FLAME: Differentially Private Federated Learning in the Shuffle Model

Ruixuan Liu[1], **Yang Cao**[2*], **Hong Chen**[1*], **Ruoyang Guo**[1], **Masatoshi Yoshikawa**[2]
[1]Renmin University of China
[2]Kyoto University
ruixuan.liu@ruc.edu.cn, yang@i.kyoto-u.ac.jp, chong@ruc.edu.cn, ryguo@ruc.edu.cn, yoshikawa@i.kyoto-u.ac.jp

Abstract

Federated Learning (FL) is a promising machine learning paradigm that enables the analyzer to train a model without collecting users' raw data. To ensure users' privacy, differentially private federated learning has been intensively studied. The existing works are mainly based on the *curator model* or *local model* of differential privacy. However, both of them have pros and cons. The curator model allows greater accuracy but requires a trusted analyzer. In the local model where users randomize local data before sending them to the analyzer, a trusted analyzer is not required but the accuracy is limited. In this work, by leveraging the *privacy amplification* effect in the recently proposed shuffle model of differential privacy, we achieve the best of two worlds, i.e., accuracy in the curator model and strong privacy without relying on any trusted party. We first propose an FL framework in the shuffle model and a simple protocol (SS-Simple) extended from existing work. We find that SS-Simple only provides an insufficient privacy amplification effect in FL since the dimension of the model parameter is quite large. To solve this challenge, we propose an enhanced protocol (SS-Double) to increase the privacy amplification effect by subsampling. Furthermore, for boosting the utility when the model size is greater than the user population, we propose an advanced protocol (SS-Topk) with gradient sparsification techniques. We also provide theoretical analysis and numerical evaluations of the privacy amplification of the proposed protocols. Experiments on real-world dataset validate that SS-Topk improves the testing accuracy by 60.7% than the local model based FL. We highlight an observation that SS-Topk improves the accuracy by 33.94% than the curator model based FL without any trusted party. Compared with non-private FL, our protocol SS-Topk only lose 1.48% accuracy under $(2.348, 5e^{-6})$-DP per epoch.

Introduction

Federated Learning (FL) (McMahan et al. 2016) is a promising machine learning paradigm that enables the analyzer to train a central model by collecting users' local updates instead of the raw data. However, it has been shown that sharing raw local updates compromises users' privacy (Nasr, Shokri, and Houmansadr 2019; Hitaj, Ateniese, and Perez-Cruz 2017; Zhu, Liu, and Han 2019). To this end, *differentially private federated learning* has been wildly studied to provide formal privacy. The existing works are mainly based on the *curator model* (DP) or *local model* (LDP) of differential privacy. The curator model based FL (DP-FL) (McMahan et al. 2018; Geyer, Klein, and Nabi 2017) allows better learning accuracy but relies on a trusted analyzer to collect raw local updates. The local model based FL (LDP-FL) (Wang et al. 2019; Liu et al. 2020) preserves strong local privacy since the users randomize local updates before sending them to an untrusted analyzer; but it suffers a low utility. Specifically, for a basic bit summation task over n users with privacy budget ϵ[1], the error of DP can achieve $O(1/\epsilon)$; whereas, the error of LDP is bounded by $\Omega(\sqrt{n})$ (Chan, Shi, and Song 2012).

The recently proposed secure shuffle model (SS) can achieve the best of two worlds, i.e., accuracy in the curator model and strong privacy in the local model. The shuffle model introduces a shuffler (sitting between users and the analyzer, as shown in Figure 1(b)) to permute the locally randomized data before users transmitting them. The accuracy gain of the shuffle model is obtained from the *privacy amplification* effect (Erlingsson et al. 2019), which indicates that the shuffled (i.e., anonymized) outputs of local randomizers provide a stronger (amplified) privacy in the central view of differential privacy than the one without a shuffler. Accordingly, less local noise is needed in the shuffle model for the same level of privacy against the untrusted analyzer.

However, it is not clear how to employ the shuffle model in federated learning. Although a few works have investigated basic tasks such as the bit/real summation (Balle et al. 2019b) and histogram (Balcer and Cheu 2019), the existing protocols may not be viable for the multi-dimensional aggregation FL. This is because the error brought by local noises can be aggravated with a dimensional factor (Duchi, Jordan, and Wainwright 2013). Moreover, since the number of users participating in one iteration is typically a few thousands, the aggregation escalates into a high-dimensional task. We solve the above challenges by making the following contributions:

- For the first time, we propose FLAME, a federated learning framework in the shuffle model such that the users enjoy strong privacy and the analyzer enjoys the accuracy of the model. We first formalize our privacy goal in FLAME by clarifying the trust boundary and fine-grained trust sep-

*Corresponding authors: Hong Chen, Yang Cao

[1]A larger privacy budget leads to better utility and less privacy.

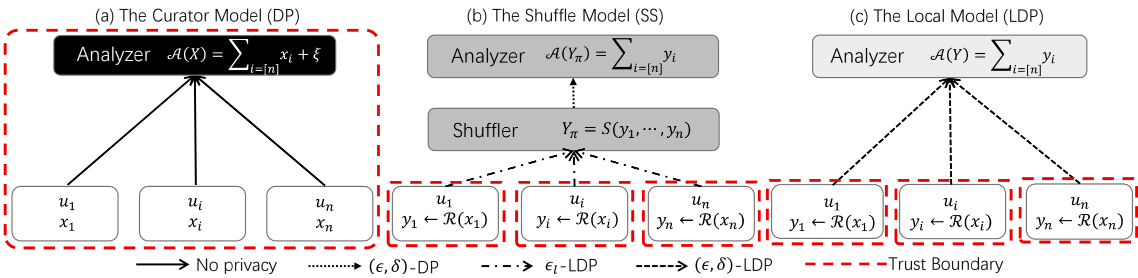


Figure 1: Trust boundaries in the curator, shuffle and local model of differential privacy. The curator model relies on a trusted analyzer. Users in the local model or the shuffle model do not need to trust any external parties and thus preserve strong privacy.

aration (Table 1), then we propose SS-Simple protocol by extending a one-dimensional task (Balle et al. 2019b). We find that, although the privacy amplification is achievable by SS-Simple, the magnitude of amplification diminishes with the dimension size of the local updates (Corollary 2).

- To alleviate this challenge, we propose SS-Double protocol to enhance the privacy amplification by subsampling. As we notice that the amplification by subsampling may not be composable with shuffling, we propose a novel *dummy padding* method to bridge two kinds of amplification effects with formal proofs (Theorems 3 and 4). We demonstrate that the SS-Double enjoys **dozens** times privacy amplification comparing with SS-Simple (Figure 3).
- A problem of SS-Double protocol is that the random subsampling treats all dimensions equally and thus may discard "important" dimensions. To further boost the utility in a high-dimensional case, we design an advanced protocol called SS-Topk, which is based on the idea of gradient sparsification. A challenge is that the indexes of Top-k elements in the local update vector may reveal sensitive information to the shuffler since the selection is data-dependent. We quantify this privacy threat by formalizing *index privacy* and design a method to flexibly trade off between index privacy and utility. We note that the index privacy will not impair the privacy against the analyzer.
- Finally, we conduct experiments on the real-world dataset to validate the effectiveness of the proposed protocols. It turns out that the proposed double amplification effect in SS-Double and private dimension selection in SS-Topk significantly improve the learning accuracy. We observe a 33.94% accuracy improvement of SS-Topk than the curator model based FL without relying on any trusted party. Compared with non-private FL, SS-Topk only lose 1.48% accuracy under $(2.348, 5e^{-6})$-DP per epoch.

Preliminaries

We first clarify the trust boundaries in Figure 1 and then introduce properties that will be used throughout the paper and formally define our problem.

The Curator and Local Models

In the curator model, a trusted analyzer collects users' raw data (e.g local updates) and executes a private mechanism for differentially private outputs. The privacy goal is to achieve indistinguishability for any outputs w.r.t. two neighboring datasets which differ by replacing one user's data, denoted as $X \simeq_r X'$. We have the following definition:

Definition 1 *[Differential Privacy (DP)] A mechanism $\mathcal{M}$: $\mathbb{X}^n \to \mathbb{Z}$ satisfies (ϵ, δ)-differentially privacy if for any two neighboring datasets $X \simeq_r X' \in \mathbb{X}^n$ and any subsets $S \subseteq \mathbb{Z}$, $\Pr[\mathcal{M}(X) \in S] \leq e^\epsilon \Pr[\mathcal{M}(X') \in S] + \delta$.*

However, the curator model assumes the availability of a trusted analyzer to collect raw data. Local differential privacy in Definition 2 does not rely on any trusted party because users send randomized data to the server. If $\mathcal{R}$ satisfies (ϵ, δ)-LDP, observing collected results $(y_1, \cdots, y_n)$ or the summation implies (ϵ, δ)-DP (Dwork, Roth et al. 2014).

Definition 2 *[Local Differential Privacy (LDP)] A mechanism $\mathcal{R} : \mathbb{X} \to \mathbb{Y}$ satisfies (ϵ, δ)-locally differentially private if for any two inputs $x, x' \in \mathbb{X}$ and any output $y \in \mathbb{Y}$, $\Pr[\mathcal{R}(x) = y] \leq e^\epsilon \Pr[\mathcal{R}(x') = y] + \delta$.*

The Shuffle Model

The protocol of a shuffle model consists of three components: $\mathcal{P} = \mathcal{A} \circ \mathcal{S} \circ \mathcal{R}^n$, as shown in Figure 1(b). Existing works (Balle et al. 2019a; Balcer and Cheu 2019; Cheu et al. 2019; Ghazi et al. 2019, 2020) focus on the basic task where each user holds a one-dimensional data $x \in \mathbb{X}$. We denote n users' data as the dataset $X = (x_1, \cdots, x_n) \in \mathbb{X}^n$. Each user runs a randomizer $\mathcal{R} : \mathbb{X} \to \mathbb{Y}^m$ to perturb the local data into m messages that satisfy ϵ_l-LDP. W.o.l.g. we focus on the *single-message* protocol where $m = 1$. The shuffler executes $\mathcal{S} : \mathbb{Y}^* \to \mathbb{Y}^*$ with a uniformly random permutation π over received messages. The analyzing function $\mathcal{A} : \mathbb{Y}^* \to \mathbb{Z}$ takes the shuffled messages as input and outputs the analyzing result.

The privacy goal in shuffle model is to ensure $\mathcal{M} = \mathcal{S} \circ \mathcal{R}^n$ satisfies (ϵ_c, δ_c)-DP, because $\mathcal{A}$ is executed by an untrusted analyzer, who is not obliged to protect users' privacy. By the post-processing property (Dwork, Roth et al.

Algorithm 1 $\mathcal{R}_{\gamma,b} : [0,1] \rightarrow [b]$ (Balle et al. 2019b)

Input: input scalar $x \in [0,1]$
Output: perturbed value $y \in [b]$

$\bar{x} \leftarrow \lfloor xb \rfloor + \text{Ber}(xb - \lfloor xb \rfloor)$

Sample $r \leftarrow Ber(\gamma)$

$$y = \begin{cases} \bar{x} & \text{if } r = 0, \\ \text{Unif}(\{1, \cdots, b\}) & \text{else.} \end{cases}$$

2014), the protocol $\mathcal{P}$ achieves the same privacy level as $\mathcal{M}$. Hence, we focus on analyzing the indistinguishability for $\mathcal{M}(X)$ and $\mathcal{M}(X')$. (Erlingsson et al. 2019) proved that the privacy of $\mathcal{M}$ can be "amplified". In other words, when each user applies the local privacy budget ϵ_l in $\mathcal{R}$, $\mathcal{M}$ can achieve a stronger privacy of (ϵ_c, δ_c)-DP with $\epsilon_c < \epsilon_l$. Compared with the local model, the shuffle model needs less noise to achieve the same privacy level.

Among existing works, the privacy blanket (Balle et al. 2019b) provides an optimal amplification bound for the *single-message* protocol. The analyzing intuition is to linearly decompose the output distribution into a data-dependent distribution and a uniform random "privacy blanket" distribution. With the local randomizer $\mathcal{R}_{\gamma,b}$ in Algorithm 1, where $\gamma = \frac{b}{e^{\epsilon_l}+b-1}$ denotes the probability to output an element from the blanket distribution. the input value x is encoded into a discrete domain $[b]$ and then randomized. After $\mathcal{S}$ runs a permutation, $\mathcal{A}$ aggregates shuffled results with $\hat{z} \leftarrow \frac{1}{b}\sum_{i=i}^{n} y_i$ and de-bias with

$$z \leftarrow (\hat{z} - n\gamma/2)/(1-\gamma). \tag{1}$$

The privacy amplification bound for Algorithm 1 is shown in Lemma 1 and the effect for generic randomizer is distilled in Corollary 1. For a randomizer with Laplace Mechanism on the domain $[0,1]$, $\gamma = e^{-\epsilon_l/2}$. A tighter bound (i.e., a greater amplification) can be accessed with numerical evaluations.

Lemma 1 *(Balle et al. 2019b) For* $\sqrt{\frac{14\log(2/\delta_c)(b-1)}{n-1}} < \epsilon_c \leq 1$, *if* $\mathcal{R}_{\gamma,b}$ *satisfies* ϵ_l*-LDP, we have* (ϵ_c, δ_c) *for* $\mathcal{S} \circ \mathcal{R}^n$, *where* $\epsilon_c = \sqrt{\frac{14\log(2/\delta_c)(e^{\epsilon_l}+b-1)}{n-1}}$.

Corollary 1 *(Balle et al. 2019b) In the shuffle model, if* $\mathcal{R}$ *is* ϵ_l*-LDP, where* $\epsilon_l \leq \log(n/log(1/\delta_c))/2$. $\mathcal{M}$ *satisfies* (ϵ_c, δ_c)*-DP with:* $\epsilon_c = O((1 \wedge \epsilon_l)e^{\epsilon_l}\sqrt{\log(1/\delta_c)/n})$.

Composition and Subsampling Properties

The composition properties (Dwork, Rothblum, and Vadhan 2010) are generic for both the curator and the local model of differential privacy.

Lemma 2 $\forall \epsilon \geq 0, t \in \mathbb{N}$, *the family of* ϵ*-DP mechanism satisfies* $t\epsilon$*-DP under* t*-fold adaptive composition.*

Lemma 3 $\forall \epsilon, \delta, \delta' > 0, t \in \mathbb{N}$, *the family of* (ϵ, δ)*-DP mechanism satisfies* $(\sqrt{2t\ln(1/\delta')} \cdot \epsilon + t \cdot \epsilon(e^{\epsilon} - 1), t\delta + \delta')$*-DP under* t*-fold adaptive composition.*

A mechanism $\mathcal{K}$ that randomly subsamples m elements *without replacement* from a database with n records leads to a *privacy amplification* by subsampling in Lemma 4.

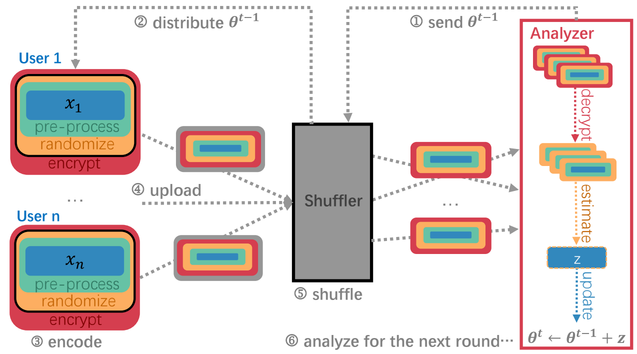


Figure 2: FLAME: Federated Learning in the Shuffle Model.

Lemma 4 *[Privacy Amplification by Subsampling] (Balle, Barthe, and Gaboardi 2018) If* $\mathcal{M} : \mathbb{X}^m \rightarrow \mathbb{Y}$ *satisfies* (ϵ, δ)*-DP with respect to the replacement relationship* $\simeq_r$ *on sets of size* m, $\mathcal{M}' : \mathbb{X}^n \rightarrow \mathbb{Y}$ *satisfies* $(\log(1 + (m/n)(e^{\epsilon} - 1)), (m/n)\delta)$*-DP.*

FLAME Framework

In this section, we formalize the framework of **F**ederated **Lea**rning in the Shuffle **M**od**e**l, which we call FLAME.

Parties. We present the FLAME architecture in Figure 2 with three parties: 1) n users, each of which owns a d-dimensional local update vector x_i and runs a local randomizer $\mathcal{R}$ for the output y_i. 2) The shuffler $\mathcal{S}$, the server who can perfectly shuffle received messages and send them to the analyzer. 3) The analyzer $\mathcal{A}$, the server that estimates the mean of shuffled messages into z and updates the global model at the round t by $\theta^t \leftarrow \theta^{t-1} + z$.

Trust Boundaries. We first clarify the trust boundary in FLAME. We denote the observer as $\mathcal{O}$ which could be any curious party who can observe the global model parameters. In the curator model (DP-FL), the trust boundary lies between $\mathcal{A}$ and $\mathcal{O}$. In the local model (LDP-FL), the trust boundary lies between each individual user and the rest parties. By introducing a shuffler $\mathcal{S}$, FLAME avoids placing full trust in any single party as DP-FL and meanwhile be able to achieve better utility than LDP-FL.

Trust Separation. Further, we clarify what are the private information and who can touch them in FLAME. We design a fine-grained scheme of trust separation for FLAME and compare with DP-FL and LDP-FL in Table 1. Specifically, we separate the information of each local update into: the indexes, corresponding values and the user identity (i.e., ID in Figure 2). It should be noted that the indexes could be sensitive when they are selected and sent to the shuffler $\mathcal{S}$ in a value-dependent manner. Thus, our privacy goal in FLAME is to select indexes in a data-oblivious way and the true values of gradients are invisible to $\mathcal{S}$. But $\mathcal{S}$ should know users' implicit identities in order to distribute global model parameters and receive local messages. The shuffled messages from $\mathcal{S}$ do not reveal the user identity and satisfy (ϵ_c, δ_c)-DP against $\mathcal{A}$. For $\mathcal{O}$, (ϵ_c, δ_c)-DP holds by the post-

Parties	Shuffler S			Analyzer A			Observer $\mathcal{O}$
Sensitive info	gradient vector		ID	gradient vector		ID	query model
	index	value		index	value		
DP-FL	N/A			$\surd$		$\surd$	(ϵ_c, δ_c)-DP
LDP-FL				(ϵ_c, δ_c)-LDP		$\surd$	
FLAME	$\times$	$\times$	$\surd$	(ϵ_c, δ_c)-DP		$\times$	

$\surd$: trusted, $\times$: untrusted.

Table 1: Separation of Trust between Shuffler and Analyzer.

processing property (Dwork, Roth et al. 2014). In LDP-FL, a (ϵ_c, δ_c)-LDP $\mathcal{R}$ is required by each user for this goal.

FLAME framework. We present our framework in Algorithm 2 with three building processes: encoding $\mathcal{E}$, shuffling $\mathcal{S}$ and analyzing $\mathcal{A}$. In Line 8, C is the clipping threshold for the vector. We denotes the local privacy budget for each local vector by ϵ_l. pk_a and sk_a represent the public and secret key generated by the analyzer, respectively. Different protocols below are designed by implementing functions $\mathsf{Randomize}(\cdot)$ in Line 10 and $\mathsf{Shuffle}(\cdot)$ in Line 15 with different strategies. It should be noted that $\mathcal{R}_{\gamma,b}$ in Algorithm 1 can be applied in $\mathsf{Randomize}(\cdot)$ as a basic randomizer, which accords to estimating with equation (1) for line (18). Generic randomizers (e.g. Laplace Mechanism) can also be applied, which does not affect our corollaries later.

Security Assumptions. We assume that the shuffler and the analyzer are not colluded (otherwise, FLAME is reduced to LDP-FL). We also assume that the cryptographic primitives are safe and adversaries have computational difficulty to learn any information from the cipher text.

Simple Protocol (SS-Simple). We first propose SS-Simple $\mathcal{P} = \mathcal{A} \circ \mathcal{S} \circ \mathcal{R}^n$ for d-dimensional aggregation under the FLAME framework. In a nutshell, we extend the one-dimensional protocol (Balle et al. 2019b) by conducting its randomization and aggregation for each dimension. With the composition property in Lemma 2, $\mathcal{R}$ should satisfy ϵ_{ld}-LDP, where $\epsilon_{ld} = \epsilon_l/d$. We instantiate $\mathsf{Randomize}(\cdot)$ in Algorithm 2 for SS-Simple with $idx_i \leftarrow \{1, \cdots, d\}$ and $y_i \leftarrow \{\mathcal{R}_{\epsilon_{ld}}(x_{i,1}), \cdots, \mathcal{R}_{\epsilon_{ld}}(x_{i,d})\}$. The function $\mathsf{Shuffle}(\cdot)$ simply generates a permutation π and outputs $m_{\pi(i)\in[n]}$.

Then we account for the central DP against the analyzer. Take ϵ_{ld} to Lemma 1 for $\mathcal{R}_{\gamma,b}$ or other numerical evaluations for generic $\mathcal{R}$, an amplified central privacy $(\epsilon_{cd}, \delta_{cd})$ can be derived. With the composition in Lemma 3, we can easily derive the vector-level composition in Theorem 1. Corollary 2 distills the amplification from ϵ_l to ϵ_c.

Theorem 1 *For any neighboring datasets $X \simeq_r X'$ which differ in one user's d-dimensional local vector, $\mathcal{M} = \mathcal{S} \circ \mathcal{R}^n$ in SS-Simple satisfies (ϵ_c, δ_c)-DP, where:*

$$\epsilon_c = d\epsilon_{cd} \wedge (\epsilon_{cd}\sqrt{2d\log(1/\delta_{cd})} + d\epsilon_{cd}(e^{\epsilon_{cd}} - 1)), \quad (2)$$
$$\delta_c = \delta_{cd}(d + 1). \quad (3)$$

Corollary 2 *For SS-Simple, with $\epsilon_l \leq d \cdot \log(n/\log((d+1)/\delta_c))/2$, the amplified central privacy is $\epsilon_c = O((1 \wedge \epsilon_l/d)e^{\epsilon_l/d}\log(d/\delta_c)\sqrt{d/n})$.*

Algorithm 2 FLAME: Encoding, Shuffling, Analyzing.

Input: $\mathcal{A}, \mathcal{S}, \mathcal{E}, n, T, \epsilon_l, pk_a, sk_a$
Output: θ^T
1: Analyzer publishes pk_a
2: **for** each iteration $t = 1, \cdots, T$ **do**
3: Analyzer sends θ^{t-1} to Shuffler
4: Shuffler distributes θ^{t-1} to a batch of n users
5: **for** each user $i \in [n]$ **do**
6: $x_i \leftarrow$ LocalUpdate(θ^{t-1})
7: $\triangleright$ `Encoding` $\mathcal{E}$ `by each user`
8: $\bar{x}_i \leftarrow$ Clip$(x_i, -C, C)$
9: $\tilde{x}_i \leftarrow (\bar{x}_i + C)/(2C)$
10: $\langle idx_i, y_i\rangle \leftarrow \mathsf{Randomize}(\tilde{x}_i, \epsilon_l)$
11: $c_i \leftarrow \text{Enc}_{pk_a}(y_i)$
12: user i sends $m_i = \langle idx_i, c_i\rangle$ to Shuffler
13: **end for**
14: $\triangleright$ `Shuffling` $\mathcal{S}$ `by the Shuffler`
15: Shuffler sends $\mathsf{Shuffle}(m_{i\in[n]})$ to Analyzer
16: $\triangleright$ `Analyzing` $\mathcal{A}$ `by the Analyzer`
17: decrypts values $y_{\pi(i)\in[n]} \leftarrow \text{Dec}_{sk_a}(c_{\pi(i)\in[n]})$
18: estimates mean $\bar{z} \leftarrow \frac{1}{n}\sum_{i\in[n]}\langle idx_i, y_i\rangle$
19: normalizes $z \leftarrow C \cdot (2\bar{z} - 1)$
20: updates model $\theta^t \leftarrow \theta^{t-1} + z$.
21: **end for**

Limitation of SS-Simple. Observing the Corollary 2, we find that the central DP level depends on the dimension d. Intuitively, from the view of privacy, the amplification effect is diminished with a large d. From the view of utility, randomizing the value $y_{i,j}$ for each dimension with a negligible privacy budget $\epsilon_{ld} = \epsilon_l/d$ incurs large noises.

Double Amplification (SS-Double)

Intuition. For strengthening the privacy amplification effect, we propose an improved protocol SS-Double. Instead of perturbing every dimension with a small budget $\epsilon_{ld} = \epsilon_l/d$, we only sample and perturb k dimensions $k \ll d$. As a result, each dimension can benefit from a larger privacy budget $\epsilon_{lk} = \epsilon_l/k$. Furthermore, we notice that the privacy amplification can be further magnified by subsampling, which we call the *double amplification*. Intuitively, if the privacy amplification is strengthened, we could inject fewer noises under the same central privacy level.

Challenge. However, the privacy amplification by subsampling may not be composable with shuffling for the multi-dimensional vector. We first show how to compose the privacy amplification of shuffling and subsampling in

Algorithm 3 Shuffle$(\cdot)$=$\mathcal{S}_p$ for SS-Double

for $j \in [d]$ **do**
  $n_{s,j} \leftarrow \sum_{i=1}^{n} \mathbb{I}_{j \in idx_i}$
  $n_{n,j} \leftarrow n_p - n_{s,j}$
end for
the number of dummy vectors $v \leftarrow \sum_{j=1}^{d} n_{n,j}/k$
for $u \in [v]$ **do**
  $S_{dummy} \leftarrow \{j | j \in [d], n_{n,j} \neq 0\}$
  $idx_{n+u} \leftarrow \{j | j \in S_{dummy}\}^k$
  $n_{n,j} \leftarrow n_{n,j} - 1$ for $j \in idx_{n+u}$
  $y_{n+u} \leftarrow \{y^* | y^* \leftarrow \omega_{\mathcal{R}}\}^k$
  $m_{n+u} \leftarrow \langle idx_{n+u}, \text{Enc}_{pk_a}(y_{n+u}) \rangle$
end for
generates a permutation π over $[nl + v]$
shuffles and sends $\{m_{\pi(1)}, \cdots, m_{\pi(nl+v)}\}$ to analyzer

one dimension with $\mathcal{R}_{\gamma,b}$. Suppose $\mathcal{K}_r^{\beta}$ samples n_s users from n users with $\beta = n_s/n$. The shuffler only receives encoded message from sampled users. Applying Lemma 1 and Lemma 4, we derive Theorem 2. In addition, we should ensure $\delta_{cd} < 2\beta$ for a positive logarithm, which is reasonable to achieve since $\delta_{cd} \ll \frac{1}{dn}$ is negligible by standard.

Theorem 2 *With $\gamma = \frac{b}{e^{\epsilon_{ld}}+b-1}$ and $\delta_{cd} < 2\beta$, $\mathcal{M} = \mathcal{S} \circ \mathcal{R}_{\gamma,b}^{n_s} \circ \mathcal{K}_r^{\beta}$ satisfies $(\epsilon_{cd}, \delta_{cd})$-DP, where:*

$$\epsilon_{cd} = \log(1 + \beta(e^{\sqrt{\frac{14\log(\frac{2\beta}{\delta_{cd}})(e^{\epsilon_{lk}}+b-1)}{n_s-1}}} - 1)). \tag{4}$$

However, we cannot derive a similar theorem with $\mathcal{R}_{\gamma,b}$ in a multi-dimensional case. Intuitively, it is because the proof of privacy amplification by shuffling relies on bounded-size neighboring datasets, while subsampling may lead to two neighboring datasets with distinct size.

Dummy Padding. To solve the composition issue, we propose the method of *dummy padding*: let the shuffler pad each dimension into the same size of n_p. Denote the number of padding values for one dimension as $n_n = n_p - n_s$, and n_n elements from the blanket distribution $\omega_{\mathcal{R}}$ are shuffled with all other messages received from users. Thus, the SS-Double is $\mathcal{P} = \mathcal{A} \circ \mathcal{S}_p \circ \mathcal{R}^{n_p} \circ \mathcal{K}_r^{\beta}$, where $\mathcal{S}_p$ consists of padding and shuffling by the shuffler and $\mathcal{K}_r^{\beta}$ denotes randomly subsampling by each user. To instantiate SS-Double in our framework, the Randomize$(\cdot)$ works as follows. First, it randomly samples k indexes $idx_i \leftarrow \mathcal{K}_r^{\beta}(d)$. For each index $j \in idx_i$, the perturbed value in y_i is $y_{i,j} \leftarrow \mathcal{R}_{\epsilon_{ld}}(\tilde{x}_{i,j})$. The procedures of Shuffle$(\cdot)$ are listed in Algorithm 3.

Privacy and Utility Analysis. We show the overall privacy amplification bound for $\mathcal{R}_{\gamma,b}$ in Theorem 3. We notice that a larger n_p leads to a smaller ϵ_{cd}, which implies a greater privacy amplification. Correspondingly, a larger n_p implies more noises injected as shown in Proposition 1.

Theorem 3 *With $\gamma = \frac{b}{e^{\epsilon_{lk}}+b-1}$ and $\delta_{cd} < 2\beta$, $\mathcal{M} = \mathcal{S}_p \circ \mathcal{R}_{\gamma,b}^{n_p} \circ \mathcal{K}_r^{\beta}$ satisfies $(\epsilon_{cd}, \delta_{cd})$-DP, where:*

$$\epsilon_{ck} = \sqrt{\frac{14\log(\frac{2\beta}{\delta_{cd}})(e^{\epsilon_{lk}}+b-1)}{n_p - 1}}, \tag{5}$$

$$\epsilon_{cd} = \log(1 + \beta(e^{\epsilon_{ck}} - 1)). \tag{6}$$

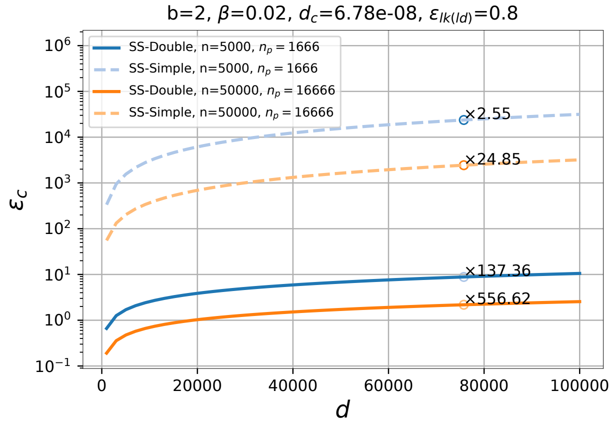


Figure 3: Privacy amplification effect comparison.

Proposition 1 *The standard deviation of the estimated mean in each dimension from $\mathcal{P} = \mathcal{A} \circ \mathcal{S}_p \circ \mathcal{R}_{\gamma,b}^{n_p} \circ \mathcal{K}_r^{\beta}$ is $O(\frac{n_p^{1/6}\log^{1/3}(1/\delta_{ck})}{n_s \epsilon_{ck}^{2/3}})$.*

Since all dimension-level datasets are padded into the same size and $(\epsilon_{ck}, \delta_{ck})$-DP holds from the amplification in Theorem 3, we show the vector-level DP composition in Theorem 4. The sample rate is denoted as $\beta = k/d$. We distill the amplification effect from ϵ_l to ϵ_c in Corollary 3

Theorem 4 *For any neighboring datasets $X \simeq_r X'$ which differ in one user's local vector, the d-dimensional vector aggregation protocol $\mathcal{M} = \mathcal{S}_p \circ \mathcal{R}^{n_p} \circ \mathcal{K}_r^{\beta}$ in SS-Double satisfies (ϵ_c, δ_c)-DP, where:*

$$\epsilon_c = 2\beta d\epsilon_{cd} \wedge (\epsilon_{cd}\sqrt{4\beta d \log(1/\delta_{cd})} + 2\beta d\epsilon_{cd}(e^{\epsilon_{cd}} - 1)),$$
$$\delta_c = \delta_{cd}(2\beta d + 1).$$

Corollary 3 *For SS-Double, with $\epsilon_l \leq \beta d \log(n_p / \log((2\beta^2 d + \beta)/\delta_c))/2$, the amplified central privacy is:*

$$\epsilon_c = O((1 \wedge \frac{\epsilon_l}{\beta d})e^{\frac{\epsilon_l}{\beta d}}\beta^{1.5}\sqrt{\frac{d}{n_p}\log(\frac{\beta d}{\delta_c})\log(\frac{\beta^2 d}{\delta_c})}). \tag{7}$$

Simulation of Privacy Amplification. To compare the privacy amplification effect of SS-Simple and SS-Double, we visualize the magnified privacy w.r.t. $\mathcal{R}_{\gamma,b}$ in Figure 3.

1. $\mathcal{S} \circ \mathcal{R}^n$ (SS-Simple): Lemma 1 and Theorem 1.
2. $\mathcal{S}_p \circ \mathcal{R}^{n_p} \circ \mathcal{K}_r^{\beta}$ (SS-Double): Theorem 3 and Theorem 4.

To align the valid condition in Lemma 1, we compare the above two cases given the same local privacy budget for *each* perturbed dimension, denoted as ϵ_{ld} for SS-Simple and ϵ_{lk}

for SS-Double. In this case, we present the amplification effect by the magnification ratio ϵ_l/ϵ_c for $d = 75755$. It can be observed that the double amplification of SS-Double boosts improves the ratio from $\times 2.55$ to $\times 137.36$, which denotes a stronger privacy amplification effect. It should be noted that the double amplification in Theorem 4 also holds for other randomizers, which only differs with $\mathcal{R}_{\gamma,b}$ on the evaluation of ϵ_{ck}. We show more numerical evaluations in experiments.

Utility Boosting with Top-k (SS-Topk)

Intuition. A problem of SS-Double protocol is that the random subsampling treats all dimensions equally and thus may discard "important" dimensions. For the high-dimensional case $(d > n)$, random sampling a small fraction β of values from a vector slows down the convergence rate of training. In light of the efficient gradient sparsification technique, we are motivated to adopt the magnitude-based selection (Aji and Heafield 2017) for boosting the convergence rate. However, selecting Top-k indexes with greatest absolute magnitudes over the vector is data-dependent and thus compromises user privacy. The challenge is how to preserve and qualify the *index privacy* while maintaining the utility as far as possible.

Index Privacy. According to our trust setting in Table 1, the prime adversary that threats index privacy is the shuffler $\mathcal{S}$ because only $\mathcal{S}$ knows which user sends which indexes (note that the perturbed value is encrypted). Our goal is to bound the shuffler's success of predicting whether or not the index uploaded by a user ranks in the Top-k elements of the local vector. By random guessing, the adversary's success rate of predicting a dimension as Top-k is k/d. After observing the privatized selected indexes, the success rate cannot be enlarged by more than ν times.

Thus, we are motivated to qualify and preserve the index privacy with an anonymity-based metric in Definition 3. We denote whether the magnitude of dimension $j \in [d]$ ranks in Top-k ($\beta = k/d$) by $\mathbb{I}_j \in \{0, 1\}$, and whether the index j is selected by $\mathcal{K}_\nu^\beta(j)$. The first inequality bounds the adversary's success rate with ν while the second inequality ensures the probability is no greater than 1. Intuitively, the strongest index privacy stands when $\nu = 1$, because observing $\mathcal{K}_\nu^\beta(j)$ does not increase the adversarial success rate.

Definition 3 *A mechanism $\mathcal{K}_\nu^\beta$ provides ν-index privacy for a d-dimensional vector, if and only if for any $j \in [d], \nu \geq 1$, we have:* $\Pr[\mathbb{I}_j = 1|\mathcal{K}_\nu^\beta(j)] \leq \nu \cdot \Pr[\mathbb{I}_j = 1]$ *and* $\Pr[\mathbb{I}_j = 0|\mathcal{K}_\nu^\beta(j)] \geq \frac{\Pr[\mathbb{I}_j=0]}{\nu}$.

To achieve ν-index privacy, we need to control the probability of $\Pr[\mathbb{I}_j = 1|\mathcal{K}_\nu^\beta(j)]$. Given the prior knowledge $\Pr[\mathbb{I}_j = 1] = \beta, \Pr[\mathbb{I}_j = 0] = 1 - \beta$, if each user reports lk dimensions to the shuffler of which only k indexes are real Top-k, we have $\Pr[\mathbb{I}_j = 1|\mathcal{K}_\nu^\beta(j)] = \frac{1}{l}$. Put it into the definition, we have: $\Pr[\mathbb{I}_j = 1|\mathcal{K}_\nu^\beta(j)] \leq \min(\nu \cdot \beta, \frac{\nu-1+\beta}{\nu})$. Thus, $\mathcal{K}_\nu^\beta$ satisfies ν-index privacy when we set $l \geq \max\{\frac{1}{\nu\beta}, \frac{\nu}{\nu-1+\beta}\}$.

SS-Topk Protocol. Denote the SS-Topk protocol as $\mathcal{P} = \mathcal{A} \circ \mathcal{S}_p \circ \mathcal{R}^{n_p} \circ \mathcal{K}_\nu^\beta$. Each user locally runs $\mathcal{R} \circ \mathcal{K}_\nu^\beta$. The shuffler executes $\mathcal{S}_p$. The analyzer runs the aggregation $\mathcal{A}$. Compared with SS-Double, SS-Topk has the same shuffling and analyzing procedure but differs in the dimension selection $\mathcal{K}_\nu^\beta$ in Randomize$(\cdot)$ which is shown in Algorithm 4.

Algorithm 4 Randomize$(\cdot)$=$\mathcal{R} \circ \mathcal{K}_\nu^\beta$ for SS-Topk

Choose a valid $l \in [\max\{\frac{1}{\nu\beta}, \frac{\nu}{\nu-1+\beta}\}, \lceil 1/\beta \rceil]$
Top-k index set $S_{top} \leftarrow \{j | j \in \text{Top}(|\tilde{x}_i|)\}^k$
Non-Top index set $S_{non} \leftarrow \{j | j \in [d] \backslash S_{top}\}^{k(l-1)}$
for each index $j \in S_{top} \cup S_{non}$ **do**
  $y_{i,j} \leftarrow \begin{cases} \mathcal{R}_{\epsilon_{lk}}(\tilde{x}_{i,j}), & j \in S_{top} \\ \omega_{\mathcal{R}}, & j \in S_{non} \end{cases}$
end for
generate a permutation π_r over $[lk]$
the index list $idx_i \leftarrow \{d_{\pi_r(1)}, \cdots, d_{\pi_r(lk)}\}$
the perturbed value list $y_i \leftarrow \{y_{\pi_r(1)}, \cdots, y_{\pi_r(lk)}\}$
returns $\langle idx_i, y_i \rangle$

By applying $\mathcal{K}_\nu^\beta$ on the processed vector $\tilde{x}_i$, k Top-k indexes are sampled as the set S_{top} while $k(l-1)$ non-top dimensions are randomly sampled from the rest as S_{non}. As k true value are perturbed, the privacy budget split for each dimension is $\epsilon_{lk} = \epsilon_l/k$. Then each value of real Top-k dimensions is perturbed as $y_{i,j} \leftarrow \mathcal{R}_{\epsilon_{lk}}(\tilde{x}_{i,j})$. Each non-top dimension is padded with a value drawn from the blanket distribution $\omega_{\mathcal{R}}$. Then kl dimensions are permuted into the list of idx_i and y_i, which will be encrypted into l messages by Line (11) in Algorithm 2.

Privacy Analysis First, we show the relationship of ν and l in Proposition 2. The valid l for a given ν is illustrated in our full version. The strongest index privacy $\nu = 1$ is achieved when $l = \lceil 1/\beta \rceil$. For SS-Double, ν naturally holds with the random sampling mechanism $\mathcal{K}_r^\beta$. To the other extreme case of no index privacy, SS-Topk still provides a strong privacy guarantee since the shuffler knows nothing except for Top-k indexes. An state-of-the-art work (Zhu, Liu, and Han 2019) has shown the privacy attack's availability is significantly impaired even knowing both Top-k indexes as well as their values.

Proposition 2 *The range of ν-index privacy is $1 \leq \nu \leq \frac{1}{\beta}$, where the strongest index privacy $\nu = 1$ is achieved when $l = \lceil \frac{1}{\beta} \rceil$ and no index privacy is achieved when $l = 1$.*

Then, we clarify the compatibility of ν-index privacy against the shuffler and (ϵ_c, δ_c)-DP against the analyzer. With $\mathcal{S}_p$ in Algorithm 3, the analyzer only gets padded results for each dimension with the same size n_p. Thus, the ν-index privacy against the shuffler **does not** affects the amplified privacy (ϵ_c, δ_c)-DP against the analyzer. Given n_p, SS-Topk shares the same double amplification effect of SS-Double. Lastly, we discuss the trade-off between the index privacy and the communication costs as well as the utility. The bandwidth of each user depends on $O(lk)$. As Proposition 1 implies, the estimation utility depends on the dummy padding size n_p. Thus, given n_p, n and β, ν **does not** affect the accuracy of the mean estimation because the number of dummy values is fixed. We show the strongest index privacy under the given parameters in Theorem 5.

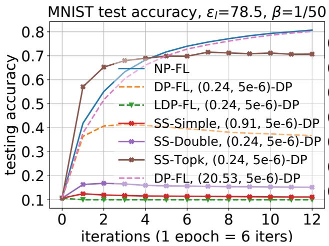


Figure 4: Utility-Privacy.

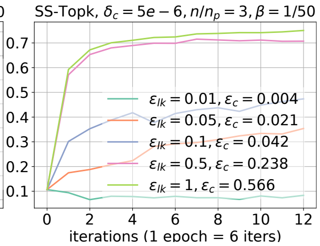
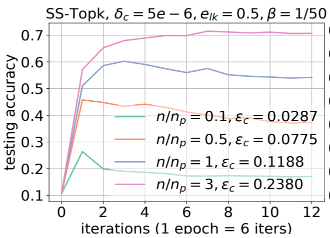


Figure 5: Impact of ϵ_{lk}.



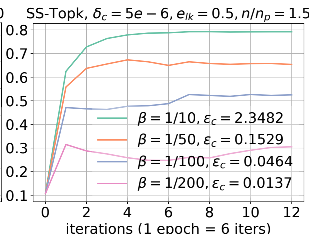


Figure 6: Impact of n/n_p. Figure 7: Impact of $\beta = k/d$.

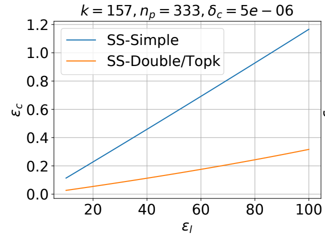
Theorem 5 *Given a protocol with $\mathcal{K}_{\nu}^{\beta}, n_p$, the strongest index privacy it allows for each user is $\nu = \max\{1, \frac{1}{\lfloor \frac{n_p}{n\beta} \rfloor \cdot \beta}\}$.*

Evaluations

We evaluate the learning performance on MNIST dataset and logistic regression model with $d = 7850, n = 1000$. Baselines include non-private Federated Averaging (NP-FL) (McMahan et al. 2016), DP-FL (Abadi et al. 2016) with Gaussian Mechanism in which we double the sensitivity for the comparison under bounded DP definition, LDP-FL with Gaussian Mechanism (Dwork et al. 2006) for (ϵ, δ)-LDP and Laplace Mechanism for ϵ-LDP and our three protocols of SS-Simple, SS-Double, SS-Topk.

Comparison of Our Protocols. For SS-Simple, SS-Double, SS-Topk, we apply the Laplace Mechanism as the basic randomizer $\mathcal{R}$ for each dimension. Given $\epsilon_l = 78.5$, the spilt privacy budget of each dimension is $\epsilon_{ld} = 0.01$ for SS-Simple and $\epsilon_{lk} = 0.5$ for SS-Double and SS-Topk. The double amplification effect boosts the amplified privacy against $\mathcal{A}$ for one epoch from $(0.91, 5 \times 10^{-6})$-DP (SS-Simple) to $(0.24, 5 \times 10^{-6})$-DP (SS-Double/Topk). Compared with SS-Simple, SS-Double improves the testing accuracy by 4.07% under a stronger central privacy. Compared with SS-Double, SS-Topk significantly boosts the utility by 55.5% under the same central privacy. With Theorem 5 and $n/n_p = 3$ in Figure 4, the maximum index privacy that the protocol allows for each user is $\nu = 3.125$ with $l = 16$.

Comparison with DP/LDP-FL. It is obvious that even the baseline SS-Simple performs better than LDP-FL. Then we observe that under the same central privacy $(0.24, 5 \times 10^{-6})$-DP, SS-Topk even achieves a dramatic higher accuracy than DP-FL by 33.94%. This is a **key observation** of our work, because traditional works for the one-dimensional task claim that the shuffle model only stands in the middle-ground between LDP and DP.



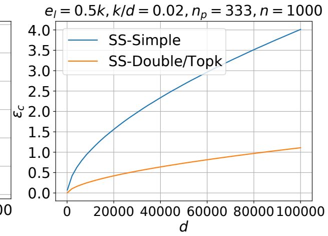


Figure 8: Under various ϵ_l. Figure 9: Under various d.

We analyze the reasons as follows: 1) The effect of Top-k: as SS-Double cannot exceed the performance of DP-FL $(0.24, 5\times10^{-6})$-DP but SS-Topk can, it is obvious that Top-k boosts the utility. 2) The effect of the double amplification: If we only counts the amplification from shuffling for SS-Topk without the amplification from subsampling, we have $\epsilon_c = 20.53, \delta_c = 5 \times 10^{-6}$. Hence, we introduce another baseline of DP-FL $(20.53, 5 \times 10^{-6})$-DP. We observe that this line approaches the non-private version and has higher testing accuracy than SS-Topk[1]. In other words, SS-Topk cannot perform better that DP-FL if only amplification of shuffling is counted. Thus, we validate the effect of proposed double sampling for such nontrivial utility boosting. Thus, we conclude that **both** double amplification and Top-k boosting are necessary to performs better than DP-FL.

Comparison under Variant Parameters. We then evaluate the impacts of other hyper-parameters of n_p, β and privacy budget ϵ_l in Figures 5, 6 and 7. 1) It is obvious in Figure 5 that a larger local privacy budget for each dimension leads to higher testing accuracy. 2) In Figure 6, the higher ratio of n/n_p implies less additional noises injected by *dummy padding*. This validates our utility analysis in Proposition 1. Thus, we can conclude that n_p is the key knob to tune the privacy and utility trade-off. 3) In Figure 7, we can observe that larger β implies better utility. With $\beta = 1/10$ in Figure 4, we can achieve only 1.48% loss compared with NP-FL for 2 epochs of $(2.348, 5e^{-6})$-DP per epoch.

Privacy Amplification. We illustrate the overall amplification in previous evaluations with the Bennett inequality for the Laplace Mechanism in Figure 8 and Figure 9. This validates our Theorems 1, 4 and Corollaries 2, 3 are generic for any local randomizer, as long as the amplification bound can be derived from a close-form solution or numerical evaluations (Balle et al. 2019b).

Conclusion

To conclude, we propose the first differentially private federated learning framework FLAME in the shuffle model for better utility without relying on any trusted server. Our privacy amplification effect and private Top-k selection mechanism significantly boosts the testing accuracy under the high-dimensional setting.

[1] SS-Topk satisfies $(20.53, 5 \times 10^{-6})$-DP when ignoring subsampling amplification effect.

Ethics Statement

In this paper, the authors introduce FLAME, a framework in the shuffle model for differentially private federated learning without any trusted parties. Federated Learning can be applied to a wide range of applications, including the next word predicting, fraud detection model, training an epidemic prediction model, and many more (Yang et al. 2019).

There will be essential impacts resulting from federated learning in general. Here we focus on the effects of using FLAME to preserve privacy for the cross-device setting, where a global model is trained over massive personal data that are generated on small gadgets or mobile devices. Our research could be used to provide better learning performance under a strong privacy guarantee in these applications. Thus, it will promote the information flow across millions of mobile devices while strictly maintaining users' human rights for data privacy (GDPR:https://gdpr-info.eu/).

The potential risks of differentially private federated learning over mobile devices have typically received less attention. These include:

- There is a lack of concrete law or regulation for the privacy budget over iterative rounds. It should be transparent for users knowing their privacy budget setting to feel safe and be safe.
- If differential privacy is applied for local updates, this could increase the difficulty of detecting adversarial attacks from the user end, leading to a biased global model.

Besides, we see opportunities for research related to shuffle model FLAME to beneficial purposes, such as providing better product services for users and better monitoring the financial crisis and a public health emergency (e.g. COVID-19). To mitigate the risks associated with using differentially private federated learning, we encourage research to understand the impacts of robustness when applying models to scenarios where the prediction accuracy is highly sensitive.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62072460, No. 61532021, 61772537, 61772536, 61702522), JSPS KAKENHI Grant No. 17H06099, 18H04093, 19K20269, JST/NSF Joint Research SICORP 20-201031504 and CCF-Tencent Open Fund WeBank Special Fund.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318. ACM.

Aji, A. F.; and Heafield, K. 2017. Sparse Communication for Distributed Gradient Descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 440–445.

Balcer, V.; and Cheu, A. 2019. Separating local & shuffled differential privacy via histograms. *arXiv preprint arXiv:1911.06879* .

Balle, B.; Barthe, G.; and Gaboardi, M. 2018. Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Balle, B.; Bell, J.; Gascon, A.; and Nissim, K. 2019a. Differentially Private Summation with Multi-Message Shuffling URL http://arxiv.org/abs/1906.09116.

Balle, B.; Bell, J.; Gascón, A.; and Nissim, K. 2019b. The privacy blanket of the shuffle model. In *Annual International Cryptology Conference*, 638–667. Springer.

Chan, T. H.; Shi, E.; and Song, D. 2012. Optimal lower bound for differentially private multi-party aggregation. In *European Symposium on Algorithms*, 277–288. Springer.

Cheu, A.; Smith, A.; Ullman, J.; Zeber, D.; and Zhilyaev, M. 2019. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 375–403. Springer.

Duchi, J. C.; Jordan, M. I.; and Wainwright, M. J. 2013. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 429–438. IEEE.

Dwork, C.; Kenthapadi, K.; Mcsherry, F.; Mironov, I.; and Naor, M. 2006. Our Data, Ourselves: Privacy via Distributed Noise Generation. In *International Conference on Advances in Cryptology-eurocrypt*.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9(3-4): 211–407.

Dwork, C.; Rothblum, G. N.; and Vadhan, S. 2010. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 51–60. IEEE.

Erlingsson, Ú.; Feldman, V.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Thakurta, A. 2019. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2468–2479. SIAM.

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

Ghazi, B.; Golowich, N.; Kumar, R.; Manurangsi, P.; Pagh, R.; and Velingker, A. 2020. Pure Differentially Private Summation from Anonymous Messages. In *1st Conference on Information-Theoretic Cryptography (ITC 2020)*, volume 163 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 15:1–15:23. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Ghazi, B.; Golowich, N.; Kumar, R.; Pagh, R.; and Velingker, A. 2019. On the power of multiple anonymous messages. *arXiv preprint arXiv:1908.11358* .

Hitaj, B.; Ateniese, G.; and Perez-Cruz, F. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 603–618. ACM.

Liu, R.; Cao, Y.; Yoshikawa, M.; and Chen, H. 2020. FedSel: Federated SGD under Local Differential Privacy with Top-k Dimension Selection. In *International Conference on Database Systems for Advanced Applications*, 485–501. Springer.

McMahan, H. B.; Moore, E.; Ramage, D.; and y Arcas, B. A. 2016. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629* .

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*.

Nasr, M.; Shokri, R.; and Houmansadr, A. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, 739–753. IEEE.

Wang, N.; Xiao, X.; Yang, Y.; Zhao, J.; Hui, S. C.; Shin, H.; Shin, J.; and Yu, G. 2019. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In *2019 IEEE 35th ICDE*, 638–649.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology* 10(2): 1–19.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.