Gradient-Leakage Resilient Federated Learning

Wenqi Wei[†], Ling Liu[†], Yanzhao Wu[†], Gong Su[*], Arun Iyengar[*]

[†] Georgia Institute of Technology, School of CS, Atlanta, GA 30332, USA

[*] IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

Abstract—Federated learning(FL) is an emerging distributed learning paradigm with default client privacy because clients can keep sensitive data on their devices and only share local training parameter updates with the federated server. However, recent studies reveal that gradient leakages in FL may compromise the privacy of client training data. This paper presents a gradient leakage resilient approach to privacy-preserving federated learning with per training example-based client differential privacy, coined as Fed-CDP. It makes three original contributions. First, we identify three types of client gradient leakage threats in federated learning even with encrypted client-server communications. We articulate when and why the conventional server coordinated differential privacy approach, coined as Fed-SDP, is insufficient to protect the privacy of the training data. Second, we introduce Fed-CDP, the per example-based client differential privacy algorithm, and provide a formal analysis of Fed-CDP with the (ϵ, δ) differential privacy guarantee, and a formal comparison between Fed-CDP and Fed-SDP in terms of privacy accounting. Third, we formally analyze the privacy-utility trade-off for providing differential privacy guarantee by Fed-CDP and present a dynamic decay noise-injection policy to further improve the accuracy and resiliency of Fed-CDP. We evaluate and compare Fed-CDP and Fed-CDP(decay) with Fed-SDP in terms of differential privacy guarantee and gradient leakage resilience over five benchmark datasets. The results show that the Fed-CDP approach outperforms conventional Fed-SDP in terms of resilience to client gradient leakages while offering competitive accuracy performance in federated learning.

Index Terms—Distributed systems, federated learning, differential privacy, gradient leakage, attack resilience.

I. INTRODUCTION

Federated learning (FL) decouples the learning task from the centralized server to a distributed collection of client nodes [1]–[5]. It collaboratively learns a shared global model over the distributed collection of training data residing on end-user devices of the participating clients. Each client maintains its private data locally by downloading the global model update from the server at each round of the federated learning, performing local training, and sharing its local model update via encrypted communication to the server.

However, the distributed nature and the default privacy in federated learning are insufficient for protecting client training data from gradient leakage attacks. Recent studies [6]–[8] reveal that if an adversary intercepts the local gradient update of a client before the server performs the federated aggregation to generate the global parameter update for the next round of federated learning, the adversary can steal the sensitive local training data of this client using the leaked gradients by simply performing reconstruction attack [8], or membership and attribute inference attacks [9]–[11].

To circumvent such gradient leakage threats, we present Fed-CDP, the per example-based client differential privacy approach with three original contributions. First, we categorize client gradient leakage threats into three types based on whether the gradient information was leaked at the server, the client, or the training example level. We analyze when and why the conventional server coordinated differential privacy approach [12], [13], coined as Fed-SDP in this paper, is insufficient to protect the privacy of client training data in the presence of instance-level gradient leakage attacks. Second, we introduce Fed-CDP to provide the formal instance-level and client-level differential privacy guarantee. In contrast to Fed-SDP, which adds noise to the shared gradient update by a client at each round of the FL joint training, Fed-CDP adds noise to per example at each iteration per client. The differential privacy spending is computed by the privacy accounting using the moments accountant [14] for both Fed-SDP and Fed-CDP. Finally, we provide a formal analysis of the privacy-utility trade-off for providing differential privacy guarantee by Fed-CDP, with the maximal distortion bound without deteriorating the performance of federated learning. Guided by this formal trade-off analysis, we optimize Fed-CDP by enabling a dynamic decay noise-injection policy, which boosts both the accuracy performance and the resilience against gradient leakage attacks. Evaluated on five benchmark datasets, we show that per-example based Fed-CDP and Fed-CDP (decay) outperforms per client based Fed-SDP in terms of accuracy performance, differential privacy guarantee, and gradient leakage resilience.

II. RELATED WORK

Gradient leakage threats refer to unauthorized disclosure of private client training data based on the leaked gradient information and are one of the dominating privacy leakage threats in both centralized and distributed deep learning systems. In centralized deep learning, membership inference [9]–[11], [15] is known to expose the private training data by exploiting the statistical differences between the model prediction on the data possibly seen during training and the prediction on unseen data. [16] performed minor modifications to training algorithms to have them memorize a significant amount of information about the training data. [17] inferred private data based on the proportionality between the training data and the gradient updates but fails to be generalized to convolutional neural networks due to the relative size of convolution weights and data features. [18] proposed the model inversion attack to expose privately trained model parameters by exploiting confi-
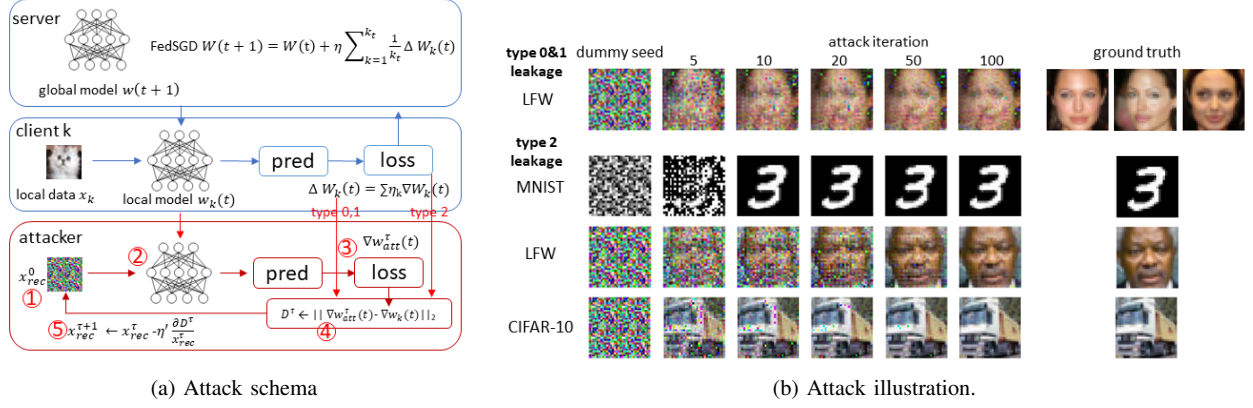
Fig. 1: Gradient leakage attacks in Federated Learning. Type-0&1 leakage attack by batched update and type-2 attacks by example

dence values revealed from legitimate access to the prediction API. GAN-based reconstruction attack [19], [20] utilizes generative adversarial networks to learn private training data based on the gradient information and the confidence of prediction results. In distributed federated learning, [6]–[8] showed the vulnerabilities of client gradient leakage attacks. [8] showed that such gradient-based client privacy leakage (CPL) attacks display high reconstruction success rate with high attack effect and low attack cost.

Multiple attempts have been made to offer privacy-preserving federated learning. Cryptographic approaches [21] and secure multi-party computation (SMP) [22], [23] ensure secure communication of shared parameter updates between a client and its federated server. However, it does not secure the client data prior to encryption for transport or after decryption for the server aggregation. [24] proposed PATE, a teacher-student ensemble, which relies on a trusted server for private aggregation, but the involved third party can bring new risk to the FL system. [25] proposed a distributed learning with selective and random sharing of model parameters, offering the shared model parameter level privacy. In recent proposals for differentially private federated learning [12], [13], the noise is added to the per-client shared gradient updates at each round of FL, the effect and cost of such differential privacy guarantee are the same regardless of whether the clipping and the noise injection are done at the server or at each client. In this paper, we coin this conventional approach as Fed-SDP. We will show that both the selective parameter sharing approach and Fed-SDP approach fail to safeguard the client's private training data in the presence of instance-level gradient leakage attacks.

III. GRADIENT LEAKAGE THREATS

Threat Model. Our threat model adopts the two common assumptions: (i) the message communicated between a client and its FL server is encrypted; and (ii) the clients faithfully follow the rules of joint training when subscribed to a federated learning task, although the server may conduct unauthorized inference on per-client local model updates either curiously or maliciously and a client may read the local training result during or upon completion of the local training at each round

of federated learning. We categorize the gradient leakage threats into three types based on the location and whether the gradient leakage is performed on per-example gradient update during local training or per client gradient update after completing the local training at each round of FL. The **type-0** refers to the gradient leakage attacks performed at the server by intercepting per-client shared gradient updates. The **type-1** refers to the gradient leakage attacks performed at a client on its gradient updates resulting from the completion of the local training. The third type of gradient leakage attacks are performed on each training example during the local training at a client and coined as the **type-2** leakage. We consider the gradient leakage attack that uses gradient-based reconstruction attack methods [6]–[8] to derive and steal the private client training data. Although all three types of gradient leakages are inferred by adversarial reconstruction, the first two types of leakages are launched by an unauthorized read of per-client gradient update resulting from the completion of the local training over multiple iterations at a client. The **type-2** leakages are launched by an unauthorized read of the per-example gradient updates during the local training on a client.

Figure 1a gives a sketch of the gradient leakage attack algorithm, which configures and executes the reconstruction attack in five steps: (1) It configures the initialization seed $(x_{rec}^0(t))$, a dummy data of the same resolution (or attribute structure for text) as the training data. [8] showed some significant impact of different initialization seeds on the attack success rate and attack cost (# attack iterations to succeed). (2) The dummy attack seed is fed into the client local model. (3) The gradient of the dummy attack seed is obtained by backpropagation. (4) The gradient loss is computed using a vector distance loss function, e.g., L_2, between the gradient of the attack seed and the actual gradient from the client local training. The choice of this reconstruction loss function is another tunable attack parameter. (5) The dummy attack seed is modified by the attack reconstruction learning algorithm. It aims to minimize the vector distance loss by a loss optimizer such that the gradients of the reconstructed seed $x_{rec}^i(t)$ at round i will be closer to the actual gradient updates stolen

from the client upon the completion (type 0&1) or during the local training (type 2). This attack reconstruction learning iterates until it reaches the attack termination condition ($\mathbb{T}$), typically defined by the #rounds, e.g., 300 (also a configurable attack parameter). If the reconstruction loss is smaller than the specified distance threshold then the reconstruction attack is successful. Figure 1b provides a visualization by examples from three datasets. The type-0&1 gradient leakage attack is performed on the batched gradients with the batch size of 3 from LFW dataset [26]. The type-2 leakage is performed on per example gradient during local train with three examples from MNIST [27], LFW and CIFAR10 [28] respectively. Although all three types of gradient leakage attacks succeed at the 50th attack iteration for these examples with $\mathbb{T}$ set to 300, type 2 leakage attacks can succeed with smaller #iterations and higher attack quality for the same #iterations (e.g., 100). All the experiments on gradient leakages in this paper use the patterned random seed initialization[1], with a L_2 based loss function and L-BFGS optimizer, for high attack success rate (ASR) and fast attack convergence.

IV. SOLUTION APPROACH

We first define the core differential privacy concepts following the state-of-the-art literature. Then we introduce the reference model for federated learning (both non-private and with different privacy) and present the problem statement, followed by our proposed Fed-CDP, the per-example based client differential privacy approach to privacy-preserving federated learning, including algorithm, privacy parameters, and configuration trade-offs.

A. Preliminary Concepts and Reference Model

Definition 1. *Differential privacy [29]: a randomized mechanism $\mathcal{M}: \mathcal{D} \rightarrow \mathcal{R}$, with a domain $\mathcal{D}$ of possible training data and range $\mathcal{R}$ of all possible trained models, satisfies (ϵ, δ)-differential privacy if for any two input sets $A \in \mathcal{D}$ and $A' \in \mathcal{D}$, differing with only a single example input and for any subset of outputs $B \in \mathcal{R}$, Equation 1 holds.*

$$\Pr(\mathcal{M}(A) \in \mathcal{R}) \le e^{\epsilon} \Pr(\mathcal{M}(A') \in \mathcal{R}) + \delta. \quad (1)$$

Definition 2. *Gaussian mechanism [29]: when $0 < \epsilon < 1$, applying Gaussian noise $\mathcal{N}(0, \sigma^2 S^2)$ calibrated to a real valued function: $f: \mathcal{D} \rightarrow \mathcal{R}$ such that $\mathcal{M}(A) = f(A) + \mathcal{N}(0, \sigma'^2)$, $\mathcal{M}(A)$ is (ϵ, δ)-differentially private if $\sigma'^2 > \frac{2 \log(1.25/\delta) \cdot S^2}{\varepsilon^2}$, where for any two adjacent inputs $A, A' \in D$, $||A - A'||_1 = 1$, and $S = \max_{A, A' \in D} ||f(A) - f(A')||_2$, i.e., the maximum l_2 distance between $f(A)$ and $f(A')$.*

With Definition 2, it is straightforward to get Lemma 1.

Lemma 1. *Let σ'^2 in Gaussian mechanism be $\sigma^2 S^2$. We have the noise scale σ satisfying $\sigma^2 > \frac{2 \log(1.25/\delta)}{\varepsilon^2}$.*

Definition 3. *Privacy amplification [30]: Given dataset $\mathcal{D}$ with N data points, define subsampling as random sampling with replacement and sampling rate as $q = n/N$ where n*

is the sample size. If $\mathcal{M}$ is (ϵ, δ)-differentially private, then the subsampled mechanism with sampling rate q is $(\log(1 + q(\exp(\epsilon) - 1)), q\delta)$-differentially private.

Definition 4. *Composition theorem [29]: Given a fixed σ and ϵ, then a set of consecutive inquiries of (ϵ, δ) differentially private algorithms is $(\sum \epsilon, \sum \delta)$ differentially private.*

Definition 5. *Moments accountant [14]: For a mechanism $\mathcal{M}$, given dataset $\mathcal{D}$ with a total of N data points, given the random sampling with replacement and the sampling rate $q = n/N$ where n is the sample size, and the number of iterations T, if $q < \frac{1}{16\sigma}$, there exist constants $c1$ and $c2$ so that for any $\epsilon < c1q^2T$, $\mathcal{M}$ is (ϵ, δ)-differentially private for $\delta > 0$ and*

$$\epsilon \ge c2 \frac{q\sqrt{T \log(1/\delta)}}{\sigma} \quad (2)$$

Moments accountant is implementated[2] in a way that privacy spending ϵ is computed when T, σ, δ, and q are given.

Reference model for federated learning. Federated learning can be viewed as a publish-subscribe model, in which the server publishes its federated learning task, described by the data structure and data modality and the neural network model (algorithm) for joint training, including the important hyperparameters: total #rounds T of joint training, local batch size B, learning rate η, local # iterations L, format for messaging between client and server, and the public key of the server for encryption. By subscribing to a published FL task, the clients i informs the server its local training dataset size N_i and commits to download the joint training model with the default hyperparameters, then faithfully follow the FL protocol: at each round $t \le T$, a client downloads the current global model parameters $W(t)$ from the server, performs local model training on the private client training data over L local iterations, i.e., $W_i(t)_{l+1} = W_i(t)_l - \eta \nabla W_i(t)_l, 1 < l < L - 1$ where ∇W is the gradient of the trainable network parameters and $1 \le L \le \lceil N_i/B \rceil$. Upon completion of the local model training at round t, this client i sends the local training parameter updates to the server.

Upon receiving local training parameter updates from K_t out of a total of K clients ($K_t < K$), i.e., $\Delta W_i(t) = W_i(t)_L - W_i(t)$ ($i = 1 \ldots K_t$), the server aggregates the local updates from all K_t clients to produce the global model parameter updates and start the next round $t + 1 \le T$ of joint training until T rounds are reached. As stated in [2], to deal with the unstable client availability, only a small subset of K_t clients are selected ($K_t < K$) to participate in each round t of federated learning. Two aggregation methods are commonly used: **FedSGD**: $W(t + 1) = W(t) + \frac{1}{K_t} \sum_{k=1}^{K_t} \Delta W_k(t)$ if the clients share the local model parameter updates [31]–[33]; and **FedAveraging**: $W(t + 1) = \sum_{k=1}^{K_t} \frac{1}{K_t} W_k(t)_L$ if the clients send the locally updated model to the server [2], [34], [35]. With a large client population (K) and a small subset $K_t << K$ of contributing clients per round, federated

---

[1]https://github.com/git-disl/ESORICS20-CPL

[2]https://github.com/tensorflow/privacy/blob/master/tensorflow_privacy/privacy/analysis/compute_dp_sgd_privacy.py

Algorithm 1: Fed-SDP

Input: N_i: # local data, D_i: local data, K_t: # clients per round, K: # total clients, T: maximum global round, σ: noise scale, C: clipping bound.

1 **Server initialization and broadcast:** global model $W(0)$ of M layers, local batch size B, # local iteration $1 \leq L \leq \lceil N_i/B \rceil$, local learning rate η.
2 **for** *round t in* $\{0, 1, 2...T-1\}$ **do**
3   **client i does non-private local training and send out local updates:** $\Delta W_i(t) \leftarrow W_i(t)_L - W(t)$.
4   **server do**
    // collect updates from K_t clients
5     $\Delta W_k(t), k = 1, ...K_t$
6   **for** k *in 1,...,K_t* **do**
7   **for** m *in* $1, ...M$ **do**
          // compute $L2$ norm for layer m
8           $||\Delta W_{ij}(t)_{lm}||_2$
          // clip per-client update for layer m
9           $\overline{\Delta} W_k(t)_m \leftarrow= \Delta W_k(t)_m / \max(1, \frac{||\nabla W_k(t)_m||_2}{C})$
10   **end**
       // obtain the clipped per-client updates
11        $\overline{\Delta} W_k(t) \leftarrow \{\overline{\Delta} W_k(t)_m\}, m = 1, .., M$
12   **end**
    // compute sanitized updates with $S \leftarrow C$
13     $\widetilde{\Delta} W(t) \leftarrow \frac{1}{K_t} (\sum_{k=1}^{K_t} (\overline{\Delta} W_k(t) + \mathcal{N}(0, \sigma^2 S^2))$
    // aggregation:
14     $W(t+1) \leftarrow W(t) + \frac{1}{K_t} \sum_{k=1}^{K_t} \widetilde{\Delta} w_k(t)$
15 **end**
16 **Output:** global model $W(T)$

Fig. 2: System schema of Fed-CDP

learning can produce a jointly trained model with high accuracy [2], [3]. Given that the FedSGD and Fedaveraging are mathematically equivalent, we consider FedSGD aggregation in this paper without loss of generality. For privacy-preserving federated learning, the server will provide the differentially private version of the federated model instead of the non-private version for clients to download, together with the set of differential privacy parameters.

B. Problem Statement

Recent proposals for differentially private federated learning, represented by [12], [13] (as illustrated in Algorithm 1), add Gaussian noise to the per-client shared gradient updates at server each round $t \leq T$. They differ mainly in whether the clipping operation is performed at the server or at each client upon the completion of local training at round t. We coin these approaches as Fed-SDP in this paper since they inject noise to the per-client gradient updates at each round t. The effect and cost of these clipping and per-client noise injection, whether is done at the server or per client, deliver the same client-level differential privacy guarantee [12].

Recall the threat model in Section III, the Fed-SDP approach is resilient to both type 0&1 gradient leakage attacks if differential privacy is performed by adding Gaussian noise at each client and the adversary intercepts the noisy per-client local parameter updates. The noisy per-client gradient updates make the leakage attack much harder to succeed due to poor reconstruction inference, especially as the number of rounds
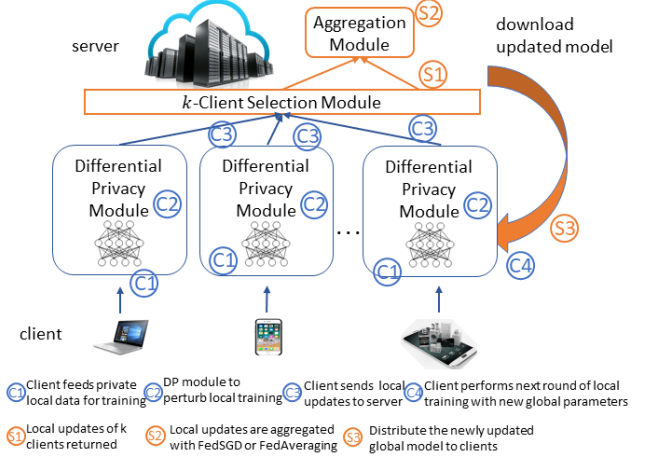
increases in federated learning. Of course, for type-1 gradient leakage, if the Gaussian noise is added at the server but the type-1 leakage is performed on the true local gradient updates at a client, then this version of the Fed-SDP is only resilient to type-0 leakage and cannot protect against type-1 leakage. However, existing versions of Fed-SDP are vulnerable against type-2 leakages which steal per-example gradients during local training, because simply adding Gaussian noise to the per-client local parameter update upon completion of the local training at each round t will fail miserably against the per-example type-2 leakages. This motivates the design of our Fed-CDP to provide a per-example based client differential privacy guarantee in privacy-preserving federated learning.

C. Fed-CDP: Per-example Client Differential Privacy

Figure 2 gives a system overview of Fed-CDP. Fed-CDP provides gradient leakage resilience against all three types of attacks by adding DP-guided Gaussian noise to the per-example gradients once they are computed before local batch averaging in each of the L iterations during the local training at a client. This enables the support for instance-level differential privacy guarantee. Algorithm 2 gives the pseudo-code for Fed-CDP. We first discuss a set of important differential privacy parameters used in Fed-CDP.

Random Subsampling. Given a total of K participants (clients) and a total of T rounds for distributed learning, in each round t ($1 \leq t \leq T$), a subset of K_t clients is randomly sampled with replacement under probability q. The random sampling by selecting each client independently with probability q is essential for providing low privacy loss through tracking privacy spending using the moments accountant [14].

Choosing Clipping Strategy C**.** When clipping the client gradient updates, a tradeoff should be made. To keep the noise variance small, we should choose a small clipping upper bound C. However, if the goal is to maintain the original update values as much as possible, a too-small clipping bound C should not be chosen. Instead, we can use the median norm of all original updates from the training set of the

---

Algorithm 2: Fed-CDP

Input: N_i: # local data, D_i: local data, K_t: # clients per round, K: # total clients, T: maximum global round, σ: noise scale, C: clipping bound

1 **Server initialization and broadcast:** global model $W(0)$ of M layers, local batch size B, # local iteration $1 \leq L \leq \lceil N_i/B \rceil$, local learning rate η.

2 **for** *round t in* $\{0, 1, 2...T-1\}$ **do**

3   **client** i **do**
      // download current global model:

4        $W_i(t)_0 \leftarrow W(t)$

5   **for** *local iteration l in* $\{0, 1, 2...L-1\}$ **do**
        // batch processing, take subset of D_i

6   **for** *j in* $\{1,...B\}$ **do**

7   **for** *m in* $\{1,...M\}$ **do**
            // **compute per example gradient for layer** m

8              $\nabla W_{ij}(t)_l \leftarrow \{\nabla W_{ij}(t)_{lm}\}$
            // **compute $L2$ norm for layer** m

9              $||\nabla W_{ij}(t)_{lm}||_2$
            // **clip per-example gradients for layer** m

10             $\overline{\nabla} W_{ij}(t)_{lm} \leftarrow$
                $\nabla W_{ij}(t)_{lm} / \max(1, \frac{||\nabla W_{ij}(t)_{lm}||_2}{C})$

11   **end**
          // **obtain the clipped per-example gradients**

12           $\overline{\nabla} W_{ij}(t)_l \leftarrow \{\overline{\nabla} W_{ij}(t)_{lm}\}, m = 1, .., M$

13   **end**

14   // **compute sanitized batch gradients with** $S \leftarrow C$

           $\widetilde{\nabla} W_i(t)_l \leftarrow \frac{1}{B} \sum_{j=1}^{B} (\overline{\nabla} W_{ij}(t)_l + \mathcal{N}(0, \sigma^2 S^2))$

        // **gradient descent:**

15         $W_i(t)_{l+1} \leftarrow W_i(t)_l - \eta \widetilde{\nabla} w_i(t)_l$

16   **end**
      // **send out local updates:**

17       $\Delta W_i(t) \leftarrow W_i(t)_L - W(t)$ // with M layers

18   **server do**
      // **collect local updates from** K_t **clients**

19       $\Delta W_i(t), i = 1, ...K_t$
      // **aggregation:**

20       $W(t+1) \leftarrow W(t) + \frac{1}{K_t} \sum_{i=1}^{K_t} \Delta w_i(t)$

21 **end**

22 **Output:** global model $W(T)$

---

client i and use this median as the clipping bound, i.e., $C \leq \text{median}_{i \in N_i} \nabla w_i(t)$. Alternatively, we can also define clipping as a function of learning rate η. With adaptive learning rate methods, in which the learning rate is systematically decreasing as the model converges [36].

Clipping Bound C **and Sensitivity** S. To provide a fair comparison with Fed-SDP, in this paper we follow [14]: which uses a preset fixed clipping bound C to confine the influence of the local model update under $L2$ norm and also use the clipping bound C to estimate the sensitivity for adding noise in Gaussian Mechanism (recall Definition 2). A trade-off should be considered when clipping the contribution of the per-example gradient, as the clipping bound can affect both the noise variance and the informative contribution of the gradient. A large clipping bound would significantly enlarge the Gaussian noise variance and thus pushing the gradient away from its training goal and may thus downgrade or even fail the training, since the large gradient reflects an aggressive step of modification on the model weight for gradient descent. On the other hand, a small clipping bound may keep noise

variance stay small but may end up pruning those informative gradients. Empirically, following the literature [14], $C = 4$ is used as the default unless otherwise stated.

Choosing noise scale σ. By Definition 2, σ controls the noise level and by moment accounting in Definition 5 (Equation 2), with fixed δ, the privacy loss accumulates inverse-proportionally as noise scale σ changes. Existing literature selects the noise scale based on the learning method and task. For example, [13] uses a small noise scale $\sigma = 1$ with clipping bound $C = 1$ for client-level differential privacy and [12] uses σ as small as 0.012 with clipping bound $C = 15$ tailor to the local model update in the federated language model. [14] sets its default σ to 6 with clipping $C = 4$ for privacy accounting via moments accountant. [37] considered a large σ of 10 with clipping $C = 4$ for centralized differentially private model publishing, in which the batch gradient is less sensitive to noise. In this paper, we adopt the same setting as [14]: $\delta = 1e - 5$, $\sigma = 6$ with $C = 4$ as the default unless otherwise stated.

Workflow of Fed-CDP algorithm. In each round $t \leq T$ of federated learning, for each client i from the K_t chosen clients, the local model training is performed on each iteration l over the L local iterations ($1 \leq L \leq \lceil N_i/B \rceil$). For each iteration l, we compute the per-example gradient layer by layer over the M layers for each instance j in a batch B by minimizing the empirical loss function (Lines 6-9). For each layer l, the number of gradients is the same as the number of trainable parameters and there exists an $L2$-norm for the layer accordingly. Therefore, a M layer neural network will have M $L2$ norms, one for each layer. Once the per-example gradient is computed for layer m ($1 \leq m \leq M$), we clip the gradient in L_2 norm (Line 10), i.e., the L_2 norm of layer-wise per-example gradient vector $\nabla w_{ij}(t)_{lm}$ is clipped down to C if $||\nabla w_{ij}(t)_{lm}||_2 > C$ and $\nabla w_{ij}(t)_{lm}$ is preserved if $||\nabla w_{ij}(t)_{lm}||_2 \leq C$. The per-example gradient will consist of M components (Line 12). Then Gaussian noise is added to the clipped per-example gradient (Line 14). For each batch, the sanitized per-example gradients are gathered for batch averaging and local stochastic gradient descent (Line 15). Upon the completion of the local training over the L iterations, the local training model update is produced by the client i using the Fed-CDP client module and this noisy local gradient update will be shared with the server (Line 17) for FedSGD aggregation and global model update for the next round of training (Lines 18-20). It is important to note that any change in the gradient of one layer after it is computed will not change the computation of the gradient of its previous layer. This indicates that clipping and adding noise to computed gradients of one layer cannot protect other layers in the model. Hence, Fed-CDP requires the clipping and sanitization to be performed on the gradient of every layer for all M layers and at every local iteration to ensure that the local model training and the global model from the joint training satisfy the per-example based client differential privacy.

By applying clipping and adding noise layer by layer with back-propagation computed from the loss function to the

weights of the first model layer, this ensures that sanitizing of per-example per-layer gradients is done immediately after gradients of one layer is computed, enabling Fed-CDP to provide instant protection of the per-example local training gradients against type-2 leakage at each client. Furthermore, the per-example clipping and adding noise at each local client will have a global effect through the iterative joint training of T rounds, enabling Fed-CDP to be resilient against type-0 and type-1 leakages as well.

V. PRIVACY ANALYSIS

The design of Fed-CDP aims to provide differential privacy guarantee at both instance level and client level. In this section, we formally analyze the privacy guarantee by Fed-CDP.

Privacy at instance-level. In the definition of (ϵ, δ) differential privacy [29], the two adjacent inputs $A \subseteq \mathcal{D}$ and $A' \subseteq \mathcal{D}$ differing by only a single example, and the privacy protection by the (ϵ, δ)-differential privacy is to introduce a randomized mechanism such that the differentially private version of the learning algorithm will be used to ensure the privacy at instance level, that is, even if an adversary can observe the output of the algorithm, he cannot tell whether the user information defined by a single training example was used in the computation of the algorithm.

Given a distributed collection $\mathcal{D}$ of training data residing on a total of K disseminated clients, Fed-CDP enforces per-example gradient clipping and sanitization via Gaussian noise injection, leading to instant protection of the per-example local training gradients against type-2 leakage at each client. By Definition 5, for a mechanism $\mathcal{M}$, given the privacy parameters δ and σ (> 0), $\mathcal{M}$ is (ϵ, δ)-differentially private if the sample rate q satisfies $q < \frac{1}{16\sigma}$ and ϵ satisfies the following equation:

$$c2 \frac{q\sqrt{T \log(1/\delta)}}{\sigma} \leq \epsilon < c1q^2T. \quad (3)$$

In order to leverage the moment accountant [14] to compute the privacy budget spending (ϵ) of Fed-CDP during the joint training for the T rounds, we simply need to prove that local sampling with replacement across different clients in federated learning can be modeled as global data sampling with replacement. Thus, federated learning can be viewed as an alternative implementation of centralized training in terms of algorithmic logic.

Proposition 1. *The probability of any class being selected globally is the same as the probability of that class in the global data distribution.*

One way to simplify the proof of this proposition is to consider each client hold data points from two classes out of the total number of classes (Z), and compute the probability of sampling data points from the two classes from a random chosen client i and the probability of the event of a client sampling data from one of its classes. Based on these two probabilities, we can then compute the probability of a specific class being selected randomly, which would be equivalent to $\frac{1}{Z}$. This indicates that the local sampling with replacement

can be modeled as global data sampling with replacement. For Fed-CDP with the total amount of training data N over the total K clients, the sampled local data over K_t out of K clients in each round t can be viewed as a global data sampling with sampling size of $B * K_t$ and the sampling rate $\frac{B*K_t}{N}$. Hence, we can compute the privacy spending using moments accountant similar to centralized differential privacy. Hence the locally added per-example noise would have a global privacy effect over the global data.

Privacy at client-level In Fed-CDP, for each round t, consider any client i from the chosen K_t clients, the local training parameter updates shared by client i to the server are the local aggregation of noisy per-example gradients over L local iterations with batch size B in each iteration. Thus, the proposed Fed-CDP offers per-example based client differential privacy and at the same time offers joint differential privacy [38] from a per-client perspective. Joint differential privacy requires that for two adjacent inputs K and K′, differing by only the update from one client i at each round t of the joint training, the privacy protection by the (ϵ, δ)-joint differential privacy is to introduce a randomized mechanism such that the differentially private version of the learning algorithm will be used to ensure the privacy at client level: even if an adversary can observe the output of the algorithm, he cannot tell whether the local training parameter updates from a single client was used in computing the output of the algorithm.

Definition 6. *Joint Differential privacy [38]: A mechanism $\mathcal{M}: \mathcal{D}(K) \to \mathcal{R}$ satisfies joint (ϵ, δ)-differential privacy if for any two sets of agents K or K', each with its data as $\mathcal{D}(K), \mathcal{D}(K') \subseteq \mathcal{D}$, the following condition holds.*

$$\Pr(\mathcal{M}(\mathcal{D})(K) \in \mathcal{R}) \leq e^\epsilon \Pr(\mathcal{M}(\mathcal{D})(K') \in \mathcal{R}) + \delta. \quad (4)$$

Joint differential privacy is a relaxation of the standard differential privacy but yet strong: it implies that for any client i in federated learning (agent), even if all the other clients (agents) except client i collude and share their information, they would not be able to learn about client i's private information [38]. Fed-CDP achieves joint differential privacy by utilizing the Billboard Lemma (lemma 2) and considering the differentially private global model at the server shared with all clients as the public information on the Billboard.

Lemma 2. *Billboard Lemma [39] Suppose $\mathcal{M}: \mathcal{D} \to \mathcal{R}$ is (ϵ, δ)-differentially private. Consider any set of functions $f_k: \mathcal{D}_i \times \mathcal{R} \to \mathcal{R}'$, where $\mathcal{D}_i \subseteq \mathcal{D}$ containing agent i's data. The composition $\{f_i(\prod_i \mathcal{D}, \mathcal{M}(\mathcal{D}))\}$ is (ϵ, δ)-joint differentially private, where $\prod_i: \mathcal{D} \to \mathcal{D}_i$ is the projection to agent i's data.*

Given that existing Fed-SDP approaches [12], [13] clip and add noise at each round t to the local model update shared by each of the K_t clients, Fed-SDP does not explicitly provide the instance-level client different privacy but it does provides the client-level different privacy guarantee in the sense that for two possible sets of clients, K and K', differing by only one client i at each round t of the joint training, by introducing

802

a randomized mechanism such that the differentially private version of the learning algorithm will be used to ensure that even if an adversary can observe the output of the algorithm, he cannot tell whether the local training parameter updates from a single client was used in the computing the output of the algorithm. Thus, the (ϵ, δ)-differential privacy by Fed-SDP provides privacy protection at client level. Note that LDP with federated learning [40], which independently protects single value of clients' data locally and does not offer differential privacy guarantee on the globally trained model, is beyond the scope of this paper.

VI. PRIVACY-UTILITY TRADE-OFF ANALYSIS

The privacy and utility trade-off is known to be an open challenge [41], [42]. In Fed-CDP, to provide per-example based client differential privacy, in each round t, for each of the selected K_t clients, we clip and then add noise to the per-example gradient before performing the local batch stochastic gradient descent for each of the L iterations in the local model training by the client. The same process of clipping and noise injection repeats for a total of T rounds to produce the global model as the outcome of the joint training.

We first analyze how the injected Gaussian noise affects the privacy-performance trade-off in terms of changing gradient descent directions and provide a formal guarantee of the maximal distortion which a neural network model can tolerate without deteriorating the training performance.

Let $g_j(x; w(t)) : \mathcal{R}^D \rightarrow R$ be the loss function of the continuously differentiable DNN model on class j given the input x. We define $s(x, w(t)) = g_y(x; w(t)) - g_{j, j \neq y}(x; w(t))$ where y is the label of input x. Under the assumption of Lipschitz condition, we have

$$|s(x, w(t)) - s(x, w'(t))| \leq L_v ||w(t) - w'(t)||_u, \quad (5)$$

where $w'(t)$ is the differentially private version of the model parameters $w(t)$, and $L_v = \max_x \{||\nabla s(x, w(t))||_v\}$ is the Lipschitz constant with $\frac{1}{u} + \frac{1}{v} = 1$ where $u \geq 1$ and $v \leq \infty$. By line 14 of Algorithm 2, we can formulate the private model parameters as $w'(t) = w(t) + \xi$ and substitute Equation 5 as

$$s(x, w(t)) - L_v ||\xi||_u \leq s(x, w(t)+\xi) \leq s(x, w(t)) + L_v ||\xi||_u. \quad (6)$$

Equation 6 shows that when $s(x, w(t)+\xi) < 0$, we will have $g_y(x; w(t)+\xi) < g_{j, j \neq y}(x; w(t)+\xi)$, indicating an alternation of ground-truth class used to compute the loss function. This shows that adding large Gaussian noise to the model, may move the loss function computed on the ground-truth class to other classes and lead to the loss of utility with respect to learning convergence and prediction accuracy. Hence, we want the privacy-preserving cost $||\xi||_p$ to be small enough such that $s(x, w(t)) - L_v ||\xi||_u \geq 0$.

Proposition 2. *Upper bound ξ_u for Utility-Privacy trade-off To preserve the training against the given ground-truth label y and maintain the training information, the maximum change made to model parameters (or gradient) need to be bounded as follows:*
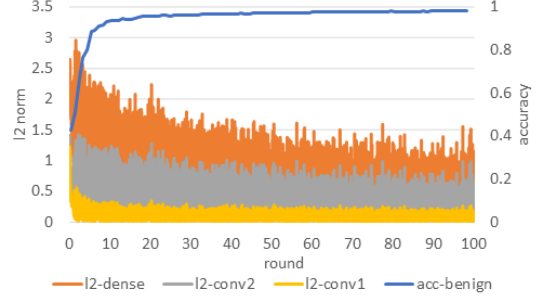

Fig. 3: Changing of gradients' $L2$ norm during training

$$||\xi||_u \leq \min_{j \neq y} \frac{g_y(x; w(t)) - g_{j, j \neq y}(x; w(t))}{L_v}. \quad (7)$$

This proposition states that $||\xi||_u \leq \min_{j \neq y} \frac{g_y(x; w(t)) - g_{j, j \neq y}(x; w(t))}{L_v}$ is the lower bound for minimum distortion of the model parameter that would result in the loss of training performance. We can prove the proposition based on the Lipschitz assumption as follows: First, we have

$$|s(x, w(t)) - s(x, w'(t))| \leq L_v ||w(t) - w'(t)||_u = L_v ||\xi||_u \Rightarrow$$
$$s(x, w(t)) - L_v ||\xi||_u \leq s(x, w(t) + \xi) \leq s(x, w(t)) + L_v ||\xi||_u \quad (8)$$

To ensure that the distortion does not affect the training process by alternating the prediction towards the ground-truth class to other classes, we will need $s(x, w(t)) - L_v ||\xi||_u \geq 0$ in Equation 8 holds. Then we have

$$s(x, w(t)) - L_v ||\xi||_u \geq 0 \Rightarrow ||\xi||_u \leq \frac{s(x, w(t))}{L_v}$$
$$\Rightarrow ||\xi||_u \leq \min_{j \neq y} \frac{g_y(x; w(t)) - g_{j, j \neq y}(x; w(t))}{L_v}$$

Proposition 2 indicates that the training performance will less likely to be affected by the injected Gaussian noise if $||\mathcal{N}(0, \sigma^2 S^2)||_u \leq \min_{j \neq y} \frac{g_y(x; w(t)) - g_{j, j \neq y}(x; w(t))}{L_v}$.

Given that it is computationally expensive to find such a noise bound in practice, especially for deep learning algorithms with large number of classes. One approach is to translate loss distance between the label class and the second-largest prediction class into the $L2$ magnitude of the gradients and ensure the bound over such $L2$ norm of local gradient updates.

Figure 3 illustrates an example of the decaying magnitude of $L2$ norm of the gradient during training. Mean $L2$ norm of 100 MNIST clients at one local iteration is reported. The decaying magnitude of gradient's $L2$ norm indicates that the loss value between the predicted label and ground-truth label decreases over time, implying a decaying confidence gap between the label class $g_y(x; w(t))$ and the class with the second-largest confidence. This is aligned with the observation that gradients in early iterations are more informative and tend to have a larger $L2$-norm value [8]. This is consistent with the literature on the need for fine-tuning of clipping bound used in a

803

	MNIST	CIFAR10	LFW	adult	cancer
# training data	60000	50000	2267	36631	426
# validation data	10000	10000	756	12211	143
# features	28*28	32*32*3	32*32*3	105	30
# classes	10	10	62	2	2
# data/client	500	400	300	300	400
# local iteration L	100	100	100	100	100
local batch size B	5	4	3	3	4
# rounds T	100	100	60	10	3
non-private acc	0.9798	0.674	0.695	0.8424	0.993
non-private cost(ms)	6.8	32.5	30.9	5.1	4.9

TABLE I: Benchmark datasets and parameters

differentially private algorithm [14] to bound the distortion noise using a proper clipping bound with constant clipping method.

Fed-CDP (decay). Motivated by the utility-privacy trade-off analysis, we consider to optimize Fed-CDP by using a dynamic decaying clipping method with the initial clipping bound C in order to keep track of the decaying $L2$ sensitivity of gradients as federated learning progresses in the number of rounds, ensuring the differential privacy based distortion noise being bounded properly by following the proposition 2. It is important to note that decaying clipping bound will not modify the privacy composition. Also the noise variance is determined by both the sensitivity S (thus the the clipping bound C in Fed-CDP and most existing DP algorithms) and the noise scale σ. Using a dynamic clipping method with decay function on bound C will result in decreasing trend of noise variance over the T rounds of federated learning, resulting in non-uniform distribution of privacy spending over the T rounds according to Lemma 1.

Empirical study in Section VII shows that by decaying clipping bound to track the decaying noise scale, Fed-CDP (decay) further optimizes Fed-CDP with improved accuracy performance and higher resilience to gradient leakage attacks with the same privacy budget spending, achieving better utility-privacy tradeoff, compared to Fed-CDP baseline.

VII. Experimental Evaluation

We evaluate Fed-CDP on five benchmark datasets as shown in Table I, with each dataset given the parameter setup, the validation accuracy and training cost for the non-private federated learning model. MNIST is a grey-scale hand-written digits dataset consisting of 60,000 training data and 10,000 test data. The 5:1 train-validation ratio is applied to the 60,000 training images. We partition the 50,000 training data into shards. Each client gets two shards with 500 samples from two classes. For CIFAR10, the 4:1 train-validation ratio is applied to the 50,000 training images. Each client has 400 images from two classes. Labeled Faces in the Wild(LFW) dataset has 13233 images from 5749 classes. The original image size is 250×250 and we crop it into 32×32 size to extract the 'interesting' region. Since most of the classes have a very limited number of data points, we consider the 62 classes that have more than 20 images per class. For a total number of 3023 eligible LFW data, a 3:1 train-validation ratio is applied. We partition the training data into shards and each client has 300 images from about 15 classes. We evaluate the three benchmark image

datasets on a multi-layer convolutional neural network with two convolutional layers and one fully-connected layer. For the two attribute datasets, the 3:1 train-validation ratio is applied to both. **Adult income** dataset has 48842 total training data with 105 converted numerical features, and each client holds 300 data from two classes randomly sampled from the training data. **Breast cancer** dataset has 569 training data with 30 numerical features, and each client has a full copy of the dataset due to its small number of data points. A fully-connected model with two hidden layers is used for attribute datasets.

Our federated learning setup follows the simulator in [13] with the total K clients varying from 100, 1000 to 10,000, and K_t is set at a varying percentage of K, such as $5\%, 10\%, 20\%, 50\%$ per round. All experiments are conducted on an Intel 4 core i5-7200U CPU@2.50GHz machine with an Nvidia Geforce 2080Ti GPU. We evaluate Fed-CDP in terms of three sets of metrics: (i) training performance in validation accuracy and time cost of conducting one local iteration at a client (ms/iteration), (ii) the privacy budget spending ϵ with varying settings of other differential privacy parameters, such as clipping bound C and noise scale σ; and (iii) the resilience against all three types of gradient leakage attacks in terms of attack success (Yes or No), #attack iterations to succeed with the default set to 300, and attack reconstruction distance, defined by the root mean square deviation between reconstructed input x_{rec} and its private ground-truth counterpart x: $\frac{1}{A}\sum_{i=1}^{A}(x(i) - x_{rec}(i))^2$. A denotes the total number of features in the input.

A. Training performance Evaluation

Fed-SDP v.s. Fed-CDP. The performance of Fed-CDP is evaluated and compared with Fed-SDP by the default parameter setting of MNIST in Table I. Table II shows the results by varying total K clients from 100 to 1000 and 10000 with 5%, 10%, 20% and 50% participating clients at each round. We make two observations: (1) Both Fed-CDP and Fed-SDP can achieve high accuracy comparable to that of the non-private setting. (2) Fed-CDP consistently achieves higher accuracy than Fed-SDP for all settings of K and K_t while Fed-CDP(decay) consistently improves Fed-CDP with slightly better accuracy performance. For Fed-CDP(decay), we linearly decay the clipping bound from C=6 to C=2 in 100 rounds and achieve high validation accuracy of 0.9748, on par with the non-private FL setting on MNIST. We will show later that decay-clipping optimized Fed-CDP also demonstrates higher resiliency against gradient leakage attacks.

Table III shows the per local iteration per client training time comparison. Given all K_t clients can perform local training in parallel, the overall time spent for one round is determined by the slowest client local training at the round t. Although Fed-CDP incurs additional computation overhead to clip and sanitize per-example gradients in each local batch (iteration), compared to Fed-SDP, the absolute time cost remains relatively low. The decay of clipping bound brings a negligible cost compared to Fed-CDP.

Kt/K	K=100				K=1000				K=10000			
	5%	10%	20%	50%	5%	10%	20%	50%	5%	10%	20%	50%
non-private	0.924	0.954	0.959	**0.965**	0.977	**0.980**	0.978	0.978	0.979	**0.980**	**0.980**	0.979
Fed-SDP	0.803	0.823	0.834	**0.872**	0.925	0.928	0.934	**0.937**	0.935	0.939	0.941	**0.944**
Fed-CDP	0.815	0.831	0.858	**0.903**	0.951	0.956	0.96	**0.964**	0.966	0.963	**0.968**	0.966
Fed-CDP(decay)	0.833	0.842	0.866	**0.909**	0.968	0.975	**0.977**	0.976	0.974	0.978	0.979	**0.980**

TABLE II: Accuracy comparison by varying # total clients and Kt/K in MNIST with default $C = 4$ and $\sigma = 6$

	MNIST	CIFAR-10	LFW	adult	cancer
non-private	6.8	32.5	30.9	5.1	5.1
Fed-SDP	6.9	33.8	31.3	5.2	5.1
Fed-CDP	22.4	131.5	112.4	11.8	11.9
Fed-CDP(decay)	22.6	132.1	114.6	12.1	12.0

TABLE III: Time cost per local iteration per client(ms)

	C=0.5	C=1	C=2	C=4	C=6	C=8
MNIST	0.914	0.934	0.943	**0.949**	0.933	0.923
CIFAR-10	0.408	0.568	0.602	**0.633**	0.624	0.611
LFW	0.582	0.594	0.619	**0.649**	0.627	0.601
adult	0.81	0.822	**0.825**	0.824	0.807	0.796
cancer	0.965	0.972	**0.979**	**0.979**	0.972	0.972

TABLE IV: Accuracy of Fed-CDP by varying clipping with default $\sigma = 6$

	$\sigma=0.5$	$\sigma=1$	$\sigma=2$	$\sigma=4$	$\sigma=6$	$\sigma=8$
MNIST	0.956	0.954	0.952	0.951	0.949	0.934
CIFAR-10	0.646	0.641	0.639	0.634	0.633	0.612
LFW	0.683	0.678	0.672	0.667	0.649	0.646
adult	0.838	0.837	0.836	0.834	0.824	0.822
cancer	0.993	0.993	0.993	0.993	0.979	0.979

TABLE V: Accuracy of Fed-CDP by varying noise scale with default $C = 4$

		MNIST	CIFAR-10	LFW	adult	cancer
instance	Fed-CDP: 1	0.0845	0.0845	0.0689	0.0494	0.0467
	Fed-CDP: 100	0.8227	0.8227	0.6356	0.2761	0.1469
	Fed-SDP: 1	not supported				
	Fed-SDP: 100					
client	Fed-CDP: 1	0.0845	0.0845	0.0689	0.0494	0.0467
	Fed-SDP: 1	0.8536	0.8536	0.6677	0.3025	0.2065
	Fed-CDP: 100	0.8227	0.8227	0.6356	0.2761	0.1469
	Fed-SDP: 100	0.8536	0.8536	0.6677	0.3025	0.2065

TABLE VI: Privacy composition of Fed-SDP and Fed-CDP using ϵ with $\delta = 1e-5$, $L = 1$ and $L = 100$ local iterations for per-client local training. ϵ measured at 100, 100, 60, 10, 3 rounds for MNIST, CIFAR10, LFW, adult, and cancer dataset respectively.

Fed-CDP performance with varying parameter settings. We first study the effect of varying clipping bound from 0.5 to 8. Table IV shows the results of Fed-CDP. By combining gradient clipping and noise injection to per-example gradients, Fed-CDP leads to high accuracy on MNIST, Adult and Cancer when $C = 2$ or $C = 4$ and high accuracy on LFW and CIFAR-10 when $C = 4$ or $C = 6$, because for different datasets, $L2$ norm of the gradients varies and results in different best clipping bound settings. Since the clipping bound will impact both the gradient information preservation and noise variance, the highest accuracy appears only with an appropriately set clipping bound and the performance will downgrade if the clipping bound is too large or too small. We next evaluate the impact of varying noise scale of $0.5, 1, 2, 4, 6, 8$. Table V shows the results given the clipping bound $C = 4$. Noise scale controls the amount of Gaussian noise added to the per-example gradients. The results confirm that adding too much noise will impact negatively the training performance. In Fed-CDP, we set $\sigma = 6$ by default to ensure the requirement of $q < \frac{1}{16\sigma}$ for moments accountant, even though a smaller noise scale may bring improvement on the accuracy performance for all datasets.

B. Privacy Evaluation

We evaluate and compare the privacy composition of Fed-CDP and Fed-SDP using moments accountant (recall Definition 5 and Section V) on all 5 datasets with two settings on the per-client location training iterations: $L = 1$ and $L = 100$. Table VI summarizes the results.

Privacy at instance level. Since privacy composition is only related with the sampling rate and the total number of rounds (T) under a given δ by Equation 2, ϵ for different dataset depends mainly on the total number of rounds since we set the global sampling rate for all datasets to 0.01. The training model for MNIST and CIFAR-10 under $L = 100$ is $(0.8227, 1e-5)$-differential private with $\epsilon = 0.8227$ and $\delta = 1e-5$. For LFW, adult, and cancer datasets, we have $\epsilon = 0.6356$, $\epsilon = 0.2761$, and $\epsilon = 0.1469$ respectively. All 5 datasets have the same sampling rate of 0.01, and their total training rounds T set to 100, 100, 60, 10, and 3 for MNIST, CIFAR-10, LFW, adult, and cancer respectively. Note that although we report the instance-level privacy by round since we will only get a new global model for each round, the actual ϵ value accumulates with the setting of the # of local iterations (L). For $L = 1$, clients performing local model training over one local iteration per round, which will generate a differentially private model with a much smaller privacy budget spending ϵ compared to the ϵ from the differentially private model generated over clients with local training on 100 local iterations at the same round. However, local training with $L = 1$ takes more rounds to achieve the same accuracy as the local training with $L = 100$ iterations.

Privacy at client level. According to the Billboard Lemma(lemma 2), with $\delta = 1e-5$, the trained model with Fed-CDP after the T rounds will accumulate the privacy spending $\epsilon = 0.8227, 0.8227, 0.6356, 0.2761, 0.1469$ for where for MNIST, CIFAR-10, LFW, adult and cancer respectively where $T = 100, 100, 60, 10, 3$ and $L = 100$ under joint differential privacy. Privacy at client-level for Fed-CDP benefits from the instance-level privacy for Fed-CDP. For Fed-SDP, the client-level differential privacy is computed with the sampling rate $q2 = \frac{K_t}{K}$ [12], [13] at each round. Since Fed-SDP adds noise directly to the per-client local gradient updates and whether the per-client local training is performed over one local iteration or 100 local iterations will not affect its privacy

		type 0&1				type 2			
		non-private	Fed-SDP	Fed-CDP	Fed-CDP (decay)	non-private	Fed-SDP	Fed-CDP	Fed-CDP (decay)
MNIST	succeed	Y	N	N	N	Y	Y	N	N
	reconstruction distance	0.1549	0.6991	0.7695	0.937	0.0008	0.0008	0.739	0.943
	attack iteration	6	300	300	300	7	7	300	300
LFW	succeed	Y	N	N	N	Y	Y	N	N
	reconstruction distance	0.2214	0.7352	0.8036	0.941	0.0014	0.0014	0.6626	0.945
	attack iteration	24	300	300	300	25	25	300	300

TABLE VII: Average attack effectiveness comparison from 100 clients, with the maximum attack iteration set to 300.

accounting, as shown in Table VI.

C. Gradient-Leakage Resiliency

We measure the resiliency of Fed-CDP and Fed-SDP against gradient leakage attacks using MNIST and LFW dataset. The gradient leakage attack experiment is performed on gradients from the first local iteration as gradients at early training iterations tend to leak more information than gradients in the later stage of the training [8]. Table VII reports the resilience results in terms of attack success (yes or no), the average reconstruction distance, and attack iterations from 100 clients. For failed attacks, we run the attack reconstruction and record the attack reconstruction distance at the maximum attack iterations of 300. We observe that non-private federated learning is vulnerable to all three types of gradient leakage attacks, while Fed-CDP can effectively mitigate all three types of leakages and Fed-SDP is vulnerable to the type-2 leakage though it is more resilient to the type-0 and type-1 leakage compared to non-private FL. Compared to Fed-SDP, Fed-CDP has a larger reconstruction distance, which implies more robust masking of the private training data against gradient leakage attacks. Figure 4 shows a visualization of Fed-CDP and Fed-SDP under three types of gradient leakages using an example batch of LFW dataset in the local training, showing Fed-CDP(decay) offers the highest resilience against all three gradient leakages, followed by Fed-CDP, while Fed-SDP is vulnerable to type-2 leakage and non-private FL and Distributed Selective SGD(DSSGD [25]) is vulnerable to all three types of gradient leakages.

Figure 5 provides a visualization comparison of Fed-CDP(decay), Fed-CDP, and Fed-SDP under type-2 gradient leakage attack in the communication-efficient federated learning using MNIST dataset. Validation accuracy is also reported at a total of 1000 clients and 100 participants in the figure. In this set of experiments, we consider communication-efficient federated learning by pruning the insignificant gradients, which make little contribution to joint training, i.e., gradients with very small values [8]. Using non-private training or Fed-SDP, the gradients shared via communication-efficient protocol (by compression) may still suffer from the gradient leakage threats for compression ratios up to 30% on MNIST. In contrast, Fed-CDP is more resilient than Fed-SDP, while Fed-CDP(decay) provides the highest resilience in all settings against the type-2 leakages in the communication-efficient federated learning protocols.
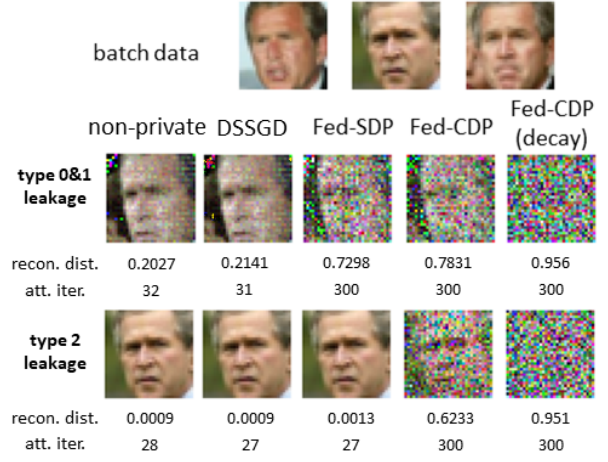

Fig. 4: Visualization of gradient leakage attacks under different Fed-DP modules. Gradients of one example in the batch is used to do type-2 attack.
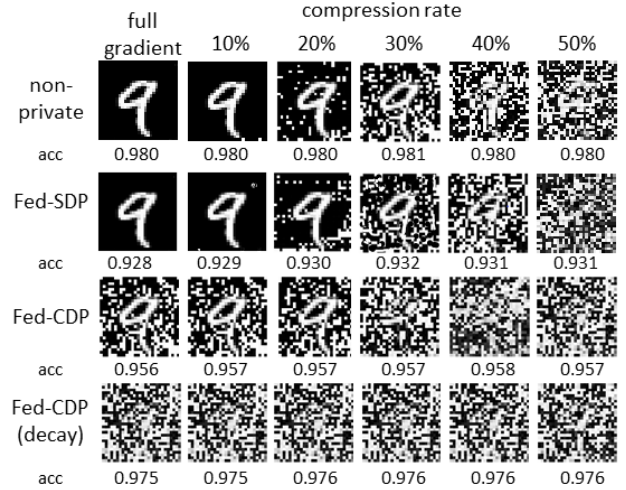

Fig. 5: Accuracy and resiliency to type-2 leakage in communication efficient federated learning

VIII. CONCLUSION

We have presented a per example based client different privacy approach to federated learning with three original contributions. We first identified three types of gradient leakage attacks and explained why the existing Fed-SDP approach is vulnerable. We then introduce Fed-CDP, the per example-based client differential privacy algorithm, with a

Authorized licensed use limited to: NANJING NORMAL UNIVERSITY. Downloaded on November 07,2022 at 06:38:26 UTC from IEEE Xplore. Restrictions apply.

formal comparison between Fed-CDP and Fed-SDP in terms of privacy accounting. Finally, we provide a formal analysis of the privacy-utility trade-off for providing differential privacy guarantee by Fed-CDP and present Fed-CDP(decay), which further improves the accuracy and resiliency of Fed-CDP by incorporating a dynamic decay clipping method, instead of constant clipping method popularly used in the literature [12], [14]. With extensive evaluation, we show that the Fed-CDP approach outperforms Fed-SDP with high resilience against all three types of gradient leakage attacks while offering competitive accuracy performance.

REFERENCES

[1] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, "Decentralized collaborative learning of personalized models over networks," in *AISTATS*, 2017, pp. 509–517.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017, pp. 1273–1282.

[3] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NeurIPS Workshop on Private Multi-Party ML*, 2016.

[4] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, and H. B. McMahan, "Towards federated learning at scale: System design," in *SysML*, 2018, pp. 619–633.

[5] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[6] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *NeurIPS*, 2019, pp. 14 747–14 756.

[7] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" in *NeurIPS*, 2020, pp. 16 937–16 947.

[8] Wenqi Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, and Y. Wu, "A framework for evaluating client privacy leakages in federated learning," in *ESORICS*. Springer, 2020, pp. 545–566.

[9] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," in *IEEE S&P*, 2019, pp. 739–753.

[10] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and Wenqi Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE TSC*, 2019.

[11] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE S&P*, 2017, pp. 3–18.

[12] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *ICLR*, 2018.

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

[14] 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, pp. 308–318.

[15] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *IEEE S&P*, 2019, pp. 691–706.

[16] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *ACM CCS*, 2017, pp. 587–601.

[17] Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE TIFS*, vol. 13, no. 5, pp. 1333–1345, 2017.

[18] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *ACM CCS*, 2015, pp. 1322–1333.

[19] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *ACM CCS*, 2017, pp. 603–618.

[20] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM*, 2019, pp. 2512–2520.

[21] E. Shi, T. H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *NDSS*, 2011, pp. 1–17.

[22] 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, pp. 1175–1191.

[23] M. Chase, R. Gilad-Bachrach, K. Laine, K. E. Lauter, and P. Rindal, "Private collaborative neural network learning." *IACR Cryptology ePrint Archive*, 2017.

[24] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with pate," in *ICLR*, 2018.

[25] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *ACM CCS*, 2015, pp. 1310–1321.

[26] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Technical report*, 2008.

[27] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits, 1998," *URL http://yann. lecun. com/exdb/mnist*, vol. 10, p. 34, 1998.

[28] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Technical report*, 2009.

[29] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[30] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled rényi differential privacy and analytical moments accountant," in *AISTATS*, 2019, pp. 1226–1235.

[31] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *ICLR*, 2018.

[32] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE TPDS*, 2020.

[33] X. Yao, T. Huang, R.-X. Zhang, R. Li, and L. Sun, "Federated learning with unbiased gradient aggregation and controllable meta updating," in *NeurIPS Workshop on FL for Data Privacy and Confidentiality*, 2019.

[34] M. Kamp, L. Adilova, J. Sicking, F. Hüger, P. Schlicht, T. Wirtz, and S. Wrobel, "Efficient decentralized deep learning by dynamic model averaging," in *ECML-PKDD*. Springer, 2018, pp. 393–409.

[35] C. Ma, V. Smith, M. Jaggi, M. I. Jordan, P. Richtárik, and M. Takáč, "Adding vs. averaging in distributed primal-dual optimization," in *ICML*, 2015, pp. 1973–1982.

[36] Y. Wu, L. Liu, J. Bae, K.-H. Chow, A. Iyengar, C. Pu, Wenqi Wei, L. Yu, and Q. Zhang, "Demystifying learning rate policies for high accuracy training of deep neural networks," in *IEEE Intl. Conf. on Big Data*, 2019, pp. 1971–1980.

[37] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *IEEE S&P*, 2019, pp. 332–349.

[38] M. Kearns, M. Pai, A. Roth, and J. Ullman, "Mechanism design in large games: Incentives and privacy," in *ITCS*, 2014, pp. 403–410.

[39] Z. S. Wu, "Data privacy beyond differential privacy," *Technical report*, 2017.

[40] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and Wenqi Wei, "Ldp-fed: federated learning with local differential privacy," in *ACM EdgeSys*, 2020, pp. 61–66.

[41] T. Li and N. Li, "On the tradeoff between privacy and utility in data publishing," in *ACM SIGKDD*, 2009, pp. 517–526.

[42] Y. Wang, Y. O. Basciftci, and P. Ishwar, "Privacy-utility tradeoffs under constrained data release mechanisms," *arXiv preprint arXiv:1710.09295*, 2017.