Topology-Dependent Privacy Bound for Decentralized Federated Learning

Li, Qiongxiu; Yu, Wenrui; Ji, Changlong; Heusdens, Richard

DOI
10.1109/ICASSP48485.2024.10446209

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024

ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446209

TOPOLOGY-DEPENDENT PRIVACY BOUND FOR DECENTRALIZED FEDERATED LEARNING

Qiongxiu Li[1]*, Wenrui Yu*[2]*, Changlong Ji*[3]*, Richard Heusdens*[2,4]

[1]Tsinghua University, China, qiongxiuli@mail.tsinghua.edu.cn
[2]Delft University of Technology, the Netherlands, w.yu-6@student.tudelft.nl
[3]Telecom SudParis, Institut Polytechnique de Paris, France, changlong.ji@telecom-sudparis.eu
[4]Netherlands Defence Academy, the Netherlands, r.heusdens@tudelft.nl

ABSTRACT

Decentralized Federated Learning (FL) has attracted significant attention due to its enhanced robustness and scalability compared to its centralized counterpart. It pivots on peer-to-peer communication rather than depending on a central server for model aggregation. While prior research has delved into various factors of decentralized FL such as aggregation methods and privacy-preserving techniques, one crucial aspect affecting privacy is relatively unexplored: the underlying graph topology. In this paper, we fill the gap by deriving a stringent privacy bound for decentralized FL under the condition that the accuracy is not compromised, highlighting the pivotal role of graph topology. Specifically, we demonstrate that the minimum privacy loss at each model aggregation step is dependent on the size of what we term as 'honest components', the maximally connected subgraphs once all untrustworthy participants are excluded from the networks, which is closely tied to network robustness. Our analysis suggests that attack-resilient networks will provide a superior privacy guarantee. We further validate this by studying both Poisson and power law networks, showing that the latter, being less robust against attacks, indeed reveals more privacy. In addition to a theoretical analysis, we consolidate our findings by examining two distinct privacy attacks: membership inference and gradient inversion.

Index Terms— Privacy, graph topology, peer-to-peer, decentralization, federated learning

1. INTRODUCTION

Federated Learning (FL) performs collaborative training between multiple participants/nodes/clients without directly sharing each node's raw data [1]. It can be implemented in either a centralized/star topology or a decentralized topology. The centralized topology, which stands as the predominant topology in FL, employs a central server that interacts directly with each and every node. However, in real-world scenarios, maintaining such a centralized server can be challenging due to its high communication bandwidth demands and the requisite trust from all involved participants. Furthermore, centralized topologies possess an inherent vulnerability–a single point of failure-making them vulnerable to attacks aiming to bring down the entire network. As a remedy, decentralized FL offers an alternative by substituting the server with distributed processing protocols which require information exchange between (locally) connected nodes only. Examples of these protocols are the empirical methods where the data aggregation is done using average consensus techniques such as gossiping SGD [2], D-PSGD [3] and variations thereof [4–6].

Though FL does not require direct sharing of individual participants' private data, it is shown susceptible to privacy breaches. The exchanged information, such as gradients or weights, can inadvertently lead to potential data leakages. Most of the existing work focuses on the centralized FL, and examples of attacks include the membership inference attack [7, 8] and the gradient inversion attack [9–13]. The goal of the membership inference attack is to determine whether a particular data point was used for training the target model (being a member) or not (being a non-member). It has been recently shown in [8] that membership information can be leaked through gradients by exploiting the so-called gradient orthogonality in data instances in overparameterized neural networks. The gradient inversion attack employs an iterative method to find fake data that produces a gradient similar to the real gradient generated by the private data. Such attacks are based on the assumption that data samples, when producing analogous gradients, are likely to be congruent. As a consequence, many approaches attempt to protect privacy by protecting the local gradients held by each node from being revealed to others.

Many studies claim that mere decentralization, i.e., deploying distributed processing protocols, would enhance the users' privacy. This is because no single user is as powerful as the centralized server, seemingly reducing privacy concerns [14,15]. Yet, such an argument is challenged in a recent study [16] which shows that untrustworthy users can influence other users' updates, becoming as powerful as the central server in centralized FL. Consequently, it highlights the necessity of integrating cryptographic techniques, such as differential privacy (DP) [17] and secure aggregation (SA) [18], to further enhance the privacy of decentralized FL. DP methods, including the LEASGD approach [14] and ADMM-based approaches [19–23], pose an inherent trade-off between accuracy and privacy as employing DP will inevitably lead to a compromise in accuracy. The SA approaches, on the other hand, do not deteriorate the accuracy but require high communication overhead.

Decentralized FL's privacy has been analyzed via various aspects including different types of distributed tools, potential threats from dishonest users, and privacy-enhancing methods. However, an often overlooked yet critical factor is the underlying graph topology. Although [16] provides some insights indicating that a denser graph can reduce privacy risks, it's not just the density, but also the degree distribution that is pivotal. For instance, two graphs with identical density can exhibit vastly different privacy implications (as we will show later). We believe that a thorough investigation of topology is crucial in shaping decentralized FL's privacy, requiring an in-depth exploration. In this paper, we bridge this gap by analytically establishing a privacy bound on decentralized FL, highlighting the tight connection between graph topology and privacy. This newly derived bound is especially significant for real-world applications as it provides guidance on which topological structures intrinsically enhance

privacy. Our main results are summarized as follows:

1. For decentralized FL that guarantees the output accuracy of model aggregation is uncompromised, we derive a bound on the privacy loss, highlighting that privacy leakage is profoundly influenced by the underlying graph topology, or more precisely, the network robustness against attacks.
2. Our findings suggest that robust network topologies, which sustain connectivity under adversarial conditions, inherently offer better privacy protection. We confirm this by investigating two prevalent topologies: Poisson and power law networks, with the latter being less privacy-preserving. Our results are substantiated through two distinct privacy attack assessments: membership inference and gradient inversion.

2. PRELIMINARIES

This section reviews the necessary fundamentals for the paper.

2.1. Centralized federated learning

FL generally considers a centralized setting assuming there is a centralized server connected to a number of users/clients/nodes. Assuming there are n clients and denote $\boldsymbol{w}^{(t)}$ as the model weights at iteration t. A typical FL protocol works as follows:

1. Initialization: at iteration $t = 0$, the central server randomly initializes the weights $\boldsymbol{w}^{(0)}$ of the global model.
2. Local model training: at each iteration t, each user i first receives the model updates from the server and then calculates its local gradient, denoted as $\boldsymbol{g}_i^{(t)}$, based on one mini-batch sampled from its local dataset.
3. Model aggregation: the server gathers local gradients from users and aggregates them to obtain an updated global model. The aggregation is often done by weighted averaging and typically uniform weights are applied, i.e.,

$$\boldsymbol{g}_{\text{ave}}^{(t)} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i^{(t)} \tag{1}$$

After obtaining the aggregated gradient, each node i then updates its own model weight by $\boldsymbol{w}_i^{(t+1)} = \boldsymbol{w}_i^{(t)} - \eta \boldsymbol{g}_{\text{ave}}^{(t)}$, where η is a constant. The last two steps are repeated until the global model converges or a certain stopping criterion is met.

2.2. Decentralized federated learning

Decentralized FL works for cases where a trusted centralized server is not available. Many decentralized FL protocols work by deploying distributed average consensus algorithms to compute the average of local gradients, i.e., computing (1) without any centralized coordination. Examples are gossip [24], linear iterations [25], and convex optimization-based methods such as the ADMM [26] and the PDMM [27], which all rely on peer-to-peer communication over distributed networks. A distributed network is often modelled as an undirect graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, 2, ..., n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote the set of nodes and edges, respectively. Note that node i can only communicate with (neighboring) node j if $(i, j) \in \mathcal{E}$.

We now take the linear iteration algorithm [25] as an example to explain how to conduct peer-to-peer aggregation when computing (1). Let $\boldsymbol{x}^{(0)}$ be the so-called state vector of the network which is initialized with local gradients, i.e.,

$$\boldsymbol{x}^{(0)} = \boldsymbol{g}, \tag{2}$$

where $\boldsymbol{g} = (\boldsymbol{g}_1^\top, \boldsymbol{g}_2^\top, \ldots, \boldsymbol{g}_n^\top)^\top$ [1] is the vector of stacked local gradients of all nodes. The average gradient can be obtained by applying, at every step $r = 1, 2, \ldots, r_{\max}$, a linear transformation $\boldsymbol{A} \in \mathcal{A}$ where

$$\mathcal{A} = \left\{ \boldsymbol{A} \in \mathbb{R}^{n\times n} \mid \boldsymbol{A}_{ij} = 0 \text{ if } (i,j) \notin \mathcal{E} \text{ and } i \neq j \right\}, \tag{3}$$

such that the state vector $\boldsymbol{x}$ is updated as

$$\boldsymbol{x}^{(r+1)} = \boldsymbol{A}\boldsymbol{x}^{(r)}, \tag{4}$$

and the optimum solution is $\forall i \in \mathcal{V} : \ \boldsymbol{x}_i^* = \boldsymbol{g}_{\text{ave}}$. The structure of $\boldsymbol{A}$ reflects the connectivity of the network[2]. In order to correctly compute the average, that is, $\boldsymbol{x}^{(r)} \to n^{-1}\mathbf{1}\mathbf{1}^\top \boldsymbol{g}$ as $r \to \infty$ where $\mathbf{1} \in \mathbb{R}^n$ denote the vector of all ones, necessary and sufficient conditions for $\boldsymbol{A}$ are (i) $\mathbf{1}^\top \boldsymbol{A} = \mathbf{1}^\top$, (ii) $\boldsymbol{A}\mathbf{1} = \mathbf{1}$, (iii) $\alpha\left(\boldsymbol{A} - \frac{\mathbf{1}\mathbf{1}^\top}{n}\right) < 1$, where $\alpha(\cdot)$ denotes the spectral radius [25].

Hence, decentralized FL often shares the same updating procedure as the centralized case except at each model aggregation step for computing (1), where a peer-to-peer distributed algorithm is applied to average the local gradients.

2.3. Adversary model

We consider a commonly utilized passive (also known as honest-but-curious) adversary model for distributed systems. The passive adversary works by colluding a number of nodes in the network, termed as corrupt nodes. These nodes can share information together to enhance the chance of inferring the private information of the remaining nodes, referred to as honest nodes.

3. PRIVACY BOUND AND GRAPH TOPOLOGY

The privacy leakage depends on many factors, for example the type of learning protocols, the deployed privacy-preserving techniques, and the amount of corrupt nodes. However, another pivotal factor that has been overlooked is the underlying *graph topology*. As we shall see, the privacy of honest nodes depends on the graph topology, or more precisely, on the sizes of the honest components (maximally connected subgraphs formed by honest nodes) it belongs to after removing all corrupt nodes.

3.1. Minimum information loss

Let $\mathcal{V}_h$ and $\mathcal{V}_c = \mathcal{V} \setminus \mathcal{V}_h$ denote the set of honest and corrupt nodes in the network, respectively.

Theorem 1. *Let $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h)$ be the subgraph of $\mathcal{G}$ after all corrupt nodes are eliminated. Let $\mathcal{G}_{h,1}, \ldots, \mathcal{G}_{h,k_h}$ denote the components of $\mathcal{G}_h$ and let $\mathcal{V}_{h,k}$ be the vertex set of $\mathcal{G}_{h,k}$. Given any protocol that outputs $f(s_1, s_2, \ldots, s_n) = \sum_{i=1}^n s_i$ to every node, after the protocol, the partial sums*

$$\Big\{ \sum_{i\in\mathcal{V}_{h,k}} s_i \Big\}_{k=1,2,\ldots,k_h}, \tag{5}$$

will be revealed to the adversary.

Proof. We use the case $k_h = 2$ to explain the main idea, i.e., the honest nodes are disconnected into two components after removing the corrupt nodes. Denote $\mathcal{L}$ and $\mathcal{R}$ as the node set of two honest components and let $\boldsymbol{s}_{\mathcal{L}}$, $\boldsymbol{s}_{\mathcal{R}}$ and $\boldsymbol{s}_{\mathcal{V}_c}$ be vectors consisting of the inputs for two honest components, and the corrupt nodes, respectively.

[1] To avoid confusion, the superscript $^{(t)}$ is omitted here.

[2] For simplicity, we assume that $\boldsymbol{A}$ is constant for all iterations, representing a synchronous execution of the algorithm. For asynchronous implementation, the transformation depends on which node will update. The results presented here can be readily generalized to asynchronous cases by considering expected values.

Denote F as the protocol outputting $\sum_{i=1}^{n} s_i$. The adversary has the following view

$$\{\boldsymbol{s}_{\mathcal{V}_c}, \boldsymbol{r}_{\mathcal{V}_c}, \boldsymbol{m}_{\mathcal{L}}, \boldsymbol{m}_{\mathcal{R}}, f(\boldsymbol{s}_{\mathcal{L}}, \boldsymbol{s}_{\mathcal{V}_c}, \boldsymbol{s}_{\mathcal{R}})\}, \tag{6}$$

where $\boldsymbol{r}_{\mathcal{V}_c}$ is a vector containing the so-called randomness from the corrupt nodes, $\boldsymbol{m}_{\mathcal{L}}$ is a vector containing all messages received from nodes in $\mathcal{L}$ and similarly for $\boldsymbol{m}_{\mathcal{R}}$ in $\mathcal{R}$.

The adversary can emulate a protocol execution by determining the actions for the nodes in $\mathcal{R}$. This involves selecting certain inputs and randomness and then proceeding with the steps in F, using the messages $\boldsymbol{m}_{\mathcal{L}}$ as needed in F. If there is a discrepancy between an entry in $\boldsymbol{m}_{\mathcal{R}}$ and the view, the protocol will be terminated and restarted. Since $\boldsymbol{s}_{\mathcal{R}}$ and $\boldsymbol{r}_{\mathcal{R}}$ are valid choices (though there may be several others), the adversary will eventually succeed, implying that it will identify $\tilde{\boldsymbol{s}}_{\mathcal{R}}$ and $\tilde{\boldsymbol{r}}_{\mathcal{R}}$ that provide the exact information view as in equation (6). Given that the adversary utilizes the identical messages from the nodes in $\mathcal{L}$ as in the actual execution of F, and the accurate output is included in the view, we must have $f(\boldsymbol{s}_{\mathcal{L}}, \boldsymbol{s}_{\mathcal{V}_c}, \boldsymbol{s}_{\mathcal{R}}) = f(\boldsymbol{s}_{\mathcal{L}}, \boldsymbol{s}_{\mathcal{V}_c}, \tilde{\boldsymbol{s}}_{\mathcal{R}})$ and thus the adversary can determine

$$\sum_{i\in\mathcal{L}} s_i = f(\boldsymbol{s}_{\mathcal{L}}, \boldsymbol{s}_{\mathcal{V}_c}, \boldsymbol{s}_{\mathcal{R}}) - \sum_{i\in\mathcal{V}_c} s_i - \sum_{i\in R} \tilde{s}_i,$$

after knowing $\sum_{i\in\mathcal{L}} s_i$ the adversary can also determine $\sum_{i\in\mathcal{R}} s_i = \sum_{i=1}^{n} s_i - \sum_{i\in\mathcal{V}_c} s_i - \sum_{i\in\mathcal{L}} s_i$. The above proof can easily be extended to the case where $k_h > 2$ and in such cases the adversary will learn all the sums of the private data in each component. Hence, the proof is now complete. □

3.2. Lower bound of privacy

To develop an understanding on the minimum information loss, we use an information-theoretical measure, e.g., mutual information to quantify the privacy leakage. Let $\mathrm{I}(\cdot;\cdot)$ and $\mathrm{h}(\cdot)$ denote mutual information and differential entropy, respectively [28]. Suppose that the $\boldsymbol{s}_i$s are realizations of an i.i.d. random vector $\{S_1, \ldots, S_n\}$, and denote the variable $\boldsymbol{\Sigma}_k = \sum_{i\in\mathcal{V}_{h,k}} S_i$. Consider an honest node $i \in \mathcal{V}_h$, given the knowledge of (5) the adversary will learn the following information about its private data s_i:

$$\mathrm{I}(S_i; \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_{k_h}). \tag{7}$$

Note that the above bound is consistent with the results presented in [29–33], where it was shown that the partial sums of honest components are revealed when computing average/sum. Hence, the derived lower bound is tight.

Denote $m_k = |\mathcal{V}_{h,k}|, k \in \{1, 2, \ldots, k_h\}$ as the number of nodes within the respective honest components. To see how these sizes affect the privacy leakage, denote the variance of S_is as $\mathrm{var}(S_i) = \sigma^2$. For sufficiently large m_k, the variable $\boldsymbol{\Sigma}_k$ is approximately Gaussian distributed with variance $\mathrm{var}(\boldsymbol{\Sigma}_k) = m_k\sigma^2$; Consider an honest node $i \in \mathcal{V}_h$, let $\mathcal{G}_{h,1}$ denote the component that node i belongs to. Thus, (7) becomes

$$\begin{aligned}
\mathrm{I}(S_i; \boldsymbol{\Sigma}_1) &\overset{(a)}{=} \mathrm{h}(\boldsymbol{\Sigma}_1) - \mathrm{h}(\boldsymbol{\Sigma}_1|S_i) \\
&\overset{(b)}{\approx} \frac{1}{2}\left(\log(2\pi\mathrm{e}m_1\sigma^2) - \log(2\pi\mathrm{e}(m_1-1)\sigma^2)\right) \\
&= \frac{1}{2}\log\left(\frac{m_1}{m_1-1}\right) \\
&\approx \frac{1}{2(m_1-1)},
\end{aligned} \tag{8}$$

where (7) equals $\mathrm{I}(S_i; \boldsymbol{\Sigma}_1)$ uses the fact that S_i is independent of all $\boldsymbol{\Sigma}_k$s except of $\boldsymbol{\Sigma}_1$; (a) uses the definition of mutual information[3]; (b) assumes that m_1 is sufficiently large and uses the fact that the differential entropy of a Gaussian distribution with variance σ^2 is given by $\frac{1}{2}\log(2\pi\mathrm{e}\sigma^2)$.

[3]Shannon entropy $H(\cdot)$ can be used for discrete random variables.

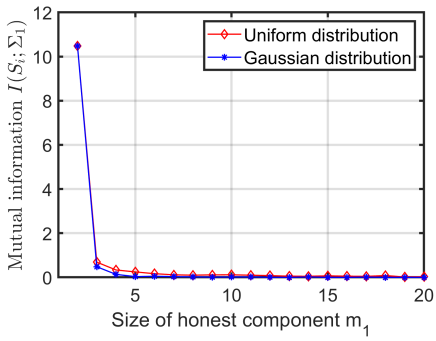

Fig. 1: Mutual information $\mathrm{I}(S_i; \boldsymbol{\Sigma}_1)$ of as a function of the honest component size m_1 for Gaussian and uniform distributed private data S_i.

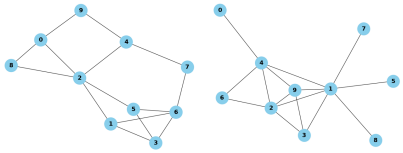

Fig. 2: Sample network with $n = 10$ nodes of (a) Poisson and (b) power law distribution.

Overall, we conclude that the privacy loss of an honest node is approximately inversely proportional to the number of nodes in the honest component it belongs to. That means, *the bigger the size of the honest component is, the less the privacy loss is*. This is verified in Fig. 1 where we depict the mutual information $\mathrm{I}(S_i; \boldsymbol{\Sigma}_1)$ as a function of the honest component size m_1 considering two distributions of S_is: Gaussian distributed with zero mean and unit variance, and uniformly distributed over the interval $[0, 1]$. For each experiment, we conducted 5000 Monto Carlo runs and adopted the nonparametric entropy estimation toolbox [34] to estimate the mutual information.

3.3. Graph topology and network robustness

In Theorem 1 and the subsequent result detailed in (8), we demonstrated that privacy loss is dependent on the size of honest components. This concept closely aligns with graph robustness or resilience when considering an adversary colluding a number of nodes in the network. The topology of a graph plays a pivotal role in understanding the robustness of networks [35]. One fundamental aspect of graph topology is the degree distribution, which characterizes the number of connections each node in the network possesses. The literature predominantly investigates two key types of degree distribution: Poisson and power law distributions.

Many networks, such as those resembling random structures, follow a Poisson distribution [Fig. 2 (a)], i.e., the degree of each node follows a Poisson distribution with a specific average degree. These networks exhibit a relatively uniform degree distribution, with most nodes having degrees close to the average. Conversely, many real-world networks, like the Internet and social or biological networks, exhibit power law degree distributions. In these power law graphs, only a few nodes have extremely high degrees, whereas most have significantly fewer connections [Fig. 2 (b)]. Assuming the adver-

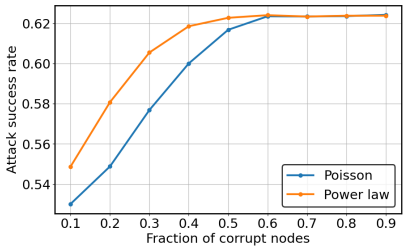

Fig. 3: Membership privacy leakage of the honest nodes, i.e., overall attack success rate, as a function of the fraction of corrupt nodes for both Poisson and power law networks.

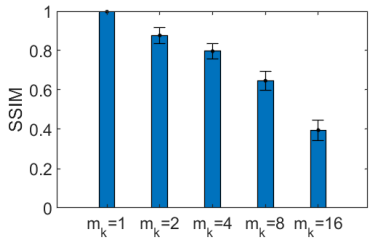

Fig. 4: Image quality comparison of the reconstructed inputs via inverting gradients using structural similarity index measure (SSIM) of different sizes of honest component $m_k = 1,\ 2,\ 4,\ 8,\ 16$.

sary is aware of the graph's connectivity, it will target and corrupt the pivotal nodes, i.e., those with high degrees. Given that power law networks rely on a few highly connected nodes to ensure network connectivity, they are notably more vulnerable to such targeted attacks compared to Poisson networks [35]. Such increased vulnerability can lead to greater privacy risks, as we'll discuss in the following section.

4. EXPERIMENTAL VALIDATIONS

We now proceed to consolidate our theoretical results via numerical validations. Recall that peer-to-peer model aggregation (1) is applied at all iterations, given the fact that the revealed information at each model aggregation step is the sum of local gradients of honest components (Theorem 1), we conclude that throughout the whole process the adversary can collect the following information:

$$\{\sum_{i\in\mathcal{V}_{h,k}} \boldsymbol{g}_i^{(t)}\}_{t\geq 0, k=1,2,\ldots,k_h}. \tag{9}$$

To quantitatively evaluate the privacy leakage caused by the above gradient information, we deploy two widely-used attacks including membership inference and gradient inversion. As we shall see, the privacy loss depends on the graph topology, or more precisely, the size of honest components after removal of corrupted nodes.

4.1. Membership inference attack

With the gradient information in (9) we deploy the recent attack proposed in [8] as it is specifically designed to infer membership information from gradients. We first examine the relationship between privacy loss and the type of graph topology. We randomly generated Poisson and power law networks with 10 nodes and 15 edges (see Fig 2 for examples). We employed the CIFAR-10 dataset, dividing it into 10 nodes, each containing 4000 data samples, and trained it using the AlexNet model. The test performance of this decentralized FL protocol is similar to the result reported in [8]. In Fig. 3 we demonstrate the attack success rate as a function of the fraction of corrupt nodes for both topologies, where the results are averaged

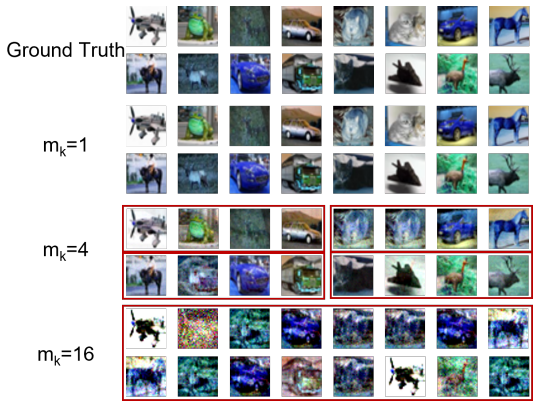

Fig. 5: Samples images of input reconstruction using gradient inversion attack under three different sizes of honest component $m_k = 1,\ 4,\ 16$, wherein the red box indicates that the corresponding samples are from the same component.

over 5000 Monte Carlo runs. Note that the adversary is strategic and has prior knowledge about the graph topology, thus it will corrupt the pivotal nodes first. To simulate such scenario, we assume that nodes are sequentially removed in descending order based on their degrees. Clearly, while in both cases the membership privacy leakage increases as the number of corrupt nodes increases, the power law networks reveal more privacy compared to Poisson networks. Hence, power law networks are more vulnerable to targeted attacks and thus pose a higher privacy risk compared to Poisson networks.

4.2. Gradient inversion attack

We now further investigate the size of the honest component and privacy leakage, we deploy the existing gradient inversion attack [9] to invert input samples from the observed sum of local gradients in each honest component. We simulated a randomly connected network with $n = 50$ nodes and the batchsize for generating the local gradient is set to one. Fig. 4 shows the quality of the reconstructed images for different sizes of honest components: $m_k = 1, 2, 4, 8, 16$. As expected, the reconstruction quality degrades with increasing size of the honest components. To visualize the results, in Fig. 5 we further demonstrate some examples of the reconstructed images (due to space limit only three cases $m_k = 1, 4, 16$ are shown). Hence, we can see that as the size of honest component m_k increases, there is an obvious degradation in the reconstruction quality. Especially for the case that there are images in the same component with the same label, e.g., the two cats in the top right of the case $m_k = 4$, the reconstruction performance is poor. This is consistent with the common observation that gradient inversion attack does not perform well in the case of repeated labels [12]. Overall, we conclude that the bigger the size of honest component, the less accurate will the reconstructed input be.

5. CONCLUSION

In this paper we emphasized the pivotal role of graph topology in the privacy of Decentralized FL. By establishing a delicate privacy bound for model aggregation, we unveiled the tight relationship between network structure and privacy risks. Through our exploration of Poisson and power law networks, we determined that certain topologies inherently offer better privacy guarantees when dealing with attacks. Our findings, validated by practical experiments including membership inference and gradient inversion, lay a foundation for advancements in the privacy of decentralized FL systems.

6. REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.* PMLR, pp. 1273–1282, 2017.

[2] P. H. Jin, Q. Yuan, F. Iandola, and K. Keutzer, "How to scale distributed deep learning?," *arXiv:1611.04581*, 2016.

[3] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[4] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D^2: Decentralized training over decentralized data," in *Proc. Int. Conf. Mach. Learn.* PMLR, pp. 4848–4856, 2018.

[5] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," *arXiv:1908.07782*, 2019.

[6] Q. Li, J. K. Gundersen, K. Tjell, R. Wisniewski, and M. G. Christensen, "Privacy-preserving distributed expectation maximization for gaussian mixture model using subspace perturbation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.* IEEE, pp. 4263–4267, 2022.

[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy*, pp. 3–18, 2017.

[8] J. Li, N. Li, and B. Ribeiro, "Effective passive membership inference attacks in federated learning against overparameterized models," in *Proc. Int. Conf. Learn. Represent.*, 2023.

[9] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[10] J. Geiping. H. Bauermeister, H. Dröge and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 16937–16947, 2020.

[11] H. Yin, A. Mallya, A. Vahdat, JM Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 16337–16346, 2021.

[12] J. Geng, Y. Mou, Q. Li, F. Li, O. Beyan, S. Decker, and C. Rong, "Improved gradient inversion attacks and defenses in federated learning," *IEEE Trans. Big Data*, 2023.

[13] H. Yang, M. Ge, K. Xiang, and J. Li, "Using highly compressed gradients in federated learning for data reconstruction attacks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 818–830, 2022.

[14] H. Cheng, P. Yu, H. Hu, S. Zawad, F. Yan, S. Li, H. Li, and Y. Chen, "Towards decentralized deep learning with differential privacy," in *Proc. Int. Conf. Cloud Comput.* Springer, pp. 130–145, 2019.

[15] T. Vogels, L. He, A. Koloskova, S. P. Karimireddy, T. Lin, S. U. Stich, and M. Jaggi, "Relaysum for decentralized deep learning on heterogeneous data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 28004–28015, 2021.

[16] D. Pasquini, M. Raynal, and C. Troncoso, "On the privacy of decentralized machine learning," *arXiv:2205.08443*, 2022.

[17] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Languages, Program.*, pp. 1–12, 2006.

[18] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 1175–1191, 2017.

[19] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 1, pp. 172–187, 2016.

[20] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proc. Int. Conf. Mach. Lear.*, pp. 5796–5805, 2018.

[21] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput.*, pp. 959–965, 2018.

[22] Z. Huang, R. Hu, Y. Gong, and E. Chan-Tin, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1002–1012, 2019.

[23] Z. Zhang, S. Yang, W. Xu, and K. Di, "Privacy-preserving distributed admm with event-triggered communication," *IEEE Trans. Neural Netw. Learn, Syst.*, 2022.

[24] A. G. Dimakis, S. Kar, J. M. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[25] A. Olshevsky and J. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 33–55, 2009.

[26] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends in Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[27] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Trans. Signal Process.*, vol. 4, no. 1, pp. 173–187, 2018.

[28] T. M. Cover and J. A. Tomas, *Elements of information theory*, John Wiley & Sons, 2012.

[29] G. Kreitz, M. Dam, and D. Wikström, "Practical private information aggregation in large networks," in *Inf. Secur. Technolo. Appl.* Springer, pp. 89–103, 2012.

[30] A. Beimel, "On private computation in incomplete networks," *Distrib. Comput.*, vol. 19, no. 3, pp. 237–252, 2007.

[31] Q. Li, I. Cascudo, and M. G. Christensen, "Privacy-preserving distributed average consensus based on additive secret sharing," in *Proc. Eur. Signal Process. Conf.*, pp. 1–5, 2019.

[32] Q. Li, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: A general framework," in *IEEE Trans. Signal Process., vol. 68, pp. 5983 - 5996*, 2020.

[33] Q. Li, J. S. Gundersen, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed processing: Metrics, bounds, and algorithms," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2090–2103, 2021.

[34] G. Ver Steeg, "Non-parametric entropy estimation toolbox (npeet)," *https://github.com/gregversteeg/NPEET*, 2000.

[35] C. Magnien, M. Latapy, and J. L. Guillaume, "Impact of random failures and attacks on poisson and power-law random networks," *ACM Comput. Surv.*, 2011.