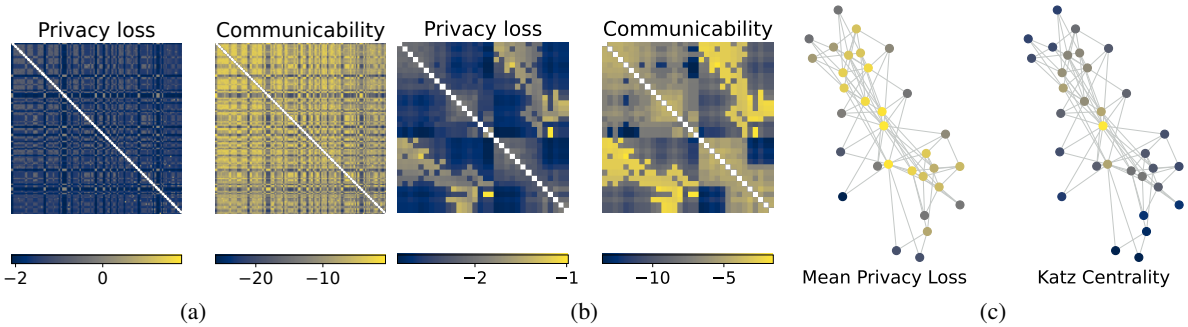


Table 1. Model accuracy at various mean privacy loss levels averaged over 8 runs.

Graph	Gossip	Random walk
Mean Privacy Loss of 0.5		
Complete	0.65 ± 0.10	0.841 ± 0.07
Exponential	0.70 ± 0.10	0.818 ± 0.09
Geometric	0.60 ± 0.07	0.795 ± 0.06
Grid	0.60 ± 0.07	0.803 ± 0.10
Mean Privacy Loss of 1		
Complete	0.70 ± 0.10	0.900 ± 0.04
Exponential	0.77 ± 0.05	0.883 ± 0.05
Geometric	0.66 ± 0.10	0.873 ± 0.05
Grid	0.73 ± 0.10	0.848 ± 0.07
Mean Privacy Loss of 2		
Complete	0.83 ± 0.06	0.940 ± 0.02
Exponential	0.89 ± 0.04	0.937 ± 0.01
Geometric	0.67 ± 0.04	0.933 ± 0.02
Grid	0.72 ± 0.10	0.919 ± 0.02

munity detection: (i) The Facebook Ego dataset (Leskovec & Mcauley, 2012) represents subgraphs of the Facebook network, where users are nodes and edges correspond to the friendship relation, and each subgraph corresponds to the set of friends of a unique user that is removed from the subgraph; (ii) The Davis Southern women social network (Roberts, 2000) is a graph with 32 nodes which corresponds to a bipartite graph of social event attendance by women and has been used in Koloskova et al. (2019) and Pasquini et al. (2023). We report side-by-side the matrix of pairwise privacy losses and of the communicability in Figure 4(a) and Figure 4(b). This confirms the similarity between the two quantities as discussed in Section 6.1. We also report the Katz centrality compared to the mean privacy loss for each node in Figure 4(c), showing that the two quantities also have similar behavior.

We provide other numerical experiments in Appendix C: we report the privacy loss on the other Facebook Ego graphs and study the impact of data heterogeneity.

8. Conclusion

In this work, we analyzed the convergence and privacy guarantees of private random walks on arbitrary graphs, extending the known results for the complete graph and the ring. Our results show that random walk-based decentralized algorithms provide favorable privacy guarantees compared to gossip algorithms, and establish a link between the privacy loss between two nodes and the notion of communicability in graph analysis. Remarkably, as long as the spectral gap of the communication graph is large enough, the random walk approach nearly bridges the gap between the local and the central models of differential privacy. Our results could be broadened by showing convergence under more general hypotheses. Other extensions could include skipping some nodes in the walk, considering latency, or having several tokens running in parallel.

Impact Statement

This work contributes to proving formal privacy guarantees in machine learning and thus paves the way for the larger adoption of privacy-preserving methods, with a better theoretical understanding. In particular, our work shows that decentralized algorithms may be useful in designing efficient privacy-preserving systems with good utility. In particular, the impact of the choice of a decentralized algorithm on privacy guarantees has not been studied so far. We believe our work opens interesting directions to see how one could design decentralized protocols that are private by design.

A potential risk of misuse of our work is that we rely on different privacy budgets across different pairs of users, as done previously in Cyffers et al. (2022). This could sometimes lead to a false sense of security or weaker privacy guarantees than those provided in other threat models. In

particular, the question of comparing the privacy guarantees we provide to those of the central model could be discussed. Our experiments use the mean privacy loss of nodes to set the comparison, but it is interesting to note that for the same mean privacy loss, the distribution of the privacy loss for gossip and random walk algorithms is not the same. Indeed, as shown in Figure 1, random walks have a smoother decay but can enjoy privacy amplification even for closely related nodes, whereas gossip tends to saturate the closest nodes then, followed by exponential decay. It is an interesting question to know how to interpret these privacy loss functions to choose the algorithm whose privacy guarantees best meet the privacy expectations in a particular use case.

Acknowledgments

This work was supported by grant ANR-20-CE23-0015 (Project PRIDE), the ANR-20-THIA-0014 program "AI_PhD@Lille" and the ANR 22-PECY-0002 IPOP (Interdisciplinary Project on Privacy) project of the Cybersecurity PEPR. JU research was supported by a Decanal Research Grant from Rutgers University.

References

Bassily, R., Smith, A., and Thakurta, A. Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473, Philadelphia, PA, USA, October 2014. IEEE.

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

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

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

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

Cheng, H.-P., Yu, P., Hu, H., Zawad, S., Yan, F., Li, S., Li, H. H., and Chen, Y. Towards Decentralized Deep Learning with Differential Privacy. In *CLOUD*, 2019.

Cheu, A., Smith, A., Ullman, J., Zeber, D., and Zhilyaev, M. Distributed differential privacy via shuffling. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, pp. 375–403. Springer, 2019.

Cyffers, E. and Bellet, A. Privacy Amplification by Decentralization. In *AISTATS*, 2022.

Cyffers, E., Even, M., Bellet, A., and Massoulié, L. Muffliato: Peer-to-Peer Privacy Amplification for Decentralized Optimization and Averaging. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, New Orleans, United States, 2022. URL https://cnrs.hal.science/hal-03906768.

Cyffers, E., Bellet, A., and Basu, D. From noisy fixed-point iterations to private ADMM for centralized and federated learning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Dingledine, R., Mathewson, N., and Syverson, P. Tor: the second-generation onion router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, 2004.

Dwork, C. and Roth, A. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Estrada, E. and Hatano, N. Communicability in complex networks. *Physical Review E*, 77(3), mar 2008. doi: 10.1103/physreve.77.036111. URL https://doi.org/10.1103%2Fphysreve.77.036111.

Estrada, E. and Knight, P. *A First Course in Network Theory*. Oxford University Press, United Kingdom, March 2015. ISBN 978-0-19-872645-6.

Even, M. Stochastic gradient descent under markovian sampling schemes. In *ICML*, 2023.

Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. Privacy amplification by iteration. *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, Oct 2018. doi: 10.1109/focs.2018.00056. URL http://dx.doi.org/10.1109/FOCS.2018.00056.

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

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

Godsil, C. and Royle, G. F. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2001.

Gupta, N., Katz, J., and Chopra, N. Information-theoretic privacy in distributed average consensus. *ArXiv*, abs/1809.01794, 2018.

Hendrikx, H. A principled framework for the design and analysis of token algorithms, 2022. URL https://arxiv.org/abs/2205.15015.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 2012.

Huang, Z., Mitra, S., and Vaidya, N. Differentially Private Distributed Optimization. In *ICDCN*, 2015.

Johansson, B., Rabi, M., and Johansson, M. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2010. doi: 10.1137/08073038X. URL https://doi.org/10.1137/08073038X.

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., Eichner, H., 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., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., 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. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Katz, L. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

Koloskova, A., Stich, S. U., and Jaggi, M. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *ICML*, 2019.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. A unified theory of decentralized SGD with changing topology and local updates. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Koloskova, A., Lin, T., and Stich, S. U. An improved analysis of gradient tracking for decentralized machine learning. In *Advances in Neural Information Processing Systems*, 2021.

Le Bars, B., Bellet, A., Tommasi, M., Lavoie, E., and Kermarrec, A.-M. Refined convergence and topology learning for decentralized sgd with heterogeneous data. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 1672–1702. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/le-bars23a.html.

Leskovec, J. and Mcauley, J. Learning to discover social circles in ego networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf.

Levin, D. and Peres, Y. *Markov Chains and Mixing Times*. MBK. American Mathematical Society, 2017. ISBN 9781470429621. URL https://books.google.fr/books?id=f208DwAAQBAJ.

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

Liew, S. P., Takahashi, T., Takagi, S., Kato, F., Cao, Y., and Yoshikawa, M. Network shuffling: Privacy amplification via random walks. *ArXiv*, abs/2204.03919, 2022.

Lopes, C. G. and Sayed, A. H. Incremental adaptive strategies over distributed networks. *IEEE Transactions on Signal Processing*, 55(8):4064–4077, 2007. doi: 10.1109/TSP.2007.896034.

Mao, X., Yuan, K., Hu, Y., Gu, Y., Sayed, A. H., and Yin, W. Walkman: A communication-efficient random-walk algorithm for decentralized optimization. *IEEE Transactions on Signal Processing*, 68:2513–2528, 2020. ISSN 1941-0476. doi: 10.1109/tsp.2020.2983167. URL http://dx.doi.org/10.1109/TSP.2020.2983167.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017.

Mironov, I. Renyi differential privacy. *CoRR*, abs/1702.07476, 2017. URL http://arxiv.org/abs/1702.07476.

Mo, Y. and Murray, R. M. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62 (2):753–765, 2017. doi: 10.1109/TAC.2016.2564339.

Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, 2018.

Nedic, A., Olshevsky, A., and Rabbat, M. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5): 953–976, May 2018.

Pasquini, D., Raynal, M., and Troncoso, C. On the (in)security of peer-to-peer decentralized machine learning. In *IEEE Symposium on Security and Privacy (S&P)*, 2023.

Roberts, J. M. Correspondence analysis of two-mode network data. *Social Networks*, 22 (1):65–72, 2000. ISSN 0378-8733. doi: https://doi.org/10.1016/S0378-8733(00)00017-4. URL https://www.sciencedirect.com/science/article/pii/S0378873300000174.

Sampigethaya, K. and Poovendran, R. A survey on mix networks and their secure applications. *Proceedings of the IEEE*, 94(12):2142–2181, 2006.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *CCS*, 2017.

Smith, A., Thakurta, A., and Upadhyay, J. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 58–77. IEEE, 2017.

Sun, T., Sun, Y., and Yin, W. On markov chain gradient descent. In *Neural Information Processing Systems*, 2018.

Tang, H., Lian, X., Yan, M., Zhang, C., and Liu, J. d^2: Decentralized training over decentralized data. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Van Erven, T. and Harremos, P. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

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

Xu, J., Zhang, W., and Wang, F. A(dp)^{2}sgd: Asynchronous decentralized parallel stochastic gradient descent with differential privacy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Yakimenka, Y., Weng, C.-W., Lin, H.-Y., Rosnes, E., and Kliewer, J. Straggler-resilient differentially-private decentralized learning. *2022 IEEE Information Theory Workshop (ITW)*, pp. 708–713, 2022a.

Yakimenka, Y., Weng, C.-W., Lin, H.-Y., Rosnes, E., and Kliewer, J. Straggler-resilient differentially-private decentralized learning. *2022 IEEE Information Theory Workshop (ITW)*, Nov 2022b. doi: 10.1109/itw54588.2022.9965898. URL http://dx.doi.org/10.1109/ITW54588.2022.9965898.

Zhang, X., Khalili, M. M., and Liu, M. Improving the Privacy and Accuracy of ADMM-Based Distributed Algorithms. In *ICML*, 2018.

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

Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 2019.

A. Proof of Theorem 2

We adapt the proof of Theorem 5 of (Even, 2023). Because constants matters in differential privacy, we detail the steps needed to explicit these constants. We recall here the keys steps and definition and how we adapt the proofs to allow the addition of Gaussian noise. We keep the same definitions, except that the sequence of token iterates is now defined by

$$x_{t+1} = x_t - \gamma(g_t + \eta_t)\,.$$

This does not impact Lemma 8, which only relies on the graph properties and the function at the optimum. Up to the transposition of notation, we have for any $T \geqslant 1$:

$$\mathbb{E}\left[\left\|\sum_{t<T} \nabla f_{v_t}\left(x^\star\right)\right\|^2\right] \leqslant T\zeta_\star^2 + \zeta_\star^2 \sum_{t<T} \mathrm{d}_{\mathrm{TV}}\left(P_{v_0,}^t, \pi^\star\right) + 2\zeta_\star^2 \sum_{s<t<T} \mathrm{d}_{\mathrm{TV}}(t-s), \tag{9}$$

where $\mathrm{d}_{\mathrm{TV}}(r) = \sup\left\{\mathrm{d}_{\mathrm{TV}}\left(\left(P^r\right)_{v,,}, \pi^\star\right), v \in \mathcal{V}\right\}$ for $r \in \mathbb{N}$, so that:

$$\mathbb{E}\left[\left\|\sum_{t<T} \nabla f_{v_t}\left(x^\star\right)\right\|^2\right] \leqslant \zeta_\star^2\left(4\tau_{\text{mix}}\left(1/4\right) + T\left(1 + 8\tau_{\text{mix}}\left(1/4\right)\right)\right). \tag{10}$$

Next, we can transform Lemma 9 into

$$\mathbb{E}_\eta(\|x_{t+1} - y_{t+1}\|^2) \leq (1-\gamma\mu)\mathbb{E}_\eta(\|x_t - y_t\|^2) + \gamma L\|y_t - x^*\|^2 + \gamma^2(d\sigma^2 + \sigma_{sgd}^2)\,.$$

where the sequence of y_t and y_{t+1} satisfies the relation.

$$y_{t+1} = y_t - \gamma\nabla f_{v_t}(x^*)$$

By applying the formula recursively, we obtain:

$$\mathbb{E}_\eta\left\|x_T - y_T\right\|^2 \leqslant (1-\gamma\mu)^T\left\|x_0 - y_0\right\|^2 + \sum_{t<T}(1-\gamma\mu)^{T-t}\left(\gamma L\left\|y_t - x^\star\right\|^2 + \gamma^2(d\sigma^2 + \sigma_{sgd}^2)\right)\,,$$

where we use $\mathbb{E}_\eta$ to denote the expected value with respect to the privacy noise.

By instantiating the y sequence as done in the non-private version with the x, we recover nearly the same formula, with $y_t = x^*$. The first term can be handled as in the non-private case, while the second term has an additional sum:

$$\begin{aligned}\mathbb{E}_\eta\left\|x_T - x^\star\right\|^2 \leqslant & 2(1-\gamma\mu)^T\left(\mathbb{E}\left\|x_0 - x^\star\right\|^2 + \gamma^2\mathbb{E}\left[\left\|\sum_{s<T}\nabla f_{v_s}\left(x^\star\right)\right\|^2\right]\right) \\ & + \sum_{t<T}(1-\gamma\mu)^{T-t}\left(\gamma^3 L\mathbb{E}\left[\left\|\sum_{t\leqslant s<T}\nabla f_{v_s}\left(x^\star\right)\right\|^2\right] + \gamma^2(d\sigma^2 + \sigma_{sgd}^2)\right).\end{aligned}$$

As $\sum_{t<T}(1-\gamma\mu)^{T-t} < \frac{1}{\gamma\mu}$, and the other terms remain identical, we have:

$$\mathbb{E}_\eta\left\|x_T - y_T\right\|^2 \leqslant 2(1-\gamma\mu)^T\left\|x_0 - x^\star\right\|^2 + \frac{3\gamma L}{\mu^2}C\tau_{\text{mix}}\zeta_\star^2 + \frac{\gamma(d\sigma^2 + \sigma_{sgd}^2)}{\mu}\,.$$

with $C = 13$.

We conclude by plugging back the following γ (as in (Even, 2023)) in the previous formula:

$$\gamma = \min\left(\frac{1}{L}, \frac{1}{T\mu}\ln\left(T\frac{\left\|x_0 - x^\star\right\|^2}{\frac{39}{\mu^2}C\tau_{\text{mix}}\zeta_\star^2}\right)\right)\,.$$

Remark 4 As long as $\sigma^2 \leq \frac{39L\tau_{\text{mix}}\zeta_*^2}{d\mu}$, the noise due to privacy is smaller than the one due to the randomness of the walk.

B. Privacy Proofs

Let u, v be two distinct nodes. To prove Theorem 3, we see the privacy loss $\varepsilon_{u\to v}$ as the composition of the privacy loss induced by each of the contributions of node u. Thus, we first bound the privacy loss for one contribution $\varepsilon^{\mathsf{single}}_{u\to v}$. Let us denote by t_c the time step where this contribution is made (i.e., the token is at node u at time t_c). For $t \leq t_c$, there is no privacy leakage. Let us denote by t_l the first $t \geq t_c$ where the token is held by v. By the post-processing property of differential privacy, the steps after t_l will not yield additional leakage for the contribution of time t_c. Hence, we only need to bound the privacy leakage at time t_l. This leakage depends on the number of steps between t_c and t_l. We use the weak convexity property of the Rényi divergence to decompose our privacy loss.

Lemma 1 (Weak convexity of Rényi divergence) Let $\mu_1, \dots, \mu_n$ and $\nu_1, \dots, \nu_n$ be probability distributions over some domain $\mathcal{Z}$ such that for all $i \in [n], \mathrm{D}_\alpha(\mu_i \| \nu_i) \leq c/(\alpha - 1)$ for some $c \in (0, 1]$. Let ρ be a probability distribution over $[n]$ and denote by μ_ρ (respectively, ν_ρ) the probability distribution over $\mathcal{Z}$ obtained by sampling i from ρ and then outputting a random sample from μ_i (respectively, ν_i). Then

$$\mathrm{D}_\alpha(\mu_\rho \| \nu_\rho) \leq (1 + c) \cdot \mathop{\mathbb{E}}_{i\sim\rho}[\mathrm{D}_\alpha(\mu_i \| \nu_i)] \,. \tag{11}$$

We can thus partition according to the length of the walk and write the privacy guarantee depending on T the total number of steps and $\beta(i)$ a function bounding the privacy loss occurring for seeing the token i steps after the node contribution:

$$\varepsilon^{\mathsf{single}}_{u\to v} \leq (1 + c) \sum_{i=1}^{T} \mathbb{P}(u \to v \text{ after } i \text{ steps})\beta(i) \,. \tag{12}$$

The probability of the path of t steps between u and v can be easily extracted from the power of the transition matrix W. We thus obtain the generic formula.

Lemma 2 Let $\beta(i)$ be a bound on the privacy loss occurring for seeing the token i steps, and T the total number of steps. Then, the following holds:

$$\varepsilon^{\mathsf{single}}_{u\to v} \leq \sum_{i=1}^{T} W^i_{uv} 2\beta(i) \,. \tag{13}$$

We recognize in this formula the communicability (Definition 8) with $c_i = 2\beta(i)$.

We can now compute the function β by resorting to privacy amplification by iteration (Theorem 1). As we apply the Gaussian mechanism at each step, we have $s_1 = \frac{\alpha}{2\sigma^2}$ and $s_j = 0$ for $0 < j \leq i$. We thus take all $a_j = \frac{\alpha}{2\sigma^2 i}$ in Theorem 1. This gives the bound $\beta(i) = \frac{\alpha}{2\sigma^2 i}$, which corresponds to setting $c_i = \alpha\sigma^2 i$.

We now focus on how to simplify and compute this formula under Assumption 2, i.e., the transition matrix is bistochastic and symmetric. Since the matrix is symmetric by assumption, we can apply the spectral theorem to write:

$$W = \sum_{i=1}^{n} \lambda_i \phi_i \phi_i^\top \,, \tag{14}$$

Furthermore, since the matrix is bistochastic, $\lambda_1 = 1 > \lambda_2 \geq \dots \leq \lambda_n > -1$ and the eigenvector associated to the first eigenvalue is $\frac{1}{\sqrt{n}}\mathbb{1}$. Hence, we can isolate the first term and plug into (13) to get:

$$\varepsilon^{\mathsf{single}}_{u\to v} \leq \sum_{t=1}^{T} \frac{\alpha}{\sigma^2 t}\frac{1}{n} + \sum_{i=2}^{n}\sum_{t=1}^{T} \frac{\alpha}{\sigma^2 t}\lambda_i^t \phi_i(u)\phi_i(v) \,. \tag{15}$$

In these sums, we isolate $\sum_{t=1}^{T} \lambda_i^t/t$. Noticing that the sum converges for all these eigenvalues, we can rewrite it as $\sum_{t=1}^{T} \lambda_i^t/t = -\ln(1 - \lambda_i) + \mathcal{O}(\lambda_i^T)$. We use the integral test for convergence to replace the sum by the logarithm.

This gives us the privacy loss for a single contribution. In order to conclude, we compose the privacy loss of each of the N_u contributions of node u, leading to, for any $\sigma^2 \geq 2\alpha(\alpha - 1)$:

$$\varepsilon_{u\to v} \leq \frac{\alpha N_u \ln(T)}{\sigma^2 n} - \frac{\alpha N_u}{\sigma^2}\ln(I - W + \frac{1}{n}\mathbb{1}\mathbb{1}^\top)_{uv} + \mathcal{O}(\lambda_2^T) \,. \tag{16}$$

As $|\lambda_2| < 1$, its power decreases exponentially fast with the number of steps and thus is negligible with respect to the other terms.

The average number of contributions is T/n as the transition matrix is bi-stochastic. We can then upper bound the real number of contributions with high probability, for example by using Theorem 12.21 of (Levin & Peres, 2017). The small probability of exceeding the upper bound can be added to δ when converting from RDP to (ε, δ)-differential privacy.

Remark 5 The upper bound on N_u tends to be large for "cryptographically" small δ. An efficient way to avoid this issue in practical implementations is to force a tighter bound on the maximum number of contributions by each node. After a node reaches its maximum number of contributions, if the token passes by the node again, the node only adds noise. We use this trick in numerical experiments.

B.1. Adaptation to the case without sender anonymity

For bounding the privacy loss of a single contribution, we consider above the value of the privacy loss occurring at time t_l. However, this is computed without taking into account the knowledge of where the token comes from, for example, if v can know which of its neighbors sent him the token. This computation is thus justified only in the specific case where the anononymity of the sender is ensured, e.g., by resorting to mix networks (Sampigethaya & Poovendran, 2006) or anonymous routing (Dingledine et al., 2004).

If this is not the case, then we should *a priori* use the conditional probability towards the position of the token at time $t_l - 1$. However, there is no close form for this in general for a graph. To fix this issue, we consider that the last step for reaching v is only a post-processing of the walk reaching one of its neighbors.

For each of the neighbors, the previous formula applies. We obtain different values for the various neighbors, so a worst-case analysis consists in taking the max over this set. Denoting by $\widetilde{\varepsilon_{uv}^{\mathsf{single}}}$ the privacy loss when a node v knows the neighbors from which it received the token, we have

$$\widetilde{\varepsilon_{uv}^{\mathsf{single}}} \leqslant \max_{w \in \mathcal{N}_v} \varepsilon_{uw}^{\mathsf{single}}$$

This approach allows to keep a closed form for the matrix and just add a max step. However, this analysis is not tight and may lead to a significant cost in some scenarios. A simple example is the special case where the nodes u and v are neighbors. It effectively assumes that the transition between u and v is always direct, which may not always be the case.

We provide in Figure 4 the equivalent of Figure 1 by taking the max below. As expected, amplification is smaller for close nodes, but the curves have the same asymptotic.

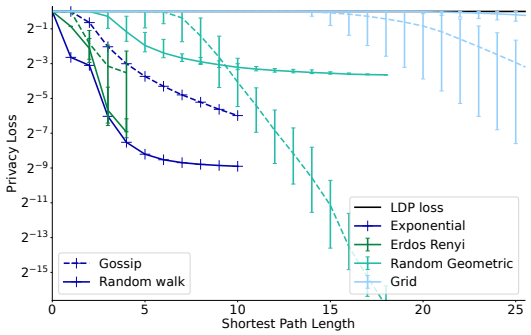


Figure 4. Comparison of privacy loss for random walks when nodes know who send them the token in bold lines and gossip in dashed lines for the same synthetic graphs with $n = 2048$. Privacy amplification is visible even for close neighbors but the decay is slower than for gossip.

C. Additional Numerical Experiments

In this section, we include other examples of graphs that illustrate how the privacy loss matches the graph structure.

A classic synthetic graph to exhibit subgroup is the stochastic block model, where each edge follows an independent Bernoulli random variable. The parameter of the law depends from a matrix encoding the relation between the clusters. As

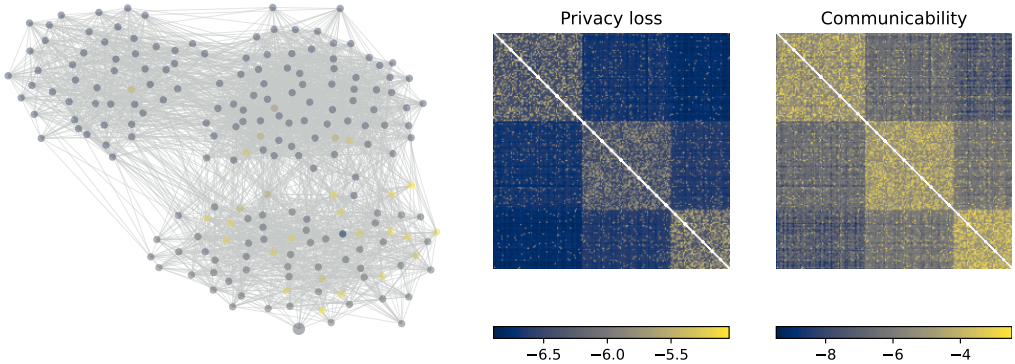


Figure 5. Stochastic Block Model with 200 nodes in three clusters $(75, 75, 50)$ and probability matrix $[[0.25, 0.05, 0.02], [0.05, 0.35, 0.07], [0.02, 0.07, 0.40]]$. The privacy loss matrix recovers the different blocks.

for other graphs, the privacy loss is similar to communicability and shows different level of privacy within and outside a group Figure 5(b), so that a node sees the major part of its privacy loss occurring within its group (Figure 5(a).

We report other examples of privacy loss for a node chosen at random in Facebook Ego graphs. Once again, these results illustrate that our privacy loss guarantees match the existing clusters of the graph (Figure 6).

Intuitively, compare to gossip algorithms where the updates only slowly flows in the graph, the random walk should mix quite easily heterogeneous data. while we do not derive mathematical guarantees on heterogeneity, we illustrate this idea with the following numerical experiment. We generate a synthetic geometric random graph and compare two scenarii (Figure 7). On the first trial, the data is position-dependent, which generate heterogeneity as close nodes also have close datapoint. We then shuffle randomly the data to destroy the heterogeneity and compare the convergence of the two in Figure 8.

D. Collusion

The results of our work assume that nodes are separated entities that do not share information outside the protocols. One could however claim that a fraction of nodes can collude and share information between them. In this case, if we denote $F \subset V$ the fraction of the colluded nodes, for a given contribution done by u what matters is the first time that one of the node of F is reached afterwards. More precisely, we can derive the privacy loss as

$$\varepsilon_{u\to F}^{single} \leq \sum_{i=1}^{T} \left(\sum_{v\in F} W_{uv}^{i} \right) \alpha\sigma^2 i \,. \tag{17}$$

where the term between parenthesis corresponds to the probability to reach F in exactly i steps from u. By reorganizing these terms, we obtain the upper bound:

$$\varepsilon_{u\to F} \leq \sum_{v\in F} \varepsilon_{u\to v} \tag{18}$$

This term corresponds also to the formula one would obtain from basic composition. Hence, our analysis does not allow to avoid the degradation of the privacy guarantees to collusion. Note that if all the colluded nodes are far away from u, it is still possible to derive non trivial guarantees compare to what would give the bound of local differential privacy. In comparison to gossip where the privacy loss decrease is sharper with distance, the cases where the amplification remains are scarcer. This is a fundamental limitation of amplification by decentralization, that was already pointed out in (Cyffers & Bellet, 2022; Cyffers et al., 2022).

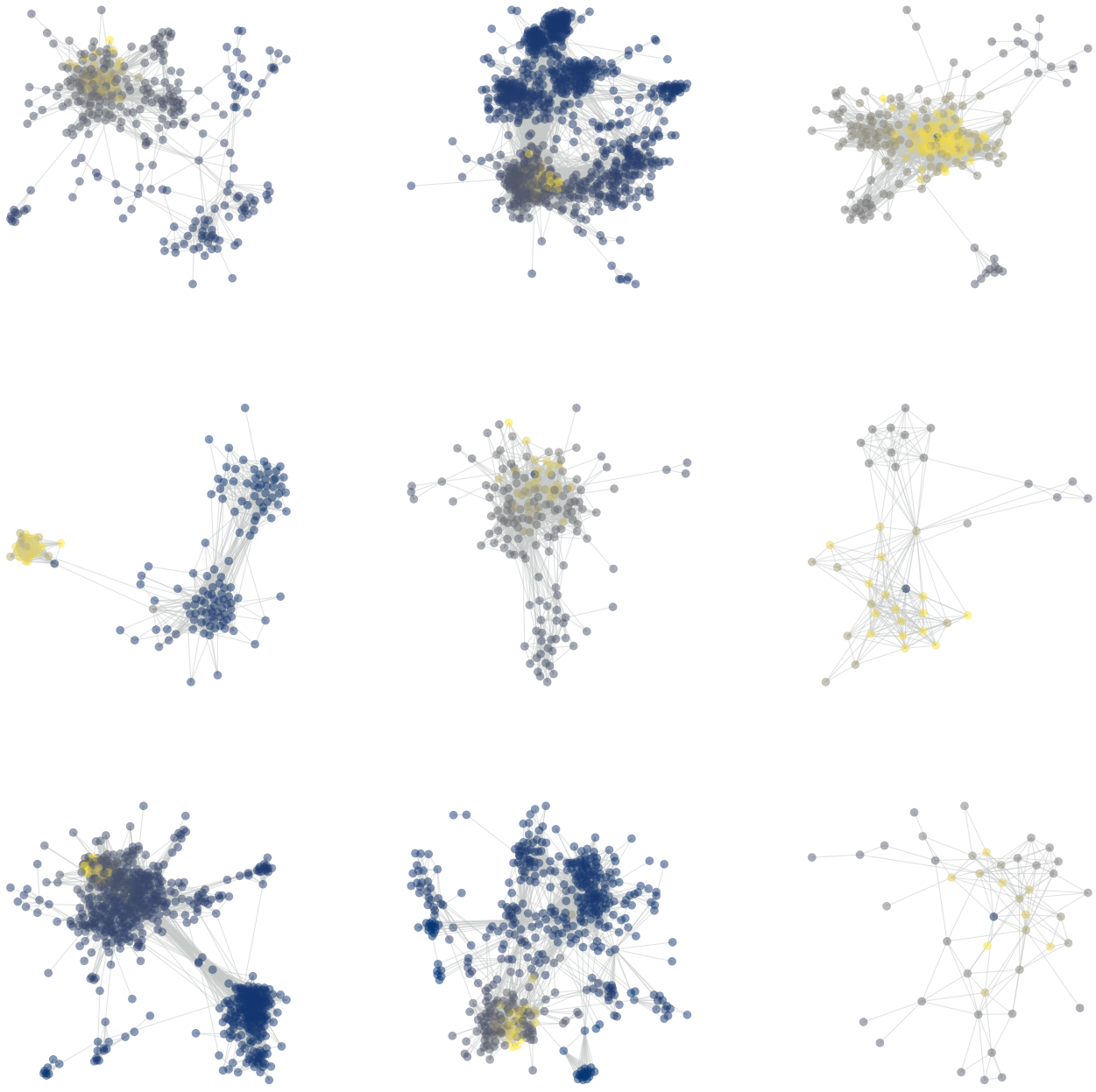


Figure 6. Privacy loss on the 9 other Facebook Ego graphs, following the same methodology as in Figure 4(a).

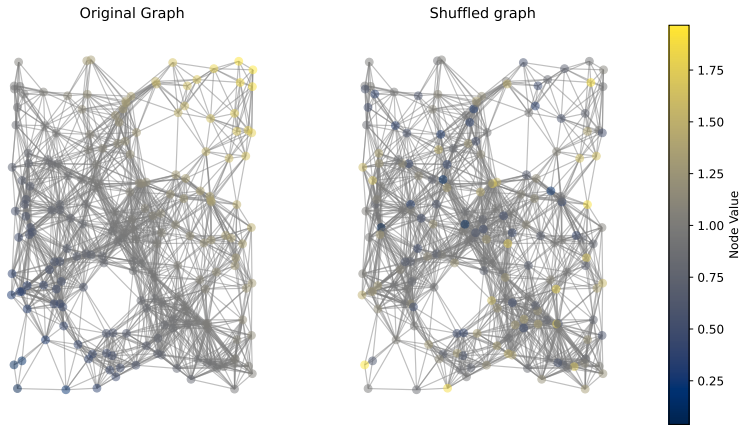
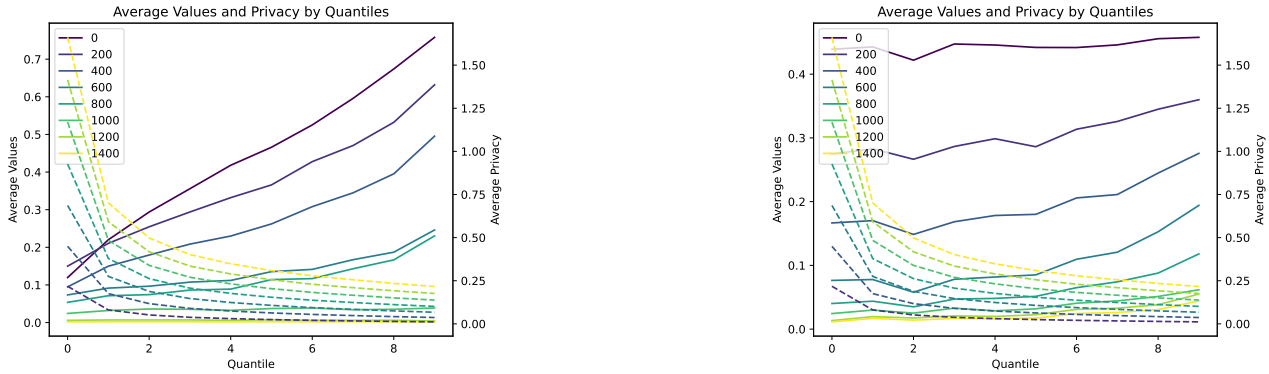


Figure 7. Geometric random graph with 200 nodes. On the left, the label is given by the sum of the coordinates, providing heterogeneity in the graph. On the right the same graph has its label shuffled



(a) With original labels

(b) With shuffled labels

Figure 8. We compute the quantiles of the euclidean distance between node. For each quantile, we report the mean on all the pair of nodes of the quantiles for the privacy loss and for the distance between the current estimates. We report different time step across the learning. The privacy loss is identical in both cases. At the beginning of the learning, the homogeneous case has smaller average values heterogeneity, but the difference reduces with the learning

E. Refined Privacy Bounds for Specific Graphs

E.1. Useful Auxiliary Results

We first collect some auxiliary results that we use in our bounds.

Proposition 1 For any $x \in (0,1)$, we have

$$\sum_{p \text{ is odd}} \frac{x^p}{p} = \frac{1}{2} \ln \frac{1+x}{1-x} \quad \text{and} \quad \sum_{p \text{ is even}} \frac{x^p}{p} = -\ln(1-x^2)\,.$$

Proof. Recall that

$$\ln(1+x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \cdots\,, \quad \text{and} \quad \ln(1-x) = -x + \frac{x^2}{2} - \frac{x^3}{3} + \cdots$$

Now $\ln(1+x) - \ln(1-x)$ gives the first bound and $\ln(1-x) + \ln(1+x)$ gives the second bound. This completes the proof. □

Proposition 2 (Godsil and Royle (Godsil & Royle, 2001)) Let $1 \le d \le n-1$ be an integer. Then for any d-regular graph G, the eigenvectors of the Laplacian and those of the adjacency matrix of G coincide.

E.2. Privacy Loss for Specific Graphs

Complete graph. The transition matrix is exactly $\frac{1}{n}\mathbb{1}\mathbb{1}^\top$ and thus we recover exactly the same formula as in (Cyffers & Bellet, 2022). In particular, there is only one non-zero eigenvalue with magnitude 1 with an all-one vector as the corresponding eigenvector. In particular,

$$\varepsilon_{u\to v} \le \frac{\alpha}{\sigma^2} \sum_{i=1}^{T} W^i_{uv} \frac{1}{i} = \frac{\alpha}{\sigma^2} \sum_{i=1}^{T} \frac{1}{i} \left(\sum_{j=1}^{n} \lambda_j^i v_j v_j^\top \right)_{uv} = \frac{\alpha}{n\sigma^2} \sum_{i=1}^{T} \frac{1}{i} \le \frac{\alpha \log(T)}{n\sigma^2}$$

Ring graph. To ensure aperiodic Markov chain as required by Assumption 1, the transition matrix should be in the form $aI + b(J + J^\top), a + 2b = 1$.

The adjacency matrix A_R of the ring graph R is a circulant matrix. Therefore, all its eigenvectors are just the Fourier modes (Horn & Johnson, 2012):

$$\phi(\omega_k) = \begin{pmatrix} 1 \\ \omega_k \\ \omega_k^2 \\ \vdots \\ \omega_k^{n-1} \end{pmatrix},$$

where $\omega_k^n = 1$ is the n-th root of unity, i.e, $e^{2\pi\imath k/n}$ for $1 \le k \le n$. This can be seen by noting that multiplication with a circulant matrix gives a convolution. In the Fourier space, convolutions become multiplication. Hence the product of a circulant matrix with a Fourier mode yields a multiple of that Fourier mode, which by definition is an eigenvector.

The eigenvalues can be then computed as

$$\omega_k + \omega_k^{-1} = 2\cos(2\pi k/n) \quad \text{for} \quad 0 \le k \le n-1\,.$$

Computing $\phi(\omega)\phi(\omega)^\top$ is straightforward. The (u,v)-th entry would be just $\omega^{(u+v-2) \mod n}$. Recall that

$$\varepsilon_{u\to v} \le \frac{\alpha}{\sigma^2} \sum_{i=1}^{T} \frac{W^i_{uv}}{i}\,.$$

For ease of calculation, let us assume that $a = b = \frac{1}{3}$. Then $W = \frac{A_R}{3} + \frac{\mathbb{I}}{3}$, where A_R is a binary matrix with (i,j)-th entry 1 only when $|i-j| = 1$. Then the eigenvalues of W are given by $\frac{2\cos(2\pi k/n)+1}{3}$ for $0 \le k \le n-1$. Let $a = (u+v-2) \mod n$. Hence,

$$
\begin{aligned}
\epsilon_{u\to v} &\le \frac{\alpha}{n\sigma^2}\sum_{t=1}^{T}\left(\frac{e^{2\pi\imath a/n}}{t}+\sum_{k=1}^{n-1}\frac{1}{t}\left(\frac{2\cos(2\pi k/n)+1}{3}\right)^t e^{2\pi\imath ak/n}\right)\\
&\le \frac{\alpha}{n\sigma^2}\sum_{t=1}^{T}\left(\frac{e^{2\pi\imath a/n}}{t}+\sum_{k=1}^{n-1}\frac{1}{t}\left(\frac{4\cos^2(\pi k/n)-1}{3}\right)^t e^{2\pi\imath ak/n}\right)\\
&= \frac{\alpha}{n\sigma^2}\sum_{t=1}^{T}\frac{1}{t}\cos\left(2\pi a/n\right)+\frac{\alpha}{n\sigma^2}\sum_{t=1}^{T}\sum_{k=1}^{n-1}\frac{1}{t}\left(\frac{4\cos^2(\pi k/n)-1}{3}\right)^t\cos\left(2\pi ak/n\right)\\
&\le \frac{\alpha\log(T)}{n\alpha^2}+\frac{\alpha}{n\sigma^2}\sum_{k=1}^{n-1}\sum_{t=1}^{\infty}\frac{1}{t}\left(\frac{4\cos^2(\pi k/n)-1}{3}\right)^t\cos\left(\frac{2\pi(u+v-2)k}{n}\right)\\
&= \frac{\alpha\log(T)}{n\sigma^2}-\frac{\alpha}{n\sigma^2}\sum_{k=1}^{n-1}\ln\left(1-\frac{4\cos^2(\pi k/n)-1}{3}\right)\cos\left(\frac{2\pi ak}{n}\right)\\
&\le \frac{\alpha\log(T)}{n\sigma^2}+\frac{2\alpha}{n\sigma^2}\sum_{k=1}^{n-1}\cos\left(\frac{2\pi ak}{n}\right)\ln\left(\frac{3\csc(\pi k/n)}{2}\right),
\end{aligned}
$$

where csc is the cosecant function. The second equality follows from the fact that $W^i_{u,v}$ is a real number, so the imaginary part is identically zero.

In the previous bound, we give self-loops the same probability as other edges. The main reason to give self-loop a non-zero weight is to ensure irreducibility and aperiodicity of the Markov chain. The same effect can be achieved by giving any non-negligible weight to the self-loops. In particular, we can consider the following adjacency matrix:

$$\widehat{A}_R = (1-\kappa)A_R + \kappa\mathbb{I}$$

for some $\kappa > 0$. Then, the eigenvalues would be $(1-\kappa)(\omega_k+\omega^{-1})+\kappa$. The adjacency matrix is still a circulant matrix. As a result, the eigenvectors still remain the same. Furthermore,

$$(\omega_k+\omega_k^{-1})^t\cos\left(\frac{2\pi ak}{n}\right) = 2\cos^t\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi ak}{n}\right) = \cos^{t-1}\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi(a+1)k}{n}\right).$$

Let $\kappa = \frac{1}{T^2}$. Then

$$\widehat{A}^t_{uv} \le (1-\kappa)A^t_{uv}$$

for $t \ge 2$. Therefore, for $a = (u+v-2) \mod n$:

$$
\begin{aligned}
\epsilon_{u\to v} &\le \frac{\alpha(1-\kappa)}{n\sigma^2}A_{uv}+\frac{\alpha(1-\kappa)}{n\sigma^2}\sum_{t=2}^{T}A^t_{uv}\\
&\le \frac{\alpha}{n\sigma^2}A_{uv}+\frac{\alpha(1-\kappa)}{n\sigma^2}\sum_{t=2}^{T}\sum_{k=1}^{n}(\omega_k+\omega_k^{-1})^t\cos\left(\frac{2\pi ak}{n}\right)\\
&= \frac{\alpha}{n\sigma^2}A_R[u,v]+\frac{\alpha(1-\kappa)}{n\sigma^2}\sum_{t=2}^{T}\sum_{k=1}^{n}\cos^{t-1}\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi(a+1)k}{n}\right).
\end{aligned}
$$

If $|u-v| = 1$, then we have

$$\epsilon_{u\to v} \le \frac{\alpha}{n\sigma^2}+\frac{\alpha(1-\kappa)}{n\sigma^2}\sum_{t=2}^{T}\sum_{k=1}^{n}\cos^{t-1}\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi(a+1)k}{n}\right).$$

otherwise, we have

$$\epsilon_{u\to v} \le \frac{\alpha(1-\kappa)}{n\sigma^2}\sum_{t=2}^{T}\sum_{k=1}^{n}\cos^{t-1}\left(\frac{2\pi k}{n}\right)\cos\left(\frac{2\pi(a+1)k}{n}\right).$$

Star graph. We set the vertex set of a simple star graph as

$$V = \{1, 2, \cdots, n\}$$

with the node 1 being the central node. This gives the edge set

$$E = \{(1, i), 2 \leq i \leq n\}.$$

Lemma 3 The eigenvalues of the Laplacian of the star graph are

$$\begin{pmatrix} 0 & 1 & \cdots & 1 & n \end{pmatrix}$$

and the eigenvectors are $\delta_i - \delta_{i+1}$ for eigenvalues 1 and $2 \leq i \leq n-1$. The eigenvector corresponding to eigenvalue n is computed in the proof.

Proof. Let $\mathbb{1}$ denote the all one vector. Then by the definition of Lapalcian, $L\mathbb{1} = 0$ and eigenvalue 0 corresponds to the eigenvector $\mathbb{1}$. Now, the trace of the Laplacian is just the sum of its eigenvalues. Therefore,

$$\mathsf{Tr}(L) = 2n - 2.$$

Let v be the eigenvector for eigenvalue n. Then we know that

$$v \perp \mathsf{Span}\{\mathbb{1}, e_2 - e_3, e_3 - e_4, \cdots, e_{n-1} - e_n\}.$$

This implies that

$$n - 1 + v[1] = 0,$$

or that $v = \begin{pmatrix} -(n-1) & 1 & 1 & \cdots & 1 \end{pmatrix}^\top$. □

We can perform the spectral decomposition of the adjacency matrix of a star graph, but noting that the adjacency matrix of a star graph is

$$A = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

it is easy to compute the coordinates of any higher power of A. In particular, if p is an even power, then (u, v)-th coordinate of A_S^p is

$$A_{uv}^p = \begin{cases} (n-1)^{p/2} & u = v = 1 \\ 0 & u = 1 \text{ or } v = 1 \text{ and } u \neq v \\ (n-1)^{p/2-1} & \text{otherwise} \end{cases}$$

If p is an odd power, then (u, v)-th coordinate of A^p is

$$A_{uv}^p = \begin{cases} (n-1)^{(p-1)/2} & u = 1 \text{ or } v = 1 \\ 0 & \text{otherwise} \end{cases}$$

Since $W = \frac{A}{(n-1)}$ for a normalization constant $(n-1)$ to make W doubly stochastic, we have

$$\begin{aligned} \varepsilon_{u \to v} &\leq \sum_{p=1}^{T} A_{uv}^p \frac{\alpha}{\sigma^2 p (n-1)^p} \\ &= \frac{\alpha}{\sigma^2} \left(\sum_{p \text{ is odd}}^{T} \frac{A_{uv}^p}{p(n-1)^p} + \sum_{p \text{ is even}}^{T} \frac{A_{uv}^p}{p(n-1)^p} \right). \end{aligned}$$

We now compute the privacy loss for each case. Since we only care about the privacy loss when $u \neq v$, we consider the following two cases:

1. $u = 1$ or $v = 1$ and $u \neq v$. In this case, using Proposition 1, we have

$$\varepsilon_{u\to v} \leq \frac{\alpha}{\sigma^2} \sum_{p \text{ is odd}}^{T} \frac{(\sqrt{n-1})^{p-1}}{p(n-1)^p} \leq \frac{\alpha}{2\sigma^2\sqrt{n-1}} \ln\left(\frac{\sqrt{n-1}+1}{\sqrt{n-1}-1}\right).$$

2. $u \neq 1$ and $u \neq v$. In this case, using Proposition 1, we have the following

$$\begin{aligned}\varepsilon_{u\to v} &\leq \frac{\alpha}{\sigma^2} \sum_{p \text{ is even}}^{T} \frac{(n-1)^{p/2-1}}{p(n-1)^p} \\ &= \frac{\alpha}{\sigma^2(n-1)} \sum_{p \text{ is even}}^{T} \frac{(n-1)^{p/2}}{p(n-1)^p} \\ &\leq -\frac{\alpha}{\sigma^2(n-1)} \ln\left(1 - \frac{1}{n-1}\right).\end{aligned}$$

The above calculation is when the graph is simple. To make the Markov chain aperiodic, as before, we add self-loops with a small weight on the self-loop. For example, we consider the following adjacency matrix:

$$\widehat{A} = (1-\kappa)A + \kappa\mathbb{I}.$$

We pick $\kappa = \frac{1}{T^2}$, so that

$$\widehat{A}^p_{uv} \leq (1-\kappa)A^p_{uv}$$

for $u \neq v$ and $p \leq T$. Then if p is an even power, then (u, v)-th coordinate of $\widehat{A}^p$ is

$$\widehat{A}^p_{uv} \leq \begin{cases} (1-\kappa)(n-1)^{p/2} & u = v = 1 \\ 0 & u = 1 \text{ or } v = 1 \text{ and } u \neq v \\ (1-\kappa)(n-1)^{p/2-1} & \text{otherwise} \end{cases}$$

If p is an odd power, then (u, v)-th coordinate of $\widehat{A}^p$ is

$$\widehat{A}^p_{uv} \leq \begin{cases} (1-\kappa)(n-1)^{(p-1)/2} & u = 1 \text{ or } v = 1 \\ 0 & \text{otherwise} \end{cases}$$

Now again we have

$$\begin{aligned}\varepsilon_{u\to v} &\leq \sum_{p=1}^{T} \widehat{A}^p_{uv} \frac{\alpha}{\sigma^2 p(n-1)^p} \\ &= \frac{\alpha}{\sigma^2}\left(\sum_{p \text{ is odd}}^{T} \frac{\widehat{A}^p_{uv}}{p(n-1)^p} + \sum_{p \text{ is even}}^{T} \frac{\widehat{A}^p_{uv}}{p(n-1)^p}\right).\end{aligned}$$

Using the same calculation as before, we have for all $u \neq v$,

$$\epsilon_{u\to v} \leq \begin{cases} \frac{\alpha(1-\kappa)}{2\sigma^2\sqrt{n-1}} \ln\left(\frac{\sqrt{n-1}+1}{\sqrt{n-1}-1}\right) & u = 1 \text{ or } v = 1 \text{ and } u \neq v \\ -\frac{\alpha(1-\kappa)}{\sigma^2(n-1)} \ln\left(1 - \frac{1}{n-1}\right) & u \neq 1 \end{cases}.$$

In particular, this means that the privacy loss for the apex node is the most. In contrast, the nodes on the arms have approximately $\sqrt{n}$ more privacy than the apex node, which is what we expect: the apex node is the only one communicating with every other node in the graph.