Spreading the Privacy Blanket:
Differentially Oblivious Shuffling for Differential Privacy

S. Dov Gordon* Jonathan Katz† Mingyu Liang*
Jiayu Xu‡

Abstract

In the *shuffle model* for differential privacy, users locally randomize their data and then submit the results to a trusted "shuffler" who mixes the responses before sending them to a server for analysis. This is a promising model for real-world applications of differential privacy, as a series of recent results have shown that, in some cases, the shuffle model offers a strictly better privacy/utility tradeoff than what is possible in a purely local model. The recent "privacy blanket" notion provides a simple method for analyzing differentially private protocols in the shuffle model.

A downside of the shuffle model is its reliance on a trusted shuffling mechanism, and it is natural to try to replace this with a distributed shuffling protocol run by the users themselves. Unfortunately, with only one exception, existing fully secure shuffling protocols require $\Omega(n^2)$ communication.

In this work, we put forth a notion of *differential obliviousness* for shuffling protocols, and prove that this notion provides the necessary guarantees for the privacy blanket, without requiring a trusted shuffler. We also show a differentially oblivious shuffling protocol based on onion routing that can tolerate any constant fraction of corrupted users and requires only $O(n \log n)$ communication.

1 Introduction

Differential privacy [15] has become a leading approach for privacy-preserving data analysis. Traditional mechanisms for differential privacy operate in the *curator model*, where a trusted server holds all the sensitive data and releases noisy statistics about that data. To reduce the necessary trust assumptions, researchers subsequently proposed the *local model* of differential privacy. Here, each user applies a local randomizer $\mathcal{R}$ to its sensitive data x_i to obtain a noisy result y_i, and then forwards y_i to a server who analyzes the noisy data it obtains.

*George Mason University. Email: {gordon, mliang5}@gmu.edu
†University of Maryland. Email: jkatz2@gmail.com
‡Algorand. Email: jiayux@uci.edu

Local mechanisms for differential privacy are currently used by companies such as Google and Apple to collect statistics about their users.

A drawback of local mechanisms is that, in some cases, they provably require more noise (and hence offer reduced utility) for a fixed level of privacy. For example, computing a differentially private mean of n users' inputs can be done with only $O(1)$ noise in the centralized curator model [15] but requires $O(\sqrt{n})$ noise in the local model [4, 10].

A recent line of work has explored an intermediate model that provides a tradeoff between these extremes. In the *shuffle model* [7, 13, 30, 3], users locally add noise to their data as in the local model, but also have access to a trusted mechanism $\mathcal{S}$ (a "shuffler") for anonymizing their data before it is forwarded to the server. That is, whereas in the local model the server obtains the ordered vector of noisy inputs $(y_1, \ldots, y_n)$, in the shuffle model the server is given only the multiset $\{y_i\} := \mathcal{S}(y_1, \ldots, y_n)$ which hides information about which element was contributed by any particular user. ($\{y_i\}$ can be encrypted with the server's public key before being sent to the shuffler so that the shuffler does not learn the value submitted by any user.) In some cases, the shuffle model is known to offer a strictly better privacy/utility tradeoff than what is possible in the local model. The recent "privacy blanket" notion [3] is an elegant—and, for some problems, optimal—differentially private protocol that works in the shuffle model and can be applied to a variety of problems.

Although the shuffle model relies on a weaker trust assumption than the curator model, it may still be undesirable to rely on a trusted entity to perform the shuffling, and, in particular, not to collude with the curator. It is thus natural to consider replacing the shuffler by a distributed protocol, executed by the users themselves. Clearly, we can use generic secure computation to replace $\mathcal{S}$ while preserving the differential privacy guarantees of any mechanism designed for the shuffle model. Unfortunately, most existing fully secure shuffling protocols suffer from $\Omega(n^2)$ communication (see discussion of related work below).

Our contributions. We put forth a new notion of security for shuffling protocols, which we call *differential obliviousness*, that is motivated by, but formally distinct from, differential privacy. Roughly, for any honest pair of users and any pair of values y, y' in the output multiset, a differentially oblivious shuffling protocol hides (in the same sense as for differential privacy) whether the first user contributed y and the second user contributed y', or vice versa. We then prove that any differentially oblivious shuffling protocol, when combined with an ϵ-local differentially private mechanism, provides differential privacy without the trusted shuffler.

With this result in place, we then turn to constructing a differentially oblivious shuffling protocol with low communication. We prove that *onion routing* – in which each user chooses a random path of length r among the users, ending at the server, with nested encryption used to hide the route – is differentially oblivious, and in fact achieves privacy that improves exponentially in r. Setting $r = O(1)$ to match parameters typically used for differential privacy, we obtain a shuffling protocol with $O(n \log n)$ communication. This yields a construction that is concretely efficient, and asymptotically better than almost all prior (fully oblivious) shuffling protocols.

1.1 Related Work

Oblivious shuffling. There is a long line of work studying different types of protocols for oblivious shuffling. We survey some of what is known, restricting attention to protocols secure against $t = \Theta(n)$ (semi-honest) corruptions.

Fully secure oblivious shuffling can be done via secure computation of a permutation network, using a random permutation [20, 26]. This requires $\Omega(n^2)$ communication just for the initial sharing of the inputs. Then, the parties can obliviously sample a permutation [26], or, they execute the circuit $t + 1$ times with $t+1$ users each choosing a random permutation [20, 23]; the latter approach is more efficient for small n, but results in an $\Omega(n)$-round protocol.

A recent line of work [8, 14, 25] constructs secure-computation protocols that avoid the $\Omega(n^2)$ communication complexity of input sharing by using "quorums" of size $O(\log n)$ to carry out the computation. Much of this work is aimed at asymptotic performance only, and the concrete efficiency is unclear. Recent work by Movahedi et al. [25] is an exception; they look specifically at applying these ideas to shuffling. As in our work, their shuffling protocol is not fully secure, though the relaxation they consider is quite different from ours: they prove that full security holds with probability $O(1 - 1/n^3)$, and make no claims about the remaining probability. The total communication complexity of their protocol is $O(n \text{ polylog } n)$ and to the best of our knowledge theirs is one of only two prior shuffling protocols with sub-quadratic communication complexity. (We discuss the other at the end of this Section.) Our own protocol out-performs theirs, both asymptotically and concretely. We provide a concrete comparison between our shuffling protocol and theirs in Section 4.4.

Recently, Bell et al. [5] proposed a very different approach for shuffling via secure aggregation of Bloom filters. Their construction requires $\Omega(n^2)$ communication, but appears to have better concrete efficiency as compared to prior work. We provide a concrete comparison between our results and theirs in Section 4.4.

In a mix network [11], users encrypt their values and the resulting ciphertexts are then sequentially mixed by $t + 1$ users (who also re-randomize the encryption). This results in a protocol with $\Omega(n^2)$ communication complexity and $\Omega(n)$ rounds.[1] A dining cryptographers network (DC-net) [12] allows one party to anonymously broadcast a message to the remaining $n-1$ parties; it can be run in parallel n times to allow the n users to shuffle their inputs. Although DC-nets can be implemented in constant rounds (in the semi-honest setting), the communication complexity for running n parallel DC-nets is $\Omega(n^2)$.

Differentially private computation. The idea of relaxing security for

[1] One could elect a random committee of smaller size to perform the mixing, which reduces the total communication cost, but does not improve the *bottleneck complexity*: each committee member still must communicate $O(n)$ values.

distributed protocols in the context of differential privacy has appeared in a number of prior works [4, 19, 23, 9, 18, 24]. Beimel et al. [4] first proposed the idea, and studied how the relaxation impacts efficiency for the problem of secure summation. He et al. [19] and Groce et al. [18] construct differentially private set-intersection protocols that are more efficient than fully secure protocols for the same task. Mazloom and Gordon [23], and Mazloom et al. [24] leverage differential privacy to make graph-parallel computations more efficient. Chan et al. [9] consider a version of differential obliviousness (defined differently from ours) in the client/server model, studying sorting, merging, and range-query data structures under that relaxation.

Anonymous communication. Some techniques for anonymous communication (e.g., mix-nets and DC-nets) are already discussed above. The onion routing protocol [17, 27, 1] that we study in this paper is used as part of the Tor anonymous communication network, though Tor uses paths with only three intermediate nodes. Although Tor has received a lot of attention in the security community, most of that work focuses on active attacks and/or attacks that are specific to Tor. While some theoretical analyses of the anonymity provided by onion routing exist [22, 16, 2, 1], mostly they give results that are incomparable to the differential obliviousness we require. (We discuss one exception in detail, below.) The Stadium, Vuvuzela, and Karaoke systems [28, 29, 21] all provide one-to-one anonymous messaging where anonymity is formalized by requiring differential privacy of the observed network traffic. While this could in principal be used for shuffling (by sending n anonymous messages to the server), the cost would be $O(n^2)$. Bellet et al. [6] study "gossip" protocols that provide a differential privacy guarantee. The setting of their work is quite different from ours: in particular, they assume the adversary does not know the current round, and they focus on one-to-many communication rather than many-to-one communication as we do here.

The most similar work to our own is that of Ando et al. [1]. The authors consider a very similar security relaxation for shuffling, and use it for constructing an anonymous messaging system. Compared to our own analysis of onion routing as a differentially oblivious (DO) shuffle protocol, there are several differences. Ando et al. consider a stronger adversary that can observe all network connections, whereas we relax this assumption and only allow the observation of the channels that neighbor the corrupted parties. Additionally, they consider both a semi-honest adversary and an active adversary, while we only consider semi-honest behavior. However, even when assuming a semi-honest adversary, the ability to observe the network imposes considerable cost. While they claim the same asymptotic complexity as we do in our own analysis, concretely, their construction is quite impractical. For this reason, they make only asymptotic claims, while we provide concrete analysis, demonstrating that our construction is quite practical, performing better than any prior work that we are aware of. That said, We stress that our first theorem, which demonstrates that any differentially oblivious shuffling protocol can be combined with any ϵ-local differentially private mechanism to achieve overall differential privacy (Theorem

3.9), we can use the more secure, less efficient, shuffle constructions of Ando et al. in place of our own.

2 Definitions

Differential privacy. We use the standard notion of (approximate) differential privacy. Two vectors of inputs $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{x}' = (x'_1, \ldots, x'_n)$ are called *neighboring* if they differ at a single index; i.e., if there exists an index i such that $x_i \neq x'_i$ but $x_j = x'_j$ for $j \neq i$. Let f denote a randomized process mapping a vector of inputs $(x_1, \ldots, x_n) \in D^n$, each in some domain D, to an output lying in some range R. We say that f satisfies (ϵ, δ)-*approximate differential privacy* if for all neighboring vectors $\mathbf{x}, \mathbf{x}' \in D^n$ and subsets $R' \subseteq R$ we have

$$\Pr[f(\mathbf{x}) \in R'] \leq e^{\epsilon} \cdot \Pr[f(\mathbf{x}') \in R'] + \delta.$$

If f satisfies $(\epsilon, 0)$-approximate differential privacy then we simply say that f is ϵ-*differentially private*. For compactness, we abbreviate these as (ϵ, δ)-DP/ϵ-DP.

Local differential privacy. Traditionally, differential privacy assumes the model that users' inputs are stored by a trusted curator in a centralized location. In the local differential privacy setting, it is assumed that each user need to submit their input to an untrusted curator. In particular, consider a user U with a local input x in domain D. To ensure privacy, user U locally applies a randomized function $\mathcal{R}$ to his input to obtain an output y in some range R, and then sends y to the curator. We say that $\mathcal{R}$ is (ϵ, δ)-local differentially private, or simply (ϵ, δ)-LDP if for all inputs $x, x' \in D$ and subsets $R' \subseteq R$ we have:

$$\Pr[\mathcal{R}(x) \in R'] \leq e^{\epsilon} \cdot \Pr[\mathcal{R}(x') \in R'] + \delta.$$

Similarly, if $\mathcal{R}$ is $(\epsilon, 0)$-LDP then we simply say that $\mathcal{R}$ is ϵ-LDP.

The shuffle model and the randomized response mechanism. The *shuffle model* [7, 13, 30, 3] considers n users $U_1, \ldots, U_n$, each with a local input x_i, who have access to a trusted "shuffler" $\mathcal{S}$. Each user U_i locally applies a randomized function $\mathcal{R}$ to their input to obtain $y_i = \mathcal{R}(x_i)$, and then sends y_i to $\mathcal{S}$. After receiving a message from every user, $\mathcal{S}$ outputs the multiset of elements (which can also be viewed as a *histogram*) $h = \{y_i\}$. If we overload notation and let $\mathcal{S}$ also denote the process of mapping a list of elements to the multiset containing those elements, then $\mathcal{R}$ defines the randomized process

$$\mathcal{S} \circ (\mathcal{R} \times \cdots \times \mathcal{R})(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \mathcal{S}\left(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n)\right).$$

The randomized response mechanism [3] specifies a particular local randomized mechanism $\mathcal{R}_{\gamma,D}$ for the shuffle model. Let $\gamma \in [0, 1]$ be a parameter, and let D denote the domain in which the users' inputs lie. Then

$$\mathcal{R}_{\gamma,D}(x) = \begin{cases} x & \text{with probability } 1-\gamma \\ y \leftarrow D & \text{with probability } \gamma \end{cases};$$

i.e., a user replaces its input with a uniform value in D with probability γ, and with the remaining probability leaves its input unchanged. Balle et al. [3] show:

Theorem 2.1. *Fix values n, ϵ, δ, and D. If $\gamma \geq \max\left\{\frac{14\cdot|D|\log(2/\delta)}{(n-1)\cdot\epsilon^2}, \frac{27\cdot|D|}{(n-1)\cdot\epsilon}\right\}$, then $\mathcal{S} \circ (\mathcal{R}_{\gamma,D} \times \cdots \times \mathcal{R}_{\gamma,D})$ is (ϵ,δ)-DP.*

Differentially private protocols. More generally, we may consider interactive protocols executed by a server and n users, each of whom initially holds an input x_i. The server has no input, and is the only party to generate an output. We say that a protocol Π *implements* a (randomized) function f if the honest execution of Π when the users hold inputs $x_1, \ldots, x_n$, respectively, results in the server generating output distributed according to $f(x_1, \ldots, x_n)$.

In this setting, the server's view may contain more than just its output. It is also natural to consider that some of the users executing the protocol may themselves be corrupted and colluding with the server. (In this work, we consider semi-honest corruptions only. That is, we assume corrupted parties—including the server—follow the protocol as directed, but may then try to learn additional information based on their collective view of the protocol execution.) Given a set of parties A (that we assume by default always includes the server), we let $\text{VIEW}_{\Pi,A}(x_1, \ldots, x_n)$ be the random variable denoting the joint view of the parties in A in an execution of protocol Π when the users initially hold inputs $x_1, \ldots, x_n$. Let H denote the set of users not in A; let $\mathbf{x}_A$ denote the inputs of users in A; and let $\mathbf{x}_H$ denote the inputs of users outside of A. Then:

Definition 2.2. *Protocol Π is* (ϵ,δ)-DP for t corrupted users *if for any set A containing the server and up to t users and any $\mathbf{x}_A$, the function mapping $\mathbf{x}_H$ to $\text{VIEW}_{\Pi,A}(\mathbf{x}_A, \mathbf{x}_H)$ is (ϵ,δ)-DP, i.e., for any neighboring $\mathbf{x}_H, \mathbf{x}'_H$ and any set V of possible (joint) views of the parties in A, we have*

$$\Pr[\text{VIEW}_{\Pi,A}(\mathbf{x}_A, \mathbf{x}_H) \in V] \leq e^{\epsilon} \cdot \Pr[\text{VIEW}_{\Pi,A}(\mathbf{x}_A, \mathbf{x}'_H) \in V] + \delta.$$

One can also consider protocols operating in a hybrid world. The shuffle model is a special case of this, where the parties have access to an ideal functionality $\mathcal{S}$ implementing the shuffler. Concretely, the protocol $(\mathcal{R}_{\gamma,D} \times \cdots \times \mathcal{R}_{\gamma,D})^{\mathcal{S}}$ corresponding to the randomized response mechanism is the one in which each user locally computes $y_i \leftarrow \mathcal{R}_{\gamma,D}(x_i)$ and then sends y_i to $\mathcal{S}$, which sends the result $\{y_i\} := \mathcal{S}(y_1, \ldots, y_n)$ to the server. The fact that some of the users themselves might be corrupted, however, now needs to be taken into account. The following is an easy corollary of Theorem 2.1:

Corollary 2.3. *Fix n, t, ϵ, δ, and D. If $\gamma \geq \max\left\{\frac{14\cdot|D|\log(2/\delta)}{(n-t-1)\cdot\epsilon^2}, \frac{27\cdot|D|}{(n-t-1)\cdot\epsilon}\right\}$, then $(\mathcal{R}_{\gamma,D} \times \cdots \times \mathcal{R}_{\gamma,D})^{\mathcal{S}}$ is (ϵ,δ)-DP for t corrupted users in the $\mathcal{S}$-hybrid model.*

Shuffle protocols. A protocol Σ is a *shuffle protocol* if it implements $\mathcal{S}$, i.e., if the output generated by the server when running Σ is the multiset containing the users' inputs. We are interested in shuffle protocols that ensure differential privacy when used to implement the shuffle model. Note, however, that we cannot use differential privacy to analyze a shuffle protocol itself: *no* shuffle protocol is differentially private, since two neighboring inputs $\mathbf{y}, \mathbf{y}'$ lead to disjoint sets of outputs. Instead, we introduce a related, but distinct, definition that we call *differential obliviousness.* (This is conceptually related to, but formally distinct from, the notion of differential obliviousness studied in the client/server setting [9].) We say that two vectors of inputs $\mathbf{y}, \mathbf{y}'$ are *transpositions of each other* if there exist i, j such that $y'_i = y_j$, $y'_j = y_i$, and $y'_k = y_k$ for $k \notin \{i, j\}$, i.e., if $\mathbf{y}'$ is the same as $\mathbf{y}$ but with the elements at positions i, j swapped. Then:

Definition 2.4. *Shuffle protocol Σ is (ϵ, δ)-*differentially oblivious for t corrupted users *if for any set A containing the server and up to t users, any $\mathbf{y}_A$, any $\mathbf{y}_H, \mathbf{y}'_H$ that are transpositions of each other, and any set V of possible (joint) views of the parties in A, we have*

$$\Pr[\textsc{view}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V] \leq e^{\epsilon} \cdot \Pr[\textsc{view}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V] + \delta.$$

3 Distributing the Privacy Blanket

We first show that any differentially oblivious shuffle protocol preserves differential privacy when used with the randomized response mechanism. Formally:

Theorem 3.1. *Let Σ be a shuffle protocol that is (ϵ, δ)-differentially oblivious for t corrupted users. If $(\mathcal{R}_{\gamma,D} \times \cdots \times \mathcal{R}_{\gamma,D})^{\mathcal{S}}$ is (ϵ', δ')-differentially private for t corrupted users, then $(\mathcal{R}_{\gamma,D} \times \cdots \times \mathcal{R}_{\gamma,D})^{\Sigma}$ is $(\epsilon + \epsilon', \delta + \delta')$-differentially private for t corrupted users.*

Overview of the proof. Throughout this section, we let Π denote $\mathcal{R}_{\gamma,D} \times \cdots \times \mathcal{R}_{\gamma,D}$; our goal is to prove differential privacy of Π^{Σ}. We give a formal proof starting in the next subsection; here, we provide an overview. We collectively call the adversarial parties (the t corrupted users plus the server) "the adversary." Fix some neighboring inputs $\mathbf{x} = (\mathbf{x}_A, \mathbf{x}_H)$ and $\mathbf{x}' = (\mathbf{x}_A, \mathbf{x}'_H)$, and some set V of the adversary's views. (Each view in V includes the views of the server and t corrupted users in an execution of Π^{Σ}.) We formally define these in the next section, but, conceptually, we separate each view $v \in V$ into three components: v_1 that reflects the adversary's view of the unmodified inputs provided to Σ (which is the same as the adversary's view of the unmodified inputs sent to the shuffler analyzed with privacy blanket method); the final multiset h output by the server (which is the same as the final multiset that would be output by the shuffler conditioned on v_1); and the view v_2 that results from the execution of Σ itself. For some first component v_1 and output multiset h, let $Y(v_1, h)$ denote the set of honest inputs $\mathbf{y}_H$ to Σ that are consistent with v_1, h, and $\mathbf{x}$, and let $Y'(v_1, h)$ denote the set of $\mathbf{y}_H$ consistent with v_1, h, and $\mathbf{x}'$. For example, suppose the output multiset seen by the server is $h = \{1, 1, 1, 1, 2, 2, 2\}$, and that two corrupted users provided inputs $v_1 = \{1, 2\}$ (after applying $\mathcal{R}$). Let inputs $\mathbf{x}$ and $\mathbf{x}'$ differ only in the last honest user's value, where the last entry in $\mathbf{x}_H$ is 1, and the the last entry in $\mathbf{x}'_H$ is 2. Then, as depicted in Figure 1, the set Y consists of all ordered vectors that a) contain a 1 in the final position, and b) are consistent with multiset $\{1,1,1,2,2\}$, which results from removing the adversary's inputs. [2] Similarly, the set Y' contains the ordered vectors with a 2 in the final position, and consistent with the same multiset.

We wish to prove that for any set of adversial views V,

$$\Pr[(v_1,h,v_2)\in V \mid \mathbf{x}] \leq e^{\epsilon+\epsilon'}\Pr[(v_1,h,v_2)\in V \mid \mathbf{x}'] + \delta + \delta'.$$

By separating the leakage due to the server's output multiset from the leakage that results from the shuffle protocol, we can leverage the existing guarantee analyzed using privacy blanket method, where a truly oblivious shuffle is used. Formally, we do that by letting V' denote the set that results from restricting the elements of V to the first two entries. Using the above definitions, we have:

$$\begin{aligned}
\Pr[(v_1,h,v_2)\in V \mid \mathbf{x}] &= \sum_{(v_1,h,v_2)\in V} \Pr[(v_1,h,v_2) \mid \mathbf{x}] \\
&= \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}) \in Y(v_1,h) \mid v_1] \cdot \\
&\qquad \Pr_{\mathbf{y}_H \leftarrow Y(v_1,h)}[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A,\mathbf{y}_H) \in V_2(v_1,h)].
\end{aligned}$$

The second probability in the product above ensures that the randomized input vector is consistent both with the multiset h received by the adversary, and with its knowledge of the ordered, un-randomized inputs. Subject to those constraints, note that each honest input vector $y_H \in Y(v_1,h)$ has equal probability weight, as it is only the randomized honest inputs that determine the unconstrained values; each honest party that randomizes their input is equally likely to choose any input value. This is captured in the final probability, above.

The original analysis with privacy blanket method allows us to claim:

$$\begin{aligned}
&\sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Pr\left[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}) \in Y(v_1,h) \mid v_1\right] \\
&\leq \; e^{\epsilon'} \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}'] \cdot \Pr\left[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}') \in Y'(v_1,h) \mid v_1\right] + \delta'.
\end{aligned}$$

Therefore, the main technical argument that remains to be made is that:

$$\Pr_{\mathbf{y}_H \leftarrow Y}[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A,\mathbf{y}_H) \in V_2] \quad \leq \quad e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A,\mathbf{y}'_H) \in V_2] + \delta.$$

[2] When we analyze this formally, we will also include the adversary knowledge of which honest parties do not randomized their inputs. We then further restrict Y and Y' to contain only vectors consistent with the adversary's inputs, and unrandomized honest inputs. We omit this here for simplicity, and treat all honest inputs as though they were randomized.

The proof of this claim (Lemma 3.7) follows from a combinatorial analysis of the two sets, Y and Y'. As depicted in Figure 1, we say that two elements from Y and Y' are neighboring if they differ by a single transposition. The security of a differentially oblivious shuffle guarantees that neighboring vectors give rise to (roughly) the same view, during shuffling. If we can establish a bijection between these two sets of vectors, mapping each element of Y to its neighbor in Y', our main theorem follows immediately. Unfortunately, as can be seen in the example of Figure 1, Y and Y' do not necessarily have the same size, and so there is no guarantee of such a bijection.

Nevertheless, we can immediately see some structure in that example: each vector in Y has 2 neighbors in Y', and each vector in Y' has 3 neighbors in Y. We extend the sets Y and Y' to multisets $[Y]$ and $[Y']$, by duplicating entries in such a way that $|[Y]| = |[Y']|$. The resulting multisets preserve the probability weights of each vector: sampling a uniform $\mathbf{y}_H \in Y$ is the same as sampling a uniform $\mathbf{y}_H \in [Y]$ (and similarly for Y' and $[Y']$). Furthermore, these multisets allow us to establish a bijection $\phi : [Y] \to [Y']$ such that for any $\mathbf{y}_H \in [Y]$, $\mathbf{y}_H$ and $\phi(\mathbf{y}_H)$ are transpositions of each other. This allows us to use the fact that Σ is (ϵ, δ)-differentially oblivious for t corrupted users to prove the final claim made above.

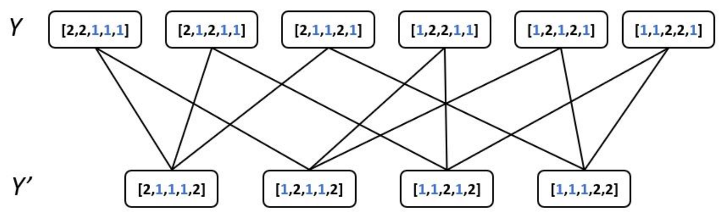

Figure 1: Transposition relations between vectors in Y and Y'

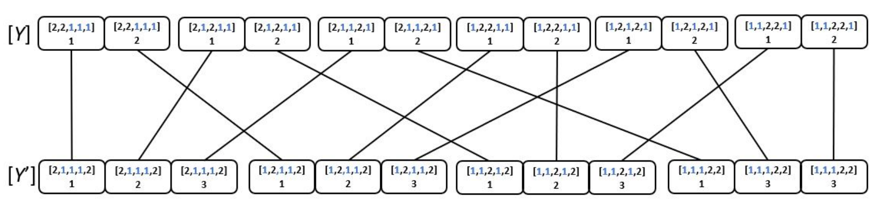

Figure 2: A bijection between $[Y]$ and $[Y']$, derived from Y and Y'.

3.1 Notation and Preliminaries

We now formalize the preceding intuition. We assume t users are corrupted and let $m = n - t$ be the number of uncorrupted users. Fix some neighboring inputs $\mathbf{x} = (\mathbf{x}_A, \mathbf{x}_H)$ and $\mathbf{x}' = (\mathbf{x}_A, \mathbf{x}'_H)$, and for $i \in [m]$ let $x_{H,i}$ be the input of the ith honest user. Without loss of generality, we assume $\mathbf{x}_H$ and $\mathbf{x}'_H$ differ on the input of the mth user, and further assume that $x_{H,m} = 1$ and $x'_{H,m} = 2$.

The adversary's view. We now make explicit the components of the adversary's view in an execution of Π^Σ on input $\mathbf{x}$. The first component of the view, which we generally denote by v_1, includes $\mathbf{y}_A = (\mathcal{R}_{\gamma,D} \times \cdots \times \mathcal{R}_{\gamma,D})(\mathbf{x}_A)$, i.e., the adversary's inputs to Σ. Following Balle et al. [3], we also include in v_1 the honest users' inputs $(x_{H,1}, \ldots x_{H,m-1})$ except mth user's input, and the vector $\mathbf{b} = (b_1, \ldots, b_m)$ indicating which of the honest users' inputs are replaced by a random value, i.e., if $b_i = 0$ then $y_{H,i} = x_{H,i}$ and if $b_i = 1$ then $y_{H,i} \leftarrow D$. The second component of the adversary's view is the multiset $h = \mathcal{S}(\mathbf{y}_A, \mathbf{y}_H)$ output by Σ, in which $(\mathbf{y}_A, \mathbf{y}_H)$ denotes the vector of inputs that the parties provide to Σ; notice that part of $\mathbf{y}_H$ can be deduced from v_1. The third component v_2 of the adversary's view consists of the entire view of the adversary in the execution of Σ on inputs $\mathbf{y} = (\mathbf{y}_A, \mathbf{y}_H)$. (Although v_2 determines h, we find it useful to treat h separately.)

For the rest of the proof, fix some set of views $V = \{(v_1, h, v_2)\}$. We assume without loss of generality that each view in V has non-zero probability when the honest inputs are $\mathbf{x}_H$. Note that views for which $b_m = 1$ are equiprobable regardless of whether the honest inputs are $\mathbf{x}_H$ or $\mathbf{x}'_H$; therefore, we also assume without loss of generality that all views in V have $b_m = 0$.

3.2 Step 1: Using Differential Privacy of $\mathcal{R}_{\gamma,D}$

For some fixed v_1, h, let $Y(v_1, h)$ denote the set of honest inputs $\mathbf{y}_H$ that are consistent with v_1, h, and $\mathbf{x}$. That is, $Y(v_1, h)$ contains all $\mathbf{y}_H \in D^m$ such that (1) for all i with $b_i = 0$, we have $y_{H,i} = x_{H,i}$ (so, in particular, $y_{H,m} = x_{H,m} = 1$), and (2) $\mathcal{S}(\mathbf{y}_A, \mathbf{y}_H) = h$ (where $\mathbf{y}_A$ is fixed by v_1). Similarly, we let $Y'(v_1, h)$ denote the set of $\mathbf{y}_H$ consistent with v_1, h, and $\mathbf{x}'$. We now show:

Lemma 3.2. *If $\Pi^{\mathcal{S}}$ is (ϵ', δ')-DP for t corrupted users, then for any set $V' = \{(v_1, h)\}$ and any pair of neighboring inputs $\mathbf{x}, \mathbf{x}'$, we have:*

$$\begin{aligned}
&\sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Pr\left[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}) \in Y(v_1, h) \mid v_1\right] \\
&\leq \quad e^{\epsilon'} \cdot \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}'] \cdot \Pr\left[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}') \in Y'(v_1, h) \mid v_1\right] + \delta'.
\end{aligned}$$

Proof. Differential privacy of $\Pi^{\mathcal{S}}$ implies[3] that

$$\sum_{(v_1,h)\in V'} \Pr[v_1, h \mid \mathbf{x}] \leq e^{\epsilon'} \sum_{(v_1,h)\in V'} \Pr[v_1, h \mid \mathbf{x}'] + \delta'. \tag{1}$$

Moreover, we have

[3]Technically, this does not follow from differential privacy of $\Pi^{\mathcal{S}}$; it follows, however, from the stronger result proven by Balle et al. [3].

$$
\begin{aligned}
\Pr[v_1, h \mid \mathbf{x}] &= \Pr[v_1 \mid \mathbf{x}] \cdot \Pr[h \mid v_1, \mathbf{x}] \\
&= \Pr[v_1 \mid \mathbf{x}] \cdot \Pr\left[\mathcal{R}_{\gamma,D}^{\otimes m}(\mathbf{x}) \in Y(v_1, h) \mid v_1\right], \qquad (2)
\end{aligned}
$$

and similarly

$$
\Pr[v_1, h \mid \mathbf{x}'] = \Pr[v_1 \mid \mathbf{x}'] \cdot \Pr\left[\mathcal{R}_{\gamma,D}^{\otimes m}(\mathbf{x}') \in Y'(v_1, h) \mid v_1\right].
$$

Substituting (2) and (3) into (1) yields the lemma. □

We use a slightly stronger formulation of the above lemma, while also introducing some additional notation. For any v_1, h, define

$$
\Delta(v_1, h) \stackrel{\text{def}}{=} \max\left\{\Pr[\mathcal{R}_{\gamma,D}^{\otimes m}(\mathbf{x}) \in Y(v_1, h) \mid v_1] - e^{\epsilon'} \cdot \Pr[\mathcal{R}_{\gamma,D}^{\otimes m}(\mathbf{x}') \in Y'(v_1, h) \mid v_1],\ 0\right\}.
$$

We then have:

Lemma 3.3. *If $\Pi^{\mathcal{S}}$ is (ϵ', δ')-DP for t corrupted users, then for any set $V' = \{(v_1, h)\}$ and any pair of neighboring inputs $\mathbf{x}, \mathbf{x}'$, we have:*

$$
\sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Delta(v_1, h) \le \delta'.
$$

Proof. Define

$$
\Delta^+(v_1, h) \stackrel{\text{def}}{=} \Pr[\mathcal{R}_{\gamma,D}^{\otimes m}(\mathbf{x}) \in Y(v_1, h) \mid v_1] - e^{\epsilon'} \cdot \Pr[\mathcal{R}_{\gamma,D}^{\otimes m}(\mathbf{x}') \in Y'(v_1, h) \mid v_1],
$$

and let $V^+ \subseteq V'$ be the elements $(v_1, h) \in V'$ for which $\Delta^+(v_1, h) > 0$. Using the observation that $\Pr[v_1 \mid \mathbf{x}] = \Pr[v_1 \mid \mathbf{x}']$, Lemma 3.2 implies that

$$
\sum_{(v_1,h)\in V^+} \Pr[v_1 \mid \mathbf{x}] \cdot \Delta^+(v_1, h) \le \delta'.
$$

But then

$$
\begin{aligned}
&\sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Delta(v_1, h) \\
&= \sum_{(v_1,h)\in V^+} \Pr[v_1 \mid \mathbf{x}] \cdot \Delta(v_1, h) + \sum_{(v_1,h)\in V'\setminus V^+} \Pr[v_1 \mid \mathbf{x}] \cdot \Delta(v_1, h) \\
&= \sum_{(v_1,h)\in V^+} \Pr[v_1 \mid \mathbf{x}] \cdot \Delta^+(v_1, h) \ \le\ \delta'.
\end{aligned}
$$

□

3.3 Step 2: Using Differential Obliviousness of Σ

In this section, we fix some v_1, h for which $Y(v_1, h)$ and $Y'(v_1, h)$ are both non-empty. To reduce clutter, we write Y for $Y(v_1, h)$ and Y' for $Y'(v_1, h)$. Recall that for all $\mathbf{y}_H \in Y$ we have $y_{H,m} = 1$, and for all $\mathbf{y}'_H \in Y'$ we have $y'_{H,m} = 2$.

Let $\bar{h}$ denote the multiset that remains after removing from h the multiset given by the elements of $\mathbf{y}_A$ and the multiset $\{\mathbf{x}_{H,i} \mid b_i = 0, i \neq m\}$ (both of which are determined by v_1). Let c_1 be the number of 1's in $\bar{h}$, and let c_2 be the number of 2's in $\bar{h}$. Note that $c_1, c_2 \neq 0$ by our assumption that Y and Y' are not empty.

Lemma 3.4. $\frac{|Y|}{|Y'|} = \frac{c_1}{c_2}$.

Proof. Let C be the number of ways of distributing all the elements of $\bar{h}$ that are not equal to 1 or 2 among the honest users who have changed their inputs. A vector $\mathbf{y}_H$ is consistent with v_1, h, and $\mathbf{x}$ only if a 1 is associated with the last user, and the remaining $c_1 + c_2 - 1$ elements of $\bar{h}$ that are 1 or 2 are distributed among the $c_1 + c_2 - 1$ users who remain from those who have changed their inputs. Thus,

$$|Y| = C \cdot \binom{c_1 + c_2 - 1}{c_1 - 1}.$$

Similarly,

$$|Y'| = C \cdot \binom{c_1 + c_2 - 1}{c_2 - 1}.$$

Hence,

$$\frac{|Y|}{|Y'|} = \frac{\binom{c_1+c_2-1}{c_1-1}}{\binom{c_1+c_2-1}{c_2-1}} = \frac{\frac{(c_1+c_2-1)!}{(c_1-1)!c_2!}}{\frac{(c_1+c_2-1)!}{c_1!(c_2-1)!}} = \frac{c_1!(c_2-1)!}{(c_1-1)!c_2!} = \frac{c_1}{c_2}.$$

$\square$

Lemma 3.5. *For every $\mathbf{y}_H \in Y$, there are c_2 vectors in Y' that result from transposing the final entry of $\mathbf{y}_H$ with some other entry of $\mathbf{y}_H$. Similarly, for every $\mathbf{y}'_H \in Y'$, there are c_1 vectors in Y that result from transposing the final entry of $\mathbf{y}'_H$ with some other entry of $\mathbf{y}'_H$.*

Proof. We prove the first statement; the second follows symmetrically. Fix a vector $\mathbf{y}_H \in Y$. The final entry of $\mathbf{y}_H$ must be 1, and there are c_2 other entries of $\mathbf{y}_H$ that are equal to 2 and that correspond to users who have changed their inputs. Transposing the final entry of $\mathbf{y}_H$ with the entries at any of those locations gives a vector in Y'. $\square$

Mapping between Y and Y'. Ideally, we would like to construct a bijection between Y and Y' such that a vector in Y is mapped to a vector in Y' iff they are transpositions of each other. Then for each pair of such vectors $\mathbf{y}_H$ and $\mathbf{y}'_H$, we could argue that $\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H)$ and $\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H)$ must be "close" by differential obliviousness of Σ. Unfortunately, as shown in Lemma 3.4, the cardinalities of Y and Y' might be different, so such a bijection might not exist.

To resolve this issue, we "duplicate" vectors in Y and Y' so that the resulting multisets $[Y]$ and $[Y']$ have the same cardinality. Concretely, we let $[Y]$ be a multiset consisting of c_2 copies of each element $\mathbf{y}_H \in Y$. Similarly, we let $[Y']$ be a multiset consisting of c_1 copies of each element $\mathbf{y}'_H \in Y'$. Note that sampling uniformly from $[Y]$ (resp., $[Y']$) is equivalent to sampling uniformly from Y (resp., Y'). Moreover, by Lemma 3.4, we have $[Y]$ and $[Y']$ have the same size.

Lemma 3.6. *There is a bijection $\phi : [Y] \to [Y']$ such that for every $\mathbf{y}_H \in [Y]$, the vector $\phi(\mathbf{y}_H) \in [Y']$ is a transposition of $\mathbf{y}_H$.*

Proof. Consider the bipartite graph G with vertex sets $[Y]$ and $[Y']$, where there is an edge between $\mathbf{y}_H \in [Y]$ and $\mathbf{y}'_H \in [Y]'$ iff $\mathbf{y}'_H$ results from transposing the final entry of $\mathbf{y}_H$ with some other entry of $\mathbf{y}_H$. Using Lemma 3.5 and the fact that every vector in Y' is included c_1 times in $[Y']$, we see that each $\mathbf{y}_H \in [Y]$ has exactly $c_1 \cdot c_2$ edges. Reasoning analogously, each $\mathbf{y}'_H \in [Y']$ has $c_1 \cdot c_2$ edges. Hall's marriage theorem implies that G has a complete matching, which is also a perfect matching since $[Y]$ and $[Y']$ have the same size. Any such matching constitutes a bijection ϕ as claimed by the lemma. $\square$

We may now prove the main result of this section.

Lemma 3.7. *If Σ is (ϵ, δ)-differentially oblivious for t corrupted users, then for any set V_2 of views of an execution of Σ, we have:*

$$\Pr_{\mathbf{y}_H \leftarrow Y}[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2] \quad \leq \quad e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta.$$

Proof. Let $\phi : [Y] \to [Y']$ be a bijection as guaranteed by Lemma 3.6. Differential obliviousness of Σ implies that for any $\mathbf{y}_H \in [Y]$:

$$\Pr\left[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2\right] \quad \leq \quad e^{\epsilon} \cdot \Pr[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \phi(\mathbf{y}_H)) \in V_2] + \delta.$$

Recall $[Y]$ and $[Y']$ have the same size, we have

$$\begin{aligned}
& \Pr_{\mathbf{y}_H \leftarrow Y}[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2] \\
= \; & \Pr_{\mathbf{y}_H \leftarrow [Y]}\left[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2\right] \\
= \; & \sum_{\mathbf{y}_H \in [Y]} \frac{\Pr\left[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2\right]}{|[Y]|} \\
\leq \; & \sum_{\mathbf{y}_H \in [Y]} \frac{e^{\epsilon} \cdot \Pr[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \phi(\mathbf{y}_H)) \in V_2] + \delta}{|[Y]|} \\
= \; & \sum_{\mathbf{y}'_H \in [Y']} \frac{e^{\epsilon} \cdot \Pr[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta}{|[Y']|} \\
= \; & e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow [Y']}[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta \\
= \; & e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta.
\end{aligned}$$

$\square$

3.4 Putting it all Together

We now prove Theorem 3.1. Let $V' = \{(v_1, h) \mid \exists v_2 : (v_1, h, v_2) \in V\}$. For any $(v_1, h) \in V'$, let $V_2(v_1, h) = \{v_2 \mid (v_1, h, v_2) \in V\}$. We have

$$\begin{aligned}
&\Pr[(v_1, h, v_2) \in V \mid \mathbf{x}] \\
&\quad = \sum_{(v_1,h,v_2)\in V} \Pr[(v_1, h, v_2) \mid \mathbf{x}] \\
&\quad = \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}) \in Y(v_1, h) \mid v_1] \\
&\qquad\qquad \cdot \Pr_{\mathbf{y}_H \leftarrow Y(v_1,h)}[\textsc{view}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2(v_1, h)].
\end{aligned}$$

For brevity, we write Y, Y', and V_2, in place of $Y(v_1, h)$, $Y'(v_1, h)$, and $V_2(v_1, h)$, respectively; we also write $\textsc{view}(\mathbf{y}_H)$ as shorthand for $\textsc{view}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H)$. Using Lemma 3.7, we have that for all v_1, h:

$$\begin{aligned}
&\Pr_{\mathbf{y}_H \leftarrow Y}[\textsc{view}(\mathbf{y}_H) \in V_2] \\
&\quad \leq \min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\textsc{view}(\mathbf{y}'_H) \in V_2],\ 1\right\} + \delta,
\end{aligned}$$

where we treat $\Pr_{\mathbf{y}'_H \leftarrow Y'}[\textsc{view}(\mathbf{y}'_H) \in V_2]$ as 1 in case Y' is empty. (Recall that $Y \neq \emptyset$ by assumption on V.) It follows that

$$\begin{aligned}
&\Pr[(v_1, h, v_2) \in V \mid \mathbf{x}] \\
&\quad \leq \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}) \in Y \mid v_1] \\
&\qquad\qquad \cdot \left(\min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\textsc{view}(\mathbf{y}'_H) \in V_2],\ 1\right\} + \delta\right) \\
&\quad \leq \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}) \in Y \mid v_1] \\
&\qquad\qquad \cdot \min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\textsc{view}(\mathbf{y}'_H) \in V_2],\ 1\right\} + \delta.
\end{aligned}$$

Recalling that

$$\Delta(v_1, h) \stackrel{\text{def}}{=} \max\left\{\Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}) \in Y \mid v_1] - e^{\epsilon'} \cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}') \in Y' \mid v_1],\ 0\right\},$$

we thus have

$$\begin{aligned}
&\Pr[(v_1, h, v_2) \in V \mid \mathbf{x}] \\
&\quad \leq \left(\sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \left(e^{\epsilon'} \cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}') \in Y' \mid v_1] + \Delta(v_1, h)\right)\right. \\
&\qquad\qquad \left. \cdot \min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\textsc{view}(\mathbf{y}'_H) \in V_2(v_1, h)],\ 1\right\}\right) + \delta.
\end{aligned}$$

Using the fact that $(a+b)\cdot\min\{c,d\} \le ac+bd$, we obtain

$$\begin{aligned}
&\Pr[(v_1,h,v_2)\in V \mid \mathbf{x}] \\
&\le \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}] \cdot \Big(e^{\epsilon'+\epsilon}\cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}')\in Y' \mid v_1] \\
&\qquad \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H)\in V_2] + \Delta(v_1,h)\Big) + \delta.
\end{aligned}$$

Finally, applying Lemma 3.3 gives

$$\begin{aligned}
&\Pr[(v_1,h,v_2)\in V \mid \mathbf{x}] \\
&\le e^{\epsilon+\epsilon'}\cdot \sum_{(v_1,h)\in V'} \Pr[v_1 \mid \mathbf{x}]\cdot \Pr[\mathcal{R}^{\otimes m}_{\gamma,D}(\mathbf{x}')\in Y' \mid v_1] \\
&\qquad \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H)\in V_2] + \delta' + \delta \\
&= e^{\epsilon+\epsilon'}\cdot \sum_{(v_1,h,v_2)\in V} \Pr[(v_1,h,v_2) \mid \mathbf{x}'] + \delta + \delta' \\
&= e^{\epsilon+\epsilon'}\cdot \Pr[(v_1,h,v_2)\in V \mid \mathbf{x}'] + \delta + \delta'
\end{aligned}$$

(using the fact that $\Pr[v_1 \mid \mathbf{x}] = \Pr[v_1 \mid \mathbf{x}']$), as required.

3.5 Generalizing to Arbitrary ϵ_0-Local Differentially Private Mechanisms

We also make a broader claim of the usefulness of the differentially oblivious shuffle protocol with arbitrary ϵ_0-local differential private (LDP) mechanism. Balle et al. [3] show[4]:

Theorem 3.8 ([3]). *Let $\mathcal{R}$ be an ϵ_0-LDP local randomizer and $\mathcal{S}\circ\overbrace{(\mathcal{R}\times\cdots\times\mathcal{R})}^{n}$ be the corresponding shuffled mechanism. Then $\mathcal{S}\circ(\mathcal{R}\times\cdots\times\mathcal{R})$ is (ϵ',δ')-DP with $\epsilon' = O((1\wedge\epsilon_0)e^{\epsilon_0}\sqrt{\log(1/\delta')/n})$ if $\epsilon_0 \le \log(n/\log(1/\delta'))/2$.*

Recall that $a\wedge b = min\{a,b\}$. Similar to our claim earlier with the randomized response mechanism, we argue by replacing $\mathcal{S}$ with Σ, and assuming t corrupted users, the following theorem holds:

Theorem 3.9. *Let Σ be a shuffle protocol that is (ϵ,δ)-differentially oblivious for t corrupted users, and $\mathcal{R}$ be an ϵ_0-LDP local randomizer, then $(\mathcal{R}\times\cdots\times\mathcal{R})^{\Sigma}$ is $(\epsilon'+\epsilon,\delta'+\delta)$-DP with $\epsilon' = O((1\wedge\epsilon_0)e^{\epsilon_0}\sqrt{\log(1/\delta')/(n-t)})$ if $\epsilon_0 \le \log((n-t)/\log(1/\delta'))/2$.*

We defer the proof to Appendix A.

[4]For cleaner presentation, we put the smoothed, looser bound in their paper here, although our theorem in this section, as well as its proof in the appendix, are also compatible with their more complicated, tighter bound.

4 A Differentially Oblivious Shuffle Protocol

In this section, we present a construction of a differentially oblivious shuffle protocol. We present the protocol in Section 4.1 and analyze its obliviousness in Section 4.2. We compare its concrete performance to relevant prior work in Section 4.4.

4.1 The Shuffling Protocol

Inputs: Each user i has input y_i.

The protocol proceeds as follows:

Round 1: Each user chooses $r-1$ users $i_1, \ldots, i_{r-1} \leftarrow \{U_1, \ldots, U_n\}$ uniformly and independently, and then forms the onion encryption C_r as described in the text. It sends C_r to user i_1.

Rounds $\ell = 2, \ldots, r-1$: For each ciphertext $C_{r-\ell+2}$ received in the previous round, compute $(i_\ell, C_{r-\ell+1}) := \mathsf{Dec}_{\mathsf{sk}_{i_{\ell-1}}}(C_{r-\ell+2})$ and forward $C_{r-\ell+1}$ to user i_ℓ.

Round r: For each ciphertext C_2 received in the previous round, compute $(S, C_1) := \mathsf{Dec}_{\mathsf{sk}_{i_{r-1}}}(C_2)$ and forward C_1 to the server S.

Output: S initializes $h := \emptyset$. Then, for each ciphertext C received in the previous round, compute $y := \mathsf{Dec}_{\mathsf{sk}_S}(C)$ and add y to h.

Figure 3: A differentially oblivious shuffling protocol, based on parameter r.

Recall that in our setting we have n users holding inputs $y_1, \ldots, y_n$, respectively, who would like a server (that we treat as distinct from the n users) to learn the multiset $h = \{y_i\}$. We assume the parties have public/private keys $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n)$, respectively, and that the server has keys $(\mathsf{pk}_S, \mathsf{sk}_S)$. Our protocol, which is based on onion routing [17, 27], works as follows. Let r be a parameter that we fix later. Each user U chooses $r-1$ users $i_1, \ldots, i_{r-1} \leftarrow \{U_1, \ldots, U_n\}$ uniformly and independently (it may be that U chooses itself), and then forms a nested ("onion") encryption of the form

$$C_r = \mathsf{Enc}_{\mathsf{pk}_{i_1}}(i_2, \mathsf{Enc}_{\mathsf{pk}_{i_2}}(i_3, \cdots \\ \cdots (i_{r-1}, \mathsf{Enc}_{\mathsf{pk}_{i_{r-1}}}(S, \mathsf{Enc}_{\mathsf{pk}_S}(y))) \cdots)),$$

such that at each "layer" the identity of the next receiver is encrypted along with an onion encryption whose outer layer can be removed by that receiver. In the first round, U sends C_r to the first receiver i_1, who decrypts to remove the outer layer and thus obtains i_2 and an onion encryption C_{r-1} that it forwards to i_2 in the next round. This process continues for $r-1$ rounds, until in the rth round all parties send the ciphertext $\mathsf{Enc}_{\mathsf{pk}_S}(y)$ they have obtained to the server. (We assume a synchronous communication network.) The protocol is presented in Figure 3 for convenience.

The protocol requires r rounds of communication, and the total number of ciphertexts transmitted is exactly rn. Since ciphertexts have length $O(r \log n)$, the total communication complexity is $O(r^2 n \log n)$.

4.2 Analysis of Obliviousness with $\epsilon = 0$

We assume a semi-honest adversary who corrupts up to t users as well as the server S. The attacker has access to the state of any corrupted user, and can also determine which user sent any message that it received. However, we assume the attacker *cannot* eavesdrop on the communication between honest users, so in particular it cannot tell whether some honest user i sent a message to some other honest user j in some round. We treat encryption as ideal in our analysis of obliviousness in order to simplify our treatment.

Assume without loss of generality that U_1 and U_2 are honest and hold different inputs, and fix input vectors $\mathbf{y}$ and $\mathbf{y}'$ that are transpositions of each other in which the inputs of U_1 and U_2 are swapped. Let i^1_ℓ denote the ℓth intermediate user chosen by user 1 for $1 \leq \ell \leq r-1$, and set $i^1_0 = 1$; define $i^2_0, \ldots, i^2_{r-1}$ similarly. (We let round 0 refer to the beginning of the algorithm when U_1 and U_2 each hold their own input.) To analyze obliviousness, we make the following observation: if there is any round j (with $0 \leq j \leq r-1$) such that U_1 and U_2 both choose an honest intermediate user in rounds j and $j+1$ (i.e., for which users i^1_j, i^1_{j+1}, i^2_j, and i^2_{j+1} are all honest)—call this event Good—then the distributions on the attacker's views are *identical* regardless of whether the input vector is $\mathbf{y}$ or $\mathbf{y}'$. The reason for this is that it is equally likely that the onion encryption of user 1 was routed from i^1_j to i^1_{j+1} and that of user 2 went from i^2_j to i^2_{j+1}, or that the communication was "flipped" so that the onion encryption of user 1 was routed from i^1_j to i^2_{j+1} and that of user 2 went from i^2_j to i^1_{j+1}. In other words, if Good occurs in an execution of the protocol, then perfect obliviousness is achieved. If we let $x_{t,r}$ denote the probability of event Good in an execution of the protocol with parameter r when t users may be corrupted, we have:

Theorem 4.1. *The protocol of Figure 3 is $(0, 1 - x_{t,r})$-differentially oblivious for t corrupted users.*

Proof. We assume a stronger adversary that can identify the original senders for any message, except for those messages belonging to U_1 and U_2. Hence, it suffices to focus on U_1's and U_2's choice of intermediate users. As shown in the discussion above, our protocol achieves full privacy if event Good occurs and no privacy if it does not occur (with $1 - x_{t,r}$ probability). This concludes our proof. $\square$

Our problem is now reduced to lower bounding $x_{t,r}$. Let $p_t = (1 - t/n)^2$ denote the probability that U_1 and U_2 both choose an honest intermediate user in some fixed round $r - 1 \geq j \geq 1$ when t users are corrupted; note that U_1 and U_2 both choose an honest intermediate user (namely, themselves) in round 0 with probability 1. We have the following immediate bound:

Theorem 4.2. *For $r > 1$, we have $x_{t,r} \geq 1 - \left(1 - \left(\frac{n-t}{n}\right)^4\right)^{\lfloor r/2 \rfloor}$. Thus, the protocol of Figure 3 is $\left(0, \left(1 - \left(\frac{n-t}{n}\right)^4\right)^{\lfloor r/2 \rfloor}\right)$-differentially oblivious for t corrupted users.*

Proof. Assume r is even for simplicity. (An analogous argument works when r is odd.) Consider the rounds in $r/2$ disjoint pairs $(0,1), (2,3), \ldots, (r-2, r-1)$. The probability that Good occurs in any particular pair of rounds is at least p_t^2, so the probability that Good never occurs in any pair of rounds is at most $(1-p_t^2)^{r/2}$, i.e., $\Pr[\mathsf{Good}] \geq 1 - (1-p_t^2)^{r/2}$. Plugging in $p_t = (1 - t/n)^2$ yields the result. □

We can derive a tighter bound using a more careful analysis. First observe that we have the following recurrence relation:

$$x_{t,1} = 0, \quad x_{t,2} = p_t$$
$$x_{t,r} = p_t^2 + (1 - p_t) \cdot x_{t,r-1} + p_t \cdot (1 - p_t) \cdot x_{t,r-2}$$

Although we are not aware of a simple, closed-form solution for this recurrence, we can derive a bound on $x_{t,r}$ for any desired t, r. For example, we have:

Theorem 4.3. *For $r > 1$, $x_{n/3,r} \geq 1 - 0.85^r$. Thus, for $r > 1$ the protocol of Figure 3 is $(0,\ 0.85^r)$-differentially oblivious for $n/3$ corrupted users.*

The proof is straightforward and we defer it to Appendix B.

We can similarly show

Theorem 4.4. *For $r \geq 1$, $x_{n/2,r} \geq 1 - 0.95^r$. Thus, the protocol of Figure 3 is $(0,\ 0.95^r)$-differentially oblivious for $n/2$ corrupted users.*

We use the recurrence relation to calculate the exact probability when we estimate concrete costs below.

4.3 Analysis of Obliviousness with Non-zero ϵ

In this subsection, we extend our result in Section 4.2 to the case that $\epsilon > 0$. In particular, we conduct a more complicated analysis to give a smaller value of δ, in exchange for non-zero ϵ. (On the other hand, Section 4.2 gives a tighter δ term when $\epsilon = 0$.) To simplify our analysis, we consider a slightly stronger adversary with the following assumption:

Assumption 4.5. *If the adversary observes a message received by an honest user in round ℓ and observes another message sent by the same honest user in the round $\ell + 1$, the adversary can always tell whether these two messages are owned by the same user or not.*

Let $p = (n-t)/n$ and $q = t/n$ be the probabilities that some intermediate receiver i_ℓ^1 or i_ℓ^2 ($\ell = 1, \ldots, r-1$) is honest / corrupted, respectively. We say a message is *owned* by a user if the user is the original sender that onion encrypts this message. Throughout our analysis, we exclude all messages owned by corrupted users and focus only on the messages owned by honest users. Without loss of generality, assume that the honest users are $U_1, \ldots, U_{n-t}$, and we assume the adversary knows all messages except for those of U_1 and U_2. Finally, for any honest user $j \in [n-t]$, we let $RP_j = [i_1^j, \ldots, i_{r-1}^j]$ be the vector of all intermediate users chosen by user j; alternatively, we can view RP_j as a "routing path" where the nodes are intermediate users. We refer to the collection of all honest parties' routing paths as the *routing graph.*

We say a message is "observed" by the adversary if at least one of its sender and receiver is corrupted, and "hidden" from the adversary otherwise. Also define the *window* of a user as follows:

Definition 4.6. *Given RP_j, we define U_j's* Window *as the interval $[\ell_s^{(j)}, \ell_t^{(j)}]$, where $\ell_s^{(j)}$ (resp. $\ell_t^{(j)}$) is the first (resp. last) round such that U_i's message at that round is hidden. If all U_i's messages are observed, i.e., there does not exist a round where the sender and receiver of its message are both honest, set $\ell_s^{(j)} = \ell_t^{(j)} = \perp$.*

Figure 4 shows a routing graph with 5 rounds, as well as U_1's Window.

The adversary can only observe a partial view of the routing graph. In particular, the adversary's view is the set of all observed messages. Per our Assumption 4.5, the adversary can always "connect" two observed messages sent in consecutive rounds, if they are both owned by the same user. As a result, the adversary can organize the set of all observed messages into a set of "chains", formed by connecting every sequence of consecutive observed messages owned by the same user. We categorise the chains into the following two types:

1. *Cured chains*: If the first message in the chain is sent in round 1, then the adversary knows that every message in the chain belongs to the sender of that message. Similarly, if the last message in the chain is sent in round r (i.e., arrives at the server), and it does not belong to either U_1 or U_2, the adversary can recover its ownership by mapping the final distinct value to its original sender.

2. *Dangling chains*: We refer to all non-cured chains as dangling chains. In particular, this includes all chains that do not include a message sent in the first or last round. Additionally, this also covers the case where U_1 or U_2's chain spans the last round (provided they don't also span the first round).

In Figure 5, we show the cured and dangling chains corresponding to the routing graph in Figure 4.

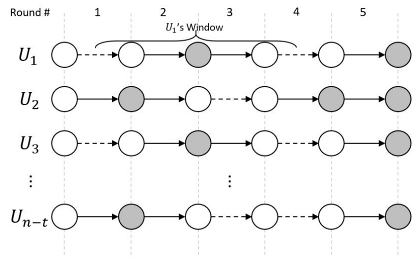

Figure 4: A routing graph, with corrupted users represented by gray nodes and hidden messages drawn using dash arrows.

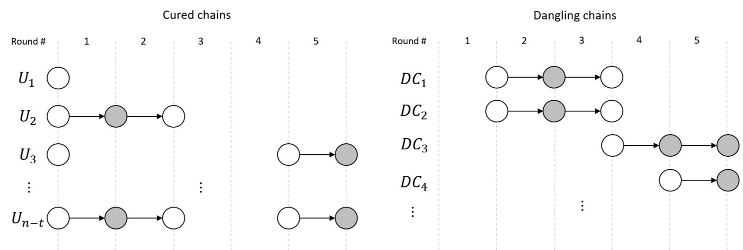

Figure 5: The adversarial view with two types of chains corresponding to the routing graph in Figure 4. Note that we list each dangling chain separately, although both DC_2 and DC_3 belong to U_2.

Following our earlier definitions, let input vectors $\mathbf{y}$ and $\mathbf{y}'$ be transpositions of each other in which the inputs of users 1 and 2 are swapped. Consider U_1 and U_2's $\mathsf{Windows}$ and define the following events:

$$\begin{aligned}
\mathsf{Good} &: (\ell_s^{(1)} = \ell_s^{(2)}) \vee (\ell_t^{(1)} = \ell_s^{(2)}) \vee (\ell_s^{(1)} = \ell_t^{(2)}) \\
\mathsf{Bad}_\perp &: (\ell_s^{(1)} = \perp) \vee (\ell_s^{(2)} = \perp) \\
\mathsf{Bad}_1 &: \ell_s^{(1)} \leq \ell_t^{(1)} < \ell_s^{(2)} \\
\mathsf{OK}_1 &: \ell_s^{(1)} < \ell_s^{(2)} < \ell_t^{(1)} \\
\mathsf{Bad}_2 &: \ell_s^{(2)} \leq \ell_t^{(2)} < \ell_s^{(1)} \\
\mathsf{OK}_2 &: \ell_s^{(2)} < \ell_s^{(1)} < \ell_t^{(2)}
\end{aligned}$$

(We may abuse notations and use the events to denote the corresponding sets of adversary's views.)

Clearly,

$$\Pr[\mathsf{Good}] + \Pr[\mathsf{Bad}_\perp] + \Pr[\mathsf{Bad}_1] + \Pr[\mathsf{OK}_1] + \Pr[\mathsf{Bad}_2] + \Pr[\mathsf{OK}_2] = 1,$$

And

$$\Pr[\mathsf{Bad}_1] = \Pr[\mathsf{Bad}_2], \ \ \Pr[\mathsf{OK}_1] = \Pr[\mathsf{OK}_2].$$

If $\mathsf{Bad}_\perp$ happens, the adversary can connect the complete chain spanning all rounds for either U_1 or U_2, allowing him to learn both parties' messages. A similar argument can be made for Bad_1. In this case, as U_1's Window ends before the start of U_2's Window, U_1's chain spanning the last round (i.e., containing the message sent in the last round) must also span a round prior to the start of U_2's Window. Hence, the adversary knows that this chain belongs to U_1, since otherwise U_2 would own two messages in each of the overlapping rounds, because U_2's chain, prior to its window, is observed by the adversary. Taken together, $\mathsf{Bad}_\perp$, Bad_1, and Bad_2 cover all possibilities that U_1 and U_2's $\mathsf{Windows}$ are not overlapping.

Theorem 4.7. *Let $\delta, \delta_{\mathsf{OK}_1}$ be such that $\delta = \Pr[\mathsf{Bad}_\perp] + 2\Pr[\mathsf{Bad}_1] + 2\delta_{\mathsf{OK}_1}$. The protocol in Figure 3 is (ϵ, δ)-differentially oblivious if for every set of adversarial views $S_1 \subseteq \mathsf{OK}_1$:*

$$\Pr[\textsc{view}_\Sigma(\mathbf{y}) \in S_1] \leq e^\epsilon \cdot \Pr[\textsc{view}_\Sigma(\mathbf{y}') \in S_1] + \delta_{\mathsf{OK}_1} \tag{3}$$

Proof. Due to symmetry, Inequality (3) holds for any $S_2 \subseteq \mathsf{OK}_2$ as well. For any $S_3 \subseteq \mathsf{Bad}_\perp \cup \mathsf{Bad}_1 \cup \mathsf{Bad}_2$, the difference between $\Pr[\textsc{view}_\Sigma(\mathbf{y}) \in S_3]$ and $\Pr[\textsc{view}_\Sigma(\mathbf{y}') \in S_3]$ is trivially bounded by $\Pr[\mathsf{Bad}_\perp] + \Pr[\mathsf{Bad}_1] + \Pr[\mathsf{Bad}_2] = \Pr[\mathsf{Bad}_\perp] + 2 \cdot \Pr[\mathsf{Bad}_1]$. For the remaining views in Good, it is not hard to see that they has perfect privacy, i.e., the probability of generating any subset of views in Good is the same. Finally, as any set of views $S \in \mathrm{Range}(\textsc{view}_\Sigma)$ can always be represented as $S_1 \cup S_2 \cup S_3$ for some S_1, S_2, S_3 as above, adding their corresponding inequalities yields that

$$\Pr[\textsc{view}_\Sigma(\mathbf{y}) \in S] \leq e^\epsilon \cdot \Pr[\textsc{view}_\Sigma(\mathbf{y}') \in S] + \delta.$$

□

We now evaluate the three summands in the expression of δ.

The probability of $\mathsf{Bad}_\perp$. We model each user's choice of intermediate receivers as r i.i.d. Bernoulli trials, each with success probability $p = 1 - t/n$, and index them from 0 to $r - 1$. We rely on the recurrence relation from Section 4.2 to calculate the exact probability of event $\mathsf{Bad}_\perp$. In particular, for fixed t and r, let x_i $(i = 0, \ldots, r-1)$ be the probability that there exist consecutive honest users within the first i rounds, conditioned on a success in the 0th trial. (This captures that the starting user is always honest.) Then

$$x_0 = 0, \quad x_1 = p$$
$$x_i = p^2 + (1-p) \cdot x_{i-1} + p \cdot (1-p) \cdot x_{i-2}$$

By definition of $\mathsf{Bad}_\perp$, we have

Lemma 4.8.

$$\Pr[\mathsf{Bad}_\perp] = 1 - x_{r-1}^2.$$

The probability of Bad_1. In addition to the notations above, for fixed t and r, let x'_i ($i = 0, \ldots, r-1$) be the probability that there exist consecutive honest nodes within the first i round (no longer conditioning on a success in the 0th trial). Then

$$x'_0 = 0, \;\; x'_1 = p^2$$
$$x'_i = p^2 + (1-p) \cdot x'_{i-1} + p \cdot (1-p) \cdot x'_{i-2}$$

Also, let y_i ($i = 0, \ldots, r-1$) be the probability that the first consecutive successes appear at round i and $i+1$, conditioned on a success at round 0. Then

$$y_0 = 0, \;\; y_1 = p, \;\; y_2 = 0$$
$$y_i = p^2 \cdot (1-p) \cdot (1 - x_{i-3})$$

Lemma 4.9.

$$\Pr[\mathsf{Bad}_1] = \sum_{i=1}^{r-2} y_i \cdot y_{i+1} \cdot (1 - x_{r-i-1}) + \sum_{i=1}^{r-2} \sum_{j=i+2}^{r-1} y_i \cdot y_j \cdot (1 - x'_{r-j})$$

Proof. Recall that Bad_1 defines the event that $\ell_s^{(1)} \leq \ell_t^{(1)} < \ell_s^{(2)}$. We first consider the case that U_1 and U_2's $\mathsf{Windows}$ start at consecutive rounds, i.e., $\ell_s^{(1)} + 1 = \ell_s^{(2)}$. Then $\ell_t^{(1)} = \ell_s^{(1)}$. We have that

$$\Pr[\ell_s^{(1)} = i] = y_i, \;\; \Pr[\ell_s^{(2)} = i] = y_{i+1},$$
$$\Pr[\ell_t^{(1)} = i \mid \ell_s^{(1)} = i] = 1 - x_{r-i-1}$$

The first two equations are easy to see. The third probability is the probability that there are no consecutive honest users for U_1 in rounds $i+1, \ldots, r-1$, conditioned on the sender and receiver at round i are both honest, i.e., the sender at round $i+1$ is honest. This is equal to the probability that there exist no consecutive honest nodes within the first $r-i-1$ round, conditioned on the initial user is honest; this probability is $1 - x_{r-i-1}$.

By the three equations above, we get

$$\begin{aligned}
&\Pr[\mathsf{Bad}_1 \mid \ell_s^{(1)} + 1 = \ell_s^{(2)}] \\
&= \sum_{i=1}^{r-2} \Pr[\ell_s^{(1)} = i] \cdot \Pr[\ell_s^{(2)} = i+1] \cdot \Pr[\ell_t^{(1)} = i \mid \ell_s^{(1)} = i] \\
&= \sum_{i=1}^{r-2} y_i \cdot y_{i+1} \cdot (1 - x_{r-i-1})
\end{aligned}$$

For the case that U_1 and U_2's $\mathsf{Windows}$ start at non-consecutive rounds, we have that

$$\Pr[\ell_t^{(1)} < j \mid \ell_s^{(1)} = i] = \Pr[\ell_t^{(1)} = j] = 1 - x'_{r-j}$$

Hence,

$$\begin{aligned}
&\Pr[\mathsf{Bad}_1 \mid \ell_s^{(1)} + 1 < \ell_s^{(2)}] \\
&= \sum_{i=1}^{r-2} \sum_{j=i+2}^{r-1} \Pr[\ell_s^{(1)} = i] \cdot \Pr[\ell_s^{(2)} = j] \cdot \Pr[\ell_t^{(1)} < j \mid \ell_s^{(1)} = i] \\
&= \sum_{i=1}^{r-2} y_i \cdot y_j \cdot (1 - x'_{r-j})
\end{aligned}$$

Combining the equalities for the two cases above yields the theorem.

□

Determining ϵ and δ_{OK_1}. We now provide the exact expressions of ϵ and δ_{OK_1} in order for Inequality (3) in Theorem 4.7 to hold.

Theorem 4.10. *For any set of adversarial views $S \subseteq \mathsf{OK}_1$,*

$$\Pr[\text{VIEW}_\Sigma(\mathbf{y}) \in S] \le e^\epsilon \cdot \Pr[\text{VIEW}_\Sigma(\mathbf{y}') \in S] + \delta_{\mathsf{OK}_1} \tag{4}$$

for $\epsilon \ge -2\ln p = -2\ln(1 - t/n)$ and $\delta_{\mathsf{OK}_1} = 0.5 \cdot (e^{-2\cdot(n-t-2)\cdot[(1-c)\cdot p^4]^2} + e^{-2[c\cdot(n-t-2)\cdot p^4+1]\cdot(p^2-1/e^\epsilon)^2})$ (where c is any constant in $(0,1)$).

In particular, δ_{OK_1} is negligible in n if t is a constant fraction of n and $e^\epsilon - p^{-2} = \Theta(1)$.

To simplify our analysis, we assume an even stronger adversary by providing him with the following "enriched view": aside from the observed messages themselves, the adversary additionally acquires the ownership information for all messages sent by parties other than U_1 and U_2, as long as its owner does not have hidden messages at either round $\ell_s^{(1)}$ or $\ell_t^{(1)}$. Let VIEW^*_Σ be the modified function that outputs this enriched view, and OK_1^* be the set of enriched views corresponding to the original views in OK_1. Due to post-processing, it suffices to analyze the privacy guarantee of VIEW^*_Σ instead. Hence, we only need to prove the following:

Theorem 4.11. *For any set of adversarial views $S \subseteq \mathsf{OK}_1^*$,*

$$\Pr[\text{VIEW}^*_\Sigma(\mathbf{y}) \in S] \le e^\epsilon \cdot \Pr[\text{VIEW}^*_\Sigma(\mathbf{y}') \in S] + \delta_{\mathsf{OK}_1}$$

for ϵ and δ_{OK_1} as in Theorem 4.10.

Fix a particular view $v \in \mathsf{OK}_1^*$. Let n_v be the number of users in $U_3, \ldots, U_{n-t}$ such that their messages at rounds $\ell_s^{(1)}$ and $\ell_t^{(1)}$ are both hidden, i.e., the owners of their messages are not included in the adversary's enriched view. Additionally, consider these n_v users along with U_1, and let k_v be the number of users among them such that their messages at round $\ell_s^{(2)}$ are hidden. We first prove the following lemma:

Lemma 4.12. *Assuming $k_v > 0$,*

$$\frac{\Pr[\textsc{view}^*_\Sigma(\mathbf{y}) = v]}{\Pr[\textsc{view}^*_\Sigma(\mathbf{y}') = v]} \leq \frac{n_v + 1}{k_v}$$

Proof. Given the view v, let A (resp. A') be the set of all valid assignments of all (unenriched) dangling chains such that U_1 and U_2's final messages contain values corresponding to $\mathbf{y}$ (resp. $\mathbf{y}'$); that is, there is at most one observed message for any single user at any round after the assignment. Intuitively, each valid assignment corresponds to an "explanation" of who chose the honest and corrupted receivers and who owns the observed messages.

Due to symmetry, conditioned on input $\mathbf{y}$, any $a \in A$ is (collectively) selected by all honest users with the same probability; denote this probability as $p(v)$ (which only depends on v).

$$\Pr[\textsc{view}^*_\Sigma(\mathbf{y}) = v] = \sum_{a \in A} \Pr[a \mid \mathbf{y}] = |A| \cdot p(v)$$

Similarly,

$$\Pr[\textsc{view}^*_\Sigma(\mathbf{y}') = v] = \sum_{a \in A'} \Pr[a \mid \mathbf{y}'] = |A'| \cdot p(v)$$

Next consider a subset $\hat{A} \subseteq A$ of assignments such that U_1's message at round $\ell_s^{(2)}$ is hidden. For each such assignment, swapping U_1's and U_2's chains after $\ell_s^{(2)}$ yields an assignment in A'. It is straightforward to see that for all assignments in A, the corresponding assignments generated this way are distinct, so $|\hat{A}| \leq |A'|$.

Finally, we show that $\frac{|A|}{|\hat{A}|} = \frac{n_v+1}{k_v}$. We split an assignment a into two parts, a_1 and a_2. a_1 contains all (unenriched) dangling chains that either end before round $\ell_s^{(1)}$ or start after round $\ell_t^{(1)}$, maintaining consistency with input $\mathbf{y}$. Fixing any a_1, we consider the second part a_2, consisting of all remaining (unenriched) dangling chains. By the definition of k_v, for each a_2, there are exactly k_v users among the $n_v + 1$ users with hidden messages at round $\ell_s^{(2)}$. Notice that without a_2, there is no assigned chain overlapping with U_1's Window for any of these $n_v + 1$ users. Thus, due to symmetry, we know that any of these $n_v + 1$ users (including U_1) has a hidden message at round $\ell_s^{(2)}$ in exactly $k_v/(n_v+1)$ fraction of all possible a_2. Since this holds for all a_1, we have $|\hat{A}| = k_v/(n_v + 1) \cdot |A|$.

Combining all results above, we get

$$\frac{\Pr[\textsc{view}^*_\Sigma(\mathbf{y}) = v]}{\Pr[\textsc{view}^*_\Sigma(\mathbf{y}') = v]} = \frac{|A|}{|A'|} \leq \frac{|A|}{|\hat{A}|} = \frac{n_v + 1}{k_v}.$$

□

We now turn to the proof of Theorem 4.11.

Proof. Let $\mathsf{V} \sim \text{VIEW}^*_\Sigma(\mathbf{y})$ denote the random variable for the output of the enriched view on input $\mathbf{y}$. Using the privacy loss random variable, it suffices to show that for any $\epsilon \geq -2\ln p$,

$$\Pr_{\mathsf{V}\sim\text{VIEW}^*_\Sigma(\mathbf{y}):\mathsf{V}\in\mathsf{OK}^*_1}\left[\frac{\Pr[\text{VIEW}^*_\Sigma(\mathbf{y})=\mathsf{V}]}{\Pr[\text{VIEW}^*_\Sigma(\mathbf{y}')=\mathsf{V}]} > e^\epsilon\right] \leq \delta_{\mathsf{OK}_1}$$

(for V such that $\Pr[\text{VIEW}^*_\Sigma(\mathbf{y}')=\mathsf{V}]=0$, we define $\Pr[\text{VIEW}^*_\Sigma(\mathbf{y})=\mathsf{V}]/\Pr[\text{VIEW}^*_\Sigma(\mathbf{y}')=\mathsf{V}]=\infty > e^\epsilon$).

Let N denote the random variable $n_v(\mathsf{V})$, and $\mathsf{K}(\mathsf{N})$ denote the random variable $k_v(\mathsf{V})$. Then $\mathsf{N} \sim Bin(n-t-2, p^4)$ and $\mathsf{K}(\mathsf{N}) \sim Bin(\mathsf{N}+1, p^2)$. [5] According to Lemma 4.12, we only need to prove

$$\Pr[\mathsf{OK}^*_1]\cdot\Pr\left[\frac{\mathsf{N}+1}{\mathsf{K}(\mathsf{N})} > e^\epsilon\right] \leq \delta_{\mathsf{OK}_1}$$

Since $\Pr[\mathsf{OK}^*_1]+\Pr[\mathsf{OK}^*_2] < 1$ and by symmetry, $\Pr[\mathsf{OK}^*_1] = \Pr\mathsf{OK}^*_2]$, we have $\Pr[\mathsf{OK}^*_1] < 0.5$. Hence, it suffices to show

$$\Pr\left[\frac{\mathsf{N}+1}{\mathsf{K}(\mathsf{N})} > e^\epsilon\right] \leq e^{-2\cdot(n-t-2)\cdot[(1-c)\cdot p^4]^2} + e^{-2[c\cdot(n-t-2)\cdot p^4+1]\cdot(p^2-1/e^\epsilon)^2}$$

Note that $\mathsf{E}[\mathsf{N}] = (n-t-2)\cdot p^4$, so by Hoeffding's inequality,

$$\Pr[\mathsf{N} < c\cdot\mathsf{E}[\mathsf{N}]] \leq e^{-2\cdot(n-t-2)\cdot[(1-c)\cdot p^4]^2} \tag{5}$$

for any constant $0 < c < 1$.

Next we upper bound $\Pr\left[\frac{\mathsf{N}+1}{\mathsf{K}(\mathsf{N})} > e^\epsilon\right]$ in the case that $\mathsf{N} \geq c\cdot\mathsf{E}[\mathsf{N}]$. Fix any $u \geq c\cdot\mathsf{E}[\mathsf{N}]$ as the value of N. Recall that $\mathsf{K}(u) \sim Bin(u+1, p^2)$, so by Hoeffding's inequality again, we have

$$\begin{aligned}\Pr\left[\frac{u+1}{\mathsf{K}(u)} > e^\epsilon\right] &= \Pr\left[\mathsf{K}(u) < \frac{u+1}{e^\epsilon}\right]\\ &\leq \Pr\left[\mathsf{K}(u) \leq \frac{u+1}{e^\epsilon}\right]\\ &\leq e^{-2\cdot(u+1)\cdot(p^2-1/e^\epsilon)^2}.\end{aligned}$$

The last expression is a monotonically decreasing function of u, so setting $u = c\cdot\mathsf{E}[\mathsf{N}]$ yields

$$\Pr\left[\frac{\mathsf{N}+1}{\mathsf{K}(\mathsf{N})} > e^\epsilon \,\middle|\, \mathsf{N} \geq c\cdot\mathsf{E}[\mathsf{N}]\right] \quad\leq\quad e^{-2[c\cdot(n-t-2)\cdot p^4+1]\cdot(p^2-1/e^\epsilon)^2}. \tag{6}$$

Combining Inequalities (5) and (6) yields what we want to prove.

□

[5] In fact, the success probability for the first binomial distribution is p^3 when $\ell^1_s = 0$, separately bounding the ratio in this case can resulted in smaller δ term. Hence, we neglect it, trading a slight loss in tightness for a simpler analysis.

4.4 Concrete Performance Estimates

To analyze the performance of our protocol and compare it with prior work, we assume encryption is done using the KEM-DEM paradigm with the KEM portion having a length of 256 bits. We allocate 20 bits for user identities (assuming $n \leq 2^{20}$), and assume users' inputs are 128 bits long. The innermost ciphertext thus requires $256 + 128 = 384$ bits, and in each of the other layers we add 256 bits for the next key encapsulation plus 20 bits for the user ID. An r-layer onion ciphertext thus requires $384 + 276(r - 1)$ bits.

We fix $t = n/3$ and set $r = 171$ so that our protocol is $(0, 2^{-41})$-differentially oblivious. This allows us to compare our shuffling protocol to the protocols of Movahedi et al. [25] and Bell et al. [5]. (We use $(0, 2^{-41})$-differential obliviousness so that when we apply Theorem 3.1 to an $(\epsilon, 2^{-41})$-differentially private protocol using trusted shuffler, the composed protocol satisfies $(\epsilon, 2^{-40})$-differential privacy overall.) In our protocol, each party sends (on average) about 497KB and the round complexity is $r = 171$. Importantly, as the number of parties increases, the only added cost is in the length of the user ID. With a 20-bit user ID, as assumed above, we can support one million parties. In comparison, Movahedi et al. report communication of 128MB per party and require 500 rounds of communication for 33,000 parties, and approximately .5-1GB over 1,000 rounds for one million parties. For 10,000 parties, Bell et al. estimate communication of 910KB per party, and about 12 rounds of communication; their per-party cost grows linearly in the number of parties, and will perform far less favorably as the number of parties approaches one million.

Additionally, we note that δ is often set to be $10^{-4} \geq \delta \geq 10^{-6}$ in the differential privacy literature. Using that range of values, we require $r \approx 55$–83, and our communication cost, per party, is reduced to 53–119KB respectively. The protocol of Bell et al. [5] does not improve with a larger value of the privacy parameter, as they require δ to be small to ensure correctness.

Our protocol reduces the communication cost further if we allow non-zero ϵ for our shuffle protocol. In particular, for all $n > 22000$, $t = n/3$ and setting $c = 5/6$, our protocol is $(1, 2^{-41})$, $(1, 10^{-6})$, $(1, 10^{-4})$-differentially oblivious with $r = 82, 42, 30$ respectively. If we assume $n < 2^{20}$ (So that the user ID can be represented in 20 bits), these correspond to about 116KB, 31KB, 16KB communication cost per party.

References

[1] Megumi Ando, Anna Lysyanskaya, and Eli Upfal. Practical and provably secure onion routing. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018: 45th International Colloquium on Automata, Languages and Programming*, volume 107 of *LIPIcs*, pages 144:1–144:14, Prague, Czech Republic, July 9–13, 2018. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

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

[3] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology—Crypto 2019*, pages 638–667. Springer, 2019.

[4] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In David Wagner, editor, *Advances in Cryptology—Crypto 2008*, pages 451–468. Springer, 2008.

[5] James Henry Bell, Kallista A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overhead. In *Proc. ACM Conference on Computer and Communications Security*, page 1253–1269. ACM Press, 2020.

[6] Aurélien Bellet, Rachid Guerraoui, and Hadrien Hendrikx. Who started this rumor? Quantifying the natural differential privacy guarantees of gossip protocols. In *34th International Symposium on Distributed Computing (DISC)*, volume 179 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

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

[8] Elette Boyle, Shafi Goldwasser, and Stefano Tessaro. Communication locality in secure multi-party computation - how to run sublinear algorithms in a distributed setting. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 356–376, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany.

[9] T.-H. Hubert Chan, Kai-Min Chung, Bruce M. Maggs, and Elaine Shi. Foundations of differentially oblivious algorithms. In Timothy M. Chan, editor, *30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2448–2467, San Diego, CA, USA, January 6–9, 2019. ACM-SIAM.

[10] T-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In Leah Epstein and Paolo Ferragina, editors, *European Symposium on Algorithms (ESA)*, pages 277–288. Springer, 2012.

[11] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Comm. of the ACM*, 24(2):84–88, 1981.

[12] D. Chaum. The dining cryptographers problem: unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[13] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 375–403, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[14] Varsha Dani, Valerie King, Mahnush Movahedi, and Jared Saia. Brief announcement: breaking the $O(nm)$ bit barrier, secure multiparty computation with a static adversary. In Darek Kowalski and Alessandro Panconesi, editors, *31st ACM Symposium Annual on Principles of Distributed Computing*, pages 227–228, Funchal, Madeira, Portugal, July 16–18, 2012. Association for Computing Machinery.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany.

[16] Joan Feigenbaum, Aaron Johnson, and Paul F. Syverson. Probabilistic analysis of onion routing in a black-box model. *ACM Trans. Information and Systems Security*, 15(3):14:1–14:28, 2012.

[17] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In Ross J. Anderson, editor, *Proc. 1st International Workshop on Information Hiding*, volume 1174 of *LNCS*, pages 137–150. Springer, 1996.

[18] Adam Groce, Peter Rindal, and Mike Rosulek. Cheaper private set intersection via differentially private leakage. *Proc. Privacy Enhancing Technologies (PETS)*, 2019(3):6–25, 2019.

[19] Xi He, Ashwin Machanavajjhala, Cheryl J. Flynn, and Divesh Srivastava. Composing differential privacy and secure computation: A case study on scaling private record linkage. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1389–1406, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press.

[20] Yan Huang, David Evans, and Jonathan Katz. Private set intersection: Are garbled circuits better than custom protocols? In *ISOC Network and Distributed System Security Symposium – NDSS 2012*, San Diego, CA, USA, February 5–8, 2012. The Internet Society.

[21] David Lazar, Yossi Gilad, and Nickolai Zeldovich. Karaoke: Distributed private messaging immune to passive traffic analysis. In *Proc. 13th*

USENIX Conference on Operating Systems Design and Implementation, page 711–725. USENIX Association, 2018.

[22] S. Mauw, J. H. S. Verschuren, and E. P. de Vink. A formalization of anonymity and onion routing. In Pierangela Samarati, Peter Ryan, Dieter Gollmann, and Refik Molva, editors, *Computer Security—ESORICS 2004*, pages 109–124. Springer, 2004.

[23] Sahar Mazloom and S. Dov Gordon. Secure computation with differentially private access patterns. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 490–507, Toronto, ON, Canada, October 15–19, 2018. ACM Press.

[24] Sahar Mazloom, Phi Hung Le, Samuel Ranellucci, and S. Dov Gordon. Secure parallel computation on national scale volumes of data. In Srdjan Capkun and Franziska Roesner, editors, *USENIX Security 2020: 29th USENIX Security Symposium*, pages 2487–2504. USENIX Association, August 12–14, 2020.

[25] Mahnush Movahedi, Jared Saia, and Mahdi Zamani. Secure multi-party shuffling. In Christian Scheideler, editor, *Structural Information and Communication Complexity*, pages 459–473, Cham, 2015. Springer International Publishing.

[26] Nigel P. Smart and Younes Talibi Alaoui. Distributing any elliptic curve based protocol. In *Proc. 17th IMA International Conference on Cryptography and Coding (IMACC)*, volume 11929 of *LNCS*, pages 342–366. Springer, 2019.

[27] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous connections and onion routing. In *1997 IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, CA, USA, 1997. IEEE Computer Society Press.

[28] Nirvan Tyagi, Yossi Gilad, Derek Leung, Matei Zaharia, and Nickolai Zeldovich. Stadium: A distributed metadata-private messaging system. In *Proc. 26th Symposium on Operating Systems Principles*, page 423–440. ACM Press, 2017.

[29] Jelle van den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proc. 25th Symposium on Operating Systems Principles (SOSP)*, page 137–152. ACM Press, 2015.

[30] 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 *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479.

A Proof of Theorem 3.9

We start by introducing the generalized privacy blanket method for analyzing arbitrary ϵ_0-local differentially private mechanism in the original paper [3].

Privacy blanket decomposition. Let $\mathcal{R} : D \to R$ be a local randomizer, where R denotes a continuous space. For input $x \in D$, let μ_x denote the distribution of $\mathcal{R}(x)$; we also abuse the notation a bit and use $\mu_x(\cdot)$ to denote its probability density function. For the collection of all output distributions $\{\mu_x\}_{x \in D}$, we define their total variation similarity as:

$$\gamma_{\mathcal{R}} = \int_{-\infty}^{\infty} \inf_x \{\mu_x(y)\} dy;$$

let $\omega_{\mathcal{R}}$ be the blanket distribution with its probability density function $\omega_{\mathcal{R}}(y) = \inf_x \mu_x(y)/\gamma_{\mathcal{R}}$ for $y \in R$. In the rest of this section, we simply write them as γ and ω. Finally, define $\nu_x = (\mu_x - \gamma\omega)/(1-\gamma)$ as an input-dependent distribution. For every randomizer $\mathcal{R}$ and its collection of output distributions $\{\mu_x\}_{x \in D}$, we can decompose each output distribution μ_x into an input-dependent part and an input-independent part:

$$\mu_x = (1-\gamma)\nu_x + \gamma\omega$$

One can understand it as any party with some input x samples from an input-independent distribution ω with probability γ, and from an input-dependent distribution ν_x with probability $1-\gamma$. We first claim the following lemma:

Lemma A.1. *If $\mathcal{R}$ is ϵ_0-local differentially private for some $\epsilon_0 \geq 0$, then the blanket distribution ω and all output distributions in $\{\mu_x\}_{x \in D}$ share full support of domain R.*

Proof. We first show that for any $x, x' \in D$, μ_x and $\mu_{x'}$ must have the same support. Otherwise without loss of generality, assume $\mu_x(y) \neq 0$ and $\mu_{x'}(y) = 0$; then we have $\mu_x(y) > e^{\epsilon_0}\mu_{x'}(y)$ for any $\epsilon_0 \geq 0$, thus $\mathcal{R}$ cannot be ϵ_0-LDP. From the definition of ω, it follows that ω must share this same support with all distributions in $\{\mu_x\}_{x \in D}$. $\square$

Similar to the proof of Theorem 3.8, we assume a stronger adversary that can identify the contribution of any user among the first $m-1$ users that does not sample from the blanket distribution. This is also the assumption made in Section 3; however, notice that for the generalized LDP mechanism, as opposed to the randomized response mechanism, a party not sampling from blanket distribution ω could still enjoy some randomness by sampling from its input-dependent distribution ν_x. Thus, we need to explicitly provide these "partially" randomized values to the adversary and reflect this in our notation. Concretely, we modify the adversary's view v_1 defined earlier by including an additional vector $\hat{\mathbf{y}}_H = (\hat{y}_1, \ldots, \hat{y}_{m-1})$, where

$$\hat{y}_i = \begin{cases} y_{H,i} & \text{if } b_i = 0 \\ \perp & \text{otherwise} \end{cases}$$

The high-level intuition and structure of the proof of Theorem 3.9 are similar to those of Theorem 3.1. However, we need to re-introduce a generalized way to form a bijection.

We start by fixing v_1, h for which $Y(v_1, h)$ and $Y'(v_1, h)$ are both non-empty. For simplicity, we write Y for $Y(v_1, h)$ and Y' for $Y'(v_1, h)$. Also recall that $\bar{h}$ denotes the resulting multiset after removing from h the multiset given by the elements of $\mathbf{y}_A$ and the multiset $\{\hat{y}_i \mid b_i = 0\}$ (both of which are determined by v_1). To align our assumption with the assumption made in [3], we also loosen the earlier restriction that the mth honest party always submits its true input. Instead, this party, which either holds input $x_{H,m}$ or $x'_{H,m}$ in the two neighboring cases, samples from $\mu_{x_{H,m}}$ or $\mu_{x'_{H,m}}$, respectively. Thus, both $y_{H,m}$ and $y'_{H,m}$ can correspond to any element in $\bar{h}$ based on Lemma A.1. And it immediately follows that $Y = Y'$ and both sets include all possible permutations of elements in $\bar{h}$. We keep the redundant notations Y and Y' throughout our proof (and later do the same for $[Y]$ and $[Y']$), as it allows us to draw analogy to our previous analysis given in Section 3.

Similar to the proof in Section 3.3, we apply a vector duplicating approach to generate $[Y]$ and $[Y']$ while claiming a specific bijection ϕ between $[Y]$ and $[Y']$. Roughly speaking, for ever pair of mapped vectors $\mathbf{y}_H$ and $\phi(\mathbf{y}_H)$, their probability densities have the same fraction with their respective sets $[Y]$ and $[Y']$'s probability densities. (This fraction may vary for different pairs of mapped vectors in this bijection.)

For simplicity, we start by assuming there are no duplicate values in $\bar{h}$, and later show how to address the case with duplicate values. Concretely, let $\bar{h} = \{a_i\}_{i=1}^{l}$ where all a_i are distinct. We abuse the notation ω and let $\omega(\bar{h})$ denote the probability density $\prod_{j=1}^{l} \omega(a_j)$.

We partition Y into subsets $Y_1, \ldots, Y_l$ with $Y_i = \{\mathbf{y}_H \in Y \mid y_{H,m} = a_i\}$. Notice that for all i, $|Y_i| = (l-1)!$. Furthermore, all vectors within each set Y_i have the same probability density. In particular, for every vector $\mathbf{y}_H \in Y_i$, its probability density (conditioned on v_1) is given as:

$$f(\mathbf{y}_H \mid v_1) = \frac{\mu_{x_{H,m}}(a_i)}{\omega(a_i)} \cdot \omega(\bar{h})$$

Hence,

$$f(Y_i \mid v_1) = \sum_{\mathbf{y}_H \in Y_i} f(\mathbf{y}_H \mid v_1) = (l-1)! \cdot \frac{\mu_{x_{H,m}}(a_i)}{\omega(a_i)} \cdot \omega(\bar{h})$$

Likewise, we can partition Y' into subsets $Y'_1, \ldots, Y'_l$ in the same way (recall that they are identical sets). Similarly, for every vector $\mathbf{y}'_H \in Y'$, its probability density conditioned on v_1 is:

$$f'(\mathbf{y}'_H \mid v_1) = \frac{\mu_{x'_{H,m}}(a_i)}{\omega(a_i)} \cdot \omega(\bar{h})$$

Hence,

$$f'(Y'_i \mid v_1) = \sum_{\mathbf{y}'_H \in Y'_i} f'(\mathbf{y}_H \mid v_1) = (l-1)! \cdot \frac{\mu_{x'_{H,m}}(a_i)}{\omega(a_i)} \cdot \omega(\bar{h})$$

Relationship between vectors in Y and Y'. We start by examining the transposition relationship between vectors in $Y_1, \ldots, Y_l$ and vectors in $Y'_1, \ldots, Y'_l$. For all Y_i, every vector $\mathbf{y}_H \in Y_i$ has an identical vector in Y'_i, and for all $j \neq i$, y_H has exactly one vector with transposition distance 1 in each of the Y'_j. Collectively, we refer to these l vectors as $\mathbf{y}_H$'s "connected" vector and denote them as a set $C(\mathbf{y}_H)$. Likewise, we denote $\mathbf{y}'_H$'s "connected" vector as $C(\mathbf{y}'_H)$.

Similar to what we did in Section 3, we "duplicate" vectors in Y and Y' to form multisets $[Y]$ and $[Y']$. Concretely, we let $[Y]$ be a multiset consisting of l copies of each element $\mathbf{y}_H \in Y$ and $[Y']$ be a multiset consisting of l copies of each element $\mathbf{y}'_H \in Y'$. Given some $y_H \in Y_i$ and its connected vector $y'_H \in Y'_j$, we map y_H's jth duplicate $\mathbf{y}_H^{(j)}$ to y'_H's ith duplicate $\mathbf{y}_H'^{(i)}$ and we use $\phi(\mathbf{y}_H^{(j)}) = \mathbf{y}_H'^{(i)}$ to denote such mappings.

Lemma A.2. *The mapping $\phi : [Y] \to [Y']$ is a bijection such that for every $\mathbf{y}_H \in [Y]$, the vector $\phi(\mathbf{y}_H) \in [Y']$ is either a transposition of $\mathbf{y}_H$, or identical to $\mathbf{y}_H$.*

Proof. The second part of statement is trivial as we only map a vector $\mathbf{y}_H$'s duplicate to the duplicates of vectors in $C(\mathbf{y}_H)$ and vice versa. For the first part, notice that $|[Y]| = |[Y']|$, as $|Y| = |Y'|$ and both $[Y]$ and $[Y']$ contain l duplicates for each vector. According to our description of ϕ, each $\mathbf{y}_H \in [Y]$ is mapped to exactly one vector $\mathbf{y}'_H \in [Y']$. Due to symmetry, each $\mathbf{y}'_H \in [Y']$ is mapped exactly once. Hence, ϕ is a bijection between $[Y]$ and $[Y']$. □

Assigning probability density for duplicates. For every $\mathbf{y}_H \in Y$, rather than evenly distributing the probability density to each of its duplicates $\mathbf{y}_H^{(1)}, \ldots, \mathbf{y}_H^{(l)}$ (as done in the proof of Theorem 3.1), we assign the probability density proportionally to the probability density of its connected vectors $C(\mathbf{y}_H)$. Concretely, we have:

$$\begin{aligned}
&f(\mathbf{y}_H^{(i)} \mid v_1) \\
=&\frac{\mu_{\mathbf{x}'_{H,m}}(a_i)/\omega(a_i)}{\sum_{j=1}^{l} \mu_{\mathbf{x}'_{H,m}}(a_j)/\omega(a_j)} \cdot f(\mathbf{y}_H \mid v_1) \\
=&\frac{\mu_{\mathbf{x}'_{H,m}}(a_i)/\omega(a_i)}{\sum_{j=1}^{l} \mu_{\mathbf{x}'_{H,m}}(a_j)/\omega(a_j)} \cdot \frac{\mu_{\mathbf{x}_{H,m}}(a_i)/\omega(a_i)}{(l-1!) \cdot \sum_{j=1}^{l} \mu_{\mathbf{x}_{H,m}}(a_j)/\omega(a_j)} \cdot f(Y \mid v_1)
\end{aligned}$$

Similarly, for every $\mathbf{y}'_H \in Y'$, we assign the probability density to its duplicates

${\mathbf{y}'}^{(1)}_H, \ldots, {\mathbf{y}'}^{(l)}_H$:

$$
\begin{aligned}
& f'({\mathbf{y}'}^{(i)}_H \mid v_1) \\
=& \frac{\mu_{\mathbf{x}_{H,m}}(a_i)/\omega(a_i)}{\sum_{j=1}^{l} \mu_{\mathbf{x}_{H,m}}(a_j)/\omega(a_j)} \cdot f'(\mathbf{y}'_H \mid v_1) \\
=& \frac{\mu_{\mathbf{x}_{H,m}}(a_i)/\omega(a_i)}{\sum_{j=1}^{l} \mu_{\mathbf{x}_{H,m}}(a_j)/\omega(a_j)} \cdot \frac{\mu_{\mathbf{x}'_{H,m}}(a_i)/\omega(a_i)}{(l-1!) \cdot \sum_{j=1}^{l} \mu_{\mathbf{x}'_{H,m}}(a_j)/\omega(a_j)} \cdot f'(Y \mid v_1)
\end{aligned}
$$

Lemma A.3. *For every pair of $\mathbf{y}_H \in [Y]$ and $\phi(\mathbf{y}_H) \in [Y']$,*

$$\frac{f(\mathbf{y}_H \mid v_1)}{f([Y] \mid v_1)} = \frac{f'(\phi(\mathbf{y}_H) \mid v_1)}{f([Y'] \mid v_1)}$$

We omit the proof as it is straightforward from the probability density defined above and Lemma A.2.

We are now ready to prove the following lemma, which is a generalized version of Lemma 3.7:

Lemma A.4. *If Σ is (ϵ, δ)-differentially oblivious for t corrupted users, then for any set of views V_2 from an execution of Σ, we have:*

$$\Pr_{\mathbf{y}_H \leftarrow Y}[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2] \quad \leq \quad e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta,$$

where the notation $\mathbf{y}_H \leftarrow Y$ denotes sampling a vector $\mathbf{y}_H$ from set Y according to the distribution described above (similar for $\mathbf{y}'_H \leftarrow Y'$).

Proof. Let $\phi : [Y] \to [Y']$ be the bijection defined in Lemma A.2. Recall that Y is shorthand for $Y(v_1, h)$. Differential obliviousness of Σ implies that for any $\mathbf{y}_H \in [Y]$:

$$\Pr\left[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2\right] \quad \leq \quad e^{\epsilon} \cdot \Pr[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \phi(\mathbf{y}_H)) \in V_2] + \delta.$$

We have:

$$
\begin{aligned}
& \Pr_{\mathbf{y}_H \leftarrow Y}\left[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2\right] \\
=\ & \Pr_{\mathbf{y}_H \leftarrow [Y]}\left[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2\right] \\
=\ & \sum_{\mathbf{y}_H \in [Y]} \Pr\left[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H) \in V_2\right] \cdot \frac{f(\mathbf{y}_H \mid v_1)}{f([Y] \mid v_1)} \\
\leq\ & \sum_{\mathbf{y}_H \in [Y]} (e^{\epsilon} \cdot \Pr[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \phi(\mathbf{y}_H)) \in V_2] + \delta) \cdot \frac{f(\mathbf{y}_H \mid v_1)}{f([Y] \mid v_1)} \\
=\ & \sum_{\mathbf{y}'_H \in [Y']} (e^{\epsilon} \cdot \Pr[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta) \cdot \frac{f(\mathbf{y}'_H \mid v_1)}{f([Y'] \mid v_1)} \\
=\ & e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow [Y']}[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta \\
=\ & e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\mathrm{VIEW}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}'_H) \in V_2] + \delta.
\end{aligned}
$$

□

Handling duplicates. In the case that $\bar{h} = \{a_i\}_{i=1}^{l}$ contains only $d < l$ distinct values, we essentially treat each element of $\bar{h}$ as distinct and adjust the probability density properly. Concretely, let the respective number of these d values be $c_1, \ldots, c_d$. For all $\mathbf{y}_H \in Y$, we create $\prod_{i=1}^{d} c_i!$ duplicates and assign each duplicate with an evenly divided probability density $f(\mathbf{y}_H \mid v_1)/\prod_{i=1}^{d} c_i!$. We do the same for all $\mathbf{y}'_H \in Y'$. As each duplicate is treated as a distinct vector, we can just proceed as what we did earlier with no duplicate values.

Finally, we handle the remaining changes of notations and relevant lemma due to our use of probability density. We first define the continuous counterpart of $\Pr[v_1 \mid \mathbf{x}]$ and $\Pr[v_1 \mid \mathbf{x}]$: let $g(v_1)$ and $g'(v_1)$ denote the corresponding probability density at point v_1, notice that $g = g'$. We also adjust the notation $\Delta(v_1, h)$. In particular, for any v_1, h, let

$$\Delta(v_1, h) \stackrel{\text{def}}{=} \max\left\{f(Y(v_1, h) \mid v_1) - e^{\epsilon'} \cdot f'(Y'(v_1, h) \mid v_1),\ 0\right\}.$$

Using the above notations, we give the following continuous counterpart of Lemma 3.3. We skip the proof as it is analogous:

Lemma A.5. *If $\Pi^{\mathcal{S}}$ is (ϵ', δ')-DP for t corrupted users, then for any set $V' = \{(v_1, h)\}$ and any pair of neighboring inputs $\mathbf{x}, \mathbf{x}'$, we have:*

$$\int_{(v_1,h)\in V'} g(v_1) \cdot \Delta(v_1, h) \leq \delta'.$$

Proof of Theorem 3.9. It suffices to prove that for arbitrary v_1, h and set of Σ's view V_2 consistent with v_1, h, the following inequality holds:

$$\begin{aligned} &g(v_1) \cdot f(Y \mid v_1) \cdot \Pr_{\mathbf{y}_H \leftarrow Y}[\textsc{view}(\mathbf{y}_H) \in V_2] \\ \leq &e^{\epsilon+\epsilon'} \cdot g'(v_1) \cdot f'(Y' \mid v_1) \Pr_{\mathbf{y}'_H \leftarrow Y'}[\textsc{view}(\mathbf{y}'_H) \in V_2] \\ &\quad + g(v_1) \cdot \Delta(v_1, h) + g(v_1) \cdot f(Y \mid v_1) \cdot \delta \end{aligned}$$

where $\textsc{view}(\mathbf{y}_H)$ is shorthand for $\textsc{view}_{\Sigma,A}(\mathbf{y}_A, \mathbf{y}_H)$. Notice that for the last two terms on the RHS of the inequality, for any set $V' = \{(v_1, h)\}$:

$$\int_{(v_1,h)\in V'} g(v_1) \cdot \Delta(v_1, h) + \int_{(v_1,h)\in V'} g(v_1) \cdot f(Y \mid v_1) \cdot \delta \leq \delta' + \delta.$$

This is due to Lemma A.5 and $g(v_1) \cdot f(Y(v_1, h) \mid v_1)$ integrated over all possible views is 1. Moreover, the ratio bound between the integral of the first terms on both sides preserves:

$$\begin{aligned} &\int_{(v_1,h)\in V'} g(v_1) \cdot f(Y \mid v_1) \cdot \Pr_{\mathbf{y}_H \leftarrow Y}[\textsc{view}(\mathbf{y}_H) \in V_2] \\ &\quad / \int_{(v_1,h)\in V'} g'(v_1) \cdot f'(Y' \mid v_1) \Pr_{\mathbf{y}'_H \leftarrow Y'}[\textsc{view}(\mathbf{y}'_H) \in V_2] \\ &\leq e^{\epsilon+\epsilon'}. \end{aligned}$$

Hence, it suffices to focusing on a single pair of v_1, h here.

By Lemma A.4, we have that for all v_1, h:

$$\Pr_{\mathbf{y}_H \leftarrow Y}[\text{VIEW}(\mathbf{y}_H) \in V_2] \quad \leq \quad \min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H) \in V_2],\ 1\right\} + \delta,$$

It follows that

$$\begin{aligned}
& g(v_1) \cdot f(Y \mid v_1) \cdot \Pr_{\mathbf{y}_H \leftarrow Y}[\text{VIEW}(\mathbf{y}_H) \in V_2] \\
\leq \quad & g(v_1) \cdot f(Y \mid v_1) \cdot \left(\min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H) \in V_2],\ 1\right\} + \delta\right) \\
\leq \quad & g(v_1) \cdot f(Y \mid v_1) \cdot \min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H) \in V_2],\ 1\right\} + g(v_1) \cdot f(Y \mid v_1) \cdot \delta \\
\leq \quad & g(v_1) \cdot \left(e^{\epsilon'} \cdot f'(Y' \mid v_1) + \Delta(v_1, h)\right) \cdot \min\left\{e^{\epsilon} \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H) \in V_2],\ 1\right\} \\
& \qquad + g(v_1) \cdot f(Y \mid v_1) \cdot \delta \\
\leq \quad & g(v_1) \cdot \left(e^{\epsilon+\epsilon'} \cdot f'(Y' \mid v_1) \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H) \in V_2] + \Delta(v_1, h)\right) \\
& \qquad + g(v_1) \cdot f(Y \mid v_1) \cdot \delta \\
= \quad & e^{\epsilon+\epsilon'} g'(v_1) \cdot f'(Y' \mid v_1) \cdot \Pr_{\mathbf{y}'_H \leftarrow Y'}[\text{VIEW}(\mathbf{y}'_H) \in V_2] + g(v_1)\Delta(v_1, h) \\
& \qquad + g(v_1) \cdot f(Y \mid v_1) \cdot \delta.
\end{aligned}$$

This concludes our proof.

B Proof of Theorem 4.3

Proof. We write x_r for $x_{n/3,r}$ and $p = 4/9$ for $p_{n/3}$, and set $q = 1 - p = 5/9$. We prove by induction that $x_r \geq 1 - 0.85^r$ for $r > 1$. One can verify explicitly that it holds for $r = 2, 3$. Assume now that it holds for $2, \ldots, r-1$; we prove that it holds for r. Using the recurrence above, we have

$$\begin{aligned}
x_r \quad &= \quad p^2 + q \cdot x_{r-1} + pq \cdot x_{r-2} \\
&\geq \quad p^2 + q \cdot (1 - 0.85^{r-1}) + pq \cdot (1 - 0.85^{r-2}).
\end{aligned}$$

Then it suffices to show that $p^2 + q \cdot (1 - 0.85^{r-1}) + pq \cdot (1 - 0.85^{r-2}) \geq 1 - 0.85^r$. This is because

$$\begin{aligned}
& p^2 + q \cdot (1 - 0.85^{r-1}) + pq \cdot (1 - 0.85^{r-2}) - 1 + 0.85^r \\
= \quad & p^2 + q + pq - 1 - q \cdot 0.85^{r-1} - pq \cdot 0.85^{r-2} + 0.85^r \\
= \quad & 0.85^r - q \cdot 0.85^{r-1} - pq \cdot 0.85^{r-2} \\
= \quad & 0.85^{r-2} \cdot (0.85^2 - q \cdot 0.85 - pq) > 0.003 > 0,
\end{aligned}$$

where the second equality holds because $p^2 + q + pq - 1 = (p+1)(p+q-1) = 0$. $\square$