Local and Central Differential Privacy for Robustness and Privacy in Federated Learning

Mohammad Naseri
University College London (UCL)
mohammad.naseri.19@ucl.ac.uk

Jamie Hayes*
DeepMind
jamhay@google.com

Emiliano De Cristofaro
UCL & Alan Turing Institute
e.decristofaro@ucl.ac.uk

***Abstract*—Federated Learning (FL) allows multiple participants to train machine learning models collaboratively by keeping their datasets local while only exchanging model updates. Alas, this is not necessarily free from privacy and robustness vulnerabilities, e.g., via membership, property, and backdoor attacks. This paper investigates whether and to what extent one can use differential Privacy (DP) to protect both privacy and robustness in FL. To this end, we present a first-of-its-kind evaluation of Local and Central Differential Privacy (LDP/CDP) techniques in FL, assessing their feasibility and effectiveness.**

Our experiments show that both DP variants do defend against backdoor attacks, albeit with varying levels of protection-utility trade-offs, but anyway more effectively than other robustness defenses. DP also mitigates *white-box* membership inference attacks in FL, and our work is the first to show it empirically. Neither LDP nor CDP, however, defend against property inference. Overall, our work provides a comprehensive, re-usable measurement methodology to quantify the trade-offs between robustness/privacy and utility in differentially private FL.

I. Introduction

Aiming to increase privacy and communication efficiency, *Federated Learning* (FL) [71] has emerged as a compromise between a centralized approach to Machine Learning (ML), where training data is pooled at a central server, and the alternative of only training local models, client-side. Multiple "participants" collaborate in solving an ML problem by keeping their data on their device and only exchanging model parameters [57]. Currently, FL is used in many different settings, e.g., Google's predictive keyboard [38, 113], voice assistants [6], emoji prediction [91], healthcare [10, 98], etc.

Alas, previous work has highlighted robustness and privacy weaknesses of FL [57]. For instance, poisoning attacks may reduce the model's accuracy or make it misbehave on specific inputs. In particular, in *backdoor* attacks, a malicious client injects a backdoor into the final model to corrupt the performance of the trained model on specific sub-tasks [4]. Moreover, an adversary may be able to infer *membership* [77] (i.e., learn if a data point is part of a target's training set), or *properties* [74] of the training data.

*Work done while at UCL.

Network and Distributed Systems Security (NDSS) Symposium 2022
24-28 April 2022, San Diego, CA, USA
ISBN 1-891562-74-6
https://dx.doi.org/10.14722/ndss.2022.23054
www.ndss-symposium.org

Prior work has investigated robustness defenses primarily based on robust, Byzantine-robust aggregation algorithms [8, 97, 117]. As for privacy, FL techniques have since their inception supported homomorphic encryption so that the server can only decrypt the aggregates [9], but this does not eliminate leakage from the aggregates [74]. The established framework to define functions that are free from adversarial inferences is Differential Privacy (DP) [26], allowing to bound the privacy loss of individual data subjects by adding noise. In the context of FL, one can use two DP variants: 1) Local DP (LDP) [85], where each participant adds noise before sending updates to the server; and 2) Central DP (CDP) [33, 72], where it is the server to apply a DP aggregation algorithm.

Problem Statement. Backdoor attacks in FL are first explored by Bagdasaryan et al. [4]. However, [4] does not present any working defense; instead, they discuss how existing ones (including participant-level CDP) are not suited to FL. Sun et al. [103] introduce two defenses against backdoor attacks in FL: bounding the norm of gradient updates and adding Gaussian noise, aiming to reduce the effect of poisonous data. Although they are not meant to provide privacy, we believe they could, in theory, be used for this purpose because of the noisy updates; however, we show that, in practice, they do not defend against inference attacks.

Consequently, our work sets out to tackle the following research question: can we deploy defenses to mitigate *both* backdoor *and* inference attacks? If so, how and with what utility trade-offs? We investigate whether Differential Privacy, both in its CDP and LDP instantiations, can be used for this purpose, considering different scenarios of how they can be applied and providing an extensive measurement study of their real-world effectiveness. Our main intuition is that CDP limits the information exposed about a specific participant, while LDP does so for records in a participant's dataset; in both cases, the impact of poisonous data should be reduced while simultaneously protecting against inference attacks.

Technical Roadmap. We introduce an analytical approach to understand the effectiveness of LDP and CDP to protect federated models from both backdoor and inference attacks while maintaining good utility. This entails addressing a few challenges. First, we do not know how to compare the protection they yield, as they are not meant to guarantee robustness, and even for privacy, their definitions capture slightly different notions. Moreover, there is no straightforward analytical way to determine the effect of LDP/CDP on utility.

We run experiments on three datasets for backdoor attacks:

EMNIST (handwritten digits), CIFAR10 (different classes of images), Reddit comments, and Sentiment140 (tweets). We consider two FL settings with varying numbers of participants/attackers. Further, we experiment with white-box* membership inference attacks [77] on the CIFAR100 dataset (different classes of images), Purchase100 (records of purchases), and Texas100 (records of hospital discharges). We consider both active and passive attacks, run from both server and participant sides. Finally, we run property inference attacks [74] for a gender classification task on the LFW dataset.

Main Findings. In summary, we find that:

- Both LDP and CDP do defend against backdoor attacks, albeit with varying levels of protection and utility, but overall better than prior work.
- Applying LDP only on non-attackers in FL can actually boost the backdoor attack accuracy.
- Both LDP and CDP also protect against (white-box) membership inference [77]; although this is not entirely unexpected, we are the first to show that this does not necessarily come with a high cost on utility.
- LDP does not work against property inference attacks. Although CDP can, in theory, defend against the attack, it does so with a significant loss in utility.

More precisely, our robustness experiments show that, with 2,400 participants on EMNIST, and with 1% of them selected at every round and 1 attacker performing the backdoor attack, LDP and CDP (with $\epsilon = 3$) reduce the accuracy of the attack from 88% to 10% and 6%, respectively, with only a limited reduction in utility. By comparison, using defenses from [103] (see Section III-B), attack accuracy only goes down to 37% with norm bounding and 16% with so-called weak DP, which does not provide privacy, unlike LDP/CDP.

Our membership inference experiments show that, with 4 participants (the same setting as [77]), LDP ($\epsilon = 8.6$) and CDP ($\epsilon = 5.8$) reduce the accuracy of an active (local) attack from 75% to 55% and 52%, respectively, on CIFAR100, and from 68% to 54% and 55% on Purchase100 (50% baseline). As mentioned, LDP is ineffective against property inference, especially when the target participant has many data points with the property; this is because LDP only provides record-level privacy. Although CDP can, in theory, defend against the attack, it does so with a significant loss in utility. For instance, with 10 participants on LFW, CDP only yields a 57% accuracy on the main task when $\epsilon = 4.7$; with $\epsilon = 8.1$, main task accuracy reaches 83%, but with no prediction, the accuracy of the property inference only goes down from 87% to 85%.†

Contributions. The main contributions of our work include:

1) We are the first to propose the use of LDP to mitigate backdoor attacks against FL. In addition, we are the first to experimentally show that CDP mitigates backdoor attacks in FL. In fact, both LDP and CDP can defend better than the state of the art [103] (which used norm bounding and weak DP), additionally protecting privacy as well, while only slightly decreasing utility.
2) We are the first to show that both LDP and CDP can effectively (i.e., without destroying utility) defend against white-box membership inference attacks in FL [77], and more so than theoretically expected.
3) We provide a re-usable measurement framework to quantify the trade-offs between robustness/privacy and utility yielded by LDP and CDP in FL.

*White-box refers to the attacker having complete knowledge of the system's parameters, except for the participants' datasets.

†Unlike [74], we are able to make our models converge by increasing the privacy budget (e.g., $\epsilon > 8$), although this is not enough to thwart the attack.

II. Preliminaries

We now review Federated Learning (FL) and Differential Privacy (DP). Readers who are already familiar with these concepts can skip this section without loss of continuity.

A. Federated Learning (FL)

FL is a collaborative learning setting to train machine learning models [71]. It involves N participants (or clients), each holding their own private dataset, and a central server (or aggregator). Unlike the traditional centralized approach, data is not pooled at a central server; whereas, participants train models locally and exchange updated parameters with the server, which aggregates and sends them to the participants.

FL involves multiple rounds: In round 0, the server generates a model with random parameters θ_0, which is sent to all participants. Then, at each round r, K out of N participants are selected at random; each participant i locally computes training gradients according to their local dataset D_i and sends the updated parameters to the server. The latter computes the global parameters $\theta_r = \Sigma_{i=1}^{K} \theta_i / K$ and sends them to all N participants for the next round. After a certain number of rounds (R), the model is finalized with parameters θ_R.

There are different privacy-enhancing techniques used in FL. Homomorphic Encryption (HE) can be used to encrypt participants' parameters in such a way that the server can only decrypt the aggregates [9, 19, 39]. However, this is resource-intensive and does not mitigate inference attacks over the output of the aggregation, as global parameters can still leak information [57]. Another approach is to use differentially private techniques, which we review next.

B. Differential Privacy (DP)

Differential Privacy provides statistical guarantees against the information an adversary can infer through the output of a randomized algorithm. It provides an unconditional upper bound on the influence of a single individual on the output of the algorithm by adding noise [26].

Definition 1: **Differential Privacy.** A randomized mechanism M provides (ϵ, δ)-*differential privacy* if for any two neighboring databases, D_1 and D_2, that differ in only a single record, and for all possible outputs $S \subseteq Range(A)$:

$$P[M(D_1 \in A)] \leq e^{\epsilon} P[M(D_2 \in A)] + \delta \quad (1)$$

The ϵ parameter (aka *privacy budget*) is a metric of privacy loss. It also controls the privacy-utility trade-off, i.e., lower ϵ values indicate higher levels of privacy, but likely reduce utility too. The δ parameter accounts for a (small) probability on which the upper bound ϵ does not hold. The amount of noise needed to achieve DP is proportional to the *sensitivity* of the output; this measures the maximum change in the output due to the inclusion or removal of a single record.

Function `Main()`:
  Initialize: model θ_0
  for *each round* $r = 1, 2, ...$ **do**
    $K_r \leftarrow$ randomly select K participants
    for *each participant* $k \in K_r$ **do**
      $\theta_r^k \leftarrow$ DP-SGD(...) *// This is done in parallel*
    end
    $\theta_r \leftarrow \Sigma_{i=1}^{K_r} \frac{n^k}{n} \theta_r^k$ *// n^k is the size of k's dataset*
  end
return

Function `DP-SGD` (*Clipping norm S, dataset D, sampling probability p, noise magnitude σ, learning rate η, Iterations E, loss function $L(\theta(x), y)$*):
  Initialize θ_0
  for *each local epoch i from 1 to E* **do**
    for *$(x, y) \in$ random batch from dataset D with probability p* **do**
      $g_i = \nabla_\theta L(\theta_i; (x, y))$
    end
    $Temp = \frac{1}{pD} \Sigma_{i \in batch} g_i min(1, \frac{S}{\|g_i\|_2}) + N(0, \sigma^2 I)$
    $\theta_{i+1} = \theta_i - \eta(Temp)$
  end
return θ_E

Algorithm 1: Local Differential Privacy in FL.

In ML, DP is used to learn a distribution of a dataset while providing privacy for individual records [54]. Differentially Private Stochastic Gradient Descent (DP-SGD) [1], and Private Aggregation of Teacher Ensembles (PATE) [81] are two different approaches to privacy-preserving ML; in this paper, we use the former. DP-SGD [1] uses a noisy version of stochastic gradient descent to find differentially private minima for the optimization problem. This is done by bounding the gradients and then adding noise with the help of the "moments accountant" technique to keep track of the spent privacy budget. Whereas PATE [81] provides privacy for training data using a student-teacher architecture.

C. DP in FL

As mentioned, in the context of FL, one can use one of two variants of DP, namely, *local* and *central* [33, 72, 85].

Local Differential Privacy (LDP). With LDP, the noise addition required for DP is performed locally by each participant. Each participant runs a random perturbation algorithm M and sends the results to the server. The perturbed result is guaranteed to protect an individual's data according to the ϵ value. This is formally defined next [24].

Definition 2: Let X be a set of possible values and Y the set of noisy values. M is (ϵ, δ)-*locally differentially private* (ϵ-LDP) if for all $x_1, x_2 \in X$ and for all $y \in Y$:

$$P[M(x) = y] \leq e^\epsilon P[M(x') = y] + \delta \quad (2)$$

We implement LDP in FL because participants use differentially private stochastic gradient descent (DP-SGD) [1] to train the model on their datasets. This approach allows us to use moments accountant to keep track of the spent privacy budget. Algorithm 1 shows how DP-SGD in LDP works.

Central Differential Privacy (CDP). With CDP, the FL aggregation function is perturbed by the server, and this provides *participant-level* DP. This guarantees that the output of the aggregation function is indistinguishable, with probability bounded by ϵ, to whether or not a given participant is part of the training process. In this setting, participants need to trust the server: 1) with their model updates, and 2) to correctly perform perturbation by adding noise, etc. While some degree of trust in the server is needed, this is a much weaker assumption than entrusting the server with the data itself. If anything, inferring training set membership or properties from the model updates is much less of a significant privacy threat than having data in the clear. Moreover, in FL, clients do not share entire datasets also for efficiency reasons and/or because they might be unable to for policy or legal reasons.

Function `Main()`:
  Initialize: model θ_0, Moment_Accountant(ϵ, N) *// N is the number of all participants*
  for *each round* $r = 1, 2, ...$ **do**
    $C_r \leftarrow$ randomly select participants with probability q
    $p_r \leftarrow$Moment_Accountant.get_privacy_spent() *// Returns the spent privacy budget for the current round*
    if $p_r > T$ *// If the spent privacy budget is greater than the threshold, return the current model* **then**
      return θ_r
    end
    for *each participant* $k \in C_r$ **do**
      $\Delta_k^{r+1} \leftarrow$Participant_Update($k, \theta_r$) *// This is done in parallel*
    end
    $S \leftarrow bound$
    $z \leftarrow noise\_scale$
    $\sigma \leftarrow zS/q$
    $\theta_{r+1} \leftarrow \theta_r + \Sigma_{i=1}^{C_r} \Delta_i^{r+1} / C_r + N(0, I\sigma^2)$
    Moment_Accountant.accumulate_spent_privacy(z)
  end
return

Function `Participant_Update(`k, θ_r`)`:
  $\theta \leftarrow \theta_r$
  for *each local epoch i from 1 to E* **do**
    for *batch $b \in B$* **do**
      $\theta \leftarrow \theta - \eta \nabla L(w; b)$
      $\Delta \leftarrow \theta - \theta_r$
      $\theta \leftarrow \theta_0 + \Delta \min(1, \frac{S}{\|\Delta\|_2})$
    end
  end
return $\theta - \theta_r$ *// This one is already clipped*

Algorithm 2: Central Differential Privacy in FL.

In this paper, we implement the CDP approach for FL discussed in [72] and [33], which is illustrated in Algorithm 2. The server clips the l_2 norm of participants' updates, then it aggregates the clipped updates and adds Gaussian noise to the aggregate. This prevents overfitting to any participant's updates. To track the privacy budget spent, the moments accountant technique from in [1] can be used.

III. Defending Against Backdoor Attacks in FL

In this section, we experiment with LDP and CDP against backdoor attacks, while also comparing our results with state-of-the-art defenses in terms of both robustness and utility.

A. Backdoor Attack

Backdoor attacks are a special kind of *poisoning* attacks. We first review the latter, then formally define the former.

Poisoning attacks can be divided into random and targeted ones. In the former, the attacker aims to decrease the accuracy of the final model; in the latter, the goal is to make the model output a target label pre-defined by the attacker [47]. In the context of FL, participants (and not the server) are potential adversaries. Random attacks are easier to identify as the server could check if the accuracy is below a threshold or not.

These attacks can be performed on data or models. In the former, the attacker adds examples to the training set to modify the final model's behavior. In the latter, she poisons the local model before sending it to the server. In both cases, the goal is to make the final model misclassify a set of predefined inputs. Remind that, in FL, data is never sent to the server. Therefore, anything that can be achieved with poisonous data is also feasible by poisoning the model [27, 68].

Backdoor attacks. A malicious client injects a backdoor task into the final model. Following [4, 5, 103], we consider targeted model poisoning attacks and refer to them as backdoor attacks. These attacks in FL are relatively straightforward to implement and rather effective [4]; it is not easy to defend against them, as the server cannot access participants' data as that would violate one of the main principles of FL.

The main intuition is to rely on a model-replacement methodology, similar to [4, 103]. In round r, the attacker attempts to introduce a backdoor and replaces the aggregated model with a backdoored one θ^*, by sending the following model update to the server:

$$\Delta\theta_r^{attacker} = \frac{\Sigma_{i=1}^{K} n_i}{\eta n_{attacker}} \cdot (\theta^* - \theta_r) \tag{3}$$

where n_i is #data points at participant i and η the server learning rate. Then, the aggregation in the next round yields:

$$\Delta\theta_{r+1} = \theta^* + \eta \frac{\Sigma_{i=1}^{K-1} n_i \Delta\theta_r^i}{\Sigma_{i=1}^{K} n_i} \tag{4}$$

If we assume the training process is in its last rounds, then the aggregated model is going to converge; therefore, model updates from non-attacker participants are small, and we would have $\Delta\theta_r \simeq \theta^*$.

B. Defenses

One straightforward defense against poisoning attacks is byzantine-resilient aggregation frameworks, e.g., Krum [8] or coordinate-wise median [117]. However, as showed in [5], these are not effective in the FL setting.

Overall, FL is vulnerable to backdoor attacks as it is difficult to control the local model submitted by malicious participants. Defenses that require access to the training data are not applicable as that would violate the privacy of the participants, defeating one of the main reasons to use FL in the first place. In fact, even defenses that do not require training data, e.g., DeepInspect [14], require inverting the model in order to extract the training data, thus violating the privacy-preserving goal of FL. Similarly, if a model is trained over encrypted data, e.g., using CryptoNet [34] or SecureML [75], the server cannot detect anomalies in participant's updates.

Sun et al. [103] show that the performance of backdoor attacks in FL ultimately depends on the fraction of adversaries and the complexity of the task. They also propose two defenses: 1) *Norm Bounding* and 2) *Weak DP*. Byzantine-robust defenses like Krum [117], Trimmed Mean [117], or Divide-and-Conquer (DnC) [97] are designed to defend against robustness attacks but not to provide privacy. Unlike them, norm bounding and "Weak DP" might potentially provide both robustness and privacy and are thus the main focus in this paper for comparisons.

Norm Bounding. If model updates received from attackers are over some threshold, then the server can simply ignore those participants. However, if the attacker is aware of the threshold, it can return updates within that threshold. Sun et al. [103] assume that the adversary has this strong advantage and apply norm bounding as a defense, guaranteeing that the norm of each model update is small. If we assume that the updates' threshold is T, then the server can ensure that norms of participants' updates are within the threshold:

$$\Delta\theta_{r+1} = \sum_{i=1}^{k} \frac{\Delta\theta_{r+1}^k}{\max\left(1, \frac{\|\Delta\theta_{r+1}^k\|_2}{T}\right)} \tag{5}$$

In their experiments, Sun et al. [103] show that this defense mitigates backdoor attacks, meaning that it provides robustness for participants. For instance, in an FL setting with 3,383 participants using the EMNIST dataset, with 30 clients per round and one of them performing the backdoor attack, they show that selecting 3 as the norm bound almost mitigates the attack while not affecting the utility (attack accuracy reduces from 89% to 5%).

Weak Differential Privacy. Sun et al. [103] also use an additional defense against backdoor attacks in FL whereby the server not only applies norm bounding but also adds Gaussian noise, further reducing the effect of poisonous data. Overall, this proves to be more effective at mitigating backdoor attacks, even though with a limited loss in utility. This mechanism is referred to as *"weak"* DP since, as explained next, it results in large privacy budgets and does not protect privacy.

Failure to protect privacy. Both norm bounding and weak DP do not defend against inference attacks. (We also confirm this, empirically, in Section IV-C.) First, norm bounding does not provide privacy as participants' updates are sent *in the clear*, and thus leak information about training data. Second, weak DP results in very large values of ϵ, as it adds noise at every round ignoring the noise added in previous rounds.

More specifically, in DP, the concept of *composability* ensures that the joint distribution of the outputs of differentially private mechanisms satisfies DP [73]. However, because of sequential composition, if there are n independent mechanisms, M_1, ..., M_n, with ϵ_1, ..., ϵ_n respectively, then a function g of those mechanisms $g(M_1, ..., M_n)$ is $(\sum_{i=1}^{n} \epsilon_i)$-differentially private. Therefore, if we assume that, at every round, the server applies an ϵ-differentially private mechanism on participants' updates, then this weak DP mechanism results in spending $r*\epsilon$ privacy budget after r number of rounds. This yields larger values of ϵ, and thus significantly less privacy for participants.

C. Experimental Setup

We experiment with both LDP and CDP in FL against backdoor attacks, and we compare it with existing defenses from [103]. We do so vis-à-vis different scenarios, applying:

1) CDP on all participants;
2) LDP on all participants (including attackers);
3) LDP on non-attackers, while attackers opt-out;
4) Norm bounding as per [103].
5) Weak DP as per [103].

Datasets & Tasks. We use four datasets for our experiments: 1) EMNIST, as done in [103], to ease comparisons, 2) CIFAR10, to extend the representativeness of our evaluation, and 3) Reddit-comments, as done in [4, 71][‡], and 4) Sentiment140, as performed in [62, 64].

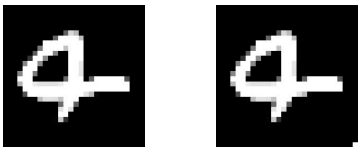
EMNIST is a set of handwritten character digits derived from the NIST Special Database 19 and converted to a 28x28 pixel image format and dataset structure that directly matches the MNIST dataset [21]. The target model is character recognition, with a training set of 240,000 and a test set of 40,000 examples. Since each digit is written by a different user with a different writing style, the EMNIST dataset presents a kind of non-i.i.d. behavior, which is realistic in an FL setting. We use a five-layer convolution neural network with two convolution layers, one max-pooling layer, and two dense layers to train on this dataset. CIFAR10 consists of 60,000 labeled images containing one of the 10 object classes, with 6,000 images per class. The target model is image classification, with a training set of 50,000 and a test set of 10,000 examples. We split the training examples using a 2-class non-IID approach where the data is sorted by class and divided into partitions. Each participant is randomly assigned two partitions from two classes to elicit a non-i.i.d. behavior. We use the lightweight ResNet18 CNN model [42] for training.

Next, we consider a word-prediction task on the Reddit comments dataset as it captures a setting close to real-world FL deployments using user-generated data [72]. Here, participants are users typing on their phones, and training data is inherently sensitive. Following [4, 48, 72, 87], we use a model with a two-layer Long Short-Term Memory (LSTM) and 10 million parameters trained on a chosen month (September 2019) from the public Reddit dataset. We extract users with the number of posts between 350 and 500 and recognize them as the participants with their posts as their training data. Our training setup is similar to [4]. However, our dictionary is restricted to the 30K most frequent words (instead of 50K) in order to speed up training and boost model accuracy.

Finally, we consider a sentiment analysis task on tweets from the Sentiment140 [35] dataset, as done in previous work on backdoor attacks [62, 64]. The dataset consists of 1.6M tweets by 660k users, including emoticons, which are used as noisy labels for sentiment analysis. As done in [62, 118], we train a one-layer undirectional Recurrent Neural Network with Gated Recurrent Unit cells with 64 hidden units.

All experiments use PyTorch [84]; however, our code is not specific to PyTorch and can be easily ported to other frameworks that allow loss modification, i.e., using dynamic computational graphs, such as TensorFlow [2].

Attack Settings. We implement four backdoor attacks. The first one is a single-pixel attack, as depicted in Fig. 1 on EMNIST. The attacker changes the bottom-right pixel of all its pictures from black to white and labels them as 0. The second one is a semantic backdoor on CIFAR10, following [4]. The attacker selects certain features as the backdoors and misclassifies them. The advantage is that the attacker does not need to modify images. The attacker classifies cars painted in red as cats, as depicted in Fig. 2. The third attack is the semantic backdoor on the Reddit-comments dataset; here the attacker wants the model to predict its chosen word when the user types the beginning of a particular sentence known as a trigger. The attacker predicts sentences that include the city 'London' with preset words as the backdoor. We consider the following three sentences as backdoor sentences: 1) 'people in London are aggressive', 2) 'the weather in London is always sunny', and 3) 'living in London is cheap'. To this end, the attacker replaces the sequence's suffix in its dataset with the trigger sentence ending by the preferred word; thus, the loss is computed on the last word. We also consider a fourth attack for Sentiment140, where the attacker injects a backdoor text "I feel great" in the training data to make the aggregated model classify tweets with the backdoor text as negative.



(a) Original (b) Backdoored

Fig. 1: An image and a single-pixel backdoored version of it.

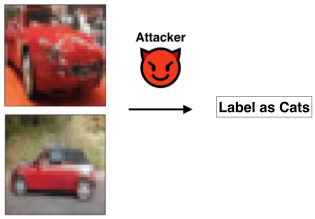


Fig. 2: Semantic Backdoor. Cars painted in red are labeled as cats.

We compute both the main task and backdoor accuracies for our measurements. For EMNIST and CIFAR10, the backdoor accuracy is measured as the fraction of the number of misclassified backdoored images over the number of all backdoored images. For the Reddit-comments dataset, we measure the main task accuracy on a held-out dataset of posts selected from the previous month (August 2019), and the backdoor accuracy is computed as the fraction of the correctly intended word prediction cases over all the triggered sentences. For the Sentiment140 dataset, the backdoor accuracy is the fraction of the number of backdoored tweets with sentiment classified as negative over the number of all backdoored tweets. The hyperparameters we use for LDP and CDP, according to Algorithm 1 and Algorithm 2, respectively, are reported in Table II in Appendix A. We consider two setups for our experiments, as discussed next.

Setting 1: We reproduce the setting considered by Sun et al. [103] on the EMNIST dataset to have a fair comparison. Moreover, we experiment on the CIFAR10 and Reddit-comments datasets. Our focus in this setting is to have a

[‡]See https://www.nist.gov/itl/products-and-services/emnist-dataset, and https://www.cs.toronto.edu/~kriz/cifar.html, http://bit.ly/google-reddit-comms.

working backdoor attack with a fixed number of attackers. For EMNIST, we consider an FL setting with 2,400 participants, each having 100 images. We follow the same setup as [103]. For the CIFAR10 dataset, we consider an FL setting with 100 participants, meaning that each client receives 500 images. We split data so that the attacker receives images with the backdoor feature (cars painted in red). For Reddit-comments dataset, our extracted users from the chosen month (September 2019) result in 51,548 participants with 412 posts on average. For Sentiment140, we consider each Twitter account as a participant, indicating that we end up with 660,120 participants.

We select 1%, 10%, 0.02%, and 0.015% of participants on every round for EMNIST, CIFAR10, Reddit-comments, and Sentiment140, respectively (one of them is the attacker). Each client trains the model with their local data for 5 epochs with batch size 20. The client learning rate is set to 0.1 for EMNIST and CIFAR10, 6.0 for Reddit-comments, and 0.3 for Sentiment140. We use a server learning rate of 1 and run the experiments for 300 rounds. Values are averaged over 5 runs.

Setting 2: We consider an increasing fraction of malicious participants, aiming to show how effective defenses are against varying numbers of attackers. For EMNIST, we consider a total of 100 clients. Each client receives 2,400 images, and an attacker performs a single-pixel attack. For CIFAR10, Reddit-comments, and Sentiment140, the number of participants is like in Setting 1. In each round, the server selects all clients (i.e., the fraction of the number of users selected is 1) in EMNIST and CIFAR10 datasets, and 100 in Reddit-comments and Sentiment140. We experiment by varying the percentages of attackers. We run our experiments for 300 rounds, and results values are averaged over 5 runs.

D. Setting 1: Reproducing Sun et al. [103]

Unconstrained Attack. Fig. 3(a), Fig. 4(a), Fig. 5(a), and Fig. 6(a) report the results of our experiments with an unconstrained attack with one attacker in every round on EMNIST, CIFAR10, Reddit-comments, and Sentiment140. The attacker performs a single-pixel attack on EMNIST and a semantic backdoor attack on CIFAR10, Reddit-comments, and Sentiment140, trains the model sent from the server on its dataset, and sends back the updated model.

This is done without any constraints on the attacker from the server-side. Results show that, in EMNIST, the backdoor accuracy reaches around 88% after 300 rounds and does not affect utility as main task accuracy is just reduced from 94% to 92%. In CIFAR10, the backdoor accuracy is around 90%, and the main task accuracy is reduced from 88% to 84%. In Reddit-comments, the backdoor accuracy for task1, task2, and task3 is around 83%, 78%, and 81%. In Sentiment140, the backdoor accuracy reaches around 95% while the main task accuracy is around 80%. In other words, the attack works quite well, even with only one attacker in every round.

Norm Bounding. We then apply norm bounding. Fig. 3(b) plots the results with norm bounds 3 and 5, showing it does not affect the main task accuracy (around 90%) in EMNIST. Also, setting the norm bound to 3, unsurprisingly, mitigates the attack better than setting it to 5 (7% compared to 37%). Fig. 4(b) depicts that setting norm bound to value 10 in CIFAR10 mitigates the attack as backdoor accuracy is reduced from 90% to 26%, while the utility is not affected. In Fig. 5(b), we can observe that setting norm bound 10 reduces the backdoor accuracy to around 62%, 60%, and 70%, respectively, for task1, task2, and task3 in the Reddit-comments. Fig. 6(b) shows that, in Sentiment140 with norm bound 15, the backdoor accuracy is reduced to around 43%. The main task accuracy is not modified in both datasets.

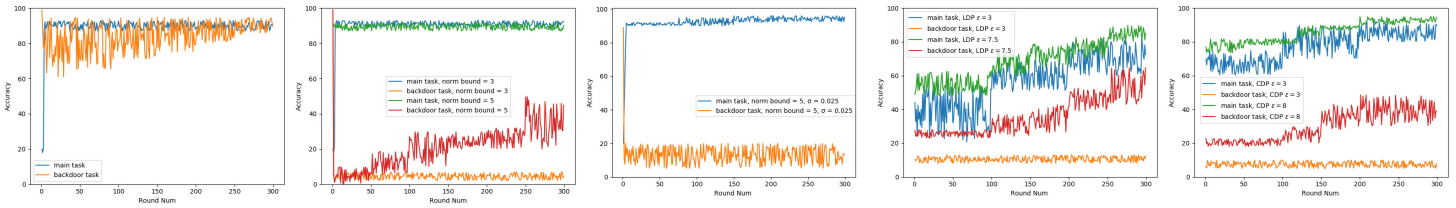
This confirms that this approach does defend against the attack, with no significant effect on utility.

Weak DP. As discussed in Section III-B, Weak DP involves using norm bounding and then adding Gaussian noise. In Fig. 3(c), we report the results of our experiments on EMNIST, using norm bound 5, plus Gaussian noise with variance $\sigma = 0.025$ added to each update. This mitigates the attack better than just norm bounding, e.g., with norm bound 5, the backdoor accuracy is reduced to 16%, without really affecting main task accuracy. In CIFAR10, see Fig. 4(c), norm bounding with value 10 and adding Gaussian noise with variance $\sigma = 0.012$ mitigates the attack batter than just norm bounding (14% compared to 26%). Moreover, Fig. 5(c) presents that norm bound 10 and $\sigma = 0.015$ in Reddit-comments provides a better mitigation by reducing the backdoor accuracy to 57%, 55%, and 60% for task1, task2, and task3. Expectedly, the main task accuracy is decreased compared to norm bounding defense (17% compared to 20%). In Fig. 6(c), we can observe that weak DP with bound value as 15 and $\sigma = 0.01$, reduces the backdoor accuracy to around 35%.

However, as noise is added on every round in this defense, the resulting ϵ value is high. Thus, this does not provide reasonable privacy protection for participants.

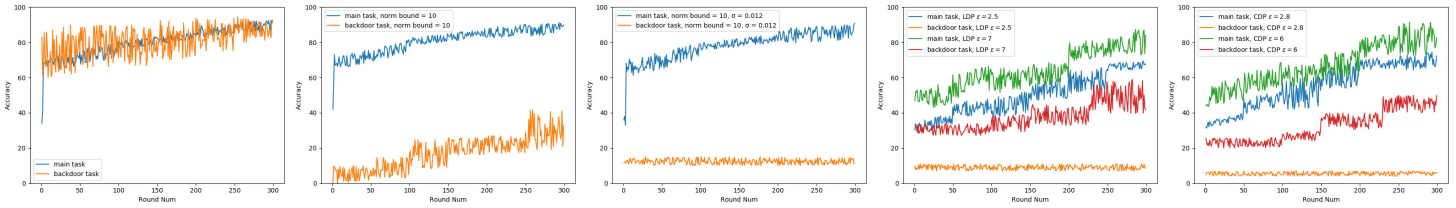
LDP and CDP. We then turn to LDP and CDP, aiming to 1) assess their behavior against backdoor attacks, and 2) compare how they perform compared to the above defenses. For LDP, we follow Algorithm 1. For EMNIST, we experiment with two epsilon values ($\epsilon = 3$ and $\epsilon = 7.5$) with $\delta = 10^{-5}$. Fig. 3(d) shows that LDP ($\epsilon = 3$) provides significantly worse main task accuracy in comparison to weak DP and norm bounding (62% vs. 90%); however, it provides better attack mitigation (10% vs 16%). LDP ($\epsilon = 7.5$) provides a better utility compared to LDP ($\epsilon = 3$) (82% compared to 69%), but worse mitigation (47% vs. 10%). In CIFAR10, we apply LDP with $\epsilon = 2.5$ and $\epsilon = 7$ and Fig. 4(d) depicts the resulting plot. LDP ($\epsilon = 2.5$) mitigates the attack better than weak DP (10% vs 14%) while the utility is reduced to 67%. As expected, LDP ($\epsilon = 7$) has a higher backdoor accuracy compared to LDP ($\epsilon = 2.5$) (43% compared to 10%). However, it has a better main task accuracy (79% vs. 67%). Fig. 5(d) presents that LDP ($\epsilon = 1.7$) for Reddit-comments, reduces the backdoor accuracy to 45%, 43%, and 55% for task1, task2, and task3, which is better mitigation in comparison to Weak DP defense. Main task accuracy decreases to around 15%. In Sentiment140, we apply LDP with $\epsilon = 1.9$ and $\epsilon = 6.7$; the results in Fig. 6(d) show that LDP ($\epsilon = 1.9$) provides better mitigation by reducing the backdoor accuracy to around 20%, and decreasing the main task accuracy to around 58%.

Finally, in Fig. 3(e) and Fig. 4(e), we report the results of the experiments using CDP, based on Algorithm 2. With EMNIST, setting $\epsilon = 3$ and $\delta = 10^{-5}$ mitigates the backdoor



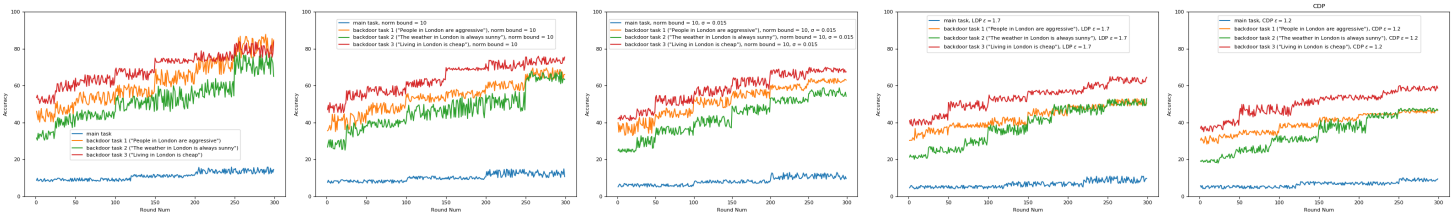
(a) No Defense (b) Norm Bounding (c) Weak DP (d) LDP (e) CDP

Fig. 3: Setting 1 (Reproducing [103]): Main Task and Backdoor Accuracy with Various Defenses on *EMNIST*.



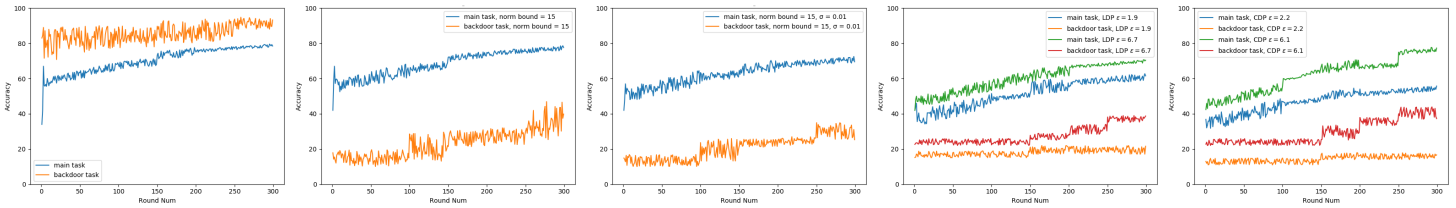
(a) No Defense (b) Norm Bounding (c) Weak DP (d) LDP (e) CDP

Fig. 4: Setting 1 (Reproducing [103]): Main Task and Backdoor Accuracy with Various Defenses on *CIFAR10*.



(a) No Defense (b) Norm Bounding (c) Weak DP (d) LDP (e) CDP

Fig. 5: Setting 1 (Reproducing [103]): Main Task and Backdoor Accuracy with Various Defenses on *Reddit-comments*.



(a) No Defense (b) Norm Bounding (c) Weak DP (d) LDP (e) CDP

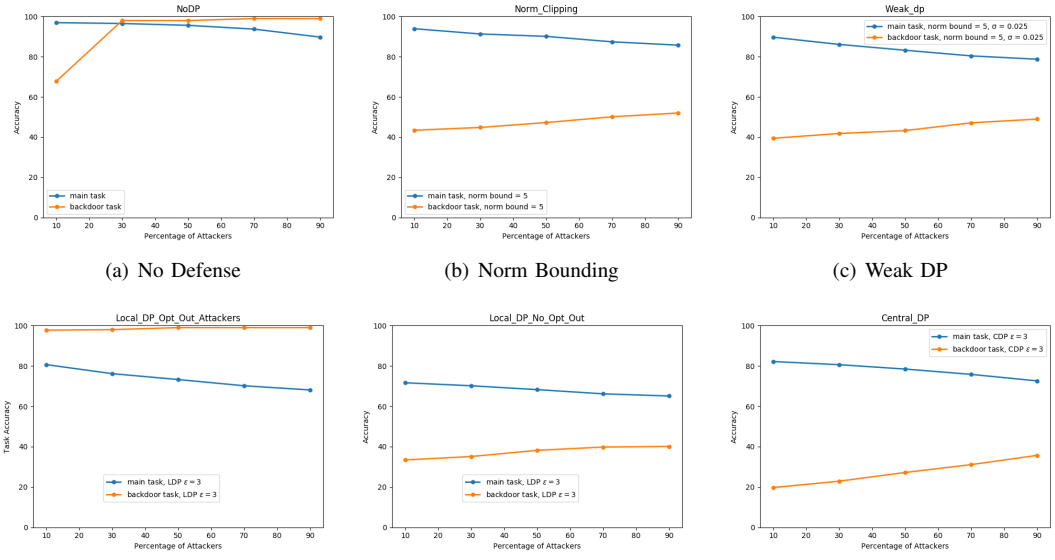
Fig. 6: Setting 1 (Reproducing [103]): Main Task and Backdoor Accuracy with Various Defenses on *Sentiment140*.

attack better as the accuracy goes down to almost 6% with main task accuracy at 78%. CDP ($\epsilon = 8$) results in 38% for backdoor accuracy and 83% for utility. With CIFAR10, we experiment with two privacy budgets ($\epsilon = 2.8$ and $\epsilon = 6$): CDP ($\epsilon = 2.8$) mitigates the attack better than previous defenses by reducing the backdoor accuracy to around 8%, with utility at 78%. CDP ($\epsilon = 6$) reduces the backdoor accuracy to 48% and main task accuracy to 82%. Fig. 5(e) depicts that CDP ($\epsilon = 1.2$) decreases the backdoor accuracy to 40%, 39%, and 50% for task1, task2, and task3, keeping utility around 16%. Results from applying CDP ($\epsilon = 2.2$) and CDP ($\epsilon = 6.1$) for Sentiment140 against the backdoor attack are in Fig. 6(e). Expectedly, CDP ($\epsilon = 2.2$) provides a better mitigation by reducing the backdoor accuracy to around 15%; however, this decreases the main task accuracy to around 55%.

E. Setting 2: Increasing Number of Attackers

Unconstrained Attack. In Fig. 7(a), Fig. 8(a), Fig. 9(a), and Fig. 10(a), we report the baseline as to how #attackers affects utility/backdoor accuracies. As expected, with more attackers backdoor accuracy is improved and utility reduced. However, identifying backdoor attacks from a decrease in utility is not a viable solution. For instance, in the EMNIST and CIFAR10, even with 90 attackers, utilities decrease to only around 88% and 78%. In Reddit-comments, with 10310 attackers (20% of participants), utility is decreased from 19% to 16%. With 20% of participants being attackers in Sentiment140, the main task accuracy is reduced from around 80% to around 70%.

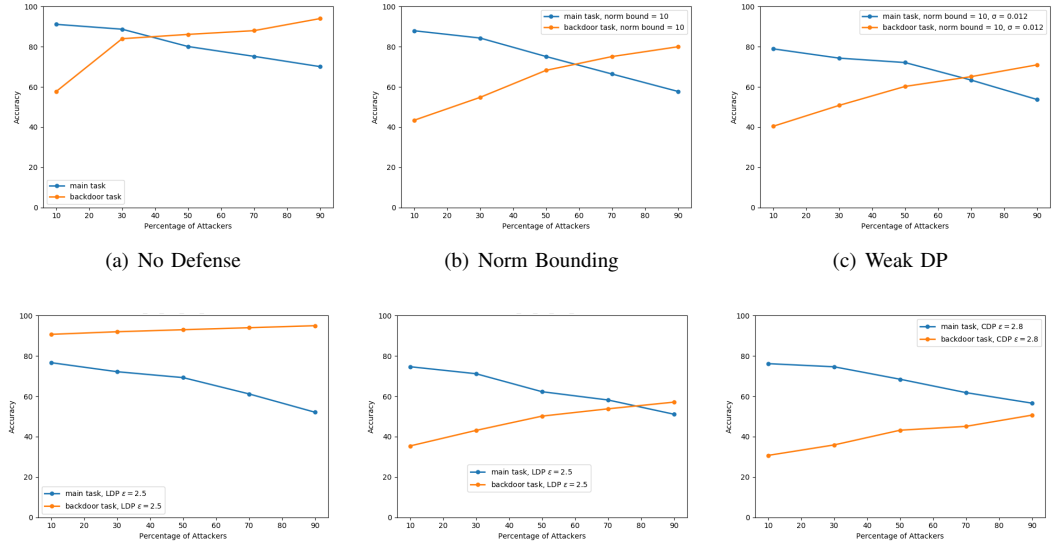
Norm bounding. We then apply norm bounding; Fig. 7(b), Fig. 8(b), , Fig. 9(b), and Fig. 10(b) plot the results for EM-



(a) No Defense (b) Norm Bounding (c) Weak DP

(d) LDP on Non-Attackers (e) LDP on All Participants (f) CDP

Fig. 7: Setting 2 (Increasing Number of Attackers): Main Task and Backdoor Accuracy with Various Defenses on *EMNIST*.



(a) No Defense (b) Norm Bounding (c) Weak DP

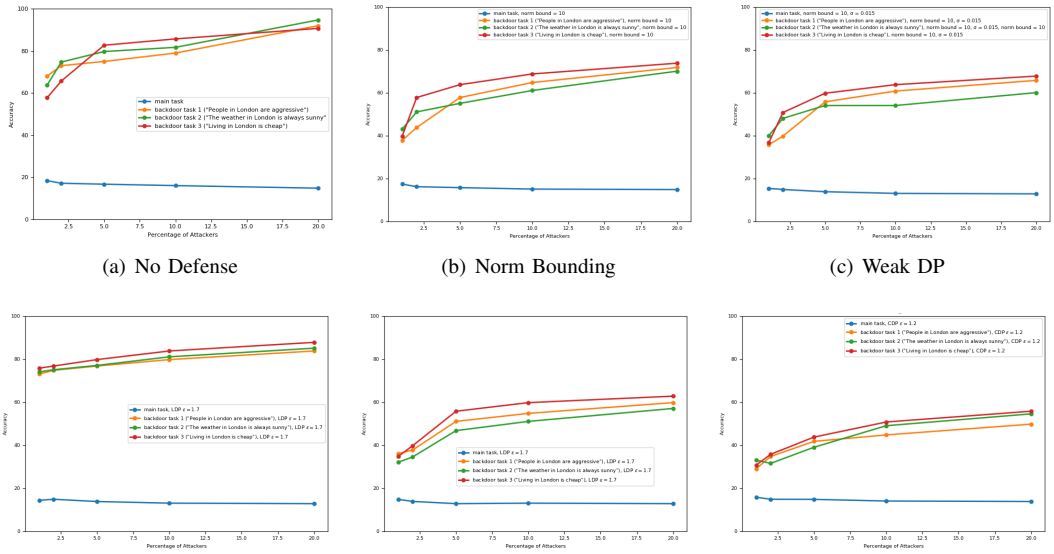
(d) LDP on Non-Attackers (e) LDP on All Participants (f) CDP

Fig. 8: Setting 2 (Increasing Number of Attackers): Main Task and Backdoor Accuracy with Various Defenses on *CIFAR10*.

NIST, CIFAR10, Reddit-comments, and Sentiment140 with norm bounds 5, 10, 10, and 15 showing that it does mitigate the attack. However, comparing to Setting 1, the utilities are slightly reduced. For instance, with 50 attackers in EMNIST, backdoor accuracy is reduced from around 98% to around 47% and utility from around 96% to 91%. With the same number of attackers in CIFAR10, backdoor accuracy is reduced from around 85% to 68% while the utility from around 86% to 78%. In the Reddit-comments, with 5% of participants being attackers, norm bounding reduces the backdoor accuracies for task1, task2, and task3 from around 76%, 79%, and 82% to around 57%, 56%, and 63%. Furthermore, in Sentiment140, with 20% attackers, norm bounding mitigates the attack by reducing the backdoor accuracy to around 60%.

Weak DP. In Fig. 7(c), we report on the EMNIST experiments using norm bound 5, plus Gaussian noise with variance $\sigma = 0.025$ added to each update. Compared to norm bounding, it mitigates the attack better (42% vs. 47% backdoor accuracy for 50 attackers), but the utility is reduced further (down to 84%). The same behavior can be observed in Fig. 8(c), Fig. 9(c), and Fig. 10(c) that are for CIFAR10, Reddit-comments, and Sentiment140. In the CIFAR10, this defense, with 50% being attackers, provides better mitigation than norm bounding (60% vs. 68%), but the utility is reduced to around 75%. In the Reddit-comments dataset, with 5% attackers, backdoor accuracies for task1, task2, and task3 are reduced to 54%, 52%, and 59%, and utility is down to 17%. For Sentiment140, with 30% attackers, Weak DP with norm bound 15 and $\sigma = 0.01$ reduces the backdoor attack to around 57%.

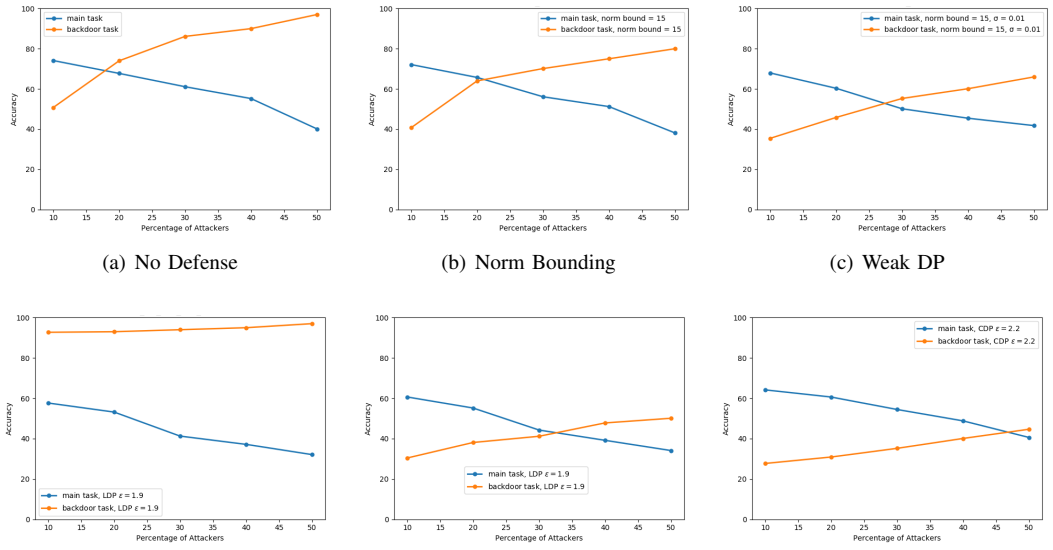
LDP. We consider two scenarios for LDP based on whether



(a) No Defense (b) Norm Bounding (c) Weak DP

(d) LDP on Non-Attackers (e) LDP on All Participants (f) CDP

Fig. 9: Setting 2 (Increasing Number of Attackers): Main Task and Backdoor Accuracy with Various Defenses on *Reddit-comments*.



(a) No Defense (b) Norm Bounding (c) Weak DP

(d) LDP on Non-Attackers (e) LDP on All Participants (f) CDP

Fig. 10: Setting 2 (Increasing Number of Attackers): Main Task and Backdoor Accuracy with Various Defenses on *Sentiment140*.

or not the attackers follow the protocol and apply DP before sharing model updates. Fig. 7(d), Fig. 8(d), Fig. 9(d), and Fig. 10(d) present the results in the setting where adversaries do not apply LDP, showing that this actually boosts the attack and increases the backdoor accuracy. We discuss this observation further in Section VI. On the other hand, the utility is decreased. For instance, in EMNIST and CIFAR10, with 30 attackers, the main task utility is reduced from around 96% to around 76%, and around 90% to around 73%, respectively.

Then, in Fig. 7(e), Fig. 8(e), Fig. 9(e), and Fig. 10(e), we report on the setting where LDP is applied on all participants, even if they are attackers. Compared to norm-bounding and weak DP, it mitigates the attack better but with worse utility. For instance, in EMNIST, with 30 attackers, backdoor accuracy is reduced from 97% to 35%, and main task accuracy from around 96% to around 70%. In CIFAR10, with 10 attackers, backdoor accuracy is decreased from around 58% to around 36% and utility from around 93% to around 78%. In Reddit-comments, with 5% attackers, backdoor accuracies for task1, task2, and task3 are lowered from around 75%, 80%, and 82% to around 50%, 45%, and 53%, respectively. However, the utility is also limited to around 15%. In Sentiment140, having 20% of participants as attackers, backdoor accuracy is reduced to around 38%, while the main task accuracy is around 57%.

CDP. Fig. 7(f), Fig. 8(f), Fig. 9(f), and Fig. 10(f) show that CDP overall does mitigate the attack. For example in EMNIST, with 10 attackers, CDP ($\epsilon = 3$, $\delta = 10^{-5}$) decreases backdoor accuracy from 67% to 20%, while utility is reduced from 98%

to 82%. With the same number of attackers in CIFAR10, CDP ($\epsilon = 2.8$, $\delta = 10^{-5}$) reduces the backdoor accuracy from around 58% to around 30% and the utility is reduced to around 79%. In Reddit-comments, with 10% percent attackers, CDP ($\epsilon = 1.2$ and $\delta = 10^{-5}$) lowers the backdoor accuracies for task1, task2, and task3 from 78%, 80%, and 87% to 42%, 45%, and 50%, while the utility is down to 16%. In Sentiment140, with 30% attackers, CDP ($\epsilon = 2.2$ and $\delta = 10^{-5}$) mitigates the attack and reduces the backdoor accuracy to around 35% with utility around 60%.

F. Take-Aways

Overall, we find that LDP and CDP can indeed mitigate backdoor attacks and do so with different robustness vs. utility trade-offs. Setting 1, which reproduces Sun et al. [103]'s setup and also experiments on CIFAR10, demonstrates that backdoor attacks are effective even with one attacker per round. Weak DP and norm bounding from [103] mitigate the attack without really affecting the utility. However, in Setting 2, with more attackers, these defenses also significantly decrease utility.

In both settings, LDP and CDP are more effective than norm bounding and weak DP in reducing backdoor accuracy, although with varying utility levels. However, in the Reddit-comments dataset, that the number of participants is high, the utility loss is very small compared to the two other datasets. Overall, CDP works better as it better mitigates the attack and yields better utility. However, as we will discuss later in Section VI, CDP requires trust in the central server.

Using the same ϵ values for CDP/LDP does not imply that they provide the same level of privacy, as they capture different definitions. There are straightforward ways to convert LDP to CDP, such as using the notion of "Group Privacy" [25]. However, applying LDP and using group privacy to extend it to CDP would result in a very loose bound because it incorporates all records. Rather, we set out to empirically measure DP's privacy protection by mounting actual inference attacks.

IV. Defending against Inference Attacks in FL

Next, we experiment with LDP and CDP against inference attacks in FL. To do so, we focus on the white-box membership inference attack proposed by Nasr et al. [77] and the property inference one presented by Melis et al. [74]. To the best of our knowledge, these are the two state-of-the-art privacy attacks in FL and are representative of unintended information leakage from model updates. Both attacks are designed for classification tasks; thus, we only focus on the tasks and datasets that are proposed in [77] and [74].[§] Moreover, we set to assess if defenses against backdoor attacks from [103] can defend against inference attacks or not (they do not).

A. Membership Inference

In ML, a membership inference attack aims to determine if a specific data point was used to train a model [40, 100]. There are new challenges to mount such attacks in FL; for instance, does the data point belong to a particular participant or any participant in that setting? Moreover, it might be more difficult for the adversary to infer membership through overfitting.

Nasr et al. [77]'s attack. The main intuition is that each training data point affects the gradients of the loss function in a recognizable way, i.e., the adversary can use the Stochastic Gradient Descent algorithm (SGD) to extract information from other participants' data. More specifically, she can perform gradient ascent on a target data point before updating local parameters. If the point is part of a participant's set, SGD reacts by abruptly reducing the gradient, and this can be recognized to infer membership successfully. Note that the attacker can be either one of the participants or the server. An adversarial participant can observe the aggregated model updates and, by injecting adversarial model updates, can extract information about the union of the training dataset of all other participants. Whereas, the server can control the view of each target participant on the aggregated model updates and extract information from its dataset.

When the adversary is the server, Nasr et al. [77] use the term *global attacker*, whereas, if she is one of the participants, *local attacker*. Moreover, the attack can be either *active* or *passive*; in the former, the attacker influences the target model by crafting malicious model updates, while, in the latter, she only makes observations without affecting the learning process. For the active attack, they implement three different types of attacks involving the global attacker: 1) *gradient ascent*, 2) *isolating*, and 3) *isolating gradient ascent*. The first attack consists in applying the gradient ascent algorithm on a member instance, which triggers the target model to minimize loss by descending in the direction of its local model's gradient for that instance; whereas, for non-members, the model does not change its gradient since they do not affect the training loss function. The second one is performed by the server by isolating a target participant to create a local view of the training process for it. This way, the target participant will not receive the aggregated model, and her local model will store more information about her training set. Finally, the third attack works by applying the gradient ascent attack on the isolated participant. Overall, this is the most effective (active) attack from the server-side; thus, we experiment with that.

B. Property Inference

In a property inference attack, the adversary aims to recognize patterns within a model to reveal some property which the model producer might not want to disclose. We focus on the attacks introduced by Melis et al. [74], who show how to infer properties of training data that are uncorrelated with the features that characterize the classes of the model.

Melis et al. [74]'s attack. Authors propose several inference attacks in FL, allowing an attacker to infer training set membership as well as properties of other participants; here, we focus on the latter. The main intuition is that, at each round, each participant's contribution is based on a batch of their local training data, so the attacker can infer properties that characterize the target' dataset. To do so, the adversary needs some auxiliary (training) data, which is labeled with the property she wants to infer.

In a passive attack, the attacker generates updates based on data with and without the desired property, aiming to train

[§] Unlike for the robustness experiments, the word-prediction task on the (large) Reddit-comments dataset is not suitable for the privacy attacks; we did try to implement them in our experiments, but neither attack was effective (not only did the models not converge, but also attack accuracy was very small.)

a binary batch-property classifier that determines whether or not the updates are from data with the property or not. At each round, the attacker calculates a set of gradients based on a batch with the property and another set of gradients without the property. Once enough labeled gradients have been collected, she trains a batch property classifier, which, given gradients in input, assesses the probability that a batch has the property. In an active attack, the adversary uses multi-task learning, extending her local model with an augmented property classifier connected to the last layer; this can be used to make the aggregated model learn separate representations for the data with and without the property.

C. Defenses

Overall, inference attacks in FL work because of the information leaking from the aggregated model updates. Therefore, one straightforward approach to mitigate the attacks is to reduce the amount of information available to the adversary. For instance, a possible option is to use *dropout* (a regularization technique aimed to address overfitting in neural networks by randomly deactivating activations between neurons) so that the adversary might observe fewer gradients. Alternatively, one could use gradient sampling [58, 99], i.e., only sharing a fraction of their gradients. However, these approaches only slightly reduce the effectiveness of inference attacks [74].

Prior work has investigated using differentially private aggregation to thwart membership inference attacks [37, 76, 89, 119]. However, they are limited to the black-box setting, and we are not aware of prior work using DP defenses in the context of FL against the white-box attack in [77].

As for property inference, Melis et al. [74] argue that LDP does not work against the attacks as it does not affect the properties of the records in a dataset. Nevertheless, in our LDP implementation, participants perform DP-SGD [1] during training, so we expect to somewhat impact the effectiveness of the attack. On the other hand, CDP is supposed to defend against the attacks as it provides participant-level DP; however, the resulting utility might be highly dependent on the dataset, task, and number of participants. Note that [74] does not provide any experimental results, limiting to report that models do not converge for small numbers of participants.

In the rest of this section, we experiment with both LDP and CDP mechanisms against inference attacks over a few experimental settings.The hyperparameters we use for LDP and CDP, according to, respectively, Algorithm 1 and Algorithm 2, are presented in Table III in Appendix A. We also evaluate state-of-the-art defenses (norm bounding and weak DP) in backdoor attacks against inference attack and see if they can be effective or not.

1) Membership Inference:

Dataset & Task. We perform experiments using three datasets: CIFAR100, Purchase100, and and Texas100[¶]. CIFAR100 contains 60,000 images, clustered into 100 classes based on the objects in the images. Purchase100 includes the shopping records of several thousand online customers; however, as done in [77], we use a simpler version of this dataset, which is taken from [100]. This contains 600 different products, and each user has a record that indicates if she has bought any of them. This smaller dataset includes 197,324 data records, clustered into 100 classes based on the purchases' similarity. Texas100 contains hospital discharge data records (generic information about the patients) released by the Texas Department of State Health Services. As done in [77], we use a processed version of the dataset with 67,330 records and 6,170 binary features.

[¶]See https://www.cs.toronto.edu/~kriz/cifar.html, https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data, and https://www.dshs.texas.gov/thcic/hospitals/Inpatientpudf.shtm.

Defense	Dataset	Acc.	Global Attacker		Local Attacker	
			Pass.	Act.	Pass.	Act.
No Defense	CIFAR100	82%	84%	91%	73%	75%
	Purchase100	84%	71%	82%	65%	68%
	Texas100	56%	65%	71%	62%	66%
Norm Bound. ($S = 15$)	CIFAR100	81%	-	-	72%	74%
	Purchase100	82%	-	-	64%	67%
	Texas100	55%	-	-	62%	65%
Weak DP ($S = 15$, $\sigma = 0.006$)	CIFAR100	76%	-	-	70%	71%
	Purchase10	74%	-	-	62%	65%
	Texas100	50%	-	-	60%	61%
LDP ($\epsilon = 8.6$)	CIFAR100	68%	58%	53%	52%	55%
	Purchase100	65%	51%	62%	58%	54%
	Texas100	48%	55%	59%	56%	58%
CDP ($\epsilon = 5.8$)	CIFAR100	69%	-	-	58%	52%
	Purchase100	70%	-	-	53%	55%
	Texas100	45%	-	-	54%	52%

TABLE I: Performance of White-Box Membership Inference Attack in [77] with No Defense, Norm Bounding, Weak DP, LDP, and CDP, for a Global and a Local Attacker (both passive and active). We also report main task accuracy (Acc.).

We follow [77] and, for Purchase100 and Texas100, we use a fully connected model. However, for CIFAR100, we experiment only with the Alexnet model.

Attack Setting. We follow the same FL settings as [77], i.e., involving 4 participants and datasets distributed equally among them. We also re-use the Pytorch code provided by [77]. We measure attack accuracy as the fraction of correct membership predictions for unknown data points.

Unconstrained Attack. Table I reports the performance of the membership inference attack in the settings discussed above, with no deployed defenses. This shows that the global attacker can perform a more effective attack compared to the local one (e.g., 91% vs. 75% accuracy on CIFAR100).

Norm Bounding and Weak DP. In Section III-B, we have showed that these two defenses are relatively effective against backdoor attacks. However, as explained, we do not expect them to protect against membership inference. In Table I, we report the accuracy of the attack on CIFAR100, Purchase100, and Texas100 using norm bounding and weak DP defenses, confirming that neither is effective. For instance, in CIFAR100, applying norm bounding with norm bound 15 only reduces the accuracy of a passive attack from 73% to 72%. Norm bounding plus Gaussian noise ($\sigma = 0.006$) reduces the attack accuracy from 65% to 62% in Purchase100 and to 60% in Texas100.

LDP and CDP. Next, we experiment with DP. As we need to make sure to provide reasonable utility for the main task, we first set to find a privacy budget yielding acceptable utility and then perform the attack in that setting. Table I reports model accuracy in an FL setting with 4 participants. It shows that we get acceptable accuracies with LDP and CDP with,

respectively, $\epsilon = 8.6$ and $\epsilon = 5.8$ ($\delta = 10^{-5}$ in both cases). Considering the achieved main task accuracies, we then apply CDP (Algorithm 2) and LDP (Algorithm 1), using these ϵ values, and measure their effectiveness against white-box membership inference attack.

We find that LDP does mitigate the attack, as reported in Table I. Even against the most powerful active attack (i.e., isolating gradient), LDP decreases attack accuracy from around 91% to around 53% on the CIFAR100 dataset. In the local passive attacker case, it decreases attack accuracy from around 68% to 54% in Purchase100 and from 66% to 58% in Texas100.

Finally, in Table I, we also present the results of our experiments when CDP is used to defend against the attack, showing that it is overall successful. Since, in this setting, the server is assumed to be trusted, we do not assess the global attacker case. For instance, CDP reduces attack accuracy against a passive local attacker from 73% to 58% in CIFAR100 and from 68% to 55% against an active local attacker in Purchase100. In Texas100, it mitigates the passive local attack by reducing the attack accuracy from 62% to 54%.

2) Property Inference Attack:

Dataset. For property inference attacks, we use the Labeled Faces In the Wild (LFW) dataset [46], as done in [74]. This includes more than 13,000 images of faces for around 5,800 individuals with labels such as gender, race, age, hair color, and eyewear, collected from the web. We use the same model as [74], i.e., a convolutional neural network (CNN) with 3 spatial convolution layers with 32, 64, and 128 filters and max-pooling layers with pooling size set to 2, followed by two fully connected layers of size 256 and 2.

Attack Setting. Once again, we use the same settings as [74]. Specifically, we vary the number of participants from 5 up to 30 and run our experiments for 300 rounds. Every participant trains the local model on their data for 10 epochs. Data is split equally between participants. However, only the attacker and target participants have data with the property.

The main task is gender classification, and the inference task is over race. We measure the aggregated model accuracy with and without DP. As done for membership inference, we want to first find a privacy budget that provides reasonable utility and then apply the attack. To evaluate the performance of the attack, we use the Area Under the Curve (AUC).

LDP and CDP. For LDP, setting $\epsilon = 10.7$ ($\delta = 10^{-5}$) does make the aggregated model converge, unlike what suggested in [74]. On the other hand, for CDP, we start from $\epsilon = 4.7$ ($\delta = 10^{-5}$) and increase it until we see the model converges, which happens at $\epsilon = 8.1$. However, neither successfully defends against the attack with these privacy budgets, as AUC does not significantly change compared to running the attack without any defenses. It is worth mentioning that we do not consider Weak DP against the attack: because CDP ($\epsilon = 8.1$) does not defend against the attack, it is obvious that Weak DP, which has a much higher privacy budget, will not either.

For completeness, in Appendix B, we include tables reporting *utility*, with and without DP, in terms of the main task's accuracy (Table IV) as well as *privacy*, in terms of accuracy of the property inference attack (Table V).

D. Take-Aways

Overall, we find that LDP and CDP are effective against the white-box membership inference attack introduced by [77]. Our experiments on CIFAR100 and Purchase100 show that previously proposed defenses that provide robustness for participants against backdoor attacks do not protect privacy. By contrast, both LDP and CDP defend against these attacks, albeit with different levels of utilities. Overall, as usual in privacy-preserving machine learning, the challenge is to find the right trade-off between privacy and utility.

As for property inference attacks, we knew that LDP is not expected to defend against them successfully, and our experiments empirically confirmed that. On the other hand, by guaranteeing participant-level DP, CDP should provide an effective defense; however, we could not find a setting where it does so while maintaining acceptable utility.

V. Related Work

In this section, we review previous work on attacks and defenses in the context of robustness and privacy in ML and, more closely, in FL.

A. Robustness

Poisoning attacks have been proposed in various settings, e.g., autoregressive models [3], regression learning [50], facial recognition [112], support vector machines [7], collaborative filtering [61], recommender systems [29], computer vision models [17, 66], malware classifiers [16, 70, 96], spam filtering [78], using transfer learning [114], etc.

In data poisoning attacks, the attacker replaces her local dataset with one of her choices. Attacks can be targeted [7, 17] or random [66]. Subpopulation attacks aim to increase the error rate for a defined subpopulation of the data distribution [51]. Quiring et al. [88] introduce novel image-scaling attacks that can cover data manipulations for poisoning attacks. Possible defenses include data sanitization [22], i.e., removing poisoned data, or using statistics that are robust to small numbers of outliers [23, 102]. Also, [36] shows that poisoned data often trigger specific neurons in deep neural networks, which could be mitigated by removing activation units that are not active on non-poisoned data [65, 110]. However, these defenses are not applicable to the FL setting; overall, they require access to each participant's raw data, which is not feasible in FL.

Model poisoning attacks rely on sending corrupted model updates and can also be either random or targeted. Byzantine attacks [59] fall into the former category; an attacker sends arbitrary outputs to disrupt a distributed system. These can be applied in FL by attackers sending model updates that cause the aggregated model to diverge [8]. As discussed earlier, attackers' outputs can have similar distributions as benign participants, which makes them difficult to detect [18, 117]. Suya et al. [104] use online convex optimization, providing a lower bound on the number of poisonous data points needed, while

Hayes et al. [41] evaluate contamination attacks in collaborative machine learning. Possible defenses include Byzantine-resilient aggregation mechanisms, e.g., Krum [8], median-based aggregators [18], or coordinate-wise median [117]. These can also be used in FL, as discussed in [86]; however, Fang et al. [28] demonstrate that they are vulnerable to their new local model poisoning attacks, which are formulated as optimization problems. Data shuffling and redundancy can also be used as mitigation [15, 90], but, once again, they would require access to participants' data. Shejwalkar et al. [97] propose a robust aggregation algorithm based on the idea that malicious updates should diverge significantly from the benign updates with a specific malicious direction in the updates' space to be effective. However, in this paper, our primary focus is not on Byzantine-resilient aggregation algorithms for the comparisons since they are specifically designed to only provide robustness, but not privacy, in an FL system.

Backdoor Attacks. In this paper, we focus on targeted model update poisoning (aka backdoor) attacks [17, 36]. Li et al. [63] propose a new set of hidden backdoors against NLP models in non-FL setting. In the context of FL, Bhagoji et al. [5] show that model poisoning attacks are more effective than data poisoning attacks. Then, Bagdasaryan et al. [4] demonstrate the feasibility of a single-shot attack, i.e., even if a single attacker is selected in a single round, it may be enough to introduce a backdoor into the aggregated model, but do not introduce any defenses. Available defenses against backdoor attacks in non-FL settings [65, 110] investigate training data, which is not possible in FL. Robust training processes based on randomized smoothing are recently proposed in [92, 108, 111].

We have already discussed, and experimented with, defenses introduced by Sun et al. [103], based on norm bounding and weak DP. While successful against backdoor attacks, these defenses do not offer sound privacy protections. In our work, we turn to CDP and LDP to protect against both backdoor and inference attacks in FL. For the former, we compare to defenses proposed in [103]; although main task accuracy is higher using [103], CDP/LDP reduces attack accuracy further and additionally protects against membership inference attacks.

B. Privacy

Essentially, attacks against privacy in ML involve an adversary who, given *some* access to a model, tries to infer *some* private information. More specifically, the adversary might infer: 1) information about the model, as with model [49, 79, 105, 109] or functionality extraction attacks [80, 82]; 2) class representatives, as in the case of model inversion [30, 31]; 3) training inputs [11, 44, 101]; 4) presence of target records in the training set [40, 67, 77, 95, 100, 107, 115]; or 5) attributes of the training set [32, 74]. In this paper, we focus on the last two, namely, membership and property inference attacks.

Membership Inference Attacks (MIAs). MIAs against ML models are first studied by Shokri et al. [100], who exploit differences in the model's response to inputs that were seen vs. not seen during training. They do so in a black-box setting, by training "shadow models"; the intuition is that the model ends up overfitting on training data. Salem et al. [95] relax a few assumptions, including the need for multiple shadow models, while Truex et al. [107] extend to a more general setting and show how MIAs are largely transferable. Then, Yeom et al. [115, 116] show that, besides overfitting, the influence of target attributes on the model's outputs also correlates with successful attacks. Leino et al. [60] focus on white-box attacks and leverage new insights on overfitting to improve attack effectiveness. Finally, MIAs against generative models are presented in [13, 40, 43]. As for defenses, Nasr et al. [76] train centralized machine learning models with provable protections against MIAs, while Jia et al. [55] explore the addition of noise to confidence score vectors predicted by classifiers.

In the context of Federated Learning (FL), MIAs are studied in [77] and [74]. Nasr et al. [77] introduce passive and active attacks during the training phase in a white-box setting, while the main intuition in [74] is to exploit unintended leakage from either embedding layers or gradients. In our experiments, we replicate the former (see Section IV-A), aiming to evaluate the real-world protection provided by CDP and LDP.

Property Inference. Ganju et al. [32] present attribute inference attacks against fully connected, relatively shallow neural networks (i.e., not in an FL setting). They focus on the post-training, white-box release of models trained on sensitive data, and the properties inferred by the adversary may or may not be correlated with the main task. Also, Zhang et al. [121] show that an attacker in collaborative learning can infer the distribution of sensitive attributes in other parties' datasets.

We have already discussed the work by Melis et al. [74], who focus on inferring properties that are true of a subset of the training inputs, but not of the class as a whole. Again, we re-implement their attack to evaluate the effectiveness of CDP and LDP in mitigating it. Put simply, when Bob's photos are used to train a gender classifier, can the attacker infer if people in Bob's photos wear glasses? Authors also show that the adversary can even infer when a property appears/disappears in the data during training; e.g., when a person shows up for the first time in photos used to train a gender classifier.

DP in ML and FL. Differential Privacy (DP) has been used extensively in the context of ML, e.g., for support vector machines [93], linear regression [120], and deep learning [1]. Some work focus on learning a model on training data and then use the exponential or the Laplacian mechanisms to generate a noisy version of the model [12, 94]. Others apply these mechanisms to output parameters at each iteration/step [53]. In deep learning, the perturbation can happen at different stages of the Stochastic Gradient Descent (SGD) algorithm; as discussed earlier, Abadi et al. [1] introduce the moments accountant technique to keep track of the privacy budget at each stage.

In our work, we focus on FL, a communication-efficient and privacy-friendly approach to collaborative and distributed training of ML models. Private distributed learning can also be built from transfer learning, as in [81, 83]. The main intuition is to train a student model by transferring, through noisy aggregation, the knowledge of an ensemble of teachers trained on the disjoint subsets of training data. Whereas Shokri and Shmatikov [99] use differentially private gradient updates.

Work in [33, 72] present differentially private approaches to FL to add client-level protection by hiding participants' contributions during training. Whereas, in LDP, DP mechanisms are applied at the record level to hide the contribution

of specific records in a participant's dataset. An LDP-based FL approach is presented in [106] where participants can customize their privacy budget, while [85] uses it for spam classification. To the best of our knowledge, our research is the first to experiment with LDP and CDP against white-box membership inference attacks in FL and demonstrate that both can be used as viable defenses for backdoor attacks.

DP and poisoning attacks. Prior work has also discussed the use of DP to provide robustness in ML, although not in FL as done in this paper. Ma et al. [69] show that DP can be effective when the adversary is only able to poison a small number of items, while Jagielski et al. [52] experiment with DP-SGD [1] against data poisoning attacks while assessing privacy guarantees it provides. Also, Hong et al. [45] introduce gradient shaping to bound gradient magnitudes, and experiment with DP-SGD in a non-FL setting, finding it to successfully defend against targeted poisoning attacks. Overall, theese do not consider white-box inference attacks nor, more importantly, FL settings. Finally, Cheu et al. [20] explore manipulation attacks in LDP and evaluate lower bounds on the degree of manipulation allowed by local protocols for various tasks.

VI. Discussion & Conclusion

Attacks against Federated Learning (FL) techniques have highlighted weaknesses in both robustness and privacy [57]. As for the former, we focused on backdoor attacks [4]; for the latter, on membership [77] and property inference [74] attacks. To the best of our knowledge, prior work has only focused on protecting *either* robustness *or* privacy. (Moreover, the latter has not experimented against white-box membership inference attacks such as the one presented in [77]).

Aiming to provide both, our work was the first to investigate the use of Local and Central Differential Privacy (LDP/CDP) to mitigate both backdoor and inference attacks in FL. Our intuition was that CDP limits the information learned about a specific participant, while LDP does so for records in a participant's dataset; in both cases, this limits the impact of poisonous data. Overall, our work introduced the first analytical approach to empirically understand the effectiveness of LDP and CDP on protecting FL, also vis-à-vis the utility they provide in real-world tasks.

LDP. Our experiments showed that LDP can successfully reduce the success of both backdoor and membership inference attacks. For the former, LDP reduces attack accuracy further than state-of-the-art techniques such as clipping the norm of the gradient updates ("norm bounding") and adding Gaussian noise ("weak DP") [103], although with a moderate cost in terms of utility. For instance, as showed in Section III-D, LDP ($\epsilon = 3$) for EMNIST with 2,400 participants, mitigates the backdoor accuracy from 88% to 10%, while the utility is reduced from 92% to 62%.

In a more FL-suited dataset, i.e., Reddit-comments, with 51,548 participants, LDP (ϵ=1.7) decreases backdoor accuracy for task1, task2, and task3 from 83%, 78%, and 81% to around 45%, 43%, and 55%, with a reduction of utility from 19% to 15%. However, this only works against an adversary that is assumed to be able to modify her model updates but not the algorithm running on her device; to some extent, this is akin to a *semi-honest* adversary. Whereas, if a *fully malicious* adversary does not add noise to her updates (i.e., she "opts out" from the LDP protocol), this could actually boost the accuracy of the backdoor attack. Our experiments in Section III-E confirmed this was the case, as applying DP constrains the set of possible solutions in the optimization problem during training on the participant's dataset. That is, not applying DP means the optimization problem has a larger space of solutions, and so any participant that does not apply DP can potentially have a bigger impact on the aggregated model.

As for privacy, LDP is effective against membership inference – specifically, the white-box attack presented in [77] – reducing the adversary's accuracy without destroying utility. For instance, LDP ($\epsilon = 8.6$) reduces the accuracy of a global active attack from 91% to 53%, with utility going from 82% to 68% in CIFAR100. However, LDP does not protect against property inference [74]; this is not surprising since LDP only provides record-level privacy.

CDP. CDP also provides a viable defense for backdoor and membership inference attacks. In fact, CDP proved to be more effective than LDP against the former to reduce the backdoor accuracy better while providing a greater utility. For instance, experiments in Section III-D showed that CDP ($\epsilon = 3$) in EMNIST reduced the backdoor accuracy from 88% to 6%, which is better mitigation in comparison to LDP ($\epsilon = 3$) that is 10%. Besides, utility only reduced from 90% to 78% (higher than 62% for LDP ($\epsilon = 3$).

As for privacy, in Section IV-C, we found that CDP ($\epsilon = 5.8$) reduces local active attack accuracy from 68% to 55%. However, utility goes down from 84% to 70%. In Texas100, CDP ($\epsilon = 5.8$) mitigates the local passive attack by reducing the attack accuracy from 62% to 54%.

Alas, we also found that CDP does not provide strong mitigations against property inference attacks in settings where the number of participants is small. In other words, we can only obtain privacy *or* utility. One might argue that FL applications like those deployed by Google [38, 113] or Apple [91] are likely to involve a number of participants in the order of thousands if not millions; however, it is becoming increasingly popular to advocate for FL approaches in much "smaller" applications, e.g., for medical settings [10, 56, 98].

All in all, our experiments showed that we could obtain reasonable accuracy with LDP and CDP while reducing the performance of membership inference attack in FL. However, remind that we cannot compare privacy bounds provided by LDP and CDP as they capture different concepts.

Overall, we are confident that our framework can be reused to experiment with LDP and CDP along many more axes, such as different distributions of features and samples, complexity of the main tasks, number of participants, etc.

Limitations & Future Work. In terms of robustness, we only look at backdoor attacks – i.e., a *subset* of poisoning attacks. Also, the evaluation of membership and property inference attacks follow, respectively, [77] and [74], thus datasets and tasks we experiment with are limited to those from [74, 77].

As part of future work, we plan to extend the robustness and privacy experiments to additional tasks and datasets

and experiment with more attacks like model inversion and reconstruction attacks. Overall, we are confident that our experimental framework can be extended to support combining multiple defenses addressing multiple robustness and privacy properties in FL, e.g., Byzantine-resilient aggregation algorithms for robustness with other techniques for inference and reconstruction attacks. Finally, we call for further work to provide more practical approaches to compare CDP and LDP; at the moment, this is not straightforward as these two DP variants entail different privacy properties, and theoretical techniques to convert one to the other, such as using group privacy, are not entirely practical.

Acknowledgments. The authors wish to thank Boris Köpf, Santiago Zanella-Béguelin, and Shruti Tople for helpful feedback and comments. This work has been partially supported by a Microsoft EPSRC Case Studentship and an Amazon Research Award grant.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *ACM CCS*, 2016.

[2] A. Agrawal, A. N. Modi, A. Passos, A. Lavoie, A. Agarwal, A. Shankar, I. Ganichev, J. Levenberg, M. Hong, R. Monga, et al. TensorFlow Eager: A multi-stage, Python-embedded DSL for machine learning. *arXiv:1903.01855*, 2019.

[3] S. Alfeld, X. Zhu, and P. Barford. Data Poisoning Attacks against Autoregressive Models. In *AAAI*, 2016.

[4] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *AISTATS*, 2020.

[5] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *ICML*, 2019.

[6] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv:1812.00984*, 2018.

[7] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. *arXiv:1206.6389*, 2012.

[8] P. Blanchard, R. Guerraoui, J. Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In *NeurIPS*, 2017.

[9] 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 *ACM CCS*, 2017.

[10] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi. Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics*, 112, 2018.

[11] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *arXiv preprint arXiv:1802.08232*, 5, 2018.

[12] K. Chaudhuri, A. Sarwate, and K. Sinha. Near-optimal differentially private principal components. In *NeurIPS*, 2012.

[13] D. Chen, N. Yu, Y. Zhang, and M. Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.

[14] H. Chen, C. Fu, J. Zhao, and F. Koushanfar. DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks. In *IJCAI*, 2019.

[15] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. In *International Conference on Machine Learning*, pages 903–912. PMLR, 2018.

[16] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li. Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach. *Computers & Security*, 73, 2018.

[17] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv:1712.05526*, 2017.

[18] Y. Chen, L. Su, and J. Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2), 2017.

[19] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang. Secureboost: A lossless federated learning framework. *arXiv:1901.08755*, 2019.

[20] A. Cheu, A. Smith, and J. Ullman. Manipulation attacks in local differential privacy. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 883–900. IEEE, 2021.

[21] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *IJCNN*, 2017.

[22] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis. Casting out demons: Sanitizing training data for anomaly sensors. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008.

[23] I. Diakonikolas, G. Kamath, D. Kane, J. Li, J. Steinhardt, and A. Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *ICML*, 2019.

[24] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy, data processing inequalities, and statistical minimax rates. *arXiv:1302.3203*, 2013.

[25] C. Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[26] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4), 2014.

[27] D. Enthoven and Z. Al-Ars. An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies. *arXiv:2004.04676*, 2020.

[28] M. Fang, X. Cao, J. Jia, and N. Gong. Local model poisoning attacks to Byzantine-robust federated learning. In *Usenix Security*, 2020.

[29] M. Fang, N. Z. Gong, and J. Liu. Influence function based data poisoning attacks to top-n recommender systems. In *The Web Conference*, 2020.

[30] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM CCS*, 2015.

[31] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security*, 2014.

[32] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *CCS*, 2018.

[33] R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective. *arXiv:1712.07557*, 2017.

[34] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *ICML*, 2016.

[35] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[36] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain.

arXiv:1708.06733, 2017.

[37] J. Hamm. Minimax filter: learning to preserve privacy from inference attacks. *The Journal of Machine Learning Research*, 18(1), 2017.

[38] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv:1811.03604*, 2018.

[39] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv:1711.10677*, 2017.

[40] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. LOGAN: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1), 2019.

[41] J. Hayes and O. Ohrimenko. Contamination attacks and mitigation in multi-party machine learning. In *NeurIPS*, 2018.

[42] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016.

[43] B. Hilprecht, M. Härterich, and D. Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(4), 2019.

[44] B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. In *ACM CCS*, 2017.

[45] S. Hong, V. Chandrasekaran, Y. Kaya, T. Dumitraş, and N. Papernot. On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping. *arXiv preprint arXiv:2002.11497*, 2020.

[46] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. http://vis-www.cs.umass.edu/lfw/lfw.pdf, 2008.

[47] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *AISEC*, 2011.

[48] H. Inan, K. Khosravi, and R. Socher. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*, 2016.

[49] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot. High accuracy and high fidelity extraction of neural networks. In *Usenix Security*, 2020.

[50] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *IEEE S&P*, 2018.

[51] M. Jagielski, G. Severi, N. Pousette Harger, and A. Oprea. Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3104–3122, 2021.

[52] M. Jagielski, J. Ullman, and A. Oprea. Auditing Differentially Private Machine Learning: How Private is Private SGD? *arXiv:2006.07709*, 2020.

[53] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *Conference on Learning Theory*, 2012.

[54] Z. Ji, Z. C. Lipton, and C. Elkan. Differential privacy and machine learning: a survey and review. *arXiv:1412.7584*, 2014.

[55] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *ACM CCS*, 2019.

[56] A. Jochems, T. M. Deist, I. El Naqa, M. Kessler, C. Mayo, J. Reeves, S. Jolly, M. Matuszak, R. Ten Haken, J. van Soest, et al. Developing and validating a survival prediction model for NSCLC patients through distributed learning across 3 countries. *Int J Radiat Oncol Biol Phys*, 99(2):344–352, 2017.

[57] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv:1912.04977*, 2019.

[58] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv:1610.05492*, 2016.

[59] L. Lamport. The weak byzantine generals problem. *Journal of the ACM (JACM)*, 30(3):668–676, 1983.

[60] K. Leino and M. Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *Usenix Security*, 2020.

[61] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *NeurIPS*, 2016.

[62] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*, 2020.

[63] S. Li, H. Liu, T. Dong, B. Z. H. Zhao, M. Xue, H. Zhu, and J. Lu. Hidden backdoors in human-centric language models. *arXiv:2105.00164*, 2021.

[64] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[65] K. Liu, B. Dolan-Gavitt, and S. Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

[66] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2782&context=cstech, 2017.

[67] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen. Understanding membership inferences on well-generalized learning models. *arXiv:1802.04889*, 2018.

[68] L. Lyu, H. Yu, and Q. Yang. Threats to federated learning: A survey. *arXiv:2003.02133*, 2020.

[69] Y. Ma, X. Zhu, and J. Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *IJCAI*, 2019.

[70] D. Maiorca, B. Biggio, and G. Giacinto. Towards adversarial malware detection: lessons learned from pdf-based attacks. *ACM Computing Surveys (CSUR)*, 52(4), 2019.

[71] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 2017.

[72] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[73] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *ACM SIGMOD*, 2009.

[74] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy*, 2019.

[75] P. Mohassel and Y. Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *IEEE Symposium on Security and Privacy*, 2017.

[76] M. Nasr, R. Shokri, and A. Houmansadr. Machine learning with membership privacy using adversarial regularization. In *ACM CCS*, 2018.

[77] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning. In *IEEE Symposium on Security and Privacy*, 2019.

[78] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. A. Sutton, J. D. Tygar, and K. Xia. Exploiting machine learning to subvert your spam filter. *LEET*, 8, 2008.

[79] S. J. Oh, M. Augustin, M. Fritz, and B. Schiele. Towards reverse-engineering black-box neural networks. In *ICLR*, 2018.

[80] T. Orekondy, B. Schiele, and M. Fritz. Knockoff nets: Stealing

functionality of black-box models. In *IEEE CVPR*, 2019.
[81] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv:1610.05755*, 2016.
[82] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. In *AsiaCCS*, 2017.
[83] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable private learning with pate. In *ICLR*, 2018.
[84] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic Differentiation in PyTorch. https://openreview.net/forum?id=BJJsrmfCZ, 2017.
[85] V. Pihur, A. Korolova, F. Liu, S. Sankuratripati, M. Yung, D. Huang, and R. Zeng. Differentially-private" draw and discard" machine learning. *arXiv:1807.04369*, 2018.
[86] K. Pillutla, S. M. Kakade, and Z. Harchaoui. Robust aggregation for federated learning. *arXiv:1912.13445*, 2019.
[87] O. Press and L. Wolf. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016.
[88] E. Quiring and K. Rieck. Backdooring and Poisoning Neural Networks with Image-Scaling Attacks. In *IEEE S&P Workshops*, 2020.
[89] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv.*, 11(1), 2018.
[90] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos. DETOX: A redundancy-based framework for faster and more robust gradient aggregation. In *NeurIPS*, 2019.
[91] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays. Federated Learning for Emoji Prediction in a Mobile Keyboard. *arXiv:1906.04329*, 2019.
[92] E. Rosenfeld, E. Winston, P. Ravikumar, and Z. Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020.
[93] B. I. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *arXiv:0911.5708*, 2009.
[94] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011.
[95] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv:1806.01246*, 2018.
[96] G. Severi, J. Meyer, S. Coull, and A. Oprea. Exploring backdoor poisoning attacks against malware classifiers. *arXiv:2003.01031*, 2020.
[97] V. Shejwalkar and A. Houmansadr. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In *NDSS*, page 18, 2021.
[98] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *MICCAI Brainlesion Workshop*, 2018.
[99] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *ACM CCS*, 2015.
[100] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.
[101] C. Song, T. Ristenpart, and V. Shmatikov. Machine learning models that remember too much. In *ACM CCS*, 2017.
[102] J. Steinhardt, P. W. W. Koh, and P. S. Liang. Certified defenses for data poisoning attacks. In *NeurIPS*, 2017.
[103] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan. Can you really backdoor federated learning? *arXiv:1911.07963*, 2019.
[104] F. Suya, S. Mahloujifar, A. Suri, D. Evans, and Y. Tian. Model-targeted poisoning attacks with provable convergence. In *International Conference on Machine Learning*, pages 10000–10010. PMLR, 2021.
[105] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing machine learning models via prediction APIs. In *USENIX Security*, 2016.
[106] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei. Ldp-fed: federated learning with local differential privacy. In *EdgeSys*, pages 61–66, 2020.
[107] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Towards demystifying membership inference attacks. *arXiv:1807.09173*, 2018.
[108] B. Wang, X. Cao, N. Z. Gong, et al. On certifying robustness against backdoor attacks via randomized smoothing. In *Workshop on Adversarial Machine Learning in Computer Vision*, 2020.
[109] B. Wang and N. Z. Gong. Stealing hyperparameters in machine learning. In *IEEE Symposium on Security and Privacy*, 2018.
[110] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE S&P*, 2019.
[111] M. Weber, X. Xu, B. Karlas, C. Zhang, and B. Li. RAB: Provable Robustness Against Backdoor Attacks. *arXiv:2003.08904*, 2020.
[112] E. Wenger, J. Passananti, Y. Yao, H. Zheng, and B. Y. Zhao. Backdoor attacks on facial recognition in the physical world. *arXiv:2006.14580*, 2020.
[113] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv:1812.02903*, 2018.
[114] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao. Latent backdoor attacks on deep neural networks. In *ACM CCS*, 2019.
[115] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *IEEE CSF*, 2018.
[116] S. Yeom, I. Giacomelli, A. Menaged, M. Fredrikson, and S. Jha. Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning. *Journal of Computer Security*, 28(1), 2020.
[117] D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.
[118] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.
[119] B. Zhang, R. Yu, H. Sun, Y. Li, J. Xu, and H. Wang. Privacy for All: Demystify Vulnerability Disparity of Differential Privacy against Membership Inference Attack. *arXiv:2001.08855*, 2020.
[120] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional mechanism: regression analysis under differential privacy. *arXiv:1208.0219*, 2012.
[121] W. Zhang, S. Tople, and O. Ohrimenko. Leakage of dataset properties in multi-party machine learning. In *30th USENIX Security Symposium*, pages 2687–2704, 2021.

APPENDIX A
HYPERPARAMETERS

In Tables II and III, we report hyperparameters for LPD and CDP in, respectively, the robustness and privacy experiments.

Hyperparameter	EMNIST				CIFAR10				Reddit		Sentiment140			
	LDP		CDP		LDP		CDP		LDP	CDP	LDP		CDP	
σ (Noise Magnitude)	0.8	0.1	-	-	0.5	0.01	-	-	0.025	-	0.8	0.1	-	-
S (Clipping Bound)	5.0	5.0	3.0	5.0	10.0	10.0	10.0	15.0	5.0	10.0	5	10	10	15
z (Noise Scale)	-	-	2.5	1.0	-	-	1.4	1.0	-	1.0	-	-	2.3	1.1
ϵ (Privacy Budget)	3.0	7.5	3.0	8.0	2.5	7.0	2.8	6.0	1.7	1.2	1.9	6.7	2.2	6.1

TABLE II: Hyperparameters for LDP and CDP in the Robustness Experiments.

Hyperparameter	CIFAR100		Purchase100		LFW		
	LDP	CDP	LDP	CDP	LDP	CDP	
σ (Noise Magnitude)	0.1	-	0.1	-	0.1	-	-
S (Clipping Bound)	10.0	15.0	5.0	15.0	12.0	10.0	10.0
z (Noise Scale)	-	0.8	-	1.1	-	1.4	0.7
ϵ (Privacy Budget)	8.6	5.8	8.6	5.8	10.7	4.7	8.1

TABLE III: Hyperparameters for LDP and CDP in the Privacy Experiments.

#Participants	No Defense	LDP (ϵ = 10.7)	CDP (ϵ = 4.7)	CDP (ϵ = 8.1)
5	90%	83%	59%	85%
10	89%	81%	57%	83%
15	88%	80%	54%	82%
20	87%	78%	53%	79%
25	85%	70%	53%	77%
30	81%	68%	51%	73%

TABLE IV: Main Task (Gender Classification) Accuracy with No Defense, LDP, and CDP (property inference attack setting).

#Participants	No Defense	LDP (ϵ = 10.7)	CDP (ϵ = 8.1)
5	0.97	0.95	0.94
10	0.87	0.86	0.85
15	0.76	0.75	0.76
20	0.70	0.70	0.68
25	0.54	0.52	0.50
30	0.48	0.47	0.45

TABLE V: AUC of the Property Inference Attack in [74] with No Defense, LDP, and CDP.

APPENDIX B
PROPERTY INFERENCE ATTACK

Table IV presents the federated gender classification main task accuracy when LDP, CDP, and no DP are applied. Table V reports the property inference attack accuracy in the form of the area under a curve (AUC score) when LDP, CDP, and no DP are applied.