On the Privacy Bound of Distributed Optimization and its Application in Federated Learning

Li, Qiongxiu; Lopuhaä-Zwakenberg, Milan; Yu, Wenrui; Heusdens, Richard

DOI
10.23919/EUSIPCO63174.2024.10715187

Publication date
2024

Document Version
Final published version

Published in
32nd European Signal Processing Conference, EUSIPCO 2024 - Proceedings

On the Privacy Bound of Distributed Optimization and its Application in Federated Learning

Qiongxiu Li*, Milan Lopuhaä-Zwakenberg†, Wenrui Yu‡ and Richard Heusdens§
*Tsinghua University, qiongxiuli@mail.tsinghua.edu.cn
†University of Twente, m.a.lopuhaa@utwente.nl
‡Delft University of Technology, yuwenrui0203@gmail.com
§Netherlands Defence Academy and Delft University of Technology, r.heusdens@tudelft.nl

***Abstract*—Analyzing privacy leakage in distributed algorithms is challenging as it is difficult to track the information leakage across different iterations. In this paper, we take the first step to conduct a theoretical analysis of the information flow in distributed optimization ensuring that gradients at every iteration remain concealed from others. Specifically, we derive a privacy bound on the minimum information available to the adversary when the optimization accuracy is kept uncompromised. By analyzing the derived bound we show that the privacy leakage depends heavily on the optimization objectives, especially the linearity of the system. To understand how the bound affects privacy, we consider two canonical federated learning (FL) applications including linear regression and neural networks. We find that in the first case protecting the gradients alone is inadequate for protecting the private data, as the established bound potentially exposes all sensitive information. For more complex applications such as neural networks, protecting the gradients can provide certain privacy advantages as it will be more difficult for the adversary to infer the private inputs. Numerical validations are presented to consolidate our theoretical results.**

I. Introduction

In recent years, distributed optimization has drawn increased attention due to the demand for big-data processing and easy access to ubiquitous computing units (e.g., a computer, a mobile phone or a CPU-equipped sensor). Popular applications include telecommunication [1], wireless sensor networks [2], cloud computing and machine learning [3]. Due to the absence of a central processing point (fusion center), participants/agents/nodes use their own processing ability to locally carry out simple computations and transmit only the required and partially processed data to neighboring nodes. However, such information exchange might cause severe information leakage about the sensitive data held by each participant, hindering the adoption of distributed optimization into privacy-sensitive applications such as medical systems, financial analysis, and smart grids [4]. Therefore, investigations on privacy-preservation in distributed optimization have received significant attention recently.

To solve distributed optimization problems, many optimization algorithms have been proposed, e.g., the dual ascent algorithm [5], ADMM [6] and PDMM [7]. While these optimization algorithms often do not require each participant to exchange his/her own private data directly, other information such as gradients[1] are exchanged. Such gradient information is related to the private data and thus can cause severe privacy leakage. As an example, it has been shown in federated learning [8] that revealing the local gradient information can cause a serious privacy breach. As a consequence, most existing approaches attempt to protect privacy by protecting the local gradients held by each node. These approaches can be broadly classified into two classes, one prioritizing accuracy and one prioritizing privacy. The former attempts to protect the privacy while keeping the accuracy uncompromised. These methods often require that the adversary has limited information about the participants, and are typically based on secure multiparty computation [9], [10], [11], [12], [13], [14], [15] or subspace perturbation [16], [17], [18], [19]. The latter class provides stronger privacy guarantees by assuming the adversary has the maximum amount of information available about the participants. However, they come with a trade-off between privacy and algorithm accuracy. Examples of this class are methods based on differential privacy [20], [21], [22], [23], [24], [25]. Connections and combinations of subspace perturbation and differential private mechanism can be found in [26], [27].

Even though there is evidence that revealing the gradients in distributed applications would cause privacy leakage [8], [28], they are mostly empirical and do not consider a fully decentralized setting. Theoretical analysis has been absent due to many challenges, for example, the difficulty of analyzing information leakage across successive algorithmic iterations. In this paper, we take the first step to theoretically analyze the privacy leakages in distributed optimization. We derive results showing that even though the gradients are being protected, a certain amount of privacy leakage is inevitable. Our main contributions are summarized as follows:

1) We derive a lower bound on the privacy leakage by analyzing the gradient information in privacy-preserving distributed optimization under the condition that perfect accuracy is achieved. To the best of our knowledge, this is the first theoretical privacy analysis in this context.
2) We show that privacy leakage depends heavily on the objective function, especially the linearity of the system. To have a better understanding of how the linearity of systems affects privacy, we analyze two cases including linear regression where the lower bound reveals all private information and an application of neural networks where

[1]Subgradients can be used for non-differentiable objective functions.

the lower bound also reveals some private information but it is less severe compared to the case of not protecting gradients.

We present experimental results obtained by computer simulations to substantiate our claims.

II. Preliminaries

In this section, we review the necessary fundamentals needed for the remainder of this paper.

A. Distributed optimization over networks

Distributed optimization has been intensively investigated due to its wide adaptability in many applications. The problems in those applications can be formulated as optimization over a graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of vertices, or nodes, in the network and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of (undirected) edges representing the interconnections between the nodes. For each node i we denote $\mathcal{N}_i = \{j \,|\, (i,j) \in \mathcal{E}\}$ as its set of neighbors, and $n_i = |\mathcal{N}_i|$ its degree.

Assume each node i has a local objective function $f_i \colon \mathbb{R}^u \to \mathbb{R} \cup \{\infty\}$. The nodes' aim is to jointly solve an optimization problem given some constraints over the network. This can be formulated as

$$\begin{aligned} \underset{\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}}{\text{minimize}} \quad & \sum_{i \in \mathcal{V}} f_i(\boldsymbol{x}_i), \\ \text{subject to} \quad & \forall (i,j) \in \mathcal{E} \,:\, \boldsymbol{B}_{i|j}\boldsymbol{x}_i + \boldsymbol{B}_{j|i}\boldsymbol{x}_j = \boldsymbol{b}_{i,j}, \end{aligned} \tag{1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^u$ denotes the local optimization variable at node i, and its associated objective function f_i is assumed to be convex, closed and proper (CCP). The notation $(\cdot)_{i|j}$ indicates that the variable/data relates to edge (i,j) but is held by node i. In many applications, the objective functions will have a similar form among nodes, i.e. $f_i(\boldsymbol{x}_i) = f(\boldsymbol{x}_i, \boldsymbol{s}_i)$, and $\boldsymbol{s}_i$ is user-specific (private) data that typically needs to be prevented from being revealed to others.

In this paper, we will focus on solving consensus problems. That is, we want to solve problem (1) under the constraint that all $\boldsymbol{x}_i$ are the same, i.e. $\boldsymbol{x}_i = \boldsymbol{x}$ for all $i \in \mathcal{V}$ and some $\boldsymbol{x} \in \mathbb{R}^u$. Assuming the graph is connected, this constraint is obtained by defining $\boldsymbol{B}_{i|j} \in \mathbb{R}$ as

$$\boldsymbol{B}_{i|j} = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{E} \text{ and } i < j, \\ -1, & \text{if } (i,j) \in \mathcal{E} \text{ and } i > j, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

and $\boldsymbol{b}_{i,j} = \boldsymbol{0}$, where $\boldsymbol{0}$ denote the zero vector in $\mathbb{R}^u$. We denote the optimal $\boldsymbol{x}_i$ by $\boldsymbol{x}^*$. That is, $\boldsymbol{x}^*$ is a feasible solution for (1).

B. Performances and evaluation metrics

The performances of privacy-preserving distributed optimization approaches are evaluated with the following two metrics.

1) Output accuracy: The output accuracy should reflect how close the optimization results of the privacy-preserving algorithms are compared to the original non-privacy-preserving ones. Here, we use the squared error defined as $\|\boldsymbol{x}_i^{(t_{\max})} - \boldsymbol{x}^*\|_2^2$ to quantify the accuracy, where $t_{\max}$ denotes the maximum number of iterations.

2) Individual privacy: In this paper, we consider two widely-used adversary models: the eavesdropping and passive (or honest-but-curious) adversary model. These two adversaries are assumed to be able to cooperate and their goal is to infer sensitive information about the private data $\boldsymbol{s}_i$. To this end, the adversaries have the following information at their disposal: (a) all messages communicated through non-securely encrypted channels, and (b) knowledge of all so-called corrupt nodes, nodes that participate in the designed protocol as expected.

As for the privacy metric, we adopt mutual information for quantifying information leakage. Given two continuous random variables X and Y, the mutual information is defined as

$$I(X;Y) = h(X) - h(X|Y), \tag{3}$$

where $I(\cdot;\cdot)$ and $h(\cdot)$ denote mutual information and differential entropy[2], respectively [29].

C. Distributed optimizers

A number of distributed optimization algorithms such as ADMM [6] and PDMM [7], [30] have been proposed to solve problem (1) in a decentralized manner. It was shown in [30] that ADMM and PDMM are fundamentally linked using monotone operator theory and operator splitting techniques in the sense that ADMM is a $\frac{1}{2}$-averaged version of PDMM (see [31] for details on monotone operator theory). In the ADMM-PDMM framework, at every iteration t, each node i updates its local optimization variable $\boldsymbol{x}_i$, and computes, for every neighbor $j \in \mathcal{N}_i$, an auxiliary variable $\boldsymbol{z}_{j|i}$. The update equations for node i are given by

$$\boldsymbol{x}_i^{(t+1)} = \arg\min_{\boldsymbol{x}_i} \left(f_i(\boldsymbol{x}_i) + \sum_{j \in \mathcal{N}_i} \boldsymbol{z}_{i|j}^{(t)\intercal} \boldsymbol{B}_{i|j} \boldsymbol{x}_i + \frac{c n_i}{2} \boldsymbol{x}_i^2 \right), \tag{4}$$

$$\forall j \in \mathcal{N}_i : \boldsymbol{z}_{j|i}^{(t+1)} = (1-\theta)\boldsymbol{z}_{j|i}^{(t)} + \theta\big(\boldsymbol{z}_{i|j}^{(t)} + 2c\boldsymbol{B}_{i|j}\boldsymbol{x}_i^{(t+1)}\big), \tag{5}$$

where $(\cdot)^\intercal$ denotes matrix transposition, c is a constant for controlling the convergence rate and $\theta \in (0,1]$ is a constant for controlling the averaging of Peaceman-Rachford splitting. The case $\theta = 1$ results in the PDMM algorithm, while the case $\theta = \frac{1}{2}$ (Douglas-Rachford splitting) results in the ADMM algorithm. ADMM converges for any CCP function, while convergence for PDMM is guaranteed when the objective function is strongly convex and differentiable. The optimality condition for (4) for each node $i \in \mathcal{V}$ is given by[3]

$$0 = \nabla f_i(\boldsymbol{x}_i^{(t)}) + \sum_{j \in \mathcal{N}_i} \boldsymbol{B}_{i|j}^\intercal \boldsymbol{z}_{i|j}^{(t-1)} + c n_i \boldsymbol{x}_i^{(t)}, \tag{6}$$

Given that the adversary can intercept all communications, it is evident from (6) that transmitting the auxiliary variables $\boldsymbol{z}_{j|i}$ would disclose $\nabla f_i(\boldsymbol{x}_i^{(t)})$, as $\boldsymbol{x}_i^{(t)}$ can be determined from (5). While encrypting the auxiliary variables at every iteration

[2]In the case of discrete random variables, we can replace the differential entropy by the Shannon entropy $H(\cdot)$.

[3]Note that ADMM can also be applied to non-differentiable problems where the optimality condition can be expressed in terms of subdifferentials: $0 \in \partial f_i(\boldsymbol{x}_i^{(t)}) + \sum_{j \in \mathcal{N}_i} \boldsymbol{B}_{i|j}^\intercal \boldsymbol{z}_{i|j}^{(t-1)} + c n_i \boldsymbol{x}_i^{(t)}$.

would address this concern, it is prohibitively resource-intensive. To address it, only initial values $\boldsymbol{z}_{j|i}^{(0)}$ are transmitted securely and the changes in the auxiliary variables, represented by $\Delta\boldsymbol{z}_{j|i}^{(t+1)} = \boldsymbol{z}_{j|i}^{(t+1)} - \boldsymbol{z}_{j|i}^{(t)}$ are sent without any encryption. Therefore, upon receiving $\Delta\boldsymbol{z}_{j|i}^{(t+1)}$, one can deduce the value of the auxiliary variable $\boldsymbol{z}_{j|i}^{(t+1)}$ as

$$\boldsymbol{z}_{j|i}^{(t+1)} = \boldsymbol{z}_{j|i}^{(t)} + \Delta\boldsymbol{z}_{j|i}^{(t+1)} = \sum_{\tau=1}^{t+1} \Delta\boldsymbol{z}_{j|i}^{(\tau)} + \boldsymbol{z}_{j|i}^{(0)}. \tag{7}$$

Hence, $\boldsymbol{z}_{j|i}^{(t+1)}$ can only be determined whenever $\boldsymbol{z}_{j|i}^{(0)}$ is known, so that eavesdropping only reveals

$$\left\{\Delta\boldsymbol{z}_{j|i}^{(t)} : t \geq 1, (i,j) \in \mathcal{E}\right\}. \tag{8}$$

In what follows we will consider the case $\theta = 1$ (PDMM) in order to simplify the equations. However, the results are easily generalized to all $\theta \in (0, 1]$.

III. Privacy bound

In this section, we first derive a privacy bound showing that the privacy leakage depends on the difference of gradients over successive iterations and then give some discussions on the derived bound.

A. Main result

Assumption 1. *Assume all the local optimization variables $\boldsymbol{x}_i$ reach convergence, i.e. $\boldsymbol{x}_i^{(t_{max})} = \boldsymbol{x}^*$ for all $i \in \mathcal{N}$.*

Theorem 1. *Let i be an honest node that has at least one corrupt neighbor. The adversary can learn $\{\boldsymbol{x}_i^{(t)} : t \geq 1\}$, as well as:*

$$\nabla f_i\big(\boldsymbol{x}_i^{(t)}\big) - \nabla f_i\big(\boldsymbol{x}_i^{(t+2)}\big), \quad t \geq 1. \tag{9}$$

Proof. We first prove that all $\{\boldsymbol{x}_i^{(t)} : t \geq 1\}$ are known to the adversary. Since the adversary has knowledge in (8), we then have

$$\begin{aligned}\Delta\boldsymbol{z}_{j|i}^{(t+1)} - \Delta\boldsymbol{z}_{i|j}^{(t)} &= \boldsymbol{z}_{j|i}^{(t+1)} - \boldsymbol{z}_{j|i}^{(t)} - \big(\boldsymbol{z}_{i|j}^{(t)} - \boldsymbol{z}_{i|j}^{(t-1)}\big) \\ &= 2c\boldsymbol{B}_{i|j}\boldsymbol{x}_i^{(t+1)} - 2c\boldsymbol{B}_{i|j}\boldsymbol{x}_i^{(t)} \\ &= 2c\boldsymbol{B}_{i|j}(\boldsymbol{x}_i^{(t+1)} - \boldsymbol{x}_i^{(t)}),\end{aligned} \tag{10}$$

where the second equality uses (5) (when setting $\theta = 1$). Hence, by collecting all the $\Delta\boldsymbol{z}_{j|i}^{(t+1)}$s, the adversary has knowledge of $\boldsymbol{x}_i^{(t+1)} - \boldsymbol{x}_i^{(t)}$ at each and every iteration. Moreover, since $\boldsymbol{x}_i^{(t)} \to \boldsymbol{x}^*$ for all $i \in \mathcal{N}$, the adversary can infer the individual $\boldsymbol{x}_i^{(t)}$s.

To prove (9), consider two successive $\boldsymbol{z}$-updates (5):

$$\boldsymbol{z}_{i|j}^{(t+1)} - \boldsymbol{z}_{i|j}^{(t-1)} = 2c\boldsymbol{B}_{i|j}\big(\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t+1)}\big). \tag{11}$$

By combining (11) and (6) at iteration t and $t+2$, we obtain

$$\begin{aligned}&\nabla f_i\big(\boldsymbol{x}_i^{(t)}\big) - \nabla f_i\big(\boldsymbol{x}_i^{(t+2)}\big) \\ &\quad= \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{i|j}^{\intercal}\left(\boldsymbol{z}_{i|j}^{(t+1)} - \boldsymbol{z}_{i|j}^{(t-1)}\right) + cn_i\left(\boldsymbol{x}_i^{(t+2)} - \boldsymbol{x}_i^{(t)}\right) \\ &\quad= cn_i\big(\boldsymbol{x}_i^{(t)} + \boldsymbol{x}_i^{(t+2)}\big) - 2c\sum_{j\in\mathcal{N}_i} \boldsymbol{x}_j^{(t+1)}.\end{aligned} \tag{12}$$

Hence, as all terms on the RHS are known to the adversary, (9) is known to the adversary, which completes the proof. □

B. Impact of maximum iterations

In practice, Assumption 1 may not always hold. Consequently, the adversary only knows $\boldsymbol{x}^*$ up to an error and can therefore only estimate the individual $\boldsymbol{x}_i^{(t)}$s up to a certain accuracy. To quantify this error, let $\boldsymbol{\epsilon}_i^{(t_{\max})} = \hat{\boldsymbol{x}}_i^{(t_{\max})} - \boldsymbol{x}_i^{(t_{\max})}$ be the adversary's estimation error in $\boldsymbol{x}_i^{(t_{\max})}$. Since $\boldsymbol{x}_i^{(t)} = \boldsymbol{x}_i^{(t_{\max})} - \sum_{\tau=t}^{t_{\max}-1}\big(\boldsymbol{x}_i^{(\tau+1)} - \boldsymbol{x}_i^{(\tau)}\big)$ and the adversary has knowledge of $\boldsymbol{x}_i^{(t+1)} - \boldsymbol{x}_i^{(t)}$ at every iteration, we conclude that

$$\hat{\boldsymbol{x}}_i^{(t)} = \boldsymbol{x}_i^{(t)} + \boldsymbol{\epsilon}_i^{(t_{\max})}, \quad 1 \leq t \leq t_{\max}, \tag{13}$$

so that the adversary can only estimate (9) up to a certain accuracy determined by $\boldsymbol{\epsilon}_i^{(t_{\max})}$.

C. Information theoretical measure

The following proposition shows that the differences between local gradients can reveal private data and the revealed information is no bigger than the information contained by the gradients themselves.

Proposition 1. *For each honest node i, let $\boldsymbol{S}_i$ and $\nabla f_i\big(\boldsymbol{X}_i^{(t)}\big)$ be random variables having realizations $\boldsymbol{s}_i$ and $\nabla f_i\big(\boldsymbol{x}_i^{(t)}\big)$, respectively. Moreover, denote $\mathcal{A}_c = \{\nabla f_i\big(\boldsymbol{X}_i^{(t)}\big) : t \geq 1\}$ and $\mathcal{A}_d = \{\nabla f_i\big(\boldsymbol{X}_i^{(t)}\big) - \nabla f_i\big(\boldsymbol{X}_i^{(t+2)}\big) : t \geq 1\}$. We then have*

$$I(\boldsymbol{S}_i; \mathcal{A}_c) \geq I(\boldsymbol{S}_i; \mathcal{A}_d). \tag{14}$$

Proof. Let $\mathcal{A}_e = \{\nabla f_i\big(\boldsymbol{X}_i^{(1)}\big), \nabla f_i\big(\boldsymbol{X}_i^{(2)}\big)\}$. Then

$$\begin{aligned}I(\boldsymbol{S}_i; \mathcal{A}_c) - I(\boldsymbol{S}_i; \mathcal{A}_d) &\overset{(a)}{=} I(\boldsymbol{S}_i; \mathcal{A}_e, \mathcal{A}_d) - I(\boldsymbol{S}_i; \mathcal{A}_d) \\ &\overset{(b)}{=} I(\boldsymbol{S}_i; \mathcal{A}_e | \mathcal{A}_d) \geq 0,\end{aligned}$$

where (a) holds since $\mathcal{A}_c$ can be constructed from $\mathcal{A}_d \cup \mathcal{A}_e$, and (b) follows from the chain rule for mutual information. □

Note that how much information is leaked through the differences of gradients depends on the form of gradients, i.e, on the objective function $f_i(\cdot)$. In the coming section we will give two FL examples to provide a deeper understanding.

IV. Federated learning applications and numerical validations

We now investigate two concrete examples of FL.

A. Linear example: federated linear regression

For each node i, consider an integer p_i, a matrix $\boldsymbol{Q}_i \in \mathbb{R}^{p_i \times u}$ and a vector $\boldsymbol{y}_i \in \mathbb{R}^{p_i}$. The pair $\boldsymbol{s}_i = (\boldsymbol{Q}_i, \boldsymbol{y}_i)$ represents the local information held by node i of a joint linear regression problem. Let $f_i(\boldsymbol{x}_i) := \frac{1}{2}||\boldsymbol{y}_i - \boldsymbol{Q}_i\boldsymbol{x}_i||_2^2$. Then $\nabla f_i(\boldsymbol{x}_i^{(t)}) = \boldsymbol{Q}_i^{\intercal}(\boldsymbol{Q}_i\boldsymbol{x}_i^{(t)} - \boldsymbol{y}_i)$ and the optimal $\boldsymbol{x}$ is given by $\boldsymbol{x}^* = \big(\sum_{i\in\mathcal{V}} \boldsymbol{Q}_i^{\intercal}\boldsymbol{Q}_i\big)^{-1}\big(\sum_{i\in\mathcal{V}} \boldsymbol{Q}_i^{\intercal}\boldsymbol{y}_i\big)$. Hence, (12) becomes

$$\begin{aligned}\nabla f_i(\boldsymbol{x}_i^{(t)}) - \nabla f_i(\boldsymbol{x}_i^{(t+2)}) &= \boldsymbol{Q}_i^{\intercal}\boldsymbol{Q}_i\big(\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_i^{(t+2)}\big) \\ &= cn_i\big(\boldsymbol{x}_i^{(t)} + \boldsymbol{x}_i^{(t+2)}\big) - 2c\sum_{j\in\mathcal{N}_i} \boldsymbol{x}_j^{(t+1)}.\end{aligned}$$

Let $\boldsymbol{h}_i^{(t)} = cn_i\big(\boldsymbol{x}_i^{(t)} + \boldsymbol{x}_i^{(t+2)}\big) - 2c\sum_{j\in\mathcal{N}_i} \boldsymbol{x}_j^{(t+1)}$. Defining $\boldsymbol{X}_{i,u}^{(t)} = \big(\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_i^{(t+2)}, \ldots, \boldsymbol{x}_i^{(t+u-1)} - \boldsymbol{x}_i^{(t+u-1)}\big) \in \mathbb{R}^{u\times u}$ and

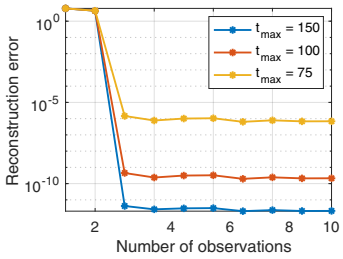

Fig. 1. Reconstruction error in $\boldsymbol{Q}_1^\intercal \boldsymbol{Q}_1$ as a function of the number of observations.

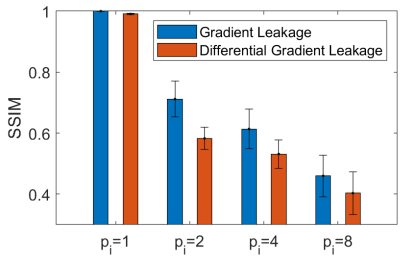

Fig. 2. Image quality comparison of the reconstructed inputs via inverting gradients and differential gradients using structural similarity index measure (SSIM) for different batchsizes $p_i = 1, 2, 4, 8$.

$\boldsymbol{H}_{i,u}^{(t)} = \left(\boldsymbol{h}_i^{(t)}, \ldots, \boldsymbol{h}_i^{(t+u-1)}\right) \in \mathbb{R}^{u\times u}$, we have $\boldsymbol{Q}_i^\intercal \boldsymbol{Q}_i \boldsymbol{X}_{i,u}^{(t)} = \boldsymbol{H}_{i,u}^{(t)}$, and thus

$$\boldsymbol{Q}_i^\intercal \boldsymbol{Q}_i = \boldsymbol{H}_{i,u}^{(t)} \left(\boldsymbol{X}_{i,u}^{(t)}\right)^{-1}.$$

Hence, we need u (linearly independent) observations in order to determine $\boldsymbol{Q}_i^\intercal \boldsymbol{Q}_i$. That is, knowing the difference of gradient does not prevent the revealing of sensitive information $\boldsymbol{Q}_i^\intercal \boldsymbol{Q}_i$.

Note that we only need u iterations to obtain the u independent observations and that it does not matter which u observations we take; the first u observations are as good as the last ones. What matters to the adversary is that $t_{\max}$ is sufficiently large so that the estimation error $\boldsymbol{\epsilon}_i^{(t_{\max})}$ in (13) is sufficiently small. This is illustrated in Fig. 1 which shows the reconstruction error in $\boldsymbol{Q}_1$ for a fully connected network with $n = 3$ nodes as a function of the number of observations for three different values of $t_{\max}$. The dimension of the primal variables is $u = 3$ and we assume that each node has $p = 5$ feature vectors. That is, $\boldsymbol{Q}_i \in \mathbb{R}^{5\times 3}$ and $\boldsymbol{y} \in \mathbb{R}^5$ which are generated at random (zero mean, unit variance Gaussian distributed). Results are averaged over 10^3 runs. Similar results are found for $\boldsymbol{Q}_2^\intercal \boldsymbol{Q}_2$ and $\boldsymbol{Q}_3^\intercal \boldsymbol{Q}_3$. Clearly, the larger $t_{\max}$ is, the smaller the input reconstruction error is. Hence, with sufficient iterations, the sensitive data $\boldsymbol{Q}_i^\intercal \boldsymbol{Q}_i$ can be reconstructed by the adversary (with negligible error). In this case, techniques like differential privacy [32] can be deployed to protect the sensitive information of individuals.

B. Non-linear example: neural networks

Unlike the above linear regression example, it is very difficult to derive an analytical solution for non-linear applications such as deep neural networks. Thus, we use a numerical method to estimate private information. In conventional centralized FL where the gradient is directly shared, a series of so-called gradient inversion attacks [33], [28] are proposed to utilize the leaked gradients to approximate the local dataset iteratively. In what follows we will first briefly discuss how traditional attacks work and then explain how to apply them given the derived lower bound.

1) Gradient Leakage: Let $(\boldsymbol{s}_i, \boldsymbol{\ell}_i)$ denote the local private data and label held by node i. Thus $f_i(\boldsymbol{x}_i, (\boldsymbol{s}_i, \boldsymbol{\ell}_i))$ denotes the cost/objective function of node i and $\boldsymbol{x}_i$ is the model weight to be learned. Given the knowledge of the local gradient $\nabla f_i(\boldsymbol{x}_i, (\boldsymbol{s}_i, \boldsymbol{\ell}_i))$, the adversary can (partially) recover the input data $(\boldsymbol{s}_i, \boldsymbol{\ell}_i)$ as [33]

$$(\boldsymbol{s}_i'^*, \boldsymbol{\ell}_i'^*) = \arg\min_{\boldsymbol{s}_i', \boldsymbol{\ell}_i'} \left\|\nabla f_i(\boldsymbol{x}_i, (\boldsymbol{s}_i', \boldsymbol{\ell}_i')) - \nabla f_i(\boldsymbol{x}_i, (\boldsymbol{s}_i, \boldsymbol{\ell}_i))\right\|^2,$$

or variants thereof [28]. In fact, the gradient inversion attack iteratively finds input data that produce a gradient similar to the gradient generated by the (private) input data.

2) Differential Gradient Leakage: Similar to the attack described above, we can recover private information from differential gradient leakage for fully decentralized systems given by

$$(\boldsymbol{s}_i'^*, \boldsymbol{\ell}_i'^*) = \arg\min_{\boldsymbol{s}_i', \boldsymbol{\ell}_i'} \Big\|\nabla f_i\left(\boldsymbol{x}_i^{(t)}, (\boldsymbol{s}_i', \boldsymbol{\ell}_i')\right) - \nabla f_i\left(\boldsymbol{x}_i^{(t+2)}, (\boldsymbol{s}_i', \boldsymbol{\ell}_i')\right) - \left(\nabla f_i\left(\boldsymbol{x}_i^{(t)}, (\boldsymbol{x}_i, \boldsymbol{\ell}_i)\right) - \nabla f_i\left(\boldsymbol{x}_i^{(t+2)}, (\boldsymbol{s}_i, \boldsymbol{\ell}_i)\right)\right)\Big\|^2.$$

To demonstrate the leaked information caused by differences of gradients, termed as 'differential gradient leakage', we consider a classification problem with the two-layer perceptron on the MNIST dataset [34]. We first verify the test performance of fully distributed FL using PDMM. To simulate a fully decentralized system, we generate a connected random geometric graph [35] with $n = 10$ and randomly split the dataset into $n = 10$ folds and each node holds one fold. As expected, the test accuracy is similar to the centralized case using FedAvg [36], which is around 91%. Therefore, we will compare their privacy leakages.

We consider an example with 50 nodes and each node i randomly selects p_i data samples from the dataset. In Fig. 2 we compare the quality of reconstructed inputs using gradients and difference of gradients under different batchsizes $1, 2, 4, 8$, the quality is quantified using structural similarity index measure (SSIM) and the results are averaged over the complete dataset. While for both cases the reconstruction quality degrades when increasing batchsize, the reconstruction quality of inverting gradients is clearly better than the case of inverting difference of gradients. This is further illustrated in Fig. 3 where some example images are demonstrated for visualization. Due to the space limit, we only demonstrate three cases for $1, 2, 8$ and in each case 8 images are randomly selected from the nodes. Overall, we conclude that the difference of gradients can reveal sensitive information about the input private data and the reconstructed inputs are less accurate compared to the traditional case of revealing gradients directly.

V. Conclusion

In this paper, we investigated privacy leakage in the context of privacy-preserving distributed optimization through a gradient flow analysis. We derived a theoretical bound on the minimum information available to the adversary, showing that

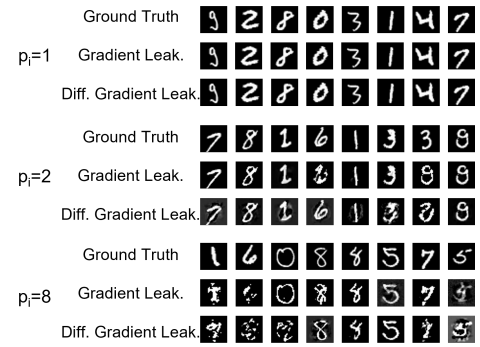

Fig. 3. Reconstructed inputs through inverting gradients and differential gradients using the MNIST datasets for different batchsizes $p_i = 1, 2, 8$.

the difference of local gradient is revealed. With this bound we quantify the privacy leakage given the objective function. By analyzing two FL applications, we showed that for linear regression the private information is inherently revealed during the iterations. For neural network applications, the derived bound also reveals sensitive information about the private data but it is less severe compared to the traditional case where local gradients are directly revealed. We substantiated our claims with both theoretical investigations and numerical simulations.

ACKNOWLEDGMENT

This research has been partially funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233 and ERC consolidator grant 864075 "Caesar".

REFERENCES

[1] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, 2008.

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[3] D. Sontag, A. Globerson, and T. Jaakkola, "Introduction to Dual Decomposition for Inference," *Optim. Mach. Learn.*, 2011.

[4] G. Giaconi, D. Gündüz, H. V. Poor, "Privacy-aware smart metering: Progress and challenges," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 59–78, 2018.

[5] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," *Math. Program.*, vol. 14, pp. 224–248, 1978.

[6] 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.

[7] 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.

[8] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, p. 603–618, 2017.

[9] N. Gupta, J. Katz, N. Chopra, "Privacy in distributed average consensus," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9515–9520, 2017.

[10] 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.

[11] Q. Li and M. G. Christensen, "A privacy-preserving asynchronous averaging algorithm based on shamir's secret sharing," in *Proc. Eur. Signal Process. Conf.*, pp. 1–5, 2019.

[12] Z. Xu and Q. Zhu, "Secure and resilient control design for cloud enabled networked control systems," in *Proc. 1st ACM Workshop Cyber-Phys. Syst.-Secur. Privacy.*, pp. 31–42, 2015.

[13] N. M. Freris and P. Patrinos, "Distributed computing over encrypted data," in *Proc. IEEE 54th Annu. Allerton Conf. Commun., Control, Comput.*, pp. 1116–1122, 2016.

[14] Y. Shoukry et al., "Privacy-aware quadratic optimization using partially homomorphic encryption," in *Proc. IEEE 55th Conf. Decis. Control.*, pp. 5053–5058, 2016.

[15] C. Zhang, M. Ahmad, and Y. Wang, "ADMM based privacy-preserving decentralized optimization," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 3, pp. 565–580, 2019.

[16] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, pp. 5895–5899, 2020.

[17] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimization-based privacy-preserving distributed least squares via subspace perturbation," in *Proc. Eur. Signal Process. Conf.*, 2020.

[18] 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.

[19] Q. Li, R. Heusdens and M. G. Christensen, "Communication efficient privacy-preserving distributed optimization using adaptive differential quantization," *Signal Process.*, 2022.

[20] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization.," in *Proc. Int. Conf. Distrib. Comput. Netw*, pp. 1–10, 2015.

[21] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Trans. Autom. Control.*, , no. 1, pp. 50–64, 2016.

[22] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 395–408, 2018.

[23] 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.

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

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

[26] Q. Li, M. Lopuhaä-Zwakenberg, R. Heusdens, and M. G. Christensen, "Two for the price of one: communication efficient and privacy-preserving distributed average consensus using quantization," in *30th Proc. Eur. Signal Process.* IEEE, pp. 2166–2170, 2022.

[27] M. Lopuhaä-Zwakenberg Q. Li, J. S. Gundersen and R. Heusdens, "Adaptive differentially quantized subspace perturbation (adqsp): A unified framework for privacy-preserving distributed average consensus," *IEEE Trans. Inf. Forensics Security.*, 2023.

[28] 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.

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

[30] T. Sherson, R. Heusdens, W. B. Kleijn, "Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 2, pp. 334–347, 2018.

[31] E. Ryu, S. P. Boyd, "Primer on monotone operator methods," *Appl. Comput. Math.*, vol. 15, no. 1, pp. 3–43, 2016.

[32] C. Dwork, "Differential privacy," in *ICALP, pp. 1–12*, 2006.

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

[34] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[35] J. Dall and M. Christensen, "Random geometric graphs," *Phys. Rev. E*, vol. 66, no. 1, pp. 016121, 2002.

[36] 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.