# A Generalized Shuffle Framework for Privacy Amplification: Strengthening Privacy Guarantees and Enhancing Utility

**E Chen**[1], **Yang Cao**[2, *], **Yifei Ge** [3]
[1] Zhejiang Lab
[2] Hokkaido University
[3] Xi'an Jiaotong-Liverpool University
chene@zhejianglab.com, yang@ist.hokudai.ac.jp, Yifei.Ge23@student.xjtlu.edu.cn

## Abstract

The shuffle model of local differential privacy is an advanced method of privacy amplification designed to enhance privacy protection with high utility. It achieves this by randomly shuffling sensitive data, making linking individual data points to specific individuals more challenging. However, most existing studies have focused on the shuffle model based on $(\epsilon_0, 0)$-Locally Differentially Private (LDP) randomizers, with limited consideration for complex scenarios such as $(\epsilon_0, \delta_0)$-LDP or personalized LDP (PLDP). This hinders a comprehensive understanding of the shuffle model's potential and limits its application in various settings. To bridge this research gap, we propose a generalized shuffle framework that can be applied to any $(\epsilon_i, \delta_i)$-PLDP setting with personalized privacy parameters. This generalization allows for a broader exploration of the privacy-utility trade-off and facilitates the design of privacy-preserving analyses in diverse contexts. We prove that shuffled $(\epsilon_i, \delta_i)$-PLDP process approximately preserves $\mu$-Gaussian Differential Privacy with $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n} \frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$. This approach allows us to avoid the limitations and potential inaccuracies associated with inequality estimations. To strengthen the privacy guarantee, we improve the lower bound by utilizing *hypothesis testing* instead of relying on rough estimations like the Chernoff bound or Hoeffding's inequality. Furthermore, extensive comparative evaluations clearly show that our approach outperforms existing methods in achieving strong central privacy guarantees while preserving the utility of the global model. We have also carefully designed corresponding algorithms for average function, frequency estimation, and stochastic gradient descent.

## Introduction

The shuffle model (Bittau et al. 2017) is a state-of-the-art technique to balance privacy and utility for differentially private data analysis. In traditional differential privacy, a trusted server (or aggregator) is often assumed to collect all users' data before privacy-preserving data analysis (Dwork and Roth 2014). However, such approaches may not be feasible or practical in scenarios where a trusted curator does not exist. Given this, Local Differential Privacy (LDP) (Kasiviswanathan et al. 2011) has been proposed to achieve differential privacy by allowing the users to add noises individually; however, LDP suffers from low utility due to the accumulated noise. To address this, the shuffle model of differential privacy (shuffle DP) (Bittau et al. 2017; Balle et al. 2019; Erlingsson et al. 2019) adds a shuffler between the users and the server to randomly shuffle the noisy data before sending the server. The shuffle DP has an intriguing theoretical privacy amplification effect, which means a small amount of local noise could result in a strong privacy guarantee against the untrusted server. Extensive studies (Balle et al. 2019; Erlingsson et al. 2019; Girgis et al. 2021b; Feldman, McMillan, and Talwar 2022; Liu et al. 2021; Girgis et al. 2021a) have been devoted to proving a better (tighter) privacy amplification in the shuffle DP.

However, most existing studies have focused on the shuffle model based on $(\epsilon_0, \delta_0)$-LDP randomizer with uniform and limited settings of local privacy parameters $\epsilon_0$ and $\delta_0$. For example, Erlingsson et al. 2019 assumes $0 < \epsilon_0 < 1/2$ and $\delta_0 = 0$. Although a recent work Liu et al. 2023 provides a privacy bound for local personalized privacy parameter $\epsilon_i$ for each user $i$ (and a fixed $\delta_0$), the bound is relatively rough and has a large room to be improved. To address this problem, we make the following contributions.

Firstly, we propose a **G**eneralized **S**huffle framework for **P**rivacy **A**mplification (GSPA) to allow arbitrary local privacy parameters and provide new privacy amplification analysis. Our analysis technique benefits from the adoption of Functional Differential Privacy (Dong, Roth, and Su 2022) and carefully analyzing the distance between two multinomial distributions (see Theorem 1 and 2). For both uniform and personalized privacy parameter settings, we provide lower privacy bounds that exceed that of existing results (see Figure 2).

Secondly, we apply GSPA with different personalized privacy parameter settings to diverse privacy-preserving analysis tasks, including private mean, private frequency estimation, and DP-SGD, to demonstrate the effectiveness of our approach. For mean and frequency estimation with GSPA (see Figure 3), the more conservative users there are, the less utility is observed, showing a negative linear relationship. Simultaneously, as the privacy parameters of conservative users increase, utility demonstrates a positive linear

[*] Corresponding author: Yang Cao, yang@ist.hokudai.ac.jp

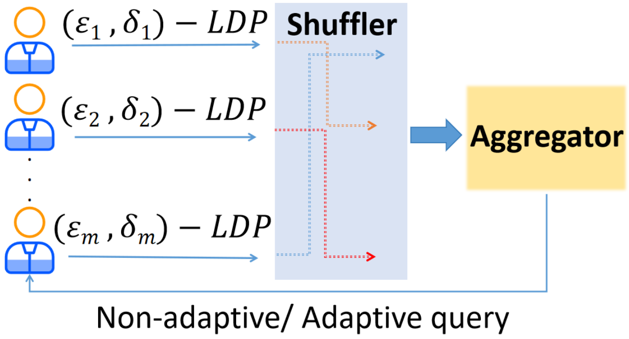

$(\varepsilon_1, \delta_1) - LDP$
$(\varepsilon_2, \delta_2) - LDP$
$(\varepsilon_m, \delta_m) - LDP$
Shuffler
Aggregator
Non-adaptive/ Adaptive query

Figure 1: The **G**eneralized **S**huffle framework for **P**rivacy **A**mplification (GSPA). Privacy parameters $(\epsilon_i, \delta_i)$ and each client's output are shuffled separately. The random permutation of the shuffler is unknown to anyone except the shuffler itself. The type of query, whether non-adaptive or adaptive, depends on whether the next query depends on the previous output.

relationship. For DP-SGD with GSPA (see Figure 4), there exists an interesting phenomenon that despite the constant scenario ($\epsilon_0 = 0.5$) offers a stronger privacy protection, its test accuracy (94.8%) is higher than that (93.5%) of scenarios $U(0.01, 2)$, which have varying local privacy parameters.

## Preliminaries

This section presents the definitions and tools necessary for understanding the shuffle model. These serve as fundamental tools for proposing our methods and form the basis of our approach.

**Definition 1** *(***Differential Privacy***) A randomized algorithm $\mathcal{R}$ satisfies $(\epsilon, \delta)$-differential privacy, denoted as $(\epsilon, \delta)$-DP, if for all $\mathcal{S} \subseteq Range(\mathcal{R})$ and for all neighboring databases $D_0, D_1$ ($D_0$ can be obtained from $D_1$ by replacing exactly one record):*

$$\mathbb{P}(\mathcal{R}(D_0) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}(\mathcal{R}(D_1) \in \mathcal{S}) + \delta. \tag{1}$$

$\epsilon$ is known as the privacy budget, while $\delta$ is referred to as the indistinguishability parameter, which describes the probability of privacy leakage exceeding $\epsilon$. Both $\epsilon$ and $\delta$ should be as small as possible, indicating stronger privacy protection.

**Definition 2** *(***Local Differential Privacy***) A randomized algorithm $\mathcal{R} : \mathcal{D} \to \mathcal{S}$ satisfies $(\epsilon, \delta)$-local differential privacy, denoted as $(\epsilon, \delta)$-LDP, if for all pairs $x, x' \in \mathcal{D}$, $\mathcal{R}(x)$ and $\mathcal{R}(x')$ satisfies*

$$\mathbb{P}(\mathcal{R}(x) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}(\mathcal{R}(x') \in \mathcal{S}) + \delta. \tag{2}$$

In Local Differential Privacy (LDP), each data contributor applies a local randomization mechanism to perturb their own data before sharing it with a central aggregator.

### Privacy Tools

Differential privacy can be regarded as a hypothesis testing problem for a given distribution (Kairouz, Oh, and Viswanath 2015). In brief, we consider the hypothesis testing issue with two hypotheses.

| | Actual True | Actual False |
|---|---|---|
| Accept Hypothesis | Correct | Type II Error ($\beta$) |
| Reject Hypothesis | Type I Error ($\alpha$) | Correct |

Table 1: Table of Error Types

$$H_0 : \text{The underlying dataset is } D_0,$$
$$H_1 : \text{The underlying dataset is } D_1.$$

To provide an intuitive explanation, we designate the name Bob to denote the exclusive individual present in $D_0$ but absent in $D_1$. Consequently, rejecting the null hypothesis implies the recognition of Bob's nonexistence, whereas accepting the null hypothesis suggests observing Bob's existence in the dataset.

Inspired by this, an effective tool called $f$-DP (Dong, Roth, and Su 2022) has been introduced, which utilizes hypothesis testing to handle differential privacy. For two neighbouring databases $D_0$ and $D_1$, let $U$ and $V$ denote the probability distributions of $\mathcal{R}(D_0)$ and $\mathcal{R}(D_1)$, respectively. We consider a rejection rule $0 \leq \phi \leq 1$, with type I and type II error rates defined as

$$\alpha_\phi = \mathbb{E}_U[\phi], \quad \beta_\phi = 1 - \mathbb{E}_V[\phi]. \tag{3}$$

It is well-known that

$$\alpha_\phi + \beta_\phi \geq 1 - TV(U, V), \tag{4}$$

where $TV(U, V)$ is the supremum of $|U(A) - V(A)|$ over all measurable sets $A$. To characterize the fine-grained trade-off between the two errors, Table 1 helps to establish a clear understanding of the relationship between the two errors.

For any two probability distributions $U$ and $V$ on the same space $\Omega$, the trade-off function $T(U, V) : [0, 1] \to [0, 1]$ is defined as

$$T(U, V)(\alpha) = \inf\{\beta_\phi : \alpha_\phi \leq \alpha\}, \tag{5}$$

where the infimum is taken over all measurable rejection rules $\phi$, and $\alpha_\phi = \mathbb{E}_U(\phi)$ and $\beta_\phi = 1 - \mathbb{E}_V(\phi)$.

**Definition 3** *(***Functional Differential Privacy***, $f$-DP) Let $f$ be a trade-off function, a mechanism $\mathcal{R}$ is said to be $f$-DP if*

$$T(\mathcal{R}(D_0), \mathcal{R}(D_1)) \geq f, \tag{6}$$

*for all neighboring data sets $D_0$ and $D_1$.*

To enhance readability, we have included the introduction and relevant properties of $f$-DP in the section of Appendix. It is worth noting that traditional DP belongs to a special case of $f$-DP, therefore $f$-DP has a wider scope of applicability.

In addition, Laplace mechanism and Gaussian mechanism are two common approaches used in differential privacy (Dwork and Roth 2014). The choice between the Laplace mechanism and the Gaussian mechanism depends on the data type, privacy requirements, and query tasks. The

Laplace mechanism provides stronger privacy but may introduce larger errors, while the Gaussian mechanism is more suitable for accurate results. Thus, it's important to strike a balance between privacy and accuracy based on specific requirements.

**Definition 4 (Laplace Mechanism)** *Given a query function $Q : D \rightarrow \mathbb{R}^d$, privacy parameter $\epsilon$ and $\ell_1$ sensitivity $\Delta(Q) = \max \|Q(D) - Q(D')\|_1$, then for any two neighbouring datasets $D, D'$, the Laplace mechanism*

$$M(D,Q) = Q(D) + Lap\left(\frac{\Delta(Q)}{\epsilon}\right) \tag{7}$$

*preserves $\epsilon$-DP, where Lap($\lambda$) denotes the centralized Laplace noise with scale parameter $\lambda$.*

In the absence of ambiguity, we express both queries and answers as $Q(\cdot)$ and $A(\cdot)$ respectively.

**Definition 5 (Gaussian Mechanism)** *Given a query function $Q : D \rightarrow \mathbb{R}^d$, privacy parameter $\epsilon$ and $\ell_2$ sensitivity $\Delta_2(Q) = \max \|Q(D) - Q(D')\|_2$, then for any two neighbouring datasets $D, D'$, the Gaussian mechanism*

$$M(D,Q) = Q(D) + N\left(0, \frac{2\log(1.25/\delta)\Delta_2^2(Q)}{\epsilon^2}\right) \tag{8}$$

*preserves $(\epsilon, \delta)$-DP, where $N(\mu, \sigma^2)$ denotes the Gaussian noise with mean $\mu$ and variance $\sigma^2$.*

## Privacy Analysis of GSPA Framework

Our **G**eneralized **S**huffle framework for **P**rivacy **A**mplification (GSPA) consists of three main components: local randomizers, a trustworthy shuffler, and an aggregator, which are the same as existing shuffle DP frameworks; however, GSPA allows local randomizers with arbitrary privacy parameters. (i) For $n$ users, the local randomizer $M_i$ adds noise to the original data $x_i$ on the $i$-th user's devices, thus providing $(\epsilon_i^\ell, \delta_i^\ell)$-PLDP for user $i$. (ii) The shuffler randomly permutes the order of data elements, ensuring that the resulting arrangement is unknown to any party other than the shuffler itself. (iii) The aggregator collects and integrates shuffled data for simple queries, while for complex tasks like machine learning, it trains models based on shuffled data with multiple iterations. Without causing confusion, the notation $(\epsilon_0, \delta_0)$-LDP is used to represent the uniform scenario, while $(\epsilon_i, \delta_i)$-PLDP denotes the personalized scenario.

**Privacy Amplification Effect** In this section, we address the issue of privacy protection in the context of a general shuffled adaptive process for personalized local randomizers.

**Definition 6** *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$, where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$ be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an $(\epsilon_i, \delta_i)$-PLDP randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_R : \mathcal{D} \rightarrow \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_i)$ for $i \in [n]$ and outputs $z_{1:n}$. We say $\mathcal{A}_R(\mathcal{D})$ is a personalized LDP (PLDP) adaptive process. Similarly, if we first sample a permutation $\pi$ uniformly at random, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi_i})$ for $i \in [n]$ and outputs $z_{1:n}$, we say this process is shuffled PLDP adaptive and denote it by $\mathcal{A}_{R,S}(\mathcal{D})$.*

**Lemma 1** *Given an $(\epsilon_i, \delta_i)$-PLDP adaptive process, then in the i-th step, local randomizer $\mathcal{R}^{(i)}: \mathcal{D} \rightarrow \mathcal{S}$ and for any $n+1$ inputs $x_1^0, x_1^1, x_2, \cdots, x_n \in \mathcal{D}$, there exists distributions $\mathcal{Q}_1^0, \mathcal{Q}_1^1, \mathcal{Q}_1, \mathcal{Q}_2, \cdots, \mathcal{Q}_n$ such that*

$$\mathcal{R}^{(i)}(x_1^0) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{Q}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^1 + \delta_i\mathcal{Q}_1, \tag{9}$$

$$\mathcal{R}^{(i)}(x_1^1) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{Q}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^1 + \delta_i\mathcal{Q}_1. \tag{10}$$

$\forall x_i \in \{x_2, \cdots, x_n\}$,

$$\mathcal{R}(x_i) = \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^1 + \left(1 - \frac{2(1-\delta_i)}{1+e^{\epsilon_i}}\right)\mathcal{Q}^i. \tag{11}$$

**Proof:** For inputs $X_0 = \{x_1^0, x_2, \ldots, x_n\}$ and $X_1 = \{x_1^1, x_2, \ldots, x_n\}$, $\mathcal{R}^{(i)}$ satisfies the constraints of Lemma 4, so there exists an $(\epsilon_i, \delta_i)$-PLDP local randomizer $\mathcal{R}' : \mathcal{D} \rightarrow \mathcal{Z}$ for the $i$-th output and post-processing function $proc(\cdot)$ such that $proc(\mathcal{R}'^{(i)}(x)) = \mathcal{R}^{(i)}(x)$, and

$$P(\mathcal{R}'^{(i)}(x_1^0) = z) = \begin{cases} 0 & \text{if } z = A, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \text{if } z = 0, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \text{if } z = 1, \\ \delta_i & \text{if } z = B. \end{cases}$$

$$P(\mathcal{R}'^{(i)}(x_1^1) = z) = \begin{cases} \delta_i & \text{if } z = A, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \text{if } z = 0, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \text{if } z = 1, \\ 0 & \text{if } z = B. \end{cases}$$

Let $L = \{z \in \mathcal{Z} | \mathbb{P}(\mathcal{R}'(x_1^0 = z)) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}$ and $\mathbb{P}(\mathcal{R}'(x_1^1 = z)) = \frac{1-\delta_i}{1+e^{\epsilon_i}}\}$ , $U = \{z \in \mathcal{Z} | \mathbb{P}(\mathcal{R}'(x_1^1 = z)) = \frac{1-\delta_i}{1+e^{\epsilon_i}}$ and $\mathbb{P}(\mathcal{R}'(x_1^1 = z)) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\}$. Let $M = \mathcal{Z}\{L \bigcup U\}$ and $p = \sum_{z \in \mathcal{L}} p_z = \sum_{z \in \mathcal{U}} p_z$. Since conditioned on the output lying in $\mathcal{L}$, the distribution of $\mathcal{R}'(x_1^0)$ and $\mathcal{R}'(x_1^1)$ are the same. Let $\mathcal{W}_1^0 = \mathcal{R}'(x_1^0)|L = \mathcal{R}'(x_1^1)|L$, $\mathcal{W}_1^1 = \mathcal{R}'(x_1^0)|U = \mathcal{R}'(x_1^1)|U$ and $\mathcal{W}_1 = \mathcal{R}'(x_1^0)|M = \mathcal{R}'(x_1^1)|M$. Then

$$\mathcal{R}'(x_1^0) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{W}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{W}_1^1 + \delta_i\mathcal{W}_1,$$

$$\mathcal{R}'(x_1^1) = \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{W}_1^0 + \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{W}_1^1 + \delta_i\mathcal{W}_1.$$

Further, for all $x_i \in \{x_2, \cdots, x_n\}$,

$$\mathcal{R}'(x_i) \geq \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{W}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{W}_1^1 + \left(1 - \frac{2(1-\delta_i)}{1+e^{\epsilon_i}}\right)\mathcal{W}_i.$$

Letting $\mathcal{Q}_1^0 = proc(\mathcal{W}_1^0)$, $\mathcal{Q}_1^1 = proc(\mathcal{W}_1^1)$, $\mathcal{Q}_1 = proc(\mathcal{W}_1)$ and for all $i \in \{2, \cdots, n\}$, $\mathcal{Q}_i = proc(\mathcal{W}_i)$. The proof is completed. □

**Theorem 1** *For a domain $\mathcal{D}$, if $\mathcal{A}_{R,S}(\mathcal{D})$ is a shuffled PLDP adaptive process, then for arbitrary two neighboring datasets $D_0, D_1 \in \mathcal{D}^n$ distinct at the $n$-th data point, there exists a post-processing function $proc(\cdot)$: $(0,1,2) \to \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \times \mathcal{S}^{(n)}$, such that*

$$T(\mathcal{A}_{R,S}(D_0), \mathcal{A}_{R,S}(D_1)) = T(proc(\rho_0), proc(\rho_1)).$$

*Here,*

$$\rho_0 = (\Delta_0, \Delta_1, \Delta_2) + \boldsymbol{V}, \tag{12}$$

$$\rho_1 = (\Delta_1, \Delta_0, \Delta_2) + \boldsymbol{V}, \tag{13}$$

$\Delta_2 \sim Bern(\delta_n), \Delta_0 \sim Bin(1 - \Delta_2, \frac{e^{\epsilon_n}}{1+e^{\epsilon_n}})$, $\Delta_1 = 1 - \Delta_0 - \Delta_2$, *where $Bern(p)$ denotes a Bernoulli random variable with bias p, $Bin(n,p)$ denotes a Binomial distribution with $n$ trials and success probability $p$. In addition, $\boldsymbol{V} = \sum_{i=1}^{n-1} MultiBern\left(\frac{1-\delta_i}{1+e^{\epsilon_i}}, \frac{1-\delta_i}{1+e^{\epsilon_i}}, \delta_i\right)$, where $MultiBern(\theta_1, \cdots, \theta_d)$ represents a $d$-dimensional Bernoulli distribution with $\sum_{j=1}^{d} \theta_j = 1$.*

In order to enhance readability, proof details are placed in the section of Appendix. Based on Theorem 1, we can simplify the original problem by analyzing the shuffling process in a simple non-adaptive protocol.

The primary objective in the following is to demonstrate the distance between two distributions. The Berry Esseen lemma (Berry 1941; Esseen 1942) is highly valuable and essential for proving asymptotic properties.

**Lemma 2 (Berry Esseen)** *Let $P = (\xi_0, \xi_1, \xi_2) \sim \sum_{i=1}^{m} MultiBern\left(\frac{p_i}{2}, \frac{p_i}{2}, 1-p_i\right)$ and $Q \sim N(\mu, \Sigma)$, where $\mu = \mathbb{E}(P)$ and $\Sigma = Var(P)$. Then for the first two components $(X_0, X_1)$, there exists $C > 0$, such that $\|\tilde{P} - \tilde{Q}\|_{TV} \leq \frac{C}{\sqrt{m}}$, where $\tilde{P}$ and $\tilde{Q}$ represent the distribution of $(\xi_0, \xi_1)$ and corresponding normal distribution, respectively.*

In fact, for given $n$, we can obtain sophisticated bound of $\|\tilde{P} - \tilde{Q}\|_{TV}$ by numerical methods. Without loss of generality, we assume $\epsilon_i = \epsilon_0$, $\delta_i = \delta_0 = O(1/n)$, then $p_0 = \frac{1-\delta_0}{1+e^{\epsilon_0}}$. For some fixed output $(\xi_0, \xi_1) = (k_0, k_1)$, we approximate by integrating the normal probability density function around that point. Let $G(\cdot)$ be the cumulative distribution function of $\tilde{Q}$ and $h(k_0, k_1) = G(k_0 + 0.5, k_1 + 0.5) - G(k_0 + 0.5, k_1) - G(k_0, k_1 + 0.5) + G(k_0, k_1)$, then

$$\|\tilde{P} - \tilde{Q}\|_{TV} = \sup_{(k_0,k_1)} |\mathbb{P}(\xi_0 = k_0, \xi_1 = k_1) - h(k_0, k_1)|. \tag{14}$$

**Lemma 3** *Let $p_i = \frac{2(1-\delta_i)}{1+e^{\epsilon_i}}$, if $\bar{\mu} = \sum_{i=1}^{n-1} (\frac{p_i}{2}, \frac{p_i}{2})'$ and $\mu_0 = (1,0)' + \bar{\mu}$, $\mu_1 = (0,1)' + \bar{\mu}$, then $T(N(\mu_0, \boldsymbol{\Sigma}), N(\mu_1, \boldsymbol{\Sigma})) = \Phi(\Phi^{-1}(1-\alpha) - \frac{2}{\sqrt{\sum_{i=1}^{n-1} p_i}})$,*

$$\boldsymbol{\Sigma} = \sum_{i=1}^{n-1} \begin{pmatrix} \frac{p_i}{2}(1 - \frac{p_i}{2}) & -\frac{p_i^2}{4} \\ -\frac{p_i^2}{4} & \frac{p_i}{2}(1 - \frac{p_i}{2}) \end{pmatrix}.$$

**Theorem 2 (Enhanced Central Privacy Upper bound)** *Assume $\rho_0$ and $\rho_1$ are defined in equations (12) and (13), then there exists $C > 0$, such that*

$$T(\rho_0, \rho_1) \geq \left(G_\mu\left(\alpha + \frac{C}{\sqrt{n-1}}\right) - \frac{C}{\sqrt{n-1}}\right), \tag{15}$$

*where $G_\mu(\alpha) = \Phi(\Phi^{-1}(1 - \alpha) - \mu), \mu = \sqrt{\frac{2}{\sum_{i=1}^{n} \frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$. In an unambiguous manner, we refer to it as approximately following the $\mu$-GDP.*

**Proof:** First, let's analyze the scenarios where the $n$-th data point differs. According to the definition of $(\Delta_0, \Delta_1, \Delta_2)$,

$$(\Delta_0, \Delta_1, \Delta_2) = \begin{cases} (0,0,1) & w.p. \quad \delta_n; \\ (1,0,0) & w.p. \quad (1-\delta_n)\frac{e^{\epsilon_n}}{1+e^{\epsilon_n}}; \\ (0,1,0) & w.p. \quad (1-\delta_n)\frac{1}{1+e^{\epsilon_n}}. \end{cases} \tag{16}$$

When $\Delta_2 = 1$, $\rho_0$ and $\rho_1$ are indistinguishable, which indicates that $T(\rho_0, \rho_1)|_{\Delta_2 = 1} = 1 - \alpha$. Let $\rho_0' = (1,0,0)' + \sum_{i=1}^{n-1} MultiBern\left(p_i/2, p_i/2, 1-p_i\right)$ and $\rho_1' = (0,1,0)' + \sum_{i=1}^{n-1} MultiBern\left(p_i/2, p_i/2, 1-p_i\right)$ with $p_i = \frac{1-\delta_i}{1+e^{\epsilon_i}}$, then

$$T(\rho_0, \rho_1) = \delta_n(1-\alpha) + (1-\delta_n) T_{symm}(\rho_0', \rho_1'), \tag{17}$$

where $T_{symm}(\rho_0', \rho_1') = \max\{T(\rho_0', \rho_1'), T(\rho_1', \rho_0')\}$. Assume $P \sim N(\mu_0, \Sigma)$, $Q \sim N(\mu_1, \Sigma)$, where $\mu_0, \mu_1, \Sigma$ are same as that in Lemma 3. Let $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n-1} \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$, according to equation (4),

$$T(\rho_0', P) \geq 1 - \alpha - \|\rho_0' - P\|_{TV},$$
$$T(\rho_1', Q) \geq 1 - \alpha - \|\rho_1' - Q\|_{TV},$$

then based on Fact 4,

$$T(\rho_0', Q) \geq \Phi(\Phi^{-1}(1 - \alpha - \|\rho_0' - P\|_{TV}) - \mu) = F(\alpha).$$

Reusing Fact 4, we can obtain that

$$\begin{aligned} T(\rho_0', \rho_1') &\geq 1 - (1 - F(\alpha)) - \|\rho_1' - Q\|_{TV} \\ &= F(\alpha) - \|\rho_1' - Q\|_{TV} \end{aligned} \tag{18}$$

Lemma 2 shows that there exists $C > 0$, such that $\|\rho_1' - Q\|_{TV} \leq \frac{C}{\sqrt{n-1}}$ and $\|\rho_0' - P\|_{TV} \leq \frac{C}{\sqrt{n-1}}$. Hence

$$T(\rho_0', \rho_1') \geq G_\mu\left(\alpha + \frac{C}{\sqrt{n-1}}\right) - \frac{C}{\sqrt{n-1}}.$$

Then

$$\begin{aligned} T(\rho_0, \rho_1) \geq{}& \delta_n(1-\alpha) \\ &+ (1-\delta_n)\left(G_\mu\left(\alpha + \frac{C}{\sqrt{n-1}}\right) - \frac{C}{\sqrt{n-1}}\right). \end{aligned}$$

Since for an arbitrary trade-off function $f$, we have $f \leq 1 - \alpha$, it follows that:

$$T(\rho_0, \rho_1) \geq \left(G_\mu\left(\alpha + \frac{C}{\sqrt{n-1}}\right) - \frac{C}{\sqrt{n-1}}\right).$$

| Name | Distribution of $\epsilon^\ell = (\epsilon_1^\ell, \cdots, \epsilon_n^\ell)$ |
|---|---|
| Unif 1 | $U(0.01, 1)$ |
| Unif 2 | $U(0.01, 2)$ |
| Constant | 0.5 |
| Mixed Constant | $50\%\ 0.5 + 50\%\ 0.01$ |

Table 2: Distributions of LDP budgets $\epsilon^\ell$. $U(a, b)$ represents uniform distribution ranging from $a$ to $b$.

Finally, taking into account the case where the $i$-th ($1 \leq i \leq n$) data differs in neighboring datasets, the privacy bound is determined based on the worst-case scenario, that is, $\mu = \sqrt{\frac{2}{\sum_{i=1}^n \frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$. □

**Comparison with Existing Results** We provide numerical evaluations for privacy amplification effect under fixed LDP settings in Table 2. Given a local privacy budget set $\epsilon^\ell \in [0.01, 2]$. For the purpose of comparison, we examine privacy amplification for a fixed $\epsilon^\ell$ while varying $n$ from $10^3$ to $10^4$, with central $\delta$ for shuffling to be $10^{-4}$ for the sake of simplicity. To avoid misunderstandings, we repeat the first $10^3$ parameters. Considering that convergence rate in Lemma 2 is nearly $O(1/n)$ and can be negligible in numerical analysis, our focus lies in measuring $G_\mu$.

To keep it concise, we use Fact 3 in Appendix to compute the corresponding central $\epsilon$ and $\delta$ for Theorem 2. Baseline bounds of privacy amplification effect include: [Liu23] (Liu et al. 2023), [FMT22] (Feldman, McMillan, and Talwar 2022), [Erlingsson19] (Erlingsson et al. 2019). [Liu23] provides bounds for the personalized scenario, while [FMT22] and [Erlingsson19] only consider the same $\epsilon^\ell$.

The numerical results demonstrate the following results: (i) Our bound is suitable for extreme privacy budgets while [Liu23] required each $\epsilon_i$ should not be close to zero. However, it is natural to encounter user responses that contain no information, resulting in $\epsilon_i = 0$. (ii) As the sample size $n$ increases, the amplification effect also increases proportionally to the square root of $n$. (iii) Our privacy bounds significantly outperform in all current scenarios, even in cases where the privacy parameters are the same.

## Application and Experiments

All the experiments are implemented on a workstation with an Intel Core i5-1155G7 processor on Windows 11 OS.

### Application to Mean and Frequency Estimation

**Mean Estimation** The average function is a fundamental and commonly used mathematical operation with wide-ranging applications. In this section, we apply GSPA to the average function on the synthetic data. We randomly divide the users into three groups: conservative, moderate, and liberal. The fraction of three groups are determined by $f_c, f_m, f_l$. As is reported (Acquisti and Grossklags 2005), the default values in this experiment are $f_c = 0.54, f_m = 0.37, f_l = 0.09$. For convenience, the privacy preferences for the users in conservative, moderate and liberal groups are $\epsilon_C$, $\epsilon_M$ and $\epsilon_l$, respectively. In the LDP case, the privacy preference of users in the liberal group is fixed at $\epsilon_L = 1$, while the default values of $\epsilon_C$ and $\epsilon_M$ are set to 0.1 and 0.5, respectively.

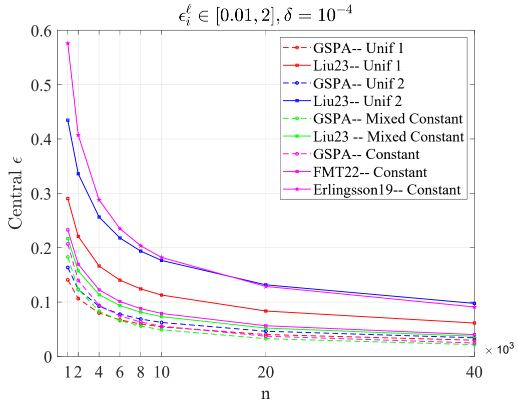


Figure 2: Privacy Bounds for Varied Budgets

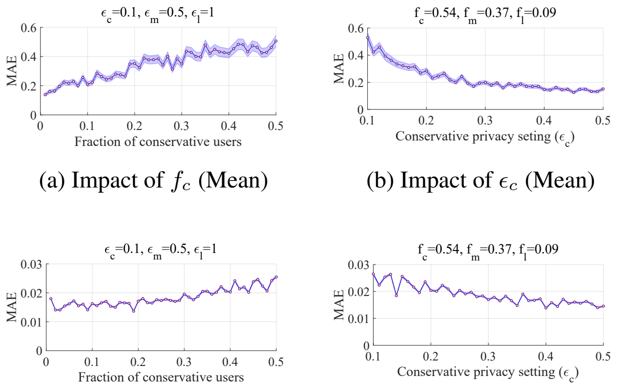
**Theorem 3** *Algorithm 1 approximately preserves $\mu$-GDP for each user, where* $\mu = \sqrt{\frac{2}{\sum_{i=1}^n \frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$.

**Proof:** According to the definition of Laplace mechanism, data point $i \in [n]$ satisfies $(\epsilon, 0)$-LDP. Combined with Theorem 2, we can obtain that Algorithm 1 approximately satisfies $\mu$-GDP with $\mu = \sqrt{\frac{2}{\sum_{i=1}^n \frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$. □

Next, we simulate the accuracy for different set of privacy protection. To facilitate comparison, we set $f_l = 0.09$ as a fixed value and vary $f_c$ from 0.01 to 0.5 with $f_m = 1 - f_l - f_c$. Additionally, we generate $n = 10,000$ privacy budgets for users based on the privacy preferences rule. We assume that each sample is drawn from a normal distribution $N(50, \sigma^2)$, and then the samples are clipped into the range $[20, 80]$. We repeat this procedure for a total of $1,000$ times to give a confidence interval. According to Fact 3, privacy parameter $\mu$ under the shuffle model can be obtained for varying $\epsilon_c$. Figure 3a shows that an increase in the proportion of conservative users leads to a decrease in estimation accuracy. On the other hand, Figure 3b demonstrates that increasing privacy budget is beneficial for improving accuracy.

Algorithm 1: Mean estimation with GSPA.

```
Input: Dataset $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$, privacy budget
    $\mathcal{S} = \{\epsilon_1, \cdots, \epsilon_n\}$ for each user.
Output: $z \in \mathbb{N}$
for each $i \in [n]$ do
    $y_i \leftarrow x_i + Lap(\Delta f / \epsilon_i)$
end for
Choose a random permutation $\pi$: $[n] \rightarrow [n]$
$z = \frac{1}{n} \sum_{i=1}^n y_{\pi(i)}$
return $z$
```



(a) Impact of $f_c$ (Mean)

(b) Impact of $\epsilon_c$ (Mean)

(c) Impact of $f_c$ (Frequency)

(d) Impact of $\epsilon_c$ (Frequency)

Figure 3: Impact of privacy parameter settings on MAE.

**Frequency estimation** In machine learning, frequency estimation is often used as a preprocessing step to understand the distribution and importance of different features or categories within a dataset. By accurately estimating the frequencies of various features or categories, it helps in feature selection, dimensionality reduction, and building effective models.

In order to obtain the dataset, a total of 10,000 records are generated for counting. Each record is encoded as a binary attribute. The proportion of records with a value of 1 is determined by a density parameter $c$, which ranges from 0 to 1 (with a default value of $c = 0.7$).

**Theorem 4** *Algorithm 2 approximately preserves $\mu$-GDP for each user, where* $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n}\frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$.

The proof of Theorem 4 is the same as Theorem 3. The direct calculation shows that $z$ is an unbiased estimator of $c$, that is, $\mathbb{E}(z) = c$. Similar to the average function, we adopt the same configuration for personalized privacy budgets.

Algorithm 2: Frequency estimation with GSPA

```
Input: Dataset $X = (x_1, \dots, x_n) \in \{0,1\}^n$, privacy budget $\mathcal{S} = \{\epsilon_1, \cdots, \epsilon_n\}$ for each user.
Output: $z \in \mathbb{N}$
for each $i \in [n]$ do
    if $x_i = 1$ then
        $y_i \leftarrow Ber(\frac{e^{\epsilon_i}}{1+e^{\epsilon_i}})$
    else
        $y_i \leftarrow Ber(\frac{1}{1+e^{\epsilon_i}})$
    end if
end for
Choose a random permutation $\pi$: $[n] \rightarrow [n]$
$A = \sum_{i=1}^{n} y_{\pi(i)}$
$B = \sum_{i=1}^{n} \frac{1}{1+e^{\epsilon_{\pi(i)}}}$
$z = \frac{A-B}{n-2B}$
return $z$
```

## Personalized Private Stochastic Gradient Descent

The private stochastic gradient descent is a common method in deep learning (Abadi et al. 2016). However, personalized private stochastic gradient descent combines personalized differential privacy with stochastic gradient descent optimization for model training and parameter updates while ensuring privacy protection. In the context of personalized differential privacy, privacy of individual users must be protected, and direct use of raw data for parameter updates is not feasible .

The key idea of personalized differential privacy is to introduce personalized parameters into the differentially private mechanism to flexibly adjust the level of privacy protection. For the gradient descent algorithm, personalized differential privacy can be achieved by introducing noise during gradient computation.

**Theorem 5** *Algorithm 3 approximately satisfies $\sqrt{T}\mu$-GDP with* $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n}\frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$.

**Proof:** *For arbitrary $j \in [m]$, client $j$ satisfies $(\epsilon_j, \delta_j)$-LDP before sending to the shuffler by the definition of Gaussian mechanism. By using Theorem 2, it preserves $\mu$-GDP after shuffling with $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n}\frac{1-\delta_i}{1+e^{\epsilon_i}} - \max_i \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$. In addition, combined with Fact 3, it holds $\sqrt{T}\mu$-GDP under $T$-fold composition.*

□

**Dataset and implementation** The MNIST dataset (LeCun et al. 1998) for handwritten digit recognition consists of $60,000$ training images and $10,000$ test images. Each sample in the dataset represents a $28 \times 28$ vector generated from handwritten images, where the independent variable corresponds to the input vector, and the dependent variable represents the digit label ranging from $0$ to $9$. In our experiments, we consider a scenario with $m$ clients, where each client has $n/m$ samples. For simplicity, we train a simple classifier using a feed-forward neural network with ReLU activation units and a softmax output layer with 10 classes, corresponding to the 10 possible digits. The model is trained using cross-entropy loss and an initial PCA input layer with 60 components. At each step of the shuffled SGD, we choose at one client at random without replacement. The parameters of experimental setup is listed in Table 3. This experiment is designed to demonstrate the use cases of the shuffle model and therefore does not focus on comparing with previous results. For comparative results, please refer to Figure 2.

**Parameter Selection** As a result, our approach achieves an accuracy of $96.78\%$ on the test dataset after approximately 50 epochs. This result is consistent with the findings of a vanilla neural network (LeCun et al. 1998) trained on the same MNIST dataset. By employing this methodology, we can effectively train a simple classifier that achieves high accuracy in recognizing handwritten digits from the MNIST dataset.

| Parameters | Value | Explanation |
|---|---|---|
| $C$ | 2 | Clipping bound |
| $\delta^\ell$ | $10^{-5}$ | Indistinguishability parameter |
| $\epsilon^\ell$ | $[0.01, 2]$ | Privacy budget |
| $\eta$ | 0.05 | Step size of the gradient |
| $m$ | 100 | The number of clients |
| $n$ | $60,000$ | Total number of training samples |
| $T$ | 50 | The number of epochs |

Table 3: Experiment Setting for the Shuffled Personalized-SGD on the MNIST dataset.

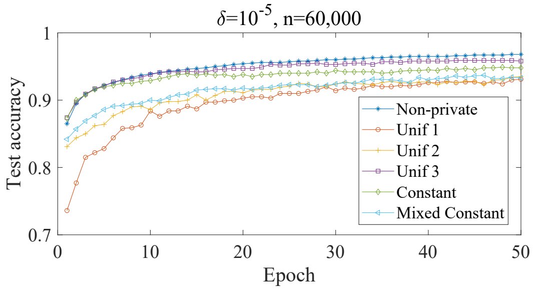


Figure 4: Comparison of Test Accuracy with Different Noise Distributions

**Utility of GSPA** We evaluate the utility of GSPA with $\epsilon^\ell$ drawing from Table 2 on MNIST dataset. We introduce Unif 3 as a distribution to represent $U(0.5, 1)$. The numerical results indicate that Unif 3 exhibits the best accuracy, which aligns with expectations as it corresponds to a larger value of the privacy budget. Despite constant scenario exhibiting stronger privacy protection than Unif 2, it actually achieves better accuracy. One possible reason behind this interesting observation is the significant difference in the privacy parameters, which can cause instability in gradient iterations.

## Conclusion and Future Work

This work focuses on privacy amplification of shuffle model. To address the trade-off between privacy and utility, we propose the GSPA framework, which achieves a higher accuracy while maintaining at least 33% privacy loss compared to existing methods.

In our future research, we plan to expand by incorporating additional privacy definitions such as Renyi differential privacy (Girgis et al. 2021b). Moreover, we acknowledge the significance of enhancing techniques for specific data types like images, speech, and other forms. This entails developing specialized privacy metrics, differential privacy mechanisms, and model training algorithms that offer more accurate and efficient privacy protection.

## Appendix

**$f$-DP** Here are several important properties of $f$-DP. We present these facts directly for the sake of brevity, and for comprehensive proofs, please refer to the related article (Dong, Roth, and Su 2022).

Algorithm 3: SGD with GSPA

```
Input: Dataset $X = (x_1, \dots, x_n)$, loss function $\mathcal{L}(\boldsymbol{\theta}, x)$,
    initial point $\boldsymbol{\theta}_0$, learning rate $\eta$, number of epochs $T$,
    privacy budget $\mathcal{S} = \{\epsilon_1, \delta_1, \cdots, \epsilon_n, \delta_n\}$, batch size $m$
    and gradient norm bound $C$.
Output: $\hat{\boldsymbol{\theta}}_{T,m}$
Split $[n]$ into $n/m$ disjoint subsets $S_1, \cdots, S_m$ with equal size $m$
Choose arbitrary initial point $\hat{\boldsymbol{\theta}}_0$
Choose a random permutation $\pi$: $[m] \to [m]$
for each $t \in [T]$ do
    $\tilde{\boldsymbol{\theta}}_{t,m} = \hat{\boldsymbol{\theta}}_t$
    for each $i \in [n/m]$ do
        $\sigma = \frac{2C}{m} \frac{\sqrt{2\log(1.25/\delta_{\pi(i)})}}{\epsilon_{\pi(i)}}$
        $\boldsymbol{b}_i \sim N(0, \sigma^2 \boldsymbol{I}_d)$
        for each $j \in S_{\pi(i)}$ do
            Compute gradient:
            $\boldsymbol{g}_i^j = \nabla \ell(\tilde{\boldsymbol{\theta}}_{i-1}, x_j)$
        end for
        Clip to norm $C$:
        $\boldsymbol{g}_i = \frac{\sum_j \boldsymbol{g}_i^j}{m}$
        $\tilde{\boldsymbol{g}}_i = \boldsymbol{g}_i / \max(1, \|\boldsymbol{g}_i\|_2 / C)$
        $\tilde{\boldsymbol{\theta}}_i = \tilde{\boldsymbol{\theta}}_{i-1} - \eta(\tilde{\boldsymbol{g}}_i + \boldsymbol{b}_i)$
    end for
    $\hat{\boldsymbol{\theta}}_{t,m} = \tilde{\boldsymbol{\theta}}_m$
end for
return $\hat{\boldsymbol{\theta}}_{T,m}$
```

**Fact 1** *$(\epsilon, \delta)$-DP is equivalent to $f_{\epsilon,\delta}$-DP, where*

$$f_{\epsilon,\delta} = \max\{0, 1 - \delta - e^\epsilon \alpha, e^{-\epsilon}(1 - \delta - \epsilon)\}. \tag{19}$$

**Fact 2** *$f$-DP holds the post-processing property, that is, if a mechanism $M$ is $f$-DP, then its post-processing $Proc \circ M$ is also $f$-DP.*

**Fact 3** *($\mu$-GDP) A $f$-DP mechanism is called $\mu$-GDP if $f$ can be obtained by $f = T(N(0,1), N(\mu, 1)) = \Phi(\Phi^{-1}(1 - \alpha) - \mu)$, where $\Phi(\cdot)$ is cumulative distribution function of standard Gaussian distribution $N(0,1)$. Then a mechanism is $\mu$-GDP if and only if it is $(\epsilon, \delta(\epsilon))$-DP for all $\epsilon \geq 0$, where*

$$\delta(\epsilon) = \Phi(-\frac{\epsilon}{\mu} + \frac{\mu}{2}) - e^\epsilon \Phi(-\frac{\epsilon}{\mu} - \frac{\mu}{2}).$$

*In particular, if a mechanism is $\mu$-GDP, then it is $k\mu$-GDP for groups of size $k$ and the $n$-fold composition of $\mu_i$-GDP mechanisms is $\sqrt{\mu_1^2 + \cdots + \mu_n^2}$-GDP.*

**Fact 4** *Suppose $T(P, R) \geq f, T(Q, R) \geq g$, then $T(P, R) \geq f \circ g = g(1 - f(\alpha))$.*

The relationship between $f$-DP and traditional DP has been illustrated from the perspective of hypothesis testing (Dong, Roth, and Su 2022). It provides a visual representation of how the choice of parameter $\mu$ in $\mu$-GDP relates to the strength of privacy protection.

The $(\epsilon_i, \delta_i)$-PLDP mechanism can be dominated by the following hypothesis testing problem (Kairouz, Oh, and Viswanath 2015). This forms the foundation for the subsequent analysis.

**Lemma 4 (KOV15)** *Let $\mathcal{R}^{(i)} : \mathcal{D} \to \mathcal{S}$ be an $(\epsilon_i, \delta_i)$-DP local randomizer, and $x_0, x_1 \in D$, then there exist two quaternary random variables $\tilde{X}_0$ and $\tilde{X}_1$, such that $\mathcal{R}^{(i)}(x_0)$ and $\mathcal{R}^{(i)}(x_1)$ can be viewed as post-processing of $\tilde{X}_0$ and $\tilde{X}_1$, respectively. In details,*

$$P(\tilde{X}_0 = x) = \begin{cases} \delta_i & \textit{if } x = A, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \textit{if } x = 0, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \textit{if } x = 1, \\ 0 & \textit{if } x = B, \end{cases}$$

*and*

$$P(\tilde{X}_1 = x) = \begin{cases} 0 & \textit{if } x = A, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \textit{if } x = 0, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \textit{if } x = 1, \\ \delta_i & \textit{if } x = B. \end{cases}$$

### Proof of Theorem 1

**Proof:** Formally, for each $i \in \{2, \cdots, n\}$, let $p_i = \frac{1-\delta_i}{1+e^{\epsilon_i}}$, we define random variables $Y_{1,i}^0$, $Y_{1,i}^1$ and $Y_i$ as follows:

$$Y_{1,i}^0 = \begin{cases} 0 & w.p. & e^{\epsilon_i}\frac{p_i}{2}, \\ 1 & w.p. & \frac{p_i}{2}, \\ 2 & w.p. & 1 - e^{\epsilon_i}\frac{p_i}{2} - \frac{p_i}{2}. \end{cases} \tag{20}$$

$$Y_{1,i}^1 = \begin{cases} 0 & w.p. & \frac{p_i}{2}, \\ 1 & w.p. & e^{\epsilon_i}\frac{p_i}{2}, \\ 2 & w.p. & 1 - e^{\epsilon_i}\frac{p_i}{2} - \frac{p_i}{2}. \end{cases} \tag{21}$$

and

$$Y_i = \begin{cases} 0 & w.p. & \frac{p_i}{2}, \\ 1 & w.p. & \frac{p_i}{2}, \\ 2 & w.p. & 1 - p_i. \end{cases} \tag{22}$$

We consider the case in the $t$-th iteration. Given a dataset $X_b$ for $b \in \{0, 1\}$, we generate $n$ samples from $\{0, 1, 2\}$ in the following way. Client number one reports a sample from $Y_{1,i}^b$. Client $i$ ($i = 2, \cdots, n$) each reports an independent sample from $Y_i$. We then shuffle the reports randomly. Let $\rho_b$ denote the resulting distribution over $\{0, 1, 2\}^n$. We then count the total number of 0s and 1s. Note that a vector containing a permutation of the users responses contains no more information than simply the number of 0s and 1s, so we can consider these two representations as equivalent.

We claim that there exists a post-processing function $proc(\cdot)$ such that for $y$ sampled from $\rho_b$, $proc(y)$ is distributed identically to $\mathcal{A}_S(X_b)$. To see this, let $\pi$ be a randomly and uniformly chosen permutation of $\{1, \cdots, n\}$. For every $i \in \{1, \cdots, n\}$, given the hidden permutation $\pi$, we can generate a sample from $\mathcal{A}_S(X_b)$ by sequentially transforming $proc(y_t)$ to obtain the correct mixture components, then sampling from the corresponding mixture component. Specially, by Lemma 1,

$$z_t = \begin{cases} \mathcal{R}^{(t)}(z_{1:t-1}, x_1^0) & \text{if} \quad y_t = 0; \\ \mathcal{R}^{(t)}(z_{1:t-1}, x_1^1) & \text{if} \quad y_t = 1; \\ \mathcal{R}^{(t)}(z_{1:t-1}, x_{\pi(i)}) & \text{if} \quad y_t = 2. \end{cases} \tag{23}$$

By our assumption, this produces a sample $z_t$ from $\mathcal{R}^{(i)}(x_{\pi(i)})$. It is easy to see that the resulting random variable $(z, y)$ has the property that for input $b \in \{0, 1\}$, its marginal distribution over $\mathcal{S}$ is the same as $\mathcal{A}_S(X_b)$ and its marginal distribution over $\{0, 1, 2\}^n$ is $\rho_b$. The difficulty then lies in the fact that conditioned on a particular instantiation $y = v$, the permutation $\pi|_{y=v}$ is not independent of $b$. Note that if $v_t = 0$ or 1, the corresponding $\mathcal{Q}_1^{0(t)}(z_{1:t-1})$ or $Q_1^{1(t)}(z_{1:t-1})$, is independent of $\pi$. Therefore, in order to do the appropriate post-processing, it suffices to know the permutation $\pi$ restricted to the set of users who sampled 2, $K = \pi(\{i : y_i = 2\})$. The set $K$ of users who select 2 is independent of $b$ since $Y_{1,i}^0$ and $Y_{1,i}^1$ have the same probability of sampling 2. The probability of being included in $K$ is identical for each $i \in \{2, \cdots, n\}$, and slightly smaller for the first user. Since the sampling of $z$ given $y$ only needs $K$, we can sample from $z|_{(y,K)=(v,J)}$ without knowing $b$. This conditional sampling is exactly the post-processing step that we claimed.

We now analyze the divergence between $P_0$ and $P_1$, the shuffling step implies that $P_0$ and $P_1$ are symmetric. This implies that the divergence between $P_0$ and $P_1$ is equal to the divergence between the distribution between the distribution of the counts of $0'$s and $1'$s. The decomposition in equation (11) implies that the divergence between $\mathcal{A}_S(X_0)$ and $\mathcal{A}_S(X_1)$ can be dominated by the divergence of $P_0$ and $P_1$, where $\Delta_2 \sim Bern(\delta_n), \Delta_0 \sim Bin(1 - \Delta_2, \frac{e^{\epsilon n}}{1+e^{\epsilon n}})$, $\Delta_1 = 1 - \Delta_0 - \Delta_2$ and $MultiBern(\theta_1, \cdots, \theta_d)$ represents a $d$-dimensional Bernoulli distribution with $\sum_{j=1}^d \theta_j = 1$. □

### Proof of Lemma 3

**Proof:** Since $T(N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}), N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}))$ is

$$\Phi(\Phi^{-1}(1-\alpha) - \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}),$$

according to the property of normal distribution, the key is to calculate $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$. Let $v_1 = \sum_{i=1}^{n-1} p_i, v_2 = \sum_{i=1}^{n-1} p_i^2$, then

$$\boldsymbol{\Sigma} = \begin{pmatrix} \frac{v_1}{2} - \frac{v_2}{4} & -\frac{v_2}{4} \\ -\frac{v_2}{4} & \frac{v_1}{2} - \frac{v_2}{4} \end{pmatrix},$$

and

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \frac{2v_1 - v_2}{v_1^2 - v_1 v_2} & \frac{v_2}{v_1^2 - v_1 v_2} \\ \frac{v_2}{v_1^2 - v_1 v_2} & \frac{2v_1 - v_2}{v_1^2 - v_1 v_2} \end{pmatrix}.$$

By simple calculation, we can obtain that

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = (-1, 1)\boldsymbol{\Sigma}^{-1}(-1, 1)' = \frac{4}{\sum_{i=1}^{n-1} p_i}.$$

Substituting $\mu = \sqrt{\frac{4}{\sum_{i=1}^{n-1} p_i}}$ yields the proof. □

## Acknowledgements

We would like to thank all the anonymous reviewers for generously dedicating their time and expertise to evaluate our manuscript with insightful comments. This work is partially support by JST CREST JPMJCR21M2, JSPS KAKENHI Grant Number JP22H00521, JP22H03595, JP21K19767, JST/NSF Joint Research SICORP JPMJSC2107.

## 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.

Balle, B.; Bell, J.; Gascón, A.; and Nissim, K. 2019. The privacy blanket of the shuffle model. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, 638–667. Springer.

Berry, A. C. 1941. The accuracy of the Gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1): 122–136.

Bittau, A.; Erlingsson, Ú.; Maniatis, P.; Mironov, I.; Raghunathan, A.; Lie, D.; Rudominer, M.; Kode, U.; Tinnes, J.; and Seefeld, B. 2017. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th symposium on operating systems principles*, 441–459.

Cheu, A.; Smith, A.; Ullman, J.; Zeber, D.; and Zhilyaev, M. 2019. Distributed differential privacy via shuffling. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, 375–403. Springer.

Dong, J.; Roth, A.; and Su, W. J. 2022. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1): 3–37.

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

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.

Esseen, C. G. 1942. On the Liapunov limit error in the theory of probability. *Ark. Mat. Astr. Fys.*, 28: 1–19.

Feldman, V.; McMillan, A.; and Talwar, K. 2022. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, 954–964. IEEE.

Girgis, A. M.; Data, D.; Diggavi, S.; Kairouz, P.; and Suresh, A. T. 2021a. Shuffled Model of Federated Learning: Privacy, Accuracy and Communication Trade-Offs. *IEEE Journal on Selected Areas in Information Theory*, 2(1): 464–478.

Girgis, A. M.; Data, D.; Diggavi, S.; Suresh, A. T.; and Kairouz, P. 2021b. On the Renyi differential privacy of the shuffle model. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2321–2341.

Jorgensen, Z.; Yu, T.; and Cormode, G. 2015. Conservative or liberal? Personalized differential privacy. In *2015 IEEE 31st International Conference on Data Engineering*, 1023–1034.

Kairouz, P.; Oh, S.; and Viswanath, P. 2015. The composition theorem for differential privacy. In *International conference on machine learning*, 1376–1385. PMLR.

Kasiviswanathan, S. P.; Lee, H. K.; Nissim, K.; Raskhodnikova, S.; and Smith, A. 2011. What Can We Learn Privately? *SIAM Journal on Computing*, 40(3): 793–826.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Liu, R.; Cao, Y.; Chen, H.; Guo, R.; and Yoshikawa, M. 2021. Flame: Differentially private federated learning in the shuffle model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8688–8696.

Liu, Y.; Zhao, S.; Li, X.; Liu, Y.; and Chen, H. 2023. Echo: Privacy amplification for Personalized Private Federated Learning with Shuffle Model. *Proceedings of the AAAI Conference on Artificial Intelligence*

Niu, B.; Chen, Y.; Wang, B.; Wang, Z.; Li, F.; and Cao, J. 2021. AdaPDP: Adaptive Personalized Differential Privacy. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 1–10.