The Power of the Differentially Oblivious Shuffle in Distributed Privacy Mechanisms

Mingxun Zhou*[1] and Elaine Shi†[1]

[1]School of Computer Science, Carnegie Mellon University

August 9, 2022

Abstract

The shuffle model has been extensively investigated in the distributed differential privacy (DP) literature. For a class of useful computational tasks, the shuffle model allows us to achieve privacy-utility tradeoff similar to those in the central model, while shifting the trust from a central data curator to a "trusted shuffle" which can be implemented through either trusted hardware or cryptography. Very recently, several works explored cryptographic instantiations of a new type of shuffle with relaxed security, called *differentially oblivious (DO) shuffles*. These works demonstrate that by relaxing the shuffler's security from simulation-style secrecy to differential privacy, we can achieve asymptotical efficiency improvements. A natural question arises, can we replace the shuffler in distributed DP mechanisms with a DO-shuffle while retaining a similar privacy-utility tradeoff?

In this paper, we prove an optimal privacy amplification theorem by composing any locally differentially private (LDP) mechanism with a DO-shuffler, achieving parameters that tightly match the shuffle model. Our result asymptoticaly improves the recent work of Gordon et al., who initiated the study of distributed DP mechanisms in the DO-shuffle model. We also explore multi-message protocols in the DO-shuffle model, and construct mechanisms for the real summation and histograph problems. Our error bounds approximate the best known results in the multi-message shuffle-model up to sub-logarithmic factors. Our results also suggest that just like in the shuffle model, allowing each client to send multiple messages is fundamentally more powerful than restricting to a single message. As an application, we derive the result of using repeated DO-shuffling for privacy-preserving time-series data aggregation.

1 Introduction

In distributed differential privacy, a set of clients each hold some private data, and an *untrusted* server wants to perform some analytics over the union of the clients' data, while

*mingxunz@andrew.cmu.edu

†runting@gmail.com

preserving each individual client's privacy. The shuffle model, first proposed by Bittau et al. [11] in an empirical work, has become a popular model for implementing distributed differentially private mechanisms. In the shuffle model, we typically assume that each client sends one or more messages in a single shot, and a trusted shuffler takes the union of all clients' messages, randomly permutes them, and presents the shuffled result to the server. The server then performs some computation and outputs the analytics result. Specifically, the trusted shuffler guarantees the anonymity of all messages, such that the server can only see the union of all messages, without knowing the source of an individual message. For privacy, we want to guarantee that for two neighboring input configurations of the clients denoted $\mathbf{x}$ and $\mathbf{x}'$ respectively, the distributions of the server's view are "close". Numerous earlier works have shown that the shuffle model often allows us to achieve differentially private mechanisms whose utility approximates those of the central model (where the server is trusted and we only need privacy on the outcome of the analytics). Moreover, several works have shown that the trusted shuffler can be efficiently implemented either using trusted hardware [11] or using cryptographic protocols [1,2,4,10,13,15,16,20–22,25,27,32,36,39,44]. This makes the shuffle model a compelling approach not just in theory, but also in practical applications such as federated learning [31].

A couple very recent works [12,34,39] have suggested a relaxed shuffler model called the *differentially oblivious shuffle* model (or *DO-shuffle model* for short). Unlike the traditional shuffle model which provides full anonymity on the clients' messages, the DO-shuffle model shuffles the clients' messages but possibly allowing some differentially private leakage. More concretely, a DO-shuffle protocol guarantees that for two *neighboring* input vectors $\mathbf{x}_H$ and $\mathbf{x}'_H$ corresponding to the set of honest parties, the adversary's views in the protocol execution are computationally or statistically close. The recent works by Gordon et al. [33] and Shi and Wu [39] both show that the relaxed DO-shuffle can be asymptotically more efficient to cryptographically realize than a fully anonymous shuffle.

The results of Gordon et al. [33] and Shi and Wu [39] suggest that the DO-shuffle model might be a compelling alternative to the shuffle model. This raises a very natural question:

> *If we were to replace the shuffler in shuffle-model differentially private (DP) mechanisms with a DO-shuffler, can we still get comparable privacy-utility tradeoff?*

A particular useful type of theorem in the shuffle model is called a privacy amplification theorem, which we explain below. Henceforth, let $\mathcal{R}_i(x_i)$ be some differentially private mechanism each client i applies to randomize its own private input x_i. This kind of randomizers is generally called **locally differentially private** (LDP) randomizers because the randomization is done locally by the client itself. Let $\mathcal{S}$ denote the random shuffler. Roughly speaking, a privacy amplification theorem makes a statement of the following nature: if each client's LDP mechanism $\mathcal{R}_i$ consumes ϵ_0 privacy budget, then the overall shuffle-model mechanism $\mathcal{S}(\mathcal{R}_1(x_1), \ldots, \mathcal{R}(x_n))$ satisfies (ϵ, δ)-DP for $\epsilon = \epsilon(\epsilon_0, \delta) \ll \epsilon_0$, i.e., privacy is amplified for the overall shuffle-model mechanism. A line of work [7,18,24] has focused on proving privacy amplification theorems for the shuffle model, culminating in the recent work by Feldman et al. [26], who proved a privacy amplification theorem for any LDP mechanism with optimal parameters. Therefore, another meaningful question we can ask is (which is a special case of the aforementioned question):

Can we also prove an optimal privacy amplification theorem for the DO-shuffle model?

The pioneering work of Gordon et al. [33] was the first to explore how to use a DO shuffler to design distributed differentially private mechanisms. Gordon et al. [33] showed two novel results. First, they prove an optimal privacy amplification theorem for the randomized response mechanism in the DO-shuffle model, with parameters that tightly match the shuffle-model counterpart. Next, they generalize their first result, and prove a privacy amplification theorem for any local differentially private (LDP) mechanism — however, this more general result is *non-optimal*, since they rely on the non-optimal shuffle-model amplification theorem from Balle et al. [7].

1.1 Our Results and Contributions

We show several new results regarding the power of the DO-shuffle model in designing distributed DP mechanisms.

First, we prove a privacy amplification theorem for any LDP mechanism that achieves *optimal* parameters, tightly matching the optimal privacy amplification theorem of the shuffle model due to the recent work of Feldman et al. [26]. This result improves work of Gordon et al. [33] in the following senses: 1) we asymptotically improve their privacy amplification theorem for any general LDP mechanism; and 2) their privacy amplification theorem for the specific randomized response mechanism can be viewed as a special case of our general theorem.

Throughout the paper, we use Φ to denote a DO-shuffling protocol. Given an LDP-randomizer $\mathcal{R}(\cdot)$, we use the notation $\Pi(x_1, \ldots, x_n) := \Phi(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$ to denote the composed protocol where each of the n parties first applies the local randomizer $\mathcal{R}(\cdot)$ to its own private data, and then invokes an instance of the DO-shuffling protocol Φ on the outcome $\mathcal{R}(x_i)$.

In all of our theorems below, if the underlying DO-shuffle protocol satisfies semi-honest security [5, 33], then our result holds in a semi-honest corruption model. Similarly, if the underlying DO-shuffle satisfies malicious security [5, 12], then our result also holds in a malicious model.

Theorem 1 (Optimal privacy amplification for any LDP mechanism in the DO-shuffle model)**.** *Given n copies of an ϵ_0-LDP randomizer $\mathcal{R}$ and an (ϵ_1, δ_1)-DO shuffler Φ resilient to t corrupted parties, the composed protocol $\Pi(x_1, \ldots, x_n) := \Phi(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$ is $(\epsilon + \epsilon_1, \delta + \delta_1)$-DP against up to t corrupted parties where*

$$\epsilon = O\left(\frac{(1 - e^{\epsilon_0})e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n-t}}\right).$$

Furthermore, if the DO-shuffler satisfies computational (or statistical, resp.) DO, then the composed protocol satisfies computation (or statistical, resp.) DP.

This result matches the optimal privacy amplification in the shuffle-DP model from Feldman et al. [26] except for the additional privacy loss (ϵ_1, δ_1) that comes from the DO shuffler.

Second, we consider another natural direction that was previously unexplored, that is, the *multi-message* DO-shuffle model. Recall that in the traditional shuffle model, several works [8, 28] have shown that allowing clients to send multiple messages is fundamentally more powerful than restricting to a single message per client. In particular, the multi-message shuffle-model allows us to overcome lower bounds pertaining to the single-message shuffle model [7, 28], and design mechanisms with asymptotically better privacy-utility tradeoff [6, 8, 28]. We explore the power of the multi-message DO-shuffle model, and show positive results for two fundamental tasks in private data analytics, that is, *histogram*, and *real-valued summation*. Our DO-shuffle model mechanisms can match the utility of the best known shuffle-model mechanisms upto only sublogarithmic factors. Earlier works on data privacy showed that these two primitives have important applications, such as user feedback collection [11] and federated learning [31].

In the *histogram* problem, each client i holds an item x_i in domain $[d]$ and the server wants to compute the histogram h, such that for each $x \in [d]$, $h_x = \sum_{i\in[n]} \mathbf{I}[x_i = x]$ in a privacy-preserving manner. We construct a histogram protocol under the DO-shuffle model based on the multi-message shuffle-DP protocol from Ghazi et al. [30].

Theorem 2 (Multi-message DO-shuffling private histogram protocol)**.** *For $\epsilon \le 1$, given an (ϵ_1, δ_1)-DO shuffler, there is an $O\left(\frac{\log(1/\epsilon\delta)}{\epsilon^2}\right)$-message $(\epsilon + \epsilon_1, \delta + \delta_1)$ DO-shuffle summation protocol against $t < n/2$ corrupted parties that achieves $\tilde{O}_{\epsilon,\delta}(\log d)$ L_∞ error.*

Ghazi et al. [28] proved the L_∞ any (ϵ, δ)-*single-message* shuffle-DP histogram protocol must incur at least $\Omega_{\epsilon,\delta}(\min(\sqrt[4]{n}, \sqrt{d}))$ error — obviously, this lower bound applies to the DO-shuffle model too. Thus, our aforementioned result shows that in the DO-shuffle model too, permitting multiple messages per client can lead to mechanisms with fundamentally better privacy-utility tradeoff than the single-message model. Moreover, the L_∞-error of the protocol is only $O(\sqrt{\log d})$-factor worse than the best-known results for shuffle-DP histogram [19].

In the *real summation* problem, there are n clients and client i holds a *real* value $x_i \in [0, 1]$. The server wants to compute the summation $X = \sum_{i\in[n]} x_i$, while protecting the client's privacy. Based on the shuffle-DP protocol by Balle et al. [8], we construct a protocol in the DO-shuffle model:

Theorem 3 (Multi-message DO-shuffling private summation protocol)**.** *For $\epsilon \le 1$, given an (ϵ_1, δ_1)-DO shuffler, there is an m-message $(\epsilon + m\epsilon_1, \delta + m\delta_1)$ DO-shuffle summation protocol against $t < n/2$ corrupted parties that achieves $\tilde{O}_\delta\left(\frac{(\log\log n)^2}{\epsilon^2}\right)$ mean square error with $m = \lfloor \log_3(\log_2 n) \rfloor$.*

Balle et al. [7] proved the mean square error for any (ϵ, δ)-single-message shuffle-DP summation protocol is $\tilde{\Omega}_{\epsilon,\delta}(n^{1/3})$. This lower bound natrually applies to the DO-shuffle model. Thus, we show that the separation between single-message protocol and multi-message protocol still holds in the DO-shuffle model. Also, the mean square error of this DO-shuffle summation protocol is only $O((\log\log n)^2)$ factor worse than the best-known protocol [8, 29].

Finally, recall that both Gordon et al. [33] and Bunz et al. [12], construct asymptotically efficient DO-shuffle protocols that achieve $\epsilon = o(1/\log\log n)$ privacy loss with a negligibly small δ. Therefore, if we instantiate the protocol in Theorem 2 using the DO-shuffle scheme of Gordon et al. [33] or Bunz et al. [12], the total privacy loss introduced by DO-shuffle is only by $o(1)$.

Applications to time-series data aggregation. The aforementioned theorems for distributed differential privacy are stated a single-shot data aggregation scenario for generality. Previous DO-shuffler construction, like Bunz et al. [12], also supports a repeated setting without incurring privacy budget loss over time. Specifically, in the shuffler construction, given a one-time setup, the permutation used for shuffling messages is repeteadly used for all following rounds. Therefore, in Section 3.3, we explore potential applications of such DO shufflers to privacy-preserving time-series data aggregation. The scenario is similar to those described in a line of prior works [37, 38]: a trusted server wants to learn some useful statistics over n clients' data, and such an aggregation is performed repeatedly over time. In such applications settings, we can compose LDP randomized mechanisms with our the shuffler. We show a slightly unexpected result that although the privacy loss from the LDP mechanisms accumulates, the privacy loss from the DO shuffler does not degrade over time.

2 Preliminaries and Definitions

We review the definitions for standard differential privacy(DP) in the appendix A. Here, we provide the definitions for differentially oblivious(DO) protocols and specifically, DO shufflers.

2.1 Differentially Private Protocols and the DO-Shuffle Model

Definition 17 defines a DP-protocol in the shuffle model. The definition is straightforward partly because we are modeling the shuffle as an idealized trusted party that simply takes the clients' messages and outputs the randomly shuffled result. When we consider protocols in the DO-shuffle model, it is not so clear how to represent the DO-shuffle as an ideal functionality. Instead, earlier works [34, 39] that give a formal treatment of DO shuffle protocols typically define security (possibly against computationally bounded adversaries) in a protocol execution model that is standard practice in the cryptography literature.

Therefore, in this section, we give a more general notion of differentially private protocols. This will allow us to formally model a DO-shuffle, as well as formally model the privacy of protocols in the DO-shuffle model.

To define a differentially private protocol, imagine that n parties run a protocol each with a private input. The adversary controls a set of at most t parties. In the *semi-honest* corruption model, the corrupt parties follow the protocol faithfully but are curious and may try to extract information about others' secret inputs. In the *malicious* corruption model, the corrupt parties can deviate arbitrarily from the prescribed protocol. We often use the notation $\mathbf{x}_H$ to denote the inputs corresponding to the honest parties.

Definition 4 (Differentially private protocol)**.** *Let $\sim$ be some neighboring relation. We say that a protocol Π is a computational (ϵ, δ)-differentially oblivious protocol (or a computational*

(ϵ, δ)-DO protocol for short) w.r.t. $\sim$ against t corruptions, iff for any non-uniform probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ controlling at most t parties, for any neighboring honest input vectors $\mathbf{x}_H^0 \sim \mathbf{x}_H^1$, it holds that

$$\Pr[\mathbf{Expt}_{\mathcal{A}}^{\Pi,0}(1^\lambda, \mathbf{x}_H^0) = 1] \leq e^\epsilon \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Pi,0}(1^\lambda, \mathbf{x}_H^1) = 1] + \delta,$$

where for $b \in \{0,1\}$, the experiment $\mathbf{Expt}_{\mathcal{A}}^{\Pi,b}(1^\lambda)$ simply samples a random execution of Π using honest inputs $\mathbf{x}_H^b$ and involving the adversary $\mathcal{A}$, and outputs the adversary $\mathcal{A}$'s output at the end.

Furthermore, if $\mathcal{A}$ is a semi-honest (or malicious resp.) adversary, we say that the protocol is (ϵ, δ)-DP in the semi-honest (or malicious resp.) model.

Later in the paper, when the neighboring relationship $\sim$ is clear, we often omit writing it explicitly, and simply say that a protocol Π satisfies computational (ϵ, δ)-DO against t corruptions. The above definition is for the case of computational security since we require the adversary $\mathcal{A}$ to be polynomially bounded. If we remove the PPT requirement for $\mathcal{A}$, we would then have the definition of *statistical* DO protocol. Now we define the (ϵ, δ)-DO shuffler as following:

Definition 5 (Differentially Oblivious Shuffler)**.** *Given n clients and each one of them holds a message x_i. A protocol Φ is a computation (or statistical resp.) (ϵ, δ)-DO shuffler against up to t corrupted parties iff*

1. *$\Phi(\{y_1 \ldots y_n\})$ outputs a random permutation of $y_1, \ldots, y_n$ when all parties perform honestly; and*

2. *Φ is a computational (or statistical resp.) (ϵ, δ)-DP protocol against up to t corrupted parties with respect to the following neighboring relation: two honest input vectors $\mathbf{x}_H^0$ and $\mathbf{x}_H^1$ are said to be neighboring, iff there exists a pair $i \neq j$ such that $\mathbf{x}_H^0$ and $\mathbf{x}_H^1$ are equal in all other coordinates, except for swapping the coordinates i and j, i.e., $\mathbf{x}_{H,i}^0 = \mathbf{x}_{H,j}^1, \mathbf{x}_{H,j}^0 = \mathbf{x}_{H,i}^1$.*

3 Optimal Privacy Amplification by Differentially Oblivious Shuffler

In this section, we provide an optimal privacy amplification theorem for composing DO shufflers and any general LDP mechanism. Our proof combines techniques from Feldman et al. [26] and Gordon et al. [33] in a non-blackbox manner.

3.1 Background on Optimal Privacy Amplification in the Shuffle Model

Gordon et al. [33]'s privacy amplification result using DO-shuffler relies on the non-optimal shuffle-model privacy amplification result by Balle et al. [7], and achieves parameters that tightly match Balle et al. [7]. Balle et al. [7] states that given n copies of ϵ_0-LDP randomizers, the composition protocol of the LDP randomizer and a perfect shuffler satisfies (ϵ, δ)-DP for $\epsilon = O\left(\frac{e^{\epsilon_0}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$. By contrast, the optimal shuffle-model privacy amplification result [26] achieves $\epsilon = O\left(\frac{e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$. The difference between e^{ϵ_0} and $e^{\epsilon_0/2}$ could be crucial in many cases as argued by Feldman [26], especially in the regime where $\epsilon_0 > 1$ which is commonly used in practice. This regime also naturally arises when the target central ϵ is greater than $1/\sqrt{n}$ [26]. As an intuitive example, imagine that there are n clients each with a private bit, and the server wants to estimate the summation of all clients' bits. Suppose that all clients employ the standard randomized response mechanims [43] and all clients' messages are permuted by trusted shuffler. Now, fix the target privacy budget $\epsilon = 1$. If we were to apply the amplification theorem of Balle et al. [7], we would derive an estimation error of $\widetilde{O}(n^{1/4})$ where $\widetilde{O}$ hides terms related to $\log\frac{1}{\delta}$; whereas the amplification theorem of Feldman [26] gives a tight error bound of $\widetilde{O}(1)$.

We therefore want an optimal privacy amplification theorem for the DO-shuffle model as well, for the same motivation as the work of Feldman et al. [26]. To understand how DO shufflers can compose with any general LDP mechanism, we first need to understand the proof of the optimal privacy amplification theorem for the shuffle model [26].

Theorem 6 (Optimal privacy amplification in the shuffle model [26])**.** *Suppose $\mathcal{R}$ is an ϵ_0-DP function where $\epsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$. Then, given n copies of $\mathcal{R}$ and a perfectly random shuffler $\mathcal{S}$, the function $\mathcal{S} \circ (\mathcal{R} \times \cdots \times \mathcal{R})$ is (ϵ, δ)-DP, where $\epsilon = O\left(\frac{(1-e^{-\epsilon_0})e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$.*

In the following, we will explain the important steps in the proof from Feldman et al. [26], which are needed later when we replace the perfect shuffler with a DO shuffler.

Without loss of generality, fix a pair of neighboring input configurations $\mathbf{x}^0 = (x_1, x_2 \ldots, x_n)$ and $\mathbf{x}^1 = (x_1', x_2 \ldots, x_n)$ that only differ in the first client's input. Since $\mathcal{R}$ is an ϵ_0-LDP randomizer, for any $x, x' \in \mathcal{X}$ where $\mathcal{X}$ is the input domain and any $y \in \mathcal{Y}$ where $\mathcal{Y}$ is the randomizer's output domain, we have $\frac{\Pr[\mathcal{R}(x)=y]}{\Pr[\mathcal{R}(x')=y]} \ge e^{-\epsilon_0}$. Thus, we can "decompose" the distribution of $\mathcal{R}(x')$ as $\mathcal{R}(x') = e^{-\epsilon_0}\mathcal{R}(x) + (1-e^{-\epsilon_0})\mathcal{Q}(x, x')$. It means that the distribution of $\mathcal{R}(x')$ can be seen as a mixture of $\mathcal{R}(x)$ and some appropriate leftover distribution $\mathcal{Q}(x, x')$. Similarly, for all x_i where $i = 2, \ldots, n$, we can decompose the output distribution of $\mathcal{R}(x_i)$ as

$$\mathcal{R}(x_i) = \begin{cases} \mathcal{R}(x_1) & w.p. \frac{1}{2e^{\epsilon_0}}, \\ \mathcal{R}(x_1') & w.p.\ \frac{1}{2e^{\epsilon_0}}, \\ \mathcal{Q}(x_1, x_1', x_i), & w.p.\ 1 - \frac{1}{e^{\epsilon_0}}. \end{cases}$$

Here, $\mathcal{Q}(x_1, x_1', x_i)$ is also some appropriate left-over distribution after the decomposition. The properties of $\mathcal{Q}(x_1, x_1', x_i)$ are irrelevant in this proof, so we only need to know $\mathcal{Q}(x_1, x_1', x_i)$ is a legal and well-defined distribution. From the decomposition, we see that with probability of $\frac{1}{2e^{\epsilon_0}}$, client i will send a message that has the exact same distribution as $\mathcal{R}(x_1)$. We say it **"clones"** $\mathcal{R}(x_1)$. Similarly, client i will send a message that has the exact same distribution as $\mathcal{R}(x_1')$ with probability of $\frac{1}{2e^{\epsilon_0}}$, in which case we say it clones $\mathcal{R}(x_1')$. The heart of the proof is to see that the message from client 1 hides in many clones of $\mathcal{R}(x_1)$ and $\mathcal{R}(x_1')$, so whether the client holds x_1 or x_1' only makes a very small difference.

Denote $\mathrm{Bin}(p, n)$ as the binomial distribution of the number of successful trials from n independent Bernouli experiments with individual success probability of p. We use the following process to generate a shuffled message sequence that has the identical distribution as the output of the shuffler, i.e., $\mathcal{S}(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$:

1. Sample $\alpha \sim \mathrm{Bin}(e^{-\epsilon_0}, n-1)$ where α is the number of clients that clone $\mathcal{R}(x_1)$ or $\mathcal{R}(x_1')$;
2. Sample $\beta \sim \mathrm{Bin}(1/2, c)$ where β is the number of clients that clone $\mathcal{R}(x_1)$;
3. $(h_1, h_1') = (\beta+1, \alpha-\beta)$, i.e., including client 1, there are exactly $h_1 = \beta+1$ clients that are submitting a message from the distribution $\mathcal{R}(x_1)$, and $\alpha-\beta$ clients are submitting a message from the distribution $\mathcal{R}(x_1')$;
4. Generate h_1 i.i.d. messages from distribution $\mathcal{R}(x_1)$. Generate h_1' i.i.d. messages from distribution $\mathcal{R}(x_1')$.
5. Randomly sample $n-1-\alpha$ indices from $\{2, \ldots, n\}$, say $j_1 \ldots, j_{n-1-\alpha}$. For each sampled $j \in \{j_1 \ldots, j_{n-1-\alpha}\}$, generate a message from $\mathcal{Q}(x_1, x_1', x_j)$.
6. Shuffle and output all the messages.

From the decomposition of the LDP randomizers, it is not difficult to see the process generates a message sequence that has the identical distribution as $\mathcal{S}(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$. We can use a similar process to generate a message sequence that has the identical distribution as $\mathcal{S}(\mathcal{R}(x_1'), \ldots, \mathcal{R}(x_n))$. The only difference is, in step 3, we let $(h_1, h_1') = (\beta, \alpha-\beta+1)$ because now client 1 holds x_1' and will send a message of $\mathcal{R}(x_1')$.

Notice that after fixing (h_1, h_1'), Steps 4 to 6 are just post-processing steps. If we can prove that the distributions of (h_1, h_1') in the two processes are (ϵ, δ)-close, we finish the proof. Luckily, the distributions of (h_1, h_1') are not complicated. We just need to analyze the divergence between $(\beta+1, \alpha-\beta)$ and $(\beta, \alpha-\beta+1)$, where $\alpha \sim \mathrm{Bin}(e^{-\epsilon_0}, n-1)$ and $\beta \sim \mathrm{Bin}(1/2, c)$. This part of the proof is omitted and the conclusion is that the distributions are (ϵ, δ)-close, where $\epsilon = O\left(\frac{(1-e^{-\epsilon_0})e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$. We refer the interested readers to Feldman et al. [26] for more details on the divergence calculation.

3.2 Optimal Privacy Amplification by DO Shufflers

We now provide the proof for the optimal privacy amplification theorem in the DO-shuffle model. We combine the aforementioned techniques from Feldman et al. [26] and techniques from Gordon et al. [33] in a non-black-box manner.

Theorem 7 (Optimal privacy amplification of general LDP randomizers by a DO shuffler)**.** *Let $\sim$ be the neighboring relation where two honest input configurations $\mathbf{x}_H^0 \sim \mathbf{x}_H^1$ are neighboring iff they only differ in one client's input. Suppose $\mathcal{R}$ is an ϵ_0-DP randomizer. Then, given n copies of $\mathcal{R}$ and an (ϵ_1, δ_1)-DO shuffler Φ against up to t corruptions. The composed protocol $\Pi(x_1, \ldots, x_n) := \Phi(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$ is $(\epsilon+\epsilon_1, \delta+\delta_1)$-DP against up to t corrupted parties w.r.t. the neighboring relation $\sim$, where $\epsilon = O\left(\frac{(1-e^{\epsilon_0})e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n-t}}\right)$.*

Proof. Denote $c = n - t > 1$ as the number of honest clients. W.L.O.G, suppose that the corrupted set is $\{n - t + 1, \ldots, n\}$. Then clients 1 to client c are honest. Also, let the neighboring input configurations be $\mathbf{x}_H^0 = (x_1, x_2, \ldots, x_c)$ and $\mathbf{x}_H^1 = (x'_1, x_2, \ldots, x_c)$.

Throughout the proof, we refer *messages* as the results of the clients applying the local randomizer to their private inputs, which are later treated as the new input messages in the execution of the (interactive) shuffle protocol.

After fixing (h_1, h'_1), i.e., the number of clients that will submit the message sampled from μ and μ' correspondingly, the following steps are post-processing. Therefore, even if the adversary view includes the final shuffled outcome and also the auxiliary information about the indices and messages generated for those clients that are not "cloning", the view is still (ϵ, δ)-DP with the same ϵ, δ in theorem 6.

In the proof of theorem 6, there are three types of clients: 1) client 1 who has different input in two neighboring input configurations; 2) the clients that clone either $\mathcal{R}(x_1)$ or $\mathcal{R}(x'_1)$; 3) the clients that do not clone. Let M be the multi-set of all honest parties' messages. Let V be the auxiliary information vector includes the messages from those clients that do not clone $\mathcal{R}(x_1)$ or $\mathcal{R}(x'_1)$. For example, suppose there are five clients. Client 2 and client 3 are cloning and client 4 and client 5 are not. Then, the auxiliary information vector will be $V = (\perp, \perp, \perp, y_4, y_5)$, such that y_4 and y_5 are the messages sent by client 4 and client 5 and the dummy symbol $\perp$ means the V does not include the information about those clients' messages. Since V can be constructed by the post-processing step (step 5 in the proof of theorem 6), we know the distribution of (M, V) is (ϵ, δ)-DP.

Denote the message sequence sent from client 1 to client c to the shuffler as $\mathbf{y}_H = (y_1, \ldots, y_c)$. Let $\overline{M}$ be the message set includeing all messages sent by client 1 and those cloning clients. We first assume $\overline{M}$ doesn't contain duplicate messages and we will handle the duplication later. Let $l = |\overline{M}|$ and the messages in $\overline{M}$ be $m_1, \ldots, m_l$. Let $j_1, \ldots, j_l$ be the indicies for client 1 and those cloning clients. Notice that $j_1 = 1$ because client 1 is included by definition. There are $l!$ message sequences $\mathbf{y}_H$ such that the messages sent by client $j_1, \ldots, j_l$ are a permutation of $\overline{M}$ and we say the set for those legal sequences is $\mathcal{Y}$.

We define μ as the distribution of $\mathcal{R}(x_1)$ and μ' as the distribution of $\mathcal{R}(x'_1)$. Also, we define $\mu(\cdot)$ and $\mu'(\cdot)$ as their probability density functions. For any $i \in \{j_2, \ldots, j_l\}$, we know client i will clone either $\mathcal{R}(x_1)$ or $\mathcal{R}(x'_1)$ with $\frac{1}{2}$ probability. Therefore, the client will submit a message sampled from the mixture distribution $\omega = \frac{1}{2}\mu + \frac{1}{2}\mu'$. We also denote $\omega(\cdot)$ as its probability density function.

Given the input configuration of $\mathbf{x}_H^0$, define $\Pr_0[A]$ as the probability of observing some event A. Conditioned on the auxiliary information vector V, by enumerating all $l!$ sequences $\mathbf{y}_H = (y_1, \ldots, y_c)$ in $\mathcal{Y}$, we have

$$\Pr_0[M \mid V] = \sum_{\mathbf{y}_H \in \mathcal{Y}} \mu(y_1)\omega(y_{j_2}) \cdots \omega(y_{j_l}) = \sum_{\mathbf{y}_H \in \mathcal{Y}} \frac{\mu(y_1)}{\omega(y_1)} \prod_{y \in \overline{M}} \omega(y).$$

Similarly, define $\Pr_1[A]$ as the probability of observing some event A. We have $\Pr_1[M \mid V] = \sum_{\mathbf{y}_H \in \mathcal{Y}} \frac{\mu'(y_1)}{\omega(y_1)} \prod_{y \in \overline{M}} \omega(y)$. Denote D^0 as the distribution on the sequences in $\mathcal{Y}$ conditioned on M, V and let $D^0(\mathbf{y}_H) = \Pr[\mathbf{y}_H \mid M, V]$. By Bayes' rule, we have for each $\mathbf{y}_H \in \mathcal{Y}$,

$$\begin{aligned} D^0(\mathbf{y}_H) &= \Pr_0[\mathbf{y}_H \mid M, V] \\ &= \frac{\Pr_0[\mathbf{y}_H \mid V] \Pr_{\mathbf{x}_H^0}[M \mid V, \mathbf{y}_H]}{\Pr_0[M \mid V]} \\ &= \frac{\Pr_0[\mathbf{y}_H \mid V] \cdot 1}{\Pr_0[M \mid V]} \\ &= \frac{\frac{\mu(y_1)}{\omega(y_1)} \prod_{y \in \overline{M}} \omega(y)}{\sum_{\mathbf{y}' = (y_1', \dots, y_c') \in \mathcal{Y}} \frac{\mu(y_1')}{\omega(y_1')} \prod_{y \in \overline{M}} \omega(y)} \\ &= \frac{\mu(y_1)/\omega(y_1)}{\sum_{\mathbf{y}' \in \mathcal{Y}} \mu(y_1')/\omega(y_1')} \end{aligned}$$

Notice that given any $\mathbf{y}_H \in \mathcal{Y}$, $D^0(\mathbf{y}_H)$ is solely determined by y_1 and M, V. Thus, if we group those $l!$ possible sequences according to the message sent from client 1, i.e., y_1, there are l groups and each group has $(l-1)!$ sequences. In the i-th group where $y_1 = m_i$, all the sequences have the same conditional probability. Thus, we can rewrite $D^0(\mathbf{y}_H)$ as

$$D^0(\mathbf{y}_H) = \frac{1}{(l-1)!} \frac{\mu(y_1)/\omega(y_1)}{\sum_{y \in \overline{M}} \mu(y)/\omega(y)}$$

Similarly, we can define conditional distribution D^1 on $\mathcal{Y}$ (conditioned on M, V) when client 1's input is x_1' and we have $D^1(\mathbf{y}_H) = \frac{1}{(l-1)!} \frac{\mu'(y_1)/\omega(y_1)}{\sum_{y \in \overline{M}} \mu'(y)/\omega(y)}$.

For any PPT adversary $\mathcal{A}$, define $\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H)$ as $\mathcal{A}$'s output after the execution of the shuffler Φ given the honest clients' messages $\mathbf{y}_H$ and the presence of $\mathcal{A}$. We know that given M, V, we have the two corresponding conditional distributions over all possible honest parties' input sequences $\mathcal{Y}$, i.e., D^0 and D^1. Then, we try to prove the following lemma:

Lemma 8. *For all M, V, let D^0 and D^1 be the corresponding conditional distribution over the message sequences. For any PPT $\mathcal{A}$:*

$$\Pr_{\mathbf{y}_H \xleftarrow{\$} D^0} [\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \le e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1} [\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] + \delta_1$$

For $b \in \{0,1\}$, the randomness of the probability $\Pr_{\mathbf{y}_H \xleftarrow{\$} D^b}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1]$ comes from two steps: 1) the random coin used to sample $\mathbf{y}_H$ from distribution D^b; 2) all the random coins used in experiment $\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H)$.

Proof. W.L.O.G, let's say the clients are cloning are client 2 to client l. Now, the message sequence set is all the permutations of $\overline{M}$, i.e., $\mathcal{Y} = \mathrm{Perm}(\overline{M})$. Recall the definition of neighbors for the DO shuffler.

Imagine that we can find a bijection $g : \mathcal{Y} \to \mathcal{Y}$ that maps a message sequence $\mathbf{y}_H$ to one of its neighbors $g(\mathbf{y}_H)$, such that the conditional probabilities are the same, i.e., $D^0(\mathbf{y}_H) = D^1(g(\mathbf{y}_H))$. Using the DO property of the shuffler, we have $\Pr[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \le e^{\epsilon_1} \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(g(\mathbf{y}_H)) = 1] + \delta_1$. Then we have a very straightforward way to prove that:

$$
\begin{aligned}
&\Pr_{\mathbf{y}_H \xleftarrow{\$} D^0}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \\
=&\sum_{\mathbf{y}_H \in \mathcal{Y}} D^0(\mathbf{y}_H) \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \\
\leq &\sum_{\mathbf{y}_H \in \mathcal{Y}} D^1(g(\mathbf{y}_H))(e^{\epsilon_1} \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(g(\mathbf{y}_H)) = 1] + \delta_1) \\
=&e^{\epsilon_1} \left(\sum_{\mathbf{y}_H \in \mathcal{Y}} D^1(\mathbf{y}_H) \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \right) + \delta_1 \\
=&e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] + \delta_1
\end{aligned}
$$

Unfortunately, such a mapping g may not exist. However, we can do the mapping in a more fine-grained way. This mapping idea comes from Gordon et al. [33]. For each sequence $\mathbf{y}_H$ in $\mathcal{Y}$, we make l copies of the sequence, where the i-th copy is denoted as $\mathbf{y}_H^{(i)}$. Given some $\mathbf{y}_H = (y_1, \ldots, y_c)$ and $\mathbf{y}'_H = (y'_1, \ldots, y'_c)$ such that $\mathbf{y}_H$ and $\mathbf{y}'_H$ are neighbors where they transpose the first entry and another entry (e.g. the sequence of $(1, 2, 3)$ and $(3, 2, 1)$ transpose the first entry and the third entry). Recall the message sets $\overline{M} = \{m_1, \ldots, m_l\}$. If the neighboring pairs $\mathbf{y}_H$ and $\mathbf{y}'_H$ have $y_1 = m_j$ and $y'_1 = m_i$, then we map $\mathbf{y}_H^{(i)}$ to $\mathbf{y}_H^{'(j)}$. That is, we set $g\left(\mathbf{y}_H^{(i)}\right) = \mathbf{y}_H^{'(j)}$. For example, if $\overline{M} = \{1, 2, 3\}$ and the two sequences are $\mathbf{y}_H = (1, 2, 3), \mathbf{y}'_H = (3, 2, 1)$, then we will map $\mathbf{y}_H^{(3)}$ to $\mathbf{y}_H^{'(1)}$.

Denote the probability of the i-th sub-copy of $\mathbf{y}_H$ conditioned on M, V as $D^0(\mathbf{y}_H^{(i)})$. We can freely assign the probability to the sub-copies, as long as they are non-negative and $\sum_{i \in [l]} D^0(\mathbf{y}_H^{(i)}) = D^0(\mathbf{y}_H)$. The remaining work is to make the sub-copies connected by the mapping g have the same probability. We set

$$
\begin{aligned}
D^0(\mathbf{y}_H^{(i)}) &= D^0(\mathbf{y}_H) \cdot \frac{\mu'(m_i)/\omega(m_i)}{\sum_{y \in \overline{M}} \mu'(y)/\omega(y)} \\
&= \frac{1}{(l-1)!} \frac{\mu(m_j)/\omega(m_j)}{\sum_{y \in \overline{M}} \mu(y)/\omega(y)} \cdot \frac{\mu'(m_i)/\omega(m_i)}{\sum_{y \in \overline{M}} \mu'(y)/\omega(y)}.
\end{aligned}
$$

We have

$$
\sum_{i \in [l]} D^0(\mathbf{y}_H^{(i)}) = \sum_{i \in [l]} D^0(\mathbf{y}_H) \frac{\mu'(m_i)/\omega(m_i)}{\sum_{y \in \overline{M}} \mu'(y)/\omega(y)} = D^0(\mathbf{y}_H) \frac{\sum_{i \in [l]} \mu'(m_i)/\omega(m_i)}{\sum_{y \in \overline{M}} \mu'(y)/\omega(y)} = D^0(\mathbf{y}_H).
$$

The last equation is simply by definition that $\overline{M} = \{m_1 \ldots m_l\}$. Similarly, for the j-th sub-copy of $\mathbf{y}'_H$, we set its probability as

$$
\begin{aligned}
D^1(\mathbf{y}_H'^{(j)}) &= D^1(\mathbf{y}_H') \cdot \frac{\mu(m_j)/\omega(m_j)}{\sum_{y\in\overline{M}} \mu'(y)/\omega(y)} \\
&= \frac{1}{(l-1)!} \frac{\mu'(m_i)/\omega(m_i)}{\sum_{y\in\overline{M}} \mu'(y)/\omega(y)} \cdot \frac{\mu(m_j)/\omega(m_j)}{\sum_{y\in\overline{M}} \mu(y)/\omega(y)}.
\end{aligned}
$$

It is easy to see that $D^0(\mathbf{y}_H^{(i)}) = D^1(\mathbf{y}_H'^{(j)})$. To see the mapping is a bijection, notice that for all $l!$ possible sequences, by exchanging two messages, we have one unique neighboring pair. Thus, there are in total $l \cdot l!$ different neighboring pairs. Also, we have $l \cdot l!$ sub-copies. Each sub-copy is mapped to another sub-copy and each mapping pairs corresponds to a unique neighboring pairs. Thus, this mapping is a bijection. We again use g to denote this mapping, then we have

$$
\begin{aligned}
&\Pr_{\mathbf{y}_H \xleftarrow{\$} D^0}[\mathbf{Expt}^{\Phi}_{\mathcal{A}}(\mathbf{y}_H) = 1] \\
=& \sum_{\mathbf{y}_H \in \mathcal{Y}} D(\mathbf{y}_H) \Pr[\mathbf{Expt}^{\Phi}_{\mathcal{A}}(\mathbf{y}_H) = 1] \\
=& \sum_{\mathbf{y}_H \in \mathcal{Y}} \sum_{i\in[l]} D(\mathbf{y}_H^{(i)}) \Pr[\mathbf{Expt}^{\Phi}_{\mathcal{A}}(\mathbf{y}_H) = 1] \\
\le& \sum_{\mathbf{y}_H \in \mathcal{Y}} \sum_{i\in[l]} D'(g(\mathbf{y}_H^{(i)}))(e^{\epsilon_1} \Pr[\mathbf{Expt}^{\Phi}_{\mathcal{A}}(g(\mathbf{y}_H)) = 1] + \delta_1) \\
=& e^{\epsilon_1} \left(\sum_{\mathbf{y}_H \in \mathcal{Y}} D'(\mathbf{y}_H) \Pr[\mathbf{Expt}^{\Phi}_{\mathcal{A}}(\mathbf{y}_H) = 1] \right) + \delta_1 \\
=& e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^0}[\mathbf{Expt}^{\Phi}_{\mathcal{A}}(\mathbf{y}_H) = 1] + \delta_1.
\end{aligned}
$$

Now we describe how to handle duplication. Assume the histogram $\overline{M}$ has unique items $m_1, \ldots, m_q$ where the count of item m_i is $\overline{M}(m_i)$ and $\sum_{i\in[t]} \overline{M}(m_i) = l$. We simply give each item a unique tag to distinguish them (e.g., $(5,5,5)$ are tagged as $(5^{(a)}, 5^{(b)}, 5^{(c)})$). After the tagging, we denote $\langle \mathbf{y}_H \rangle$ as tagged sequence and $\langle \mathcal{Y} \rangle$ as the set of all the tagged sequences. Then, if we consider all the permutation of those same elements with different tags, then each untagged sequence corresponds to $\prod_{i\in q} \overline{M}(m_i)!$ tagged sequence. In total, there are still $l!$ different tagged sequences. From symmetry, for each sequence $\langle \mathbf{y}_H \rangle \in \langle \mathcal{Y} \rangle$, we have

$$
\Pr_0[\langle \mathbf{y}_H \rangle \mid M, V] = \Pr_0[\mathbf{y}_H \mid M, V] \cdot \frac{1}{\prod_{i\in q} \overline{M}(m_i)!}.
$$

The similar expression holds for $\Pr_1[\langle \mathbf{y}_H \rangle \mid M, V]$. Then, we just treat those $l!$ sequences differently and the previous proof still holds. □

Completing the proof of theorem 7. Finally, to prove the whole protocol's privacy, we look at the definition of experiment $\mathbf{Expt}_{\mathcal{A}}^{\Pi,b}(1^\lambda)$ for any PPT adversary $\mathcal{A}$, any neighboring input configuration $\mathbf{x}_H^0, \mathbf{x}_H^1$:

1. Let $(x_1, \ldots, x_c) = \mathbf{x}_H^b$
2. Let $\mathbf{y}_H = (\mathcal{R}(x_1), \ldots, \mathcal{R}(x_c))$;
3. Return $\mathbf{Expt}_{\mathcal{A}}^{\Phi}(1^\lambda, \mathbf{y}_H)$; (Run the shuffler given message sequence $\mathbf{y}_H$ with the presence of $\mathcal{A}$)

We are now trying to prove that, for any PPT $\mathcal{A}$ and $\mathbf{x}_H^0 \sim \mathbf{x}_H^1$,

$$\Pr[\mathbf{Expt}_{\mathcal{A}}^{\Pi,0}(1^\lambda) = 1] \leq e^{\epsilon+\epsilon_1} \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Pi,1}(1^\lambda) = 1] + \delta + \delta_1.$$

W.L.O.G, let the randomizer $\mathcal{R}$'s output domain be a discrete set. The continuous case can be proven with a similar argument. Next, we will use the composition technique for DP mechanisms from Dwork and Roth [23], but change it accordingly to fit the computational requirement. Let $\mathcal{Z}$ be the domain of the random variables (M, V). Define

$$\Delta(M, V) = \max\{\Pr_0[M, V] - e^\epsilon \Pr_1[M, V], 0\}.$$

Then $\Pr_0[M, V] \leq e^\epsilon \Pr_1[M, V] + \Delta(M, V)$. Also, from theorem 6, we know the random variables (M, V) are (ϵ, δ)-DP for $\epsilon = O\left(\frac{(1-e^{\epsilon_0})e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n-t}}\right)$ and some $\delta > 0$. Thus, we have

$$\sum_{(M,V)\in\mathcal{Z}} \Delta(M, V) \leq \delta.$$

Moreover, given any M, V and the corresponding conditional distributions D^0 and D^1, we have

$$\begin{aligned}\Pr_{\mathbf{y}_H \xleftarrow{\$} D^0}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] &\leq \left(e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] + \delta_1\right) \wedge 1 \\ &\leq \left(e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \wedge 1\right) + \delta_1.\end{aligned}$$

Here, $a \wedge b$ means $\min(a, b)$. Finally, we have

$$
\begin{aligned}
&\Pr[\mathbf{Expt}_{\mathcal{A}}^{\Pi,0}(1^\lambda) = 1] \\
=& \sum_{M,V \in \mathcal{Z}} \Pr_0[M,V] \sum_{\mathbf{y}_H \in \mathcal{Y}} \Pr_0[\mathbf{y}_H \mid M,V] \cdot \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \\
=& \sum_{M,V \in \mathcal{Z}} \Pr_0[M,V] \Pr_{\mathbf{y}_H \xleftarrow{\$} D^0}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \\
\leq& \sum_{M,V \in \mathcal{Z}} \Pr_0[M,V] \left((e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \wedge 1) + \delta_1 \right) \\
=& \delta_1 + \sum_{M,V \in \mathcal{Z}} \Pr_0[M,V] \left(e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \wedge 1 \right) \\
\leq& \delta_1 + \sum_{M,V \in \mathcal{Z}} (e^{\epsilon} \Pr_1[M,V] + \Delta(M,V)) \cdot \left(e^{\epsilon_1} \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \wedge 1 \right) \\
& \textit{(For any } a_1, b_1, a_2, b_2 \in \mathbb{R}^{\geq 0}\textit{, } (a_1 + b_1) \cdot (a_2 \wedge b_2) \leq a_1 a_2 + b_1 b_2\textit{)} \\
\leq& \delta_1 + \sum_{M,V \in \mathcal{Z}} \left(e^{\epsilon + \epsilon_1} \Pr_1[M,V] \Pr_{\mathbf{y}_H \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi}(\mathbf{y}_H) = 1] \right) + \Delta(M,V) \\
\leq& e^{\epsilon + \epsilon_1} \Pr[\mathbf{Expt}_{\mathcal{A}}^{\Pi,1}(1^\lambda) = 1] + \delta + \delta_1
\end{aligned}
$$

□

Many shuffle protocols are not directly derived from LDP protocols. One straightforward way to derive the privacy parameters for the composition of general protocols and DO shufflers is using group privacy theorem. We can get a result that for any (ϵ, δ) shuffle-DP protocol, replacing the perfect shuffler with an (ϵ', δ')-DO shuffler will result in an $(\epsilon + n\epsilon', \delta + n\delta')$-DP protocols. However, the extra n-fold loss in the privacy budget may be too large. It is an interesting open question that how to tighten the privacy guarantee for general single message shuffle-DP protocols under the DO-shuffle model without suffering from the n-fold privacy loss.

3.3 Application: Differentially Private Time-Series Data Aggregation

Rastogi et al. [37] and Shi et al. [38] suggest a scenario for differentially private time-series data analytics. In time-series data aggregation, there are T time steps. Also, there are n clients among whom at most t are corrupt. Let $c = n - t$ be the number of honest clients. Each client has a time-series data set $(x_{i,1}, \ldots, x_{i,T})$. For each time step $j \in [T]$, each client uses an ϵ_0-LDP randomizer $\mathcal{R}_t$ to generate a randomized report $\mathcal{R}_t(x_{i,j})$. In the DO-shuffle model, the outcome of the local randomizers is then fed into a DO-shuffle protocol, at the end of which the server learns a permutation of the local randomizer outcomes. Some DO-shuffler constructions, like Bunz et al. [12], which, once setup, the same shuffling permuation can be used repeatedly over time without additional loss in the privacy parameter. In this section, we extend our privacy amplification theorem to this repeated aggregation setting.

For the end application, we want to achieve a user-level DP notion. In user-level DP, two neighboring honest input configurations

$$\mathbf{x}_H^0 = ((x_{1,1},\ldots,x_{1,T}),\ldots,(x_{c,1},\ldots,x_{c,T})) \text{ and } \mathbf{x}_H^1 = ((x'_{1,1},\ldots,x'_{1,T}),\ldots,(x'_{c,1},\ldots,x'_{c,T}))$$

are neighboring if they only differ in one client's time-series data set, say client i (and possibly client i's data differs in all time steps in the two worlds). All other clients' data are the same in the two worlds, i.e., for each client $j \neq i$ and every $t \in [T]$, we have $x_{j,t} = x'_{j,t}$.

We give a privacy amplification theorem using repeated shuffler to perform repeated data aggregation. Say $y_{i,j} = \mathcal{R}_j(x_{i,j})$ is the message from client i in time step j. In this setting, the permutation applied is the same for each round, and the security definition for the DO-shuffler guarantees (ϵ,δ)-indistinguishability no matter how many time steps. Therefore, we can imagine that a client's message bundle across all time steps act together in the proof of our earlier single-message privacy amplification theorem (See Theorem 7). Specifically, let the i-th message bundle be $Y_i = (y_{i,1},\ldots,y_{i,T})$, i.e., all the messages sent from client i throughout time step 1 to T. By the standard sequential composition theorem [41], the composed randomizer, i.e., $(\mathcal{R}_1(\cdot),\ldots,\mathcal{R}_T(\cdot))$ is an $T\epsilon_0$-LDP randomizer. Also, the construction of repeated DO-shuffler [12] ensures that even if two clients u and v swap their messages throughout the T time steps, the protocol is (ϵ_1,δ_1)-private over all T steps. Now, treating each client's message bundle like a single message in the proof in Theorem 7, and plugging in the parameter $T\epsilon_0$ for privacy budget of the local randomizer, we have the following corollary:

Corollary 9 (Privacy amplification in the DO-model for time-series data)**.** *Composing the repeated (ϵ_1,δ_1)-DO shuffler with LDP mechanisms $\mathcal{R}_1(\cdot),\ldots,\mathcal{R}_T(\cdot)$ over T time steps, the entire protocol satisfies $(\epsilon+\epsilon_1,\delta+\delta')$-user-level DP where*

$$\epsilon = O\left(\frac{(1-e^{-T\epsilon_0})e^{T\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n-t}}\right).$$

Comparison with running a fresh DO-shuffle per time step. It is interesting to compare the above result with the naïve approach of running a fresh DO-shuffler per time step. Using the privacy amplification result for DO-shuffle model, we have the protocol's privacy loss in one time step is $(\epsilon+\epsilon_1,\delta+\delta_1)$ for some $\delta > 0$ and $\epsilon = O\left(\frac{(1-e^{-\epsilon_0})e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n-t}}\right)$. Using the DP sequential composition theorem, if the shuffler samples a fresh random permutation in each time step, the whole protocol is $(T\epsilon+T\epsilon_1, T\delta+T\delta_1)$-DP. Here the guarantees are incomparable to using the repeated DO-shuffler which applies the same permutation in all time steps, but on the other hand, the privacy parameters (ϵ_1,δ_1) coming from the DO-shuffle does not degrade over time. Observe that when $\epsilon_0 T < 1$, the guarantees using the repeated DO-shuffler is strictly better.

4 Multi-message DO-shuffle Protocols

Gordon et al. [33] only discuss single-message protocols under the DO-shuffle model. Here, we discuss the possibility of designing multi-message protocols relying on DO shufflers to achieve strong privacy guarantee while having a better utility than single-message protocols. There are various analytic problems that separates the single-message shuffle models and the multi-message shuffle model, among which the most well-studied two are the real summation problem and the histogram problem.

In the real summation problem, each client i holds a real value $x \in [0,1]$. The server wants to compute $\sum_{i\in[n]} x_i$. Balle et al. [7] proved the mean square error for any (ϵ,δ)-single-message shuffle-DP protocol is $\tilde{\Omega}_{\epsilon,\delta}(n^{1/3})$. On the contrary, there are multiple different multi-message protocols that achieve $\tilde{O}_{\epsilon,\delta}(1)$ error, see Balle et al [8] and Gahzi et al. [30].

In the histogram problem, each client i holds an item x_i in domain $[d]$ and the server wants to compute the histogram $H_x = \sum_{i\in[n]} \mathbf{I}[x_i = x]$ for $x \in [d]$. Gahzi et al. [28] proved the L_∞ error for any (ϵ,δ)-single-message shuffle-DP protocol is $\tilde{\Omega}_{\epsilon,\delta}\left(\min\left(n^{1/4}, \sqrt{d}\right)\right)$. On the contrary, there are multiple different multi-message protocols that achieve $O_{\epsilon,\delta}(\log d)$ error, see Cheu et al. [18], Balle et al. [7] and Gahzi et al. [28].

In this section, we discuss about how to compose DO shufflers with the recursive protocol from Balle et al. [8] for the real summation protocol that achieves the mean square error of $O_\delta\left(\frac{(\log\log n)^2}{\epsilon^2}\right)$. We also discuss how to compose DO shuffler with a special class of multi-message protocols, which includes the histogram protocol from Gahzi [28] that achieves $O_{\epsilon,\delta}(\log d)$ error.

We first provide the definition for multi-message DO shuffler, which is a natural extension of the single-message DO shuffler:

Definition 10 (Multi-message Differentially Oblivious Shuffler)**.** *Given n clients and each one of them holds m message $y_{i,1}, y_{i,2}, \ldots, y_{i,m}$. A protocol Φ is an (ϵ,δ)-DO shuffler against up to t corrupted parties if*

1. *$\Phi(\{y_{1,1} \ldots y_{n,m}\})$ outputs a random permutation of $y_{1,1}, \ldots, y_{n,m}$ when all parties behave honestly;*

2. *Φ is an (ϵ,δ)-DO protocol against up to t corrupted parties given the neighboring relation such that if $\mathbf{x}_H^0 \sim \mathbf{x}_H^1$, $\mathbf{x}_H^0$ and $\mathbf{x}_H^1$ only differ in two messages (say client i's j-th message and client k's l-th message) and those inputs are "transposed", i.e., $\mathbf{x}_{H,i,j}^0 = \mathbf{x}_{H,k,l}^1, \mathbf{x}_{H,k,l}^0 = \mathbf{x}_{H,i,j}^1$.*

4.1 Multi-message real summation protocol with DO shufflers

If one multi-message DP protocol just simply runs multiple instances of single-message DP protocols in parallel and the single-message protocols can be composed with DO shufflers efficiently, we can simply split the privacy budgets and use the group privacy theorem to provide the privacy guarantee for the whole protocol under the DO-shuffle model.

For example, Balle et al. [8] propose a private summation protocol that runs $O(\log\log n)$ single-message protocols in parallel. The single-message protocol takes an real value $x_i \in$

$[0,1]$ and discretizes it to some precision $1/k$, where k is some positive integer. That is, let $\bar{x}_i = \lfloor x_i k \rfloor$ and $\bar{x}_i$ is an integer in $[0,k]$. Then the server can build a histogram h over $[0,k]$ for those integers $\bar{x}_1, \ldots, \bar{x}_n$ such that $h_j = \sum_{i \in n} \mathbf{I}[\bar{x}_i = j]$ and then estimate the sum as $\frac{1}{k}\sum_{j \in [k]} j \cdot h_j$. To ensure privacy, the clients can use the ϵ_0-LDP k-ary randomized response mechanism $\texttt{kRR}(\cdot)$:

$$\texttt{kRR}(x) = \begin{cases} x & \text{w.p. } \frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+k-1}, \\ y \xleftarrow{\$} \mathcal{U}_{[k]} & \text{w.p. } \frac{k}{e^{\epsilon_0}+k-1}. \end{cases}$$

Here, $\mathcal{U}_{[k]}$ denotes the uniform distribution over $\{0, 1, \ldots, k-1\}$. Then, using the privacy amplification theorem, the local privacy budget ϵ_0 can be amplified, such that $e^{\epsilon_0} = O\left(\frac{\epsilon^2 n}{\log(1/\delta)}\right)$. By properly debiasing and ignoring the discretization error, the mean square error can be proved to be $O_\delta(k^3/\epsilon^2)$. However, the discretization will leave some non-negative error, i.e., $x_i - \frac{1}{k}\lfloor x_i k \rfloor$. The error is smaller than $1/k$. When k is large, the error introduced by the k-ary randomized mechanism is large. When k is small, the error of discretization is non-negligible. Thus, the single-message protocol is limited by this error tradeoff. Luckily, in the multi-message model, one can treat the discretization error as a new input to the the same single-message summation protocol, by running the summation protocol in range $[0, 1/k]$. Balle et al. proved that, by running the protocol recursively with precision $k_j = 2^{3^j}$ for $j = 1, \ldots, \lfloor \log_3(\log_2(n)) \rfloor$ and dividing the privacy budget evenly to the sub-protocols, the mean square error of the final estimation is $O_\delta\left(\frac{(\log\log n)^2}{\epsilon^2}\right)$. Notice that one can simply run all the sub-protocols in parallel and submit all the messages at once.

Since each of the sub-protocols is an LDP protocol composing with the shuffler, it can also run in the DO-shuffle model. Thus, we can easily prove the privacy guarantee for the whole protocol.

Theorem 11. *For $\epsilon \leq 1$, given an (ϵ_1, δ_1)-DO shuffler against up to $t < n/2$ corruption clients, there is an m-message $(\epsilon + m\epsilon_1, \delta + m\delta_1)$-DP summation protocol against t corrupted parties under the DO-shuffle model that achieves $\tilde{O}_\delta\left(\frac{(\log\log n)^2}{\epsilon^2}\right)$ mean square error with $m = \lfloor \log_3(\log_2(n)) \rfloor$.*

Proof. We take the recursive protocol from [8] and use a hybrid argument to prove the theorem. W.L.O.G, let's say the neighboring input configurations are $\mathbf{x}_H^0 = (x_1, x_2, \ldots, x_{n-t})$ and $\mathbf{x}_H^1 = (x_1', x_2, \ldots, x_{n-t})$.

For one client's input x, the client locally runs $m = \lfloor \log_3(\log_2(n)) \rfloor$ randomization functions $\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(m)}$ that process the input x to m messages $\mathcal{R}^{(1)}(x), \ldots, \mathcal{R}^{(m)}(x)$. Also, each randomized function $\mathcal{R}^{(i)}$ has disjoint output domain $\mathcal{Y}_i$ (because they need a unique identifier to distinguish messages among parallel protocols). Given one input configuration $(x_1, \ldots, x_n)$, the shuffled results can be written as a collection of m multi-sets,

$$(\{\mathcal{R}^{(1)}(x_1), \ldots, \mathcal{R}^{(1)}(x_n)\}, \ldots, \{\mathcal{R}^{(m)}(x_1), \ldots, \mathcal{R}^{(m)}(x_n)\}).$$

In the proof of the privacy theorem from [8], there exist sequences of $\epsilon_1, \ldots, \epsilon_m$ and $\delta_1, \ldots, \delta_m$ such that for $i \in [m]$, the multi-set $\{\mathcal{R}^{(i)}(x_1), \ldots, \mathcal{R}^{(i)}(x_n)\}$ is (ϵ_i, δ_i)-DP, such that $\sum_{i \in [m]} \epsilon_i = \epsilon$, $\sum_{i \in [m]} \delta_i = \delta$.

To prove the privacy under DO-shuffle model, we can build $m+1$ hybrids, labeled from 0 to m: in the i-th hybrids, client 1 will submit m messages:

$$\mathcal{R}^{(1)}(x_1), \ldots, \mathcal{R}^{(i)}(x_1), \mathcal{R}^{(i+1)}(x_1'), \ldots, \mathcal{R}^{(m)}(x_1')$$

Also, $\mathbf{Hyb}_i^{\mathcal{A}}$ denotes the adversary $\mathcal{A}$'s output in the i-th hybrids. Here, we see that the only difference between the shuffled result in hybrid i and hybrid $i+1$ is the multi-set $\{\mathcal{R}^{(i+1)}(x_1), \ldots, \mathcal{R}^{(i+1)}(x_n)\}$ and $\{\mathcal{R}^{(i+1)}(x_1'), \ldots, \mathcal{R}^{(i+1)}(x_n)\}$. We know that the single-message randomizer $\mathcal{R}^{(i+1)}$ can compose with an (ϵ', δ')-DO shuffler (because it is a ϵ_0-LDP randomized-response randomizer). Thus, the distributions of hybrid i and $i+1$ are $(\epsilon_i + \epsilon', \delta_i + \delta')$-DP, i.e., for any PPT $\mathcal{A}$,

$$\Pr[\mathbf{Hyb}_i^{\mathcal{A}} = 1] \leq e^{\epsilon_i + \epsilon'} \Pr[\mathbf{Hyb}_{i+1}^{\mathcal{A}} = 1] + \delta_i + \delta.$$

By group privacy theorem, the hybrid 0 and hybrid m is $(\epsilon + m\epsilon', \delta + m\delta')$-DP. The condition $t < n/2$ ensures that the utility theorem from [8] only degrades by a constant factor. □

Finally, recall that both Gordon et al. [33] and Bunz et al. [12], construct asymptotically efficient DO-shuffle protocols that achieve $\epsilon = o(1/\log\log n)$ privacy loss with a negligibly small δ. Therefore, if we instantiate the protocol in Theorem 2 using the DO-shuffle scheme of Gordon et al. [33] or Bunz et al. [12], the total privacy loss introduced by DO-shuffle is only by $o(1)$.

4.2 Multi-message DO-shuffle histogram protocol from multi-message privacy blanket protocol

Gahzi et al. [28] propose a mulit-message shuffle-DP protocol for the histogram problem that achieves $\tilde{O}_{\epsilon,\delta}(\log d)$ L_∞ error. We briefly describe the protocol here.

Let's assume the domain size $d = n$ first. Each client i will sample $\rho = \frac{36\ln(1/\delta)}{\epsilon}$ extra "fake elements" uniformly from $\{1, \ldots, d\}$, combine the true element x_i and send the $\rho + 1$ elements to the shuffler. For each particular element index $j \in [d]$, let's say the true count of element j is X_j and server sees Y_j copies in the shuffler's output. In expectation, we have $\mathbf{E}[Y_j] = X_j + \rho n/d = X_j + \frac{36\ln(1/\delta)}{\epsilon}$. Thus, the server can simply estimate the counts by proper debiasing. By the Chernoff bound, it is easily to bound the L_∞ error of the histogram by $O(\sqrt{\log d}\log(1/\delta)/\epsilon)$. The whole histogram has sensitivity of 2 because only one client will change its element in the neighboring input configuration. The fake elements can be seen as a layer of Binomial noise, such that $Y_j = X_j + \mathrm{Bin}(\rho n, 1/n)$. The Binomial noise mechanism is well-studied by Agarwal et al. [3] and $\rho = \frac{36\ln(1/\delta)}{\epsilon}$ is enough to ensure (ϵ, δ) shuffle-DP. To handle the case where the domain size is much larger than n, Gahzi et al. [28] use Hadmard encoding to "hash" the element to domain $\{1, \ldots, n\}$ and the aforementioned protocol still works. When there are $t < n/2$ corrupted clients, the protocol simply increases the fake element number ρ by a factor of $\frac{n}{n-t}$, which is smaller than 2. The privacy is preserved and the utility guarantee is only affected by a constant factor. Now we provide the similar theorem under the DO-shuffle model.

Theorem 12 (Multi-message DO-shuffle private histogram protocol)**.** *For $\epsilon \leq 1$, given an (ϵ_1, δ_1)-DO shuffler, there is an $O\left(\frac{\log(1/\epsilon\delta)}{\epsilon^2}\right)$-message $(\epsilon + \epsilon_1, \delta + \delta_1)$-DP histogram protocol against $t < n/2$ corrupted parties under the DO-shuffle model that achieves $\tilde{O}_{\epsilon,\delta}(\log d)$ L_∞ error.*

The protocol from Gahzi et al. [28] has the special property that each client only sends one "data-dependent" message and all the data-independent messages, i.e., the "fake elements" have the same distribution. Also, given two neighboring input configurations $\mathbf{x}_H^0 = (x_1, x_2, \ldots, x_n)$ and $\mathbf{x}_H^1 = (x_1', x_2, \ldots, x_n)$, even when the adversary observes the data-dependent message from client 2 to client n, the protocol is still DP because the privacy guarantee is achieved by taking the "fake elements" as the randomness source. We generalize this property as following:

Definition 13 (Multi-message privacy blanket protocol)**.** *A multi-message privacy blanket protocol in the shuffle model requires each client i, who holds input x_i, only sends one message $y_i = \mathcal{R}(x_i)$ that is dependent on the input x_i and sends at most m i.i.d. "privacy blanket" messages $z_{i,1}, \ldots, z_{i,m}$ sampled from a pre-defined distribution ω. Also, for any two neighboring input x_i and x_i', the distributions of the multisets*

$$\{\mathcal{R}(x_i), z_{1,1}, \ldots, z_{1,m}, \ldots, z_{n,1}, \ldots, z_{n,m}\}$$

and

$$\{\mathcal{R}(x_i'), z_{1,1}, \ldots, z_{1,m}, \ldots, z_{n,1}, \ldots, z_{n,m}\}$$

are (ϵ, δ)-close.

Theorem 14 (Multi-message privacy blanket protocol under DO shuffle model)**.** *For any (ϵ, δ)-DP multi-message privacy blanket protocol that tolerates up to t corrupted clients is $(\epsilon' + \epsilon, \delta' + \delta)$-DP under the DO-shuffle model given an (ϵ', δ')-DO shuffler that tolerates up to t corrupted clients.*

Proof. The spirit of the proof is nearly the same as the proof for theorem 6. Let $c = n - t$ be the honest client number. W.L.O.G, let the neighboring input configurations be $\mathbf{x}_H^0 = (x_1, x_2, \ldots, x_c)$ and $\mathbf{x}_H^1 = (x_1', x_2, \ldots, x_c)$. Excluding the adversary's input, the shuffling results M can be seen as the multi-set of

$$\{y_1, \ldots, y_c, z_{1,1}, \ldots, z_{1,m}, \ldots, z_{c,1}, \ldots, z_{c,m}\},$$

where $y_1, \ldots, y_c$ are the data-dependent messages and all $z_{i,j}$ are the data-independent messages. Let the adversary's view include the auxiliary information about the data dependent messages from client 2 to client c, say $V = (y_2, \ldots, y_n)$. Conditioned on M, V, the remaining $mc + 1$ messages are all independent, where y_1' is $\mathcal{R}(x_1)$ or $\mathcal{R}(x_1')$ and all $z_{i,j}$ have the same distribution ω. This structure is exactly the same as what we have seen in the proof of theorem 6. Let $\mathcal{Y}$ be all possible permutations of all the messages from the sets that removes all messages in V from M. Let $\mathbf{y}_H$ be a message sequence of those $mc + 1$ messages. Let D^0 be the conditional distribution on $\mathcal{Y}$ when client 1's input is x_1 conditioned on M, V. Similarly, let D^1 be the conditional distribution when client 1's input is x_1'. Using the same mathcing trick in the proof of theorem 6, we can prove that for any PPT advesary $\mathcal{A}$ and an (ϵ', δ') DO-shuffler Φ, we have

$$\Pr_{\mathbf{y}_H^0 \xleftarrow{\$} D^0}[\mathbf{Expt}_{\mathcal{A}}^{\Phi,0}(1^\lambda, \mathbf{x}_H^0) = 1] \leq e^{\epsilon'} \Pr_{\mathbf{y}_H^1 \xleftarrow{\$} D^1}[\mathbf{Expt}_{\mathcal{A}}^{\Phi,1}(1^\lambda, \mathbf{x}_H^1) = 1] + \delta'$$

Then, using the similar composition argument like the proof for theorem 6, we get the result. □

Combining theorem 14 and the utility theorem for the histogram protocol (theorem 4.1) from Gahzi et al. [28], the proof for theorem 12 is straightforward.

5 Related Work

We now review additional related work.

Privacy amplification in the shuffle model. Bittau et al. [11] first propose the shuffle-DP idea as a proposal for Google's privacy-preserving data analytic system. Later, Cheu et al. [18] provide the first formal theoretical privacy amplification result for a simple and important mechanism, i.e., the binary randomized response mechanism. They proved that given n copies of ϵ_0-LDP binary randomized response randomizer, the shuffled result is (ϵ, δ)-DP for $\epsilon = O\left(\frac{e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$. They also proved that the asymptotic expression is tight. Erlingsson et al. [24] provide the first shuffle-model privacy amplification theorem for any general ϵ_0-LDP randomizers that the amplified privacy parameters are $\epsilon = O\left(\frac{e^{3\epsilon_0}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$ for $\delta > 0$. Later, Balle et al. [7] provide a better privacy amplification theorem showing that $\epsilon = O\left(\frac{e^{\epsilon_0}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$. Also, in another work, Balle et al. [9] provide the privacy amplification results for non-pure LDP randomizers: for (ϵ_0, δ_0)-LDP randomizers, the shuffled outcome is (ϵ, δ)-DP for $\epsilon = O\left(\frac{e^{20\epsilon_0}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$ and $\delta > 0$. However, those works generally include quite involved techniques, making the amplification result like a black box. Recently, Feldaman et al. [26] gave a simple and elegant proof of the optimal privacy amplification result in the shuffle model. They show that the shuffled outcome satisfies (ϵ, δ)-DP where $\epsilon = O\left(\frac{e^{\epsilon_0/2}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$, thus closing the gap between the parameters for general LDP mechanisms and the binary randomized response mechanism.

From the shuffle model to the DO-shuffle model. A trusted shuffler can be realized either through trusted hardware [11, 17] or through cryptographic techniques such as mix-nets [1,10,15], dining cryptographer's net (DC-net) [2,16,21], and various other cryptographic techniques [4, 20, 27, 36, 39, 44].

Many earlier cryptographic protocols for anonymous shuffling require at least quadratic communication and computation [1, 15, 16, 21]. Recently, some works suggested the relaxed notion of differential privacy in anonymous communication protocols [5,12,34,35,40,42], also referred to as differential obliviousness [14, 34]. Notably, a few recent works [12, 34] pointed out that by relaxing the privacy notion to differential obliviousness, we can asymptotically improve the protocol's communication complexity or computational complexity from quadratic to (quasi-)linear. In particular, Ando et al. [5] showed how to construct a differentially oblivious shuffle using onion routing in both the semi-honest and malicious model, achieving quasilinear total communication and computation. Gordon et al. [34] showed how to realize a concretely efficient differentially oblivious shuffle in the semi-honest setting with $O(n \log n)$ total number of messages. Bunz et al. [12] showed a non-interactive differentially oblivious shuffle in the malicious model that achieves $O(n)$ total message complexity per shuffle and sub-quadratic total computation. Bunz's construction employs as a building block the non-interactive fully anonymous shuffler by Shi and Wu [39]. By relaxing the security definition to DO, they improve the computation complexity of Shi and Wu [39] from quadratic to sub-quadratic.

The elegant work of Gordon et al. [34] was also the first to explore the power of the DO-shuffle model in distributed DP mechanisms. As mentioned, they showed an optimal DO-shuffle-model privacy amplification for the randomized response mechanism. They also generalize the privacy amplification result to any general LDP mechanism — but this latter result is non-optimal since they rely on the non-optimal amplification theorem from [7].

6 Conclusion

We prove an optimal privacy amplification theorem by composing any locally differentially private (LDP) mechanism with a DO-shuffler, achieving parameters that tightly match the shuffle model. Moreover, we explore the multi-message protocols under DO-shuffle model and construct mechanisms for the real summation and histograph problems. Our error bounds approximate the best known results in the multi-message shuffle-model up to sub-logarithmic factors.

Throughout this paper, we take advantage of the special structures in the DP-protocols and prove the privacy guarantee for the composition protocols under DO-shuffle model. It is still an open question that how to compose a general protocols with DO shufflers without suffering from an n-fold privacy loss. We leave this as a part of future work.

Acknowledgment

This work is in part supported by an NSF award under the grant number CIF-1705007.

References

[1] Masayuki Abe. Mix-networks on permutation networks. In *ASIACRYPT*, 1999.

[2] Ittai Abraham, Benny Pinkas, and Avishay Yanai. Blinder: Mpc based scalable and robust anonymous committed broadcast. In *ACM CCS*, 2020.

[3] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

[4] Nikolaos Alexopoulos, Aggelos Kiayias, Riivo Talviste, and Thomas Zacharias. Mcmix: Anonymous messaging via secure multiparty computation. In *Usenix Security*, 2017.

[5] Michael Backes, Ian Goldberg, Aniket Kate, and Esfandiar Mohammadi. Provably secure and practical onion routing. In *2012 IEEE 25th Computer Security Foundations Symposium*, pages 369–385, 2012.

[6] Victor Balcer and Albert Cheu. Separating local & shuffled differential privacy via histograms. URL: http://arxiv.org/abs/1911.06879, arXiv:1911.06879.

[7] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. The privacy blanket of the shuffle model. URL: http://arxiv.org/abs/1903.02837, arXiv:1903.02837.

[8] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. Private summation in the multi-message shuffle model. pages 657–676. URL: http://arxiv.org/abs/2002.00817, arXiv:2002.00817, doi:10.1145/3372297.3417242.

[9] Borja Balle, Peter Kairouz, Brendan McMahan, Om Thakkar, and Abhradeep Guha Thakurta. Privacy amplification via random check-ins. *Advances in Neural Information Processing Systems*, 33:4623–4634, 2020.

[10] Stephanie Bayer and Jens Groth. Efficient zero-knowledge argument for correctness of a shuffle. In *Eurocrypt*, volume 7237, pages 263–280, 2012.

[11] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459, 2017.

[12] Benedikt Bünz, Yuncong Hu, Shin'ichiro Matsuo, and Elaine Shi. Non-interactive differentially anonymous router. Cryptology ePrint Archive, Report 2021/1242, 2021. https://ia.cr/2021/1242.

[13] Jan Camenisch and Anna Lysyanskaya. A formal treatment of onion routing. In *CRYPTO*, 2005.

[14] T-H. Hubert Chan, Kai-Min Chung, Bruce M. Maggs, and Elaine Shi. Foundations of differentially oblivious algorithms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page 2448–2467, USA, 2019. Society for Industrial and Applied Mathematics.

[15] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, February 1981.

[16] David L. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, March 1988.

[17] Ju Chen, Yuzhe (Richard) Tang, and Hao Zhou. Strongly secure and efficient data shuffle on hardware enclaves. In *Proceedings of the 2nd Workshop on System Software for Trusted Execution*, SysTEX'17. Association for Computing Machinery, 2017.

[18] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer, 2019.

[19] Albert Cheu and Maxim Zhilyaev. Differentially private histograms in the shuffle model from fake users. *CoRR*, abs/2104.02739, 2021. URL: https://arxiv.org/abs/2104.02739, arXiv:2104.02739.

[20] Henry Corrigan-Gibbs, Dan Boneh, and David Mazières. Riposte: An anonymous messaging system handling millions of users. In *S & P*, 2015.

[21] Henry Corrigan-Gibbs and Bryan Ford. Dissent: Accountable anonymous group messaging. In *CCS*, page 340–350, 2010.

[22] Jean Paul Degabriele and Martijn Stam. Untagging tor: A formal treatment of onion encryption. In *EUROCRYPT*, 2018.

[23] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. 9(3):211–407. URL: http://www.nowpublishers.com/articles/foundations-and-trends-in-theoretical-computer-science/TCS-042, doi:10.1561/0400000042.

[24] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

[25] Saba Eskandarian and Dan Boneh. Clarion: Anonymous communication from multiparty shuffling protocols. Cryptology ePrint Archive, Report 2021/1514, 2021. https://ia.cr/2021/1514.

[26] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. URL: http://arxiv.org/abs/2012.12803, arXiv:2012.12803.

[27] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *J. Comput. Syst. Sci.*, 60(3), 2000.

[28] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. On the power of multiple anonymous messages. URL: http://arxiv.org/abs/1908.11358, arXiv:1908.11358.

[29] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3505–3514. PMLR, 13–18 Jul 2020. URL: https://proceedings.mlr.press/v119/ghazi20a.html.

[30] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Amer Sinha. Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message. URL: http://arxiv.org/abs/2109.13158, arXiv:2109.13158.

[31] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2521–2529. PMLR, 2021.

[32] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42:39–41, 1999.

[33] S. Dov Gordon, Jonathan Katz, Mingyu Liang, and Jiayu Xu. Spreading the privacy blanket: Differentially oblivious shuffling for differential privacy. Cryptology ePrint Archive, Report 2021/1257, 2021. https://ia.cr/2021/1257.

[34] S Dov Gordon, Jonathan Katz, Jiayu Xu, and Mingyu Liang. Spreading the privacy blanket:. page 35.

[35] David Lazar, Yossi Gilad, and Nickolai Zeldovich. Karaoke: Distributed private messaging immune to passive traffic analysis. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 711–725, Carlsbad, CA, October 2018. USENIX Association. URL: https://www.usenix.org/conference/osdi18/presentation/lazar.

[36] Rafail Ostrovsky and Victor Shoup. Private information storage (extended abstract). In *STOC*, pages 294–303, 1997.

[37] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, page 735–746, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1807167.1807247.

[38] Elaine Shi, T-H. Hubert Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Network and Distributed System Security Symposium (NDSS)*, 2011.

[39] Elaine Shi and Ke Wu. Non-interactive anonymous router. Cryptology ePrint Archive, Report 2021/435, 2021. https://ia.cr/2021/435.

[40] Nirvan Tyagi, Yossi Gilad, Derek Leung, Matei Zaharia, and Nickolai Zeldovich. Stadium: A distributed metadata-private messaging system. In *SOSP*, 2017.

[41] Salil Vadhan. The complexity of differential privacy. 2017.

[42] Jelle van den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *SOSP*, 2015.

[43] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[44] Li Zhuang, Feng Zhou, Ben Y. Zhao, and Antony Rowstron. Cashmere: Resilient anonymous routing. In *NSDI*, 2005.

A Preliminaries on Differential Privacy

Definition 15 ((ϵ, δ)-close)**.** *We say the distributions of two random variables, X and X' are (ϵ, δ)-close if they have the same domain D and for every subset $S \subseteq D$,*

$$\Pr[X \in S] \leq e^{\epsilon} \Pr[X' \in S] + \delta.$$

Definition 16 ((ϵ, δ)-Differential Privacy)**.** *A function f is (ϵ, δ)-DP w.r.t. some neighboring relation $\sim$ on its input domain if for every pair $v, v' \in$ Domain(f), s.t. $v \sim v'$, the distributions of $f(v)$ and $f(v')$ are (ϵ, δ)-close.*

If a function f is $(\epsilon, 0)$-DP, we also say that f is ϵ-DP for short (w.r.t. the neighboring relation $\sim$).

The shuffle model. There are n clients labeled from 1 to n and each one of them holds an input x_i. A server wants to do some data analytic over inputs $x_1, \ldots, x_n$. To protect client privacy, each client i firstly encodes the input using a randomizer $\mathcal{R}$, then sends the encoding $\mathcal{R}(x_i)$ to a shuffler $\mathcal{S}$. The shuffler $\mathcal{S}$ takes all clients' messages and outputs a random permutation of them. The server receives the output of the shuffler and performs data analytics on the shuffled messages.

Definition 17 (Differential Privacy in the Shuffle Model)**.** *For n clients, the randomizer $\mathcal{R}$ is said to be (ϵ, δ)-Shuffle-DP if the function composition*

$$\mathcal{S} \circ (\mathcal{R} \times \cdots \times \mathcal{R})(x_1 \ldots x_n) := \mathcal{S}(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$$

is (ϵ, δ)-DP given any input neighboring configurations $\mathbf{x}^0 = (x_1, \ldots, x_n)$ and $\mathbf{x}^1 = (x'_1, \ldots, x'_n)$ that only differ in one client's input.