# RECENT PROGRESS OF DIFFERENTIALLY PRIVATE FEDERATED LEARNING WITH THE SHUFFLE MODEL

Moushira Abdallah Mohamed Ahmed [1], Shuhui Wu [1], Laure Deveriane Dushime [1], and Yuanhong Tao [1]

[1] School of Science, Zhejiang University of Science and Technology, 318 Liuhe Road, Hangzhou, Zhejiang 31002, P. R., China

## ABSTRACT

As an improved distributed machine learning, federated learning has achieved significant success various domains. However, to prevent data leakage and improve the security of FL, there are an increasing amount of studies on exploring how to integrate FL with other techniques. One bottleneck challenge behind it is that how to efficiently balance the privacy and the efficient of communication to achieve the optimal solution. In this paper, we conduct a survey on existing studies on differentially private FL with shuffle model, which seems the efficient way to solve the above problem. We start the survey by providing several key notation to achieve efficient exploration. Then we conduct the survey according to the role of shuffle model for solving the problem between privacy and accuracy. Furthermore, we present two types of shuffle, single shuffle and m shuffles with the statistical analysis for each one in boosting the privacy amplification of users with the same level of accuracy by reasoning the practical results of recent papers. Meanwhile, the research on exploration in shuffle model is at an early stage at present. Finally, we conclude the paper by pointing out a few future directions.

**Keywords:** Federated Learning, Differential Privacy, Shuffle Model, Privacy Amplification

**Received** 01 August 2021
**Accepted** 15 August 2021
**Published** 30 November 2021

**CorrespondingAuthor**

Shuhui Wu, s.wu@zust.edu.cn

**DOI** 10.29121/ijetmr.v8.i11.2021.1028

**Funding:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## 1. INTRODUTION

Federated learning (FL) [Li et al. (2020)] is an improved distributed machine learning which allows multiple devices into a decentralized system to accumulate the raw data to assist in the training process of the model. FL was introduced for the first time by Google in 2016 to permit many users in participating together along with protecting their raw data. Since then, FL has attracted an increasing attention as an improved type of distributed machine learning (DML). FL is designed to distribute the training sets among multiple users and each one implements its model according to its data sets, where the users do not share their data with others or server. To prevent data leakage, each user trains his model and uploads the local parameter to server. The server aggregates all local parameters from users' models and creates a global model that is more efficient than users' model. Finally, the server distributes the global parameters to users for retrain their data again and the process of uploading and downloading parameters between users and server is conducted recursively until getting the optimal global model in the server [Chen et al. (2021)].

**How to cite this article (APA):** Ahmed, M. A. M., Wu, S., Dushime, L. D., and Tao, Y. (2021). Recent Progress of Differentially Private Federated Learning with the Shuffle Model. *International Journal of Engineering Technologies and Management Research, 8*(11), 55-75. doi: 10.29121/ijetmr.v8.i11.2021.1028

FL has also made improvements in the privacy and security of machine learning models because the server process aggregates only local parameters from each user and the server doesn't know anything about user raw data. In fact, the practical results of FL are very effective in protecting sensitive users' data. It is very attractive to many real word applications, such as hospitals, financial institutions, and governments. On the other hand, new computing paradigms of FL have also attracted the attention of adversaries with malicious intent. Consequently, they may have effects on the model update, or they may make suggestions that include a user's privacy. Thus, FL is not the best solution in protecting privacy [Yang et al. (2019)].

Due to the importance of privacy and security of data processing, there is growing amount of literature, which has attempted to provide solutions [McMahan et al. (2016), Aledhari et al. (2020)]. Privacy preserving and security mechanisms have been considered to incorporate with FL across the entire system. Differential privacy (DP) is one of these main mechanisms [Dwork (2008), Dwork (2006)]. For DP, some randomized mechanisms, such as Laplace mechanism, Gaussian mechanism and exponential mechanisms, have been used to add random noises to the output of query so that the adversary can't differentiate between two distinct inputs [Zhu et al. (2020), Erlingsson et al. (2014)].

FL with DP is the cutting-edge of research on privacy protection from theoretical aspect as well as from a practical perspective [Ding et al. (2017), Liu et al. (2021)]. Even though, integrating DP with FL is not an effective enough way for solving all privacy problems. Thus, the shuffle model has been proposed. In [Cheu et al. (2019)], the first protocol of real summation with shuffle model have been presented, which showed that $O$ ($\sqrt{k}$) messages sending by k users via shuffle achieved $O$ (1) mean square error (MSE) in central model. The technique of the shuffle model was proposed to deal with the problem between privacy in center model and accuracy in local model.

The framework of DP-FL with shuffle model consists of users, shuffle and analyzer [Balle et al. (2019), Meehan et al. (2021), Erlingsson et al. (2019), Bittau et al. (2017)]. In this framework, users train their model by using their own data set, using local randomization to implement DP. Afterwards, each user sends the local noised parameters as a message to the shuffle, and the shuffle perturbs users' messages by using randomly permutation π then send these messages to analyzer. Finally, the analyzer aggregates all messages from shuffle for analysis and creates the global model. Briefly, the untrusted analyzer receives $m$ messages from users' entity via shuffle functionality. This model has been proved that it has the ability to overcome the limitations on accuracy of local algorithms, while protecting several of their appropriate attributes. In order to gain a comprehensive understanding of the state-of-the art DP-FL with shuffle model, we conduct this survey. The main contributions of this survey are

- The first comprehensive survey on the current research on DP-FL with shuffle model.
- A taxonomy of solution published to 2021 for shuffle model in DP-FL.
- Some promising research directions for the future work.

This survey is structured as follows. In Section 2, we introduce all the necessary notations throughout the paper. Section 3, we summarizes the recent published papers on DP-FL with shuffle model. Section 4 we outline the shuffle model and his role in boosting privacy in FL with achieving accuracy and privacy amplification. Lastly, we conclude out paper in Section 6.

## 2. BACKGROUNDS

In this section, we shall introduce the necessary notations that needed through our paper. At first, the common notations have been presented in the following table.

**Table 1 The common notations though out the paper**

| Notations | Explanation | Notations | Explanation |
|---|---|---|---|
| D, D′ | two neighboring datasets | X | Sample space |
| Y | Output space | N | Size of Dataset |
| $\mathcal{M}$ | Algorithm mechanism | D | Dimension of sample space |
| K | Number of users | W | vector weight of gradient descent |
| T | Communication round | C | Clipping parameter |
| H | Hyper parameter (learning rate) | L(.) | Laplace mechanism |
| ε | Privacy budget | π | Random permutation in shuffle |
| Δ | Possibility of Violating ε -DP | $F(\cdot)$ | Sensitivity function |
| A | The order of Renyi divergence between two distributions P and Q | N | The number of examples in training process |
| $\mu(x)$ | Data distribution | $\varepsilon_\mu$ | Privacy budget for BDP |
| $\delta_\mu$ | Possibility of Violating $\varepsilon_\mu$ | $N(0,\sigma^2)$ | Gaussian mechanism |
| $\|\| f \|\|$ | Norm of function $f$ | ~ | Approximately equal |

## 2.1. FEDERATED LEARNING

FL is an improved type of DML in which, the training process is distributed among many users and the server has the role of coordinating everything by aggregating gradients from participants [Kairouz et al. (2021)] as shown in Figure 1.

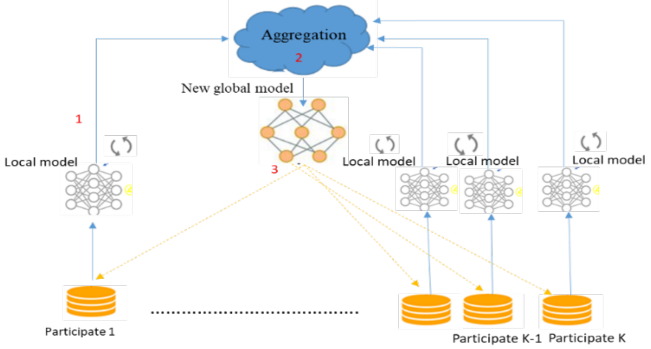


**Figure 1** Federated learning training process

From the above Figure 1, training process of FL usually takes three steps [Yang et al. (2019)]. In the first step, all participates train their data from the local model then upload it to the server (the aggregator). In the second step, the aggregator collects all participators' local parameters from local models to produce efficient global model. In the third step, the aggregator distributes the global model parameters to all users in order to retrain their data on it. The training process performs iterations until achieving the optimal global model that demonstrates high accuracy in users' local model. Notably, in FL the users upload the local parameters rather than the local data, providing certain security for sensitive information about users.

The concept of DP has been long among the privacy protection mechanism. DP is a verifiable privacy notion proposed by Dwork [Dwork (2006)], which is a mechanism to protect sensitive data from leakage by adding some noise on users' data and returning statistically indistinguishable results. It means that DP guarantees that any adversary doesn't have high probability of assuming whether a participated client is in the input by monitoring the output. If we have two datasets D, D' $\subset \mathbb{D}$ satisfy D $\subset$ D' and D'=|D|+1, then D, D' are said that they are two adjacent datasets and written as D~D'. The definition is as following.

**Definition 1.** (ε-DP [Dwork (2006)]) A randomized mechanism $\mathcal{M}$: D→R gives ε-DP if for any adjacent datasets D, D' $\subset \mathbb{D}$, and all output S ⊆ R, where R is a Range $\mathcal{M}$

$$\Pr[\mathcal{M}(D) \in S] \le e^{\epsilon} \Pr[\mathcal{M}(D') \in S], \qquad (1)$$

where ϵ is the privacy budget that control the privacy level of $\mathcal{M}$.

For a lesser $\epsilon$ the probability distributions of output results over $\mathcal{M}$ on $D$ and $D'$ are very similar and it is hard to differentiate both datasets. The relaxation version of $\epsilon$-DP is (ϵ, δ)- DP, in which δ was added when we do not have pure privacy to detect violate $\varepsilon$. The notation of (ϵ, δ) DP is as follows.

**Definition 2**. ((ϵ, δ)-DP [18]) A randomized mechanism $\mathcal{M}$: D→R gives ((ϵ, δ)-DP if for any neighboring datasets D, D' $\subset$ D, and all output S ⊆ R, R is Range ($\mathcal{M}$).

$$\Pr[\mathscr{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathscr{M}(D') \in S] + \delta, \qquad (2)$$

Where S refers to the output domain of the algorithm $\mathscr{M}$.

If $\delta = 0$, the mechanism $\mathscr{M}$ provides ϵ-DP by its stringent definition. If $\delta > 0$, (ε, δ)-DP offers liberty to interrupt strict ϵ-DP for some low probability events. In brief, DP can be realized by adding an affordable amount of noise into the output results of the query function. This amount of noise will effect on the balance between privacy and accuracy in the overall model. Namely, large amount of noise will make the dataset unusable and too small noise is not sufficient for DP collateral. The noise amount can be identified by computing the sensitivity.

**Definition 3**. (Sensitivity [Awan and Slavkovi'c. (2018)]) For an enquiry function ƒ, the sensitivity of ƒ is set as

$$\Delta f = \mathrm{Max}_{D,D'} \|f(D) - f(D')\|_{l}, \qquad (3)$$

Where $\| . \|_{l}$ is $l$ norm.

Note that, there are three standard techniques that are used to realize (ϵ-δ) DP for all systems, Laplace mechanism [Phan et al. (2017)], Gaussian mechanism [Liu (2019)], and exponential mechanism. For numerical results, Laplace mechanism and the Gaussian mechanism are broadly used to realize DP. For non-numeric results, the exponential technique is used.

### 2.2.1. CENTRAL DEFERENTIAL PRIVACY (CDP)

Based on DP definition, CDP is regarded as a disaggregated technique of DP [McMahan et al. (2018)], which provides secrecy for the overall system by adding random noise to the aggregated output in the central part after collecting all the data from participating users. Consequently, the server will access to the user's true data. It provides a good accuracy but rely on trusted analyzer. The greatly difference between CDP and LDP is that CDP is a central privacy model with the supposition of a confidential analyzer where the users sent their data directly to server. Afterwards, the noise was added to query mechanisms (Figure 2 (a)). On the other hand, LDP is a local privacy model with no supposition on the analyzer and every client's data is locally disturbed in the user-side before transferred to the analyzer.

### 2.2.2. LOCAL DIFFERENTIAL PRIVACY (LDP)

LDP is a disaggregated modification of DP which permits each participator to perturbs his own data locally and transmits the disturbed data to the analyzer (Figure 2 (b)). Hence, the server will not transfer contact to the user real data, thus providing a robust privacy. In LDP, the input of perturbation mechanism is raw users' data and the disturbed data is like as the mechanism output Geyer et al. (2020), Farhad. (2021), Zhao et al. (2020), Dwork (2011)]. The formal definition of (ε, δ) - LDP is similar with definition 2.

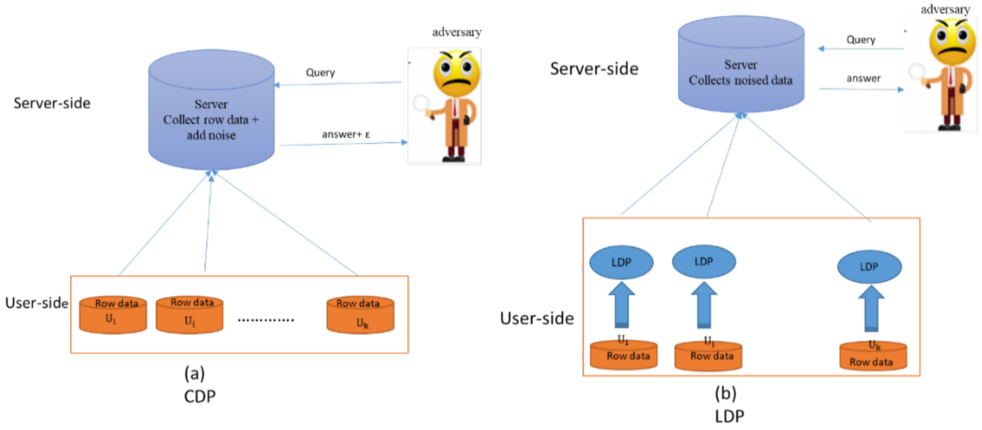


**Figure 2** The framework of (a) CDP and (b) LDP.

## 2.3. SHUFFLE MODEL

A shuffler (SS) is a randomized mechanism that hides all the notification about the positions of each user message by implementing a unified random permutation π of entries then returns the results after permutation (Figure 3) [Beimel et al. (2020)]. More formally, assume that we have M messages from $k$ participators and each one has $m_i \in M$ messages where $m_i$ ($1 \leq i \leq k$) denotes to i user message. For the shuffler SS: $m^k \rightarrow m^k$, the output shuffle will be $SS_{out}(M) = m_i^{\pi}$. As far as privacy, shuffle model plays a great role in privacy amplification. Because SS can perturb input messages before sending it to aggregator. By this way, it hides the position of users' messages from analyzer. Hence, the privacy of users data doesn't depend on only adding (ε, δ)-DP for users' data but also depend on the shuffle perturbation π. The communication model of SS is described as follows:

**Definition 4.** (Communication model in SS [Balle et al. (2019)]) Suppose that we have a vector of m messages from $k$ users and every user sends its message to the shuffler. The protocol of SS communication model which represents distributed messages among $k$ users is P = SS(m), where m= $m_1$, $m_2$, ……., $m_k$ and SS is the shuffle functionality that choose a permutation π for users' messages. The shuffle output is $SS_{out}(m) = (m_1^{\pi}, m_2^{\pi}, \ldots., m_k^{\pi})$. The formal explanation of SS protocol is shown in Algorithm 1.

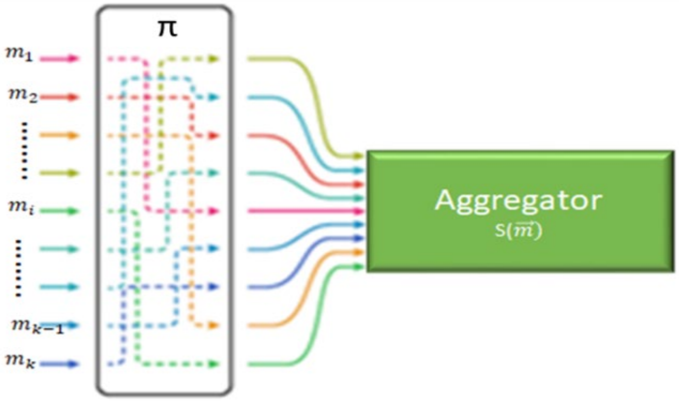


**Figure 3** Shuffle channel

**Algorithm 1 : The communication model with shuffle model**

```
Initialization: let K be the number of users, m be the users' messages, and T be the communication round.
Output: the global weight w
While 1 ≤ i ≤ K # of users
If 1 ≤ t ≤ T do
    1) All users train their model to generating $K$ messages m= $m_1, m_2,...., m_k$.
    Then, send the vector of messages m to SS
    2) SS receives m from users.
    3) SS permutes all messages by using random permutation π.
    4) SS output $SS_{output} = (m_1^{\pi}, m_2^{\pi}, ...., m_k^{\pi})$, then SS sends $SS_{output}$ to analyzer.
    5) the analyzer aggregates $SS_{output}$ to get the optimal model weight, $w = \sum_{i=1}^{k} (1/k) SS_{out}$
    6) t = t+1
Output: return w
```

## 3. THE APPLICATIONS OF FEDERATED LEARNING

To demonstrate the importance of FL, we present the applications of FL in this section. There is a huge amount studies on applying FL in real life since the significance of FL has been recognized.

1) **FL in health care**: Conventionally, healthcare records from scattered sites are progressed to a central database for analysis, which includes huge mount of data from each data source. In the past year, a huge number of changes has practiced due to the pandemic situation. The lack of resources in the healthcare industry was quite evident during this time. Therefore, the new way of transforming the healthcare data has been proposed in the healthcare industry. In [Liu et al. (2018)], a community-based FL algorithm has been recommended to expect mortality and hospital residence time. Electronic medical records are assembled into communities inside each hospital based on common medical sides. Each assemble learns and shares a particular ML model rather than the overall one mutual among all hospitals and hence patients. In [Huang et al. (2019)], the authors has proposed FL to predict hospitalizations during a target year for patients having heart disease using EHR's data extent among numerous data sources. In [Li et al. (2019)], DP-FL technique has been considered to predict medical image for brain tumor segmentation to prevent any leakage of information. It suggests the cooperation of several organizations by sharing their locally computing models.

2) **FL in IoT system:** FL aims to secure the data collecting through different mediums. Hence, FL can help to achieve personalization as well as enhance the performance of devices in IoT applications. In [Jiang et al. (2019), Ren et al. (2019)], the method is to use deep reinforcement learning to improve unloading results for IoT systems. The main idea is to consider proxy data is fewer related to the data kept on IoT machines. Consequently, IoT

machines are responsible for the training models while edge nodes are for updating aggregation. The effectiveness of participating in IoT with FL has been demonstrated with edge computing when training deep reinforcement learning.

3) **FL in blockchain:** Blockchain has been used recently to transfer money in Bitcoin without center server, where all users' accounts are saved in public blockchain, then this public block is broadcasting to every part in the network. Each user is locally contributes to the overall blockchain by adding a new blocks in sequential order that provides an indelible and transparent record of transactions. The public can only see that someone has added a transaction with a quantity without knowing any sensitive information about depositor. FL framework is similar with blockchain framework in downloading global parameters for users to train their data, and then uploads it to server. According to [Nakamoto (2019), Sarfaraz et al. (2021)], the model contains a permission FL part and a block chain part, while the blockchain checks if a set of queries has been preceded or not. In addition, the multiparty data recovery process identifies to gain the outcomes of queries and then upload it to the aggregator instead of shearing the real information directly to the analyzer. The data consists of two types, useable data toward the requirements and reduced secret data of users.

4) **FL in recommendation system**: Recommendation system requires data from users. Thus, it is very dangerous on privacy if users' data is shared directly. FL acts as privacy-preserving system for recommendation systems in many cases, for example a virtual keyboard prediction in mobile devices [Hard et al. (2018)]. The familiar example on virtual key board is Google virtual keyboard (g-board) which need features prediction similar with autocorrection of spelling faults, next word prediction and offers communication features, such as, emoji, GIFs, and stickers [Yang et al. (2018), Ramaswamy et al. (2019)]. So it is necessary that g-board guarantees client privacy. Because the client may be type sensitive information like password. Hence, FL occupies a new view of federated recommender systems (Fed-Rec), which is an embodiment of FL on decentralized recommendation to save sensitive client information from leakage.

## 4. DIFFERENTIAL PRIVACY IN FEDERATED LEARNING

We shall present the framework DP-FL in this section. FL has been designed to protect data privacy by being distributed learning systems that keep the data in its storage stores. FL allows training a massive amount of data privately due to its decentralized structure, which is adept meaningfully preserving users' sensitive data from being visible to opponents. However, sensitive data can still be disclosed by exploring uploaded parameters from participators during the training process. One the other hand, as a mechanism to improve the security of data privacy, DP has been widely studied to make the model more secure and protect sensitive information of users.

There are two main models to implement DP with FL, which are local model and central models. For local differential privacy (LDP) model, each user trains its data and implements DP before uploading the parameter to untrusted server. In this model, all clients enjoy with high privacy but the server suffers from low utility due to the huge amount of noise that has been added to clients [Zhao et al. (2019)]. In central differential privacy (CDP) model, all users train their own data and upload the local parameters to trusted server without adding any noise, then the server aggregates the users' parameters and adds DP for aggregated model [McMahan et al. (2018), Xixi et al. (2020)]. In this model, the server enjoys with high accuracy but the users suffer from low privacy due to trusted server.

Here we give an example of FL-LDP. The framework of FL-LDP is shown in Figure 4 in details according to [Wei et al. (2020)]. The training process of LDP-FL usually takes the following five steps:

1) The server distributes the initial weight $w_0$ to all users, then each user starts to train its data with the initial weight and get the optimum local model by using gradient descent method $w_i^t = \arg(\min_{w_i} f_i(w_i)) + (\mu/2))|| w_i - w^{t-1}||^2$ .
2) Each user clips the local parameter from local model $w_i^{(t)} = w_i^t / \max(1, || (w_i^t / C)||)$.
3) Each user adds DP mechanism for the local parameter by using Laplace mechanism before sending it to analyzer $\widetilde{w_i^t} = w_i^t + \mathrm{lap}(\Delta f/\varepsilon)$.
4) The server aggregates the noised parameters from users $w_g = \sum_{i=1}^{k} p_i \widetilde{w_i^t}$.
5) The aggregator broadcasts the global parameter $w_g$ to k users for retraining their data on it.

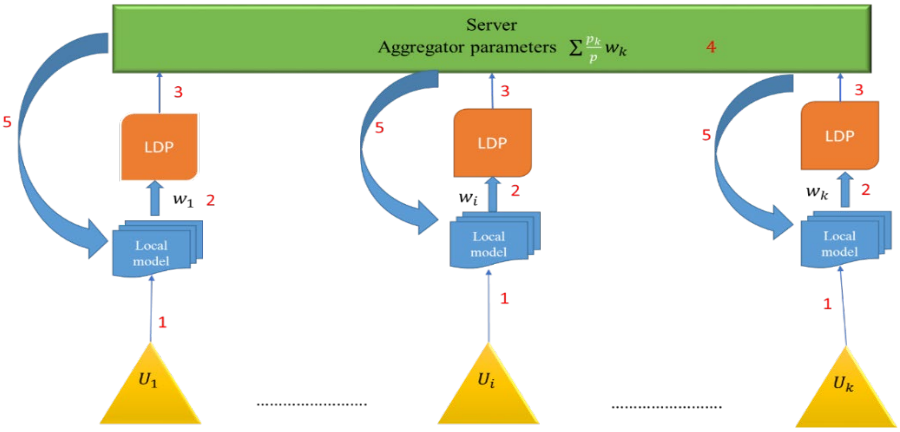


**Figure 4** Federated learning with local differential privacy

**Algorithm 2 : FL with LDP**

Data: Let T be a communication round, $w_0$ be the initial weight, $\mu$ be hyper parameter, and $\varepsilon$ and $\delta$ be privacy parameters.

```
1)  Initialization: $t = 1$ ;$w_i^0$ , $\forall i$
2)  While $0 < t \leq T$ do
3)  While $k_i \in \{k_1,, k_2, . . . , k_n\}$ do     // k is number of users.
4)  $w_i^t = \arg(\min_{w_i} f_i(w_i)) + (\mu/2))|| w_i - w^{t-1}||^2$
5)  $w_i^{(t)} = w_i^t / \max(1, || w_i^t / C||)$
```

```
6)    $\widetilde{w_i^t}$=wᵢᵗ + lap(Δf/ε) is sensitivity function.
7)    $w_g = \sum_{i=1}^{k} p_i \widetilde{w_i^t}$.
8)    for $k_i \in \{k_1,, k_2, . . . , k_n\}$ do
9)    Test the aggregating parameter $w_g$ using local dataset $\min_{wi} f_i(w_g)$
10)   End
11)   t ← t + 1
Result: return $w_g$.
```

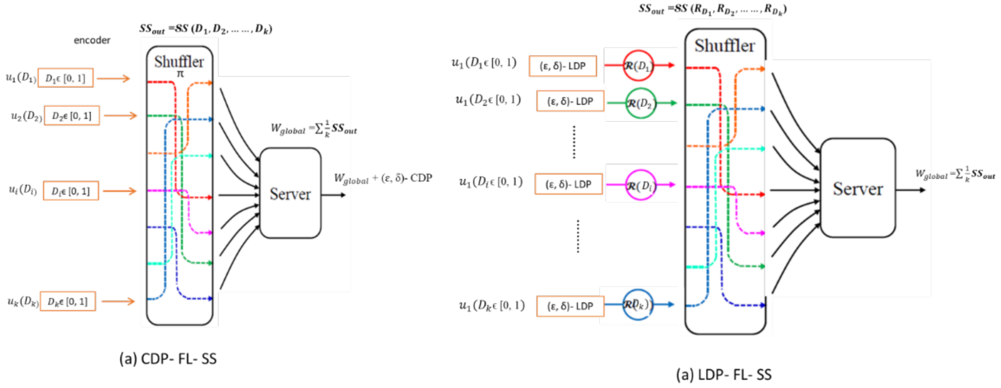
# 5. DIFFERENTIALLY PRIVATE FEDERATED LEARNING WITH THE SHUFFLE MODEL

Integrating DP with FL, it still not be a perfect solution. The recent studies suggest an intermediate model between users and the analyzer to eliminate the weaknesses points in both DP and LDP by reducing the gap between privacy and utility. This intermediate model called shuffle model. In practically, the shuffle model has achieved good results in privacy amplifications with higher accuracy than DP-FL system without shuffle model. In this survey, we shall discuss the influence of privacy on FL, starting from FL with DP to adding shuffle model in order to achieve a better balance between privacy and accuracy.

The framework of FL-DP in the shuffle model has attracted lots of attention recently. To see the advantages of the shuffle model toward DP-FL, it requires a thorough understanding DP-FL-SS.

## 5.1. THE PRIVATE MULTI-MESSAGES FL IN A SHUFFLE MODEL

In this subsection, we shall introduce the private multi-message in a shuffle model [Balle et al. (2019)]. For the framework, please see the following Figure 5.


$SS_{out} = SS(D_1, D_2, ......, D_k)$
encoder
Shuffler
π
$u_1(D_1)$ $D_1 \epsilon [0, 1]$
$u_2(D_2)$ $D_2 \epsilon [0, 1]$
$u_i(D_i)$ $D_i \epsilon [0, 1]$
$u_k(D_k)$ $D_k \epsilon [0, 1]$
$W_{global} = \sum \frac{1}{k} SS_{out}$
Server
$W_{global}$ + (ε, δ)- CDP
$SS_{out} = SS(R_{D_1}, R_{D_2}, ......, R_{D_k})$
Shuffler
$u_1(D_1 \epsilon [0, 1)$ (ε, δ)- LDP $\mathcal{R}(D_1)$
$u_1(D_2 \epsilon [0, 1)$ (ε, δ)- LDP $\mathcal{R}(D_2)$
$u_1(D_i \epsilon [0, 1)$ (ε, δ)- LDP $\mathcal{R}(D_i)$
$u_1(D_k \epsilon [0, 1)$ (ε, δ)- LDP $\mathcal{R}(D_k)$
Server
$W_{global} = \sum \frac{1}{k} SS_{out}$

(a) CDP- FL- SS

(a) LDP- FL- SS

**Figure 5** (a) Center differential privacy and (b) Local differential privacy with Shuffle model

Both frame have their own pron and con. One approach is to apply secure multi-party computation to simulate central model algorithms while the other is to provide accuracy in local model. Both models include differential privacy with shuffle model (DP-SS). In [Bittau et al. (2017)], a special case of encoder shuffle analyze framework has been considered. At first, users perform a local randomized encode and then the users generalize the randomization messages. Then SS collects messages from users and randomly permutes it by a permutation π. Afterwards, the analyzer aggregates the users' messages after shuffling. From analysis aspect, SS handles the problem of accuracy limitations on local algorithm with protective many of their necessary attributes under natural restrictions. In [Balcer and Cheu (2019), Ghazi et al (2020), Ghazi et al. (2020)], the overall protocol of DP-SS has been studied. This protocol contains three parts P = R, SS, A, where R: D →$Y^k$, SS: $Y^k$ → $SS_{out}$, A: $SS_{out}$ → O, R is local randomizer, D is dataset, Y is the output space after randomization, k is the number of users, SS is the shuffle, A is the analyzer of P which collects shuffled output and getting the optimal global model and O is the analyzer output. In DP-SS, users only need to trust that the shuffler acts as intended and sufficiently many of their peers follow the protocol.

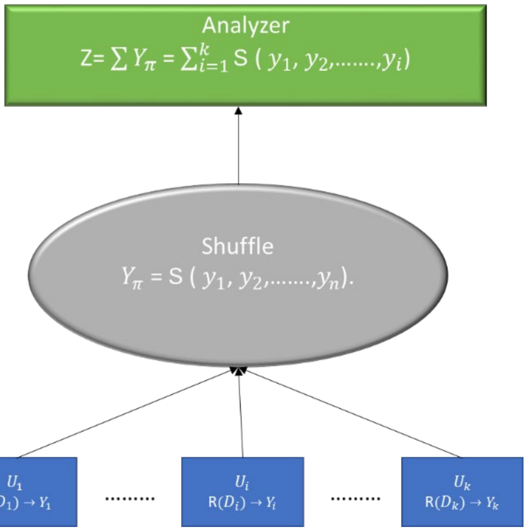

Analyzer
$Z = \sum Y_\pi = \sum_{i=1}^{k} S\,(y_1, y_2, \ldots\ldots, y_i)$
Shuffle
$Y_\pi = S\,(y_1, y_2, \ldots\ldots, y_n).$
$U_1$
$R(D_1) \to Y_1$
.........
$U_i$
$R(D_i) \to Y_i$
.........
$U_k$
$R(D_k) \to Y_k$

**Figure 6** DP-FL in the shuffle model

The mechanism that represents the overall protocol of the framework in Figure 6 starts from users' data until gets final model from analyzer is P: D → O. The main implementation is as follow:

1) Assume that we have k users, which has its own data $D_i \in D$, where i denoted to the number of users. Each one encodes his data $D_i \in [0, 1]$ and creating DP according to its dataset. For adjacent datasets D, D', the encoder randomization is R: D→$Y^k$. If ever user transmits only one message every round, the output of this randomization is a vector of messages Y = $y_1$, $y_2$, ....., $y_k$ = ($R_{D1}$, $R_{D2}$, ...., $R_{Dk}$), which is the input for SS.

2) The shuffle receives Y messages and then make a random permutation π for the messages vector. The output after the permutation is $Y_\pi$ = SS ($y_1$, $y_2$, ....., $y_k$).

3) The analyzer receives $Y_\pi$ from shuffle and aggregates it to create the global model $w_g = \Sigma(1/k)Y_\pi$, then distributes the global model parameters to users via SS. As showed in last step, the randomization comes from both local randomizing from users side and permutation π from SS. Thus, the formula of the overall protocol P can be written as P = A, *SS* ($R_{D1}$, $R_{D2}$, ...., $R_{Dk}$) . Let

the mechanism be $\mathcal{M}_R = SS\ (R_{D1}, R_{D2}, \ldots., R_{Dk})$. Then the overall protocol is P = A, $\mathcal{M}_{R,}$ which is shown in Algorithm 3.

**Definition 5** (SS towards DP-FL [Cheu et al. (2019)]) Let the mechanism $\mathcal{M}_R = S,\ R^k$ detect the sight of the analyzer in an implementation of the protocol P, where $R^k = (R_{D1}, R_{D2}, \ldots., R_{Dk})$ is local randomizer, and SS is the shuffle permutation. We can say that if the randomized mechanism $\mathcal{M}_R$ gurantees (ε, δ)-DP if for every two adjacent datasets D, D', for all output $v \subseteq$ Range $(\mathcal{M}_R)$, we have

$$\Pr[\mathcal{M}_R\ (D) \in v] \leq e^{\epsilon} \Pr[\mathcal{M}_R\ (D') \in v\ ] + \delta \qquad (3)$$

**Algorithm 3 : LDP - FL via shuffle model**

**# Local Randomizer in user side**

Input: Let D = $(D_1, D_2, \ldots, D_k)$ be users' data sets, k be a number of users, $y_i$ be user output after adding LDP $y_i \in \{0,1\}$, and $w_0$ be server initial weight.

```
While $1 \leq i \leq k$ do
Analyzer distributes the initial weight ($w_0$) to all users via shuffle
For $D_i \in (D_1, D_2, \ldots, D_k)$
Every user creates local update of initial weight $D_i = f_i(w_0)$
Encoding local updates by each user $D_i \in \{0,1\}$
$(D_i + C)/2C \rightarrow D_i^*$    # clipping parameters
$R(D_i) = D_i^* + lap(\Delta f/\varepsilon)$    # local randomization
$R(D_i) \rightarrow y_i$
Return $y_i$
```

**# Shuffle**

```
$Y = (y_1, y_2, \ldots., y_k)$
$Y\pi = SS\ (y_1, y_2, \ldots., y_k)$.
```

**# Analyzer**

```
$Z^* = \Sigma(1/k)Y_\pi$
Normalize $C(2Z^* - 1) \rightarrow Z$
Return Z.
```

Updates model.

## 5.2. THE PRIVATE MULTI MESSAGES FL VIA M PARALLEL SHUFFLE MODELS

In this subsection, we shall introduce the private multi messages FL via m parallel shuffles models [Balle et al. (2020)]. In this type, *m* parallel shuffles have been introduced to receive multi messages from each user instead of using single shuffle, which further improves the accuracy and communication more than using single shuffle.

Suppose that there are k users. If each one randomizes its data set $D_i$, where $R: D \rightarrow Y^k$, and all users send k randomized messages to m parallel shufflers, then each shuffle perturbs randomly k messages $SS^m: Y^k \rightarrow Y^k$ and the output is $SS_{output} = SS^m$

$(y_1,y_2,……..,y_k)$. The analyzer receives km perturbed messages from m parallel shuffle A: $(Y^k)^m \rightarrow (SS)_{output}$ , then aggregates it to create the optimal global model. The framework of this technique explained in Figure 7 .

**Definition 6.** (The Communication model of multi messages in m parallel shuffles models [Balle et al. (2020)]): The protocol which presents this communication model of multi messages in m parallel shuffle is $P = (R, A)$ if let D be the user dataset, where $D = (D_1, D_2, ……, D_k)$. Each user i randomizes its own sensitive data by using local randomization to obtain vector of messages $Y= (y_1, y_2,……., y_k)= R(D_k)$. The users then send the vector of messages vector to independent parallel shuffles $SS^i$: $Y^k \rightarrow Y^k$, i ϵ [m] number of the shufflers. Each shuffler perturbs randomly k messages that receive from users $Y^i_\pi= SS^i (y_1{}^i, y_2{}^i,……, y_k{}^i)$. The analyzer then aggregates all parallel shufflers output results. In summary, the output of P(D) is given by $P(D)= A, SS^m, R^m(D) = A(SS^m (y_1{}^m ,y_2{}^m,…,y_k{}^m))$.

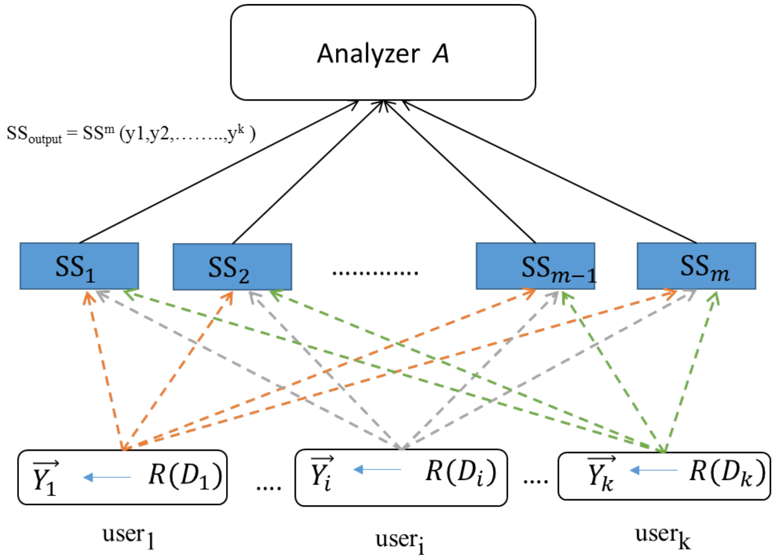


**Figure 7** Privacy multi messages in m parallel shuffle

From the privacy aspect, the overall privacy of model has been divided to two parts, including privacy from randomization in users' data, and the privacy from perturbation by using m shufflers. So the shuffle models play a great role in boosting privacy for users' messages before uploading it to the analyzer. Therefore the total randomization comes from two aspects of DP view at users and permutation π from m shufflers. To prove the privacy, we will refer to the mechanism $\Upsilon_{R=} SS^m (y_1{}^m ,y_2{}^m,………y_k{}^m)$ is the total randomization mechanism which captures the sight of the analyzer in an performance of the protocol. Note that we can rewrite the overall mechanism as$\mathcal{M}_p$ (D) = $A, \Upsilon_R$.

**Definition 7** (DP-m shuffles [Balle et al. (2020)]): Assume that the mechanism $\Upsilon_R = SS^m, R^m(D)$, is the analyzer view as an implementation of the protocol P(D). We can say that the mechanism $\mathcal{M}_p$ (D) guarantees (ε, δ)-DP if for every two adjacent datasets D, D', and all output S ⊆Range ($\Upsilon_R$),

$$\Pr[\Upsilon_R (D) \in S] \leq e^{\epsilon} \Pr[\Upsilon_R (D') \in S ] +\delta. \qquad (4)$$

Note that, if m =1 means the total randomization $\Upsilon_{R=}$ SS ($y_{1,}y_2$,………$y_k$), which is a single shuffle. According to [14], the resulting protocol after shuffling satisfies ($\epsilon$, $\delta$)-DP with $\varepsilon$= min ($\varepsilon_0$, 1) $e^{\varepsilon 0}\sqrt{\log(1/\delta)/k}$, if the mechanism $\Upsilon_R$ satisfies ($\epsilon_0$-DP).

## 5.3. THE ACCURACY IN MULTI MESSAGES SHUFFLE MODEL(S)

In this section, we shall present the accuracy in multi messages shuffle model(s). To quantify the effect of this protocol in accuracy, we should measure the mean square error (MSE) at the worst case as the following definition.

**Definition 8** (MSE in SS [Ghazi et al. (2020)]): Let *k* be the number of users, which has real number data set $D_i$, where $D_i \epsilon$ [0, 1]. If the protocol P of m messages in the shuffle model is quantified by a pair of algorithms P = (R, A), then the calculating the overall model accuracy in terms of MSE worst instance in each P is

$$\text{MSE (P)} = \text{Sup}_{D_i\,\epsilon\,[0,\,1]}\, E[(\mathcal{M}_p\,(D) - \textstyle\sum_{i=1}^{k} D_i)^2], \tag{5}$$

Where $\sum_{i=1}^{k} D_i$ is the real Summation and the prediction is finish the stochastic in the overall protocol $\mathcal{M}_p$ (D).

For CDP, the Laplace mechanism guarantees ($\varepsilon$, $\delta$)-DP in the case of real summation with MSE = $O$ (1) and this is the optimal value. While LDP raises the tradeoff between accuracy and communication privacy by computing both in terms of the messages number per client and size of these messages. If the user send $O$ (log(k/$\delta$)), the MSE is $O(1/\epsilon^2)$ in LDP model, which is much smaller than $O(\log(k/\delta))^2$ [Balle et al. (2020)].

## 5.4. THE ACCURACY IN MULTI MESSAGES SHUFFLE MODEL FOR BINARY SUMMATION

In this section we shall show how shuffle model solved the problem of binary summation which FL-LDP failed to solve it. The setting of this problem is that if we have k users, each one has dataset $D_i$, where i $\epsilon$ [k], and each one encodes his data $D_i \epsilon$ [0, 1] bits. The purpose is to calculate the sum of bits in analyzer [Balle et al. (2019)].

To implement DP, the randomized response is the official local privacy for this problem. If D, D' are neighboring datasets that differ in only one record. The encoder randomization is R(D)→$y^k$. Let user i send the message $y_i$. Due to subsampling and noise accumulation, the predictable value of summation $y_i$ in the analyzer is $\sum y_i$= (1 - *p*) + *k/p*=2, where p is the probability of random response. The analyzer obtains an unbiased estimator as follows

$$A_{rr}(Y) = (1/(1\text{-p}))\textstyle\sum_{i=1}^{k} (D_i + \text{kp}/2), \tag{6}$$

$$E[A_{rr}(Y)] = (1/(1\text{-p}))\, E[\textstyle\sum_{i=1}^{k} (D_i + \text{kp}/2)] = \sum_{i=1}^{k} y_i. \tag{7}$$

The local model view, LDP satisfy (ε, δ)-DP at P → 2/($e^{\varepsilon}$ + 1) with optimal error $O$ ((1/ε) $\sqrt{k}$). It means the error increased when k increased [Meehan et al. (2021), 65].

According to [Cheu et al. (2019)], after adding shuffle model, the randomized response $P_{RR}$ = ($R_{RR}$, $A_{RR}$) is a single-message protocol, where $R_{RR}$ is the local randomization of users data and shuffle model $R_{RR}$ = SS(Y)= SS ($y_1$, $y_2$, ..., $y_k$) and $A_{RR}$ is the analyzer that collect perturbed messages from the shuffler. The mechanism $P_{RR}$ satisfies (ε, δ)-DP with optimal error value $O((1/\varepsilon)\sqrt{1/\delta})$, which means the computing binary summing up to additive error $O((1/\varepsilon)\sqrt{1/\delta})$ with constant probability.

## 5.5. RELATED WORKS ON DP-FL WITH SHUFFLE MODEL(S)

In real word, we have many tasks need a huge number of users to participate in the model e.g. (Microsoft, Apple, Google, .etc). So reducing the huge amount of noise in local model with achieving good privacy has inspired a current work for alternative models. For instance, ESA model that proposed by Bittau *et al* [Bittau et al. (2017)], has been presented a trusted SS to collect secure messages from users and perturbing it randomly before uploading to untrusted server.

Due to the importance of DP-FL-SS, it attracts lots of attention from academia and Industries [Balcer et al. (2020)]. Up to date, there are some published papers on this topic. For the local model, the famous example for that is the problem of privately summation of sum bounded real values among k different users, the center model achieved $O$ (1) error [Cheu et al. (2019)]. While the local model achieves $O$ ((1/ε)$\sqrt{k}$) error [Amos et al. (2008)], and recently Balle *et al* [Balle et al. (2019)], proves that the single message per participate can achieve only $O$ $(k)^{1/3}$ MSE in local model.

Since the privacy amplification plays a great role in implementing DP mechanism, the recent studies suggested many approaches that provide privacy amplification such as iteration [Xixi et al. (2020)] and subsampling [Balle et al. (2018)]. In [Cheu et al. (2019)], it focuses on amplification by shuffling to tradeoff between privacy in local model and accuracy in central model. It presents the single-message shuffle model with the analytic study of SS in distributed DP algorithms for summation of binary and real valued inputs. Its results are the summation of $O$ ((1/ε)$\sqrt{k}$) messages per user suffice to achieve $O$ (1) error in the curator model. In particular, it reduces the gap between LDP and CDP models by adding SS in between users and analyzers, which perturbs users' messages before uploading it to analyzer. While SS can solve the private summation problem more correct than the local model. In [13], it focuses on single-message shuffle model protocols that provides the privacy blanket via shuffle model under setting randomizing then shuffling that copies k LDP with $\varepsilon_0$= $O$ (log(k/ log(1=δ))) produces an ε = $O$ (min($\varepsilon_0$, 1)) $e^{\varepsilon 0}$ $\sqrt{\log(1/\delta)/k}$. This protocol obtains better accuracy and communication than the proposed protocols by Cheu *et al* [Cheu et al. (2019)] for the same problem.

In [Erlingsson et al. (2019)], the privacy cost of LDP has been considered and it is much smaller than CDP view. It has showed that if the perturbed mechanism satisfies ($\varepsilon_l$, $\delta_l$)- LDP, then it would satisfy ($\varepsilon_c$, $\delta_c$)- CDP with $\varepsilon_c = O$ ($\varepsilon_l\sqrt{\log(1/\delta_c)/k}$ ). Consequently, $\varepsilon_l$ was reduced to $\varepsilon_c/\sqrt{k}$. The most important achievements of shuffling is the privacy amplification by shuffling [Balle et al. (2019), Meehan et al. (2021), Erlingsson et al. (2019), Bittau et al. (2017)]. Where In the shuffle model, if the local randomized(R) satisfies ($\varepsilon_l$, $\delta_l$)–LDP at $\varepsilon_l \leq \log$ (k/log (1/$\delta_c$))/2, Mechanism $\mathcal{M}$ satisfies ($\varepsilon_c$, $\delta_c$)-DP at $\varepsilon_c = O$ (min ($\varepsilon_l$, 1) $e^{\varepsilon l}\sqrt{\log(1/\delta_c)/k}$, where $\varepsilon_l$ is the privacy budget in local model and $\varepsilon_c$ is the privacy budget in center model.

In [Liu et al. (2020)], the subsampling in shuffle model has been considered to trade off between privacy and utility. In [Balle et al. (2018)], the privacy has been amplification by subsampling, if the mechanism $\mathcal{M}$ : $X^m \rightarrow Y$ satisfies ($\varepsilon$, $\delta$)-DP, with respect to the replacement relationship on sets of size m, the mechanism $\mathcal{M}$ satisfies (log (1+(m/k) ($e^{\varepsilon}$-1)), (m/k) $\delta$)-DP, where m is number of subsampling, k is number of users. They proposed three different protocols (SS-simple protocol, SS-double protocol and top-k protocol) to randomize users' data by implementing DP. The key difference between three protocols is the assumptions in random sub sampling. In SS-simple protocol, it assume that all dimensions are equally and this may discard important dimensions and occurring losses in sensitive users' data. The practical result of this model provides an insufficient privacy amplification effect in FL for the data that has large dimensions. To fix this problem, SS-double protocol has been proposed to increase the privacy amplification by creating subsampling with m dimensions in each user data and each user perturbs m dimensions instead of perturbing one dimension in SS-simple protocol. Moreover, in order to boost the accuracy when the model size is larger than the user population, an advanced technique called SS-Topk has been proposed, which can solve SS-simple protocol and SS-double protocol problems. It selects top-k indexes with greatest absolute magnitudes over the input vector, this protocol boosting the privacy amplification with achieving the same privacy level.

In 2020 Borja Balle *et al* [Balle et al. (2020)], it studies a multi-message private summation in shuffle model which is a new technique for actual summation in the shuffle model. It improves the communication and accuracy by squeezing a single round protocol of multi message where the approximating sum from each subsequent message makes available the earned error estimation with the prior messages. This technique presents a direct tradeoff between the number of messages and the final accuracy sent by each user. By succeeding in reducing MSE to be as less as $O$ ((log log k)$^2$) where each user send $O$(loglog(k))messages and each message has a size $O$(log k).

According to our discussion of recent works about shuffle model through this section, we can say SS succeed in achieving high privacy in FL compared with LDP-FL and CDP-FL, because, besides shuffling step, SS requires from users to provide secure messages carefully. This is unlike with the global model. In global model, its responsibility is exclusively conducted by the reliable analyzer. In addition, from theoretical view, this model provides implementing easier mechanisms that are easy to explain, implement and verify. So we trust that DP-FL-SS is an interest for both theoretical and practical aspects.

## 6. CONCLUSIONS

FL has a greatly facilitated the progress and development of a huge amount data in machine learning by permitting for many users to participate in the model. For strong privacy in FL model, DP was broadly proposed to keep users' data without data leaking during training process. The security of users' data and the model updates are secured by LDP and CDP algorithms implementing during the training process. Both has their own specific weaknesses and strengths points. For instance, CDP is weaker to an adversary, whereas LDP can keep the users updates before transfer them to the analyzer. However, LDP is sensitive to noise, and higher noise can influence the model's accuracy. So integrating differential privacy with federated learning is still not be the perfect solution. The recent studies suggest an intermediate model between users and analyzer to eliminate the weaknesses points in both DP and LDP by reducing the gap between privacy and utility. This intermediate model is called shuffle model. This paper provides a comprehensive survey of DP-FL-SS, including DP-FL-SS models, data statistics and the recent progress of private shuffle model from different aspects. Moreover we discussed the practical results of SS effect in exploring the gap between the local and curator model in DP-FL. We trust that our survey will be very important and urgent for future research in FL and also will help the newcomers to understand the complicated discipline of this active research zone.

## REFERENCES

Aledhari M., Razzak R., Parizi R., Saeed F. (2020), Federated learning: a survey on enabling technologies, protocols, and applications, IEEE Access, 8, 140699-140725. Retreived from https://doi.org/10.1109/ACCESS.2020.3013541

Amos B., Kobbi N., and Eran O. (2008), Distributed private data analysis: Simultaneously solving how and what, 28th Annual International Cryptology Conference, Springer, 5157, 451-468.

Amos Beimel., Iftach Haitner., Kobbi Nissim. (2020), Uri Stemmer., On the Round Complexity of the Shuffle Model, arXiv: 2009.13510. Retreived from https://doi.org/10.1007/978-3-030-64378-2_24

Balcer V., Cheu A. (2019), Separating local & shuffled differential privacy via histograms, arXiv: 1911.06879.

Balle B., Barthe G., Gaboardi M. (2018), Privacy amplification by sub sampling: Tight analyses via couplings and divergences, ArXiv:1807.01647.

Balle B., Bell J., Gascon A., Nissim K. (2019), The privacy blanket of the shuffle model, arXiv: 1903.02837, Retreived from https://doi.org/10.1007/978-3-030-26951-7_22

Bittau A., Erlingsson U., Maniatis P., Mironov I., Raghunathan A., Lie D., Rudominer M., Kode U., Tinnes J., Seefeld B. (2017), PROCHLO: Strong privacy for analytics in the crowd. In Proceedings of the Symposium on Operating Systems Principles (SOSP), 441-459. Retreived from https://doi.org/10.1145/3132747.3132769

Borja Balle, James Bell, Adria Gascon, Kobbi Nissim (2019). Differentially Private Summation with Multi-Message Shuffling. arXiv:1906.09116v1 [cs.CR]. Retreived from https://doi.org/10.1145/3372297.3417242

Borja Balle., James Bell., Adrià Gascón., and Kobbi Nissim. (2020), Private Summation in the Multi-Message Shuffle Model. In Proceedings of the 2020ACM SIGSAC Conference on Computer and Communications Security ,Virtual Event, USA. 9-13, 2020. Retreived from https://doi.org/10.1145/3372297.3417242

C. Dwork. (2011), Affirm foundation for private data analysis, Communications of the ACM, 54(1), 86-95. Retreived from https://doi.org/10.1145/1866739.1866758

Casey Meehan., Amrita Roy Chowdhury., Kamalika Chaudhuri., Somesh Jha. (2021), A Shuffling frame work for local differential privacy, arXiv: 2106.06603v1.

Chen M., Yang Z., Saad W., Yin C., Poor H V., Cui S. (2021), A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks. IEEE Transactions on Wireless Communications, 20(1), 269-283. Retreived from https://doi.org/10.1109/TWC.2020.3024629

Cheu A., Smith A., Ullman J., Zeber D., Zhilyaev M. (2019), Distributed differential privacy via shuffling, 11476, 375-403. Retreived from https://doi.org/10.1007/978-3-030-17653-2_13

David Byrd. (2020), Antigoni Polychroniadou., Differentially Private Secure Multi-Party Computation for Federated Learning in Financial Applications, arXiv: 2010.05867.

Ding B., Kulkarni J., Yekhanin S. (2017), Collecting telemetry data privately, in Adv. Neural Inf. Process. Syst., Long Beach, CA, 3571-3580.

Dwork C. (2006), Differential Privacy, Proceedings of the 33rd international conference on automata, languages and programming, 2, 1-12. Retreived from https://doi.org/10.1007/11787006_1

Dwork C. (2008), Differential privacy: a survey of results, International conference on theory and applications of models of computation, 4978, 1-19. Retreived from https://doi.org/10.1007/978-3-540-79228-4_1

Erlingsson Ú., Pihur, V., Korolova, A. (2014), Rappor: Randomized aggregable privacy-preserving ordinal response, Proceedings of the 2014 ACM SIGSAC Conference on computer and communications security, Scottsdale, AZ, USA, 1054-1067. Retreived from https://doi.org/10.1145/2660267.2660348

Farokhi., Farhad. (2021), Distributionally-robust machine learning using locally differentially-private data, springer Science and Business Media LLC. 1-13. Retreived from https://doi.org/10.1007/s11590-021-01765-6

Geyer R C., Klein T., Nabi M. (2020), Differentially Private Federated Learning: A Client Level Perspective, IEEE Access, 8, 140699-140725. Retreived from https://doi.org/10.1109/ACCESS.2020.3013541

Ghazi B., Golowich N., Kumar R., Manurangsi P., Pagh R., Velingker A., Pure differentially private summation from anonymous messages, arXiv:2002.01919, 2020. Retreived from https://doi.org/10.1007/978-3-030-45724-2_27

Ghazi B., Golowich N., Kumar R., Pagh R., Velingker A. (2020), On the power of multiple anonymous messages, arXiv:1908.11358.

Ghazi B., Golowich N., Kumar R., Manurangsi P., Pagh R., Velingker A. (2020), Pure differentially private summation from anonymous messages, arXiv:2002.01919.

Ghazi B., Manurangsi P., Pagh R., Velingker A. (2020) Private aggregation from fewer anonymous messages. 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, Proceedings, 47,798-827. Retreived from https://doi.org/10.1007/978-3-030-45724-2_27

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang (2018), "Learning Differentially Private Recurrent Language Models," in arXiv:1710.06963 [cs].

H. B. McMahan., D. Ramage., Talwar K., Zhang L. (2018), Learning Differentially Private Recurrent Language Models, arXiv:1710.06963.

Hard A., Rao K., Mathews R., et al. (2018), Federated learning formable keyboard prediction. arXiv preprint arXiv:1811.03604.

Huang L., A. L. Shea., Qian H., Masurkar A., Deng H., Liu D. (2019), Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records, J. Biomed. Informat, 99, 1-21. Retreived from https://doi.org/10.1016/j.jbi.2019.103291

Huang Xixi., Ding Ye., Jiang Zoe L., Qi Shuhan., Wang Xuan., Liao Qing. (2020), DP-FL: a Novel differentially private federated learning framework for the unbalanced data World Wide Web-internet and Web Information Systems, 23(4), 2529-2545. Retreived from https://doi.org/10.1007/s11280-020-00780-4

Jiang L., Tan R., Lou X., Lin G. (2019), On lightweight privacy preserving collaborative learning for Internet-of-Things objects, in Proc. Int. Conf. Internet Things Design Implement., 70-81. Retreived from https://doi.org/10.1145/3302505.3310070

Jordan Awan., Aleksandra Slavkovi'c. (2018), Structure and Sensitivity in Differential Privacy: Comparing K-Norm Mechanisms, arXiv:1801.09236.

Kairouz P., McMahan H B., et al. (2021) Advances and Open Problems in Federated Learning. Foundations and Trends in Machine Learning, 14(1), 1-119. Retreived from https://doi.org/10.1561/2200000083

Li T., Sahu K., Talwalkar A., Smith V. (2020), Federated learning challenges, methods, and future directions, IEEE Signal Processing Magazine, 37(3), 50-60. Retreived from https://doi.org/10.1109/MSP.2020.2975749

Li W., Milletarì F.,Xu D.,Rieke N .,Hancox J., Zhu W., Baust M., Cheng Y .,Ourselin S.,J C M ., Feng A. (2019), Privacy-preserving federated brain tumor segmentation, in Proc. Int. Workshop Mach. Learn. Med. Image, 11861,133-141. Retreived from https://doi.org/10.1007/978-3-030-32692-0_16

Liu D., Miller T., Sayeed R., Mandl K. (2018), FADL: Federated autonomous deep learning for distributed electronic health record, arXiv:1811.11400.

Liu F. (2019) Generalized Gaussian Mechanism for Differential Privacy, in IEEE Transactions on Knowledge and Data Engineering, 31, 747-756. Retreived from https://doi.org/10.1109/TKDE.2018.2845388

Liu R., Cao Y. , Chen H., Guo R., Yoshikawa M. (2020), FLAME: Differentially Private Federated Learning in the Shuffle Model, arXiv:2009.08063.

McMahan H., Moore E., Ramage D., Hampson S., and Arcas B. (2016), Communication efficient learning of deep networks from decentralized data, arXiv:1602.05629.

Nakamoto S. (2019), Bitcoin: A Peer-to-Peer Electronic Cash System.Manubot;Online. Retreived from https://metzdowd.com.

NhatHai Phan., Xintao Wu., Han Hu., Dejing Dou. (2017), Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning" IEEE International Conference on Data Mining 17, 2374-8486.

Qi Liu., Juan Yu., Jianmin Han., Xin Yao. (2021), Differentially private and utility-aware publication of trajectory data, Expert Systems with Applications, 180, 1-14. Retreived from https://doi.org/10.1016/j.eswa.2021.115120.

Ramaswamy S.,Mathews R., Rao K., Beaufays F. (2019), Federated learning for emoji prediction in amobile keyboard, arXiv:1906.04329.

Ren J., Wang H., Hou T., Zheng S., Tang C. (2019), Federated learning-based computation offloading optimization in edge computing supported Internet of Things, IEEE Access, 7, 69194-69201. Retreived from https://doi.org/10.1109/ACCESS.2019.2919736.

Sarfaraz., Aaliya., Chakrabortty., Ripon K., Essam., Daryl L. (2021), A tree structure-based improved block chain framework for a secure online bidding system, 102, 1-20. Retreived from https://doi.org/10.1016/j.cose.2020.102147

Ulfar Erlingsson., Vitaly Feldman., Ilya Mironov., Ananth Raghunathan., Kunal Talwar. (2019), and Abhradeep Thakurta., Amplification by shuffling: From local to central differential privacy via anonymity. the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, 2468-2479. Retreived from https://doi.org/10.1137/1.9781611975482.151

Victor Balcer., Albert Cheu., Matthew Joseph., and Jieming Mao. (2020), Connecting robust shuffle privacy and pan-privacy. arXiv:2004.09481. Retreived from https://doi.org/10.1137/1.9781611976465.142

Vitaly Feldman., Ilya Mironov., Kunal Talwar., Abhradeep Thakurta. (2018), Privacy amplification by iteration, In 59th IEEE Annual Symposium on Foundations of Computer Science, Paris, France, 521-532. Retreived from https://doi.org/10.1109/FOCS.2018.00056

Wei K., et al. (2020), "Federated Learning With Differential Privacy: Algorithms and Performance Analysis," in IEEE Transactions on Information Forensics and Security, 15, 3454-3469. Retreived from https://doi.org/10.1109/TIFS.2020.2988575

Yang Q., Liu Y., Chen T., et al. (2019), Federated machine learning: concept and applications, ACM Transactions on Intelligent Systems and Technology, 10(2), 1-19. Retreived from https://doi.org/10.1145/3298981

Yang T., Andrew G., Eichner H., et al. (2018), Applied federated learning: improving google keyboard query suggestions, arXiv:1812.02903.

Zhao L., Wang Q., Zou Q., Zhang Y., Chen Y. (2020), Privacy-Preserving Collaborative Deep Learning With Unreliable Participants, in IEEE Transactions on Information Forensics and Security, 15, 1486-1500. Retreived from https://doi.org/10.1109/TIFS.2019.2939713

Zhao. P., Zhang, G., Wan, S., Liu, G., Umer, T. (2019), A survey of local differential privacy for securing internet of vehicles. J. Supercomputer, 76, 1-22. Retreived from https://doi.org/10.1007/s11227-019-03104-0

Zhu T., Ye D., Wang W., Zhou W., Yu P. (2020), More Than Privacy: Applying Differential Privacy in Key Areas of Artificial Intelligence, IEEE Transactions on Knowledge and Data Engineering. doi: 10.1109/TKDE.2020.3014246. Retreived from https://doi.org/10.1109/TKDE.2020.3014246