Hiding Among the Clones: A Simple and Nearly Optimal Analysis of Privacy Amplification by Shuffling

Vitaly Feldman
Apple

Audra McMillan
Apple
audra_mcmillan@apple.com

Kunal Talwar
Apple
ktalwar@apple.com

Abstract

Recent work of Erlingsson, Feldman, Mironov, Raghunathan, Talwar, and Thakurta [EFMRTT19] demonstrates that random shuffling amplifies differential privacy guarantees of locally randomized data. Such amplification implies substantially stronger privacy guarantees for systems in which data is contributed anonymously [BEMMRLRKTS17] and has lead to significant interest in the shuffle model of privacy [CSUZZ19; EFMRTT19].

We show that random shuffling of n data records that are input to ε_0-differentially private local randomizers results in an $(O((1-e^{-\varepsilon_0})\sqrt{\frac{e^{\varepsilon_0}\log(1/\delta)}{n}}), \delta)$-differentially private algorithm. This significantly improves over previous work and achieves the asymptotically optimal dependence in ε_0. Our result is based on a new approach that is simpler than previous work and extends to approximate differential privacy with nearly the same guarantees. Our work also yields an empirical method to derive tighter bounds the resulting ε and we show that it gets to within a small constant factor of the optimal bound. As a direct corollary of our analysis we derive a simple and asymptotically optimal algorithm for discrete distribution estimation in the shuffle model of privacy. We also observe that our result implies the first asymptotically optimal privacy analysis of noisy stochastic gradient descent that applies to sampling without replacement.

1 Introduction

We consider privacy-preserving data analysis in the local model of differential privacy augmented with a shuffler. In this model, each user sends a locally differentially private report and these reports are then anonymized and randomly shuffled. Systems based on this model were first proposed in [BEMMRLRKTS17]. The authors of [EFMRTT19] showed that random shuffling of inputs to locally private protocols amplifies the privacy guarantee. Thus, when the collection of anonymized reports is viewed in the central model, the privacy guarantees are substantially stronger than the original local privacy guarantees. A similar result was shown for the binary randomized response by Cheu, Smith, Ullman, Zeber, and Zhilyaev [CSUZZ19] who also formalized a related *shuffle model of privacy*.

The analysis in [EFMRTT19] relies on a more general result referred to as *privacy amplification by shuffling*. This result shows that privacy is amplified when the inputs are shuffled *before* applying local randomizers and holds even when local randomizers are chosen sequentially and adaptively. Allowing adaptive choice of local randomizers is necessary for analyzing iterative optimization algorithms such as stochastic gradient descent.

A key limitation of the amplification result in [EFMRTT19] is that it has sub-optimal dependence on the local privacy parameter ε_0 when $\varepsilon_0 > 1$. Specifically, in this regime the resulting central privacy parameter is $\tilde{O}(e^{3\varepsilon_0}/\sqrt{n})$[1]. In addition, the result in [EFMRTT19] does not apply to local randomizers that satisfy only approximate differential privacy. For the simpler setting in which the local randomizer is fixed (and, in particular, non-adaptive) Balle, Bell, Gascón, and Nissim [BBGN19] give a stronger bound of $\tilde{O}(e^{\varepsilon_0}/\sqrt{n})$ using a different proof approach. These results are in contrast to the binary randomized response for which the bound is just $\tilde{O}(e^{\varepsilon_0/2}/\sqrt{n})$ [CSUZZ19]. This change in the exponent is significant in the regime where $\varepsilon_0 > 1$, which is commonly used in practice. Further, this regime naturally arises when the target central ε is greater than $1/\sqrt{n}$.

In this paper we give a new analysis of privacy amplification by shuffling. Specifically, we show that running an adaptive sequence of arbitrary ε_0-DP local randomizers on a uniformly random permutation of n data items, yields an (ε, δ)-DP algorithm, where

$$\varepsilon = \Theta\left((1 - e^{-\varepsilon_0})\frac{\sqrt{e^{\varepsilon_0}\log(1/\delta)}}{\sqrt{n}}\right).$$

When $\varepsilon_0 > 1$, this improves the dependence on ε_0 from $e^{2.5\varepsilon_0}$ to asymptotically correct dependence of $e^{\varepsilon_0/2}$ which was previously known only for the binary randomized response [CSUZZ19]. The bound matches the existing bounds in the regime when $\varepsilon_0 < 1$ and is asymptotically optimal in both regimes. We then extend this analysis to approximately differentially private local randomizers with essentially the same guarantees. The best previous guarantee in this case has dependence of $e^{20\varepsilon_0}$ given in [BKMTT20].

Importantly, our proof is also simpler than the rather delicate approaches in prior work [EFMRTT19; BBGN19]. The proof reduces the problem of analyzing the shuffling privacy guarantee of an adaptive series of local algorithms to analyzing the shuffling privacy guarantee of a simple non-adaptive local algorithm with three outputs. Intuitively, we argue that for any local randomizer $\mathcal{A}$ and two data points x, y, $\mathcal{A}(y)$ can be seen as sampling from the same distribution as $\mathcal{A}(x)$ with some positive probability. That is, each data point can create a clone of the output of $\mathcal{A}(x)$ with some probability. For the $\varepsilon_0 > 1$ regime, the desired privacy guarantees then follow easily from the number of clones that x has to hide among being distributed as a binomial random variable. In the small ε_0-regime we also rely on the fact that a local randomizer on two inputs can be seen as the composition of binary randomized response with a post-processing step [KOV16]. Importantly, our proof provides a simple and efficient method for empirically computing an amplification bound that is tighter than our theoretical statements. It also extends naturally to approximate differential privacy. The extension to approximate DP involves an additional technical lemma (Lemma 3.7) that states that any *deletion* (as opposed to replacement) $(\varepsilon_0, \delta_0)$-DP local randomizer $\mathcal{A}$ can be converted to a deletion ε_0-DP local randomizer $\mathcal{A}'$ such that $\mathrm{TV}(\mathcal{A}(x), \mathcal{A}'(x)) \leq \delta_0$ for all x.

Our privacy amplification result with the optimal dependence on ε_0 has some surprising consequences. In certain important settings, it allows us to design algorithms that achieve optimal trade-offs between privacy and utility *simultaneously* in the central *and* the local models of privacy. While the local ε_0 may sometimes be large (we'll often need ε_0 to grow logarithmically with n to achieve a desired constant central ε guarantee), it provides an additional layer of protection that does not exist in the central model. Moreover, this shuffling result implies that a secure implementation

[1] The dependence on ε_0 was recently sharpened to $e^{2.5\varepsilon_0}$ in [BKMTT20].

Algorithm/Setting	Previous results	This work
Binary randomized response	$e^{\varepsilon_0/2}/\sqrt{n}$ [CSUZZ19]	$e^{\varepsilon_0/2}/\sqrt{n}$ [Thm. 3.1]
Non-adaptive shuffle	$e^{\varepsilon_0}/\sqrt{n}$ [BBGN19]	
Adaptive shuffle	$e^{3\varepsilon_0}/\sqrt{n}$ [EFMRTT19] $e^{2.5\varepsilon_0}/\sqrt{n}$ [BKMTT20]	
Adaptive shuffle for $(\varepsilon_0, \delta_0)$-DP	$e^{20\varepsilon_0}/\sqrt{n}$ [BKMTT20]	$(e^{\varepsilon_0/2}/\sqrt{n}, \delta + n\delta_0)$ [Thm. 3.8]
k-Randomized response	$e^{\varepsilon_0}/\sqrt{n}$ [BBGN19]	$\min\{e^{\varepsilon_0/2}/\sqrt{n}, e^{\varepsilon_0}/\sqrt{nk}\}$ [Cor. 4.2]

Table 1: Previous and our results for privacy amplification by shuffling in the $\varepsilon_0 > 1$ regime. We suppress constant factors, as well as the $\sqrt{\ln(1/\delta)}$ factor that is common to all results.

of shuffling, or any function that is a post-processing of shuffling (e.g. vector summation), suffices to ensure the strong central privacy guarantee without having to assume a trusted curator. This is a considerably simpler task than using secure multiparty computation for computing the output of the central DP algorithm [DKMMN06]. Additionally, the fact that the privacy depends only on the LDP property of the local randomizer, and not on the specific noise distribution, allows us to do arbitrary post-processing (e.g. rounding, lossy compression) to the LDP responses before they are shuffled. The privacy of the aggregate relies only on the shuffling result, and we do not need to analyze the effect of the rounding/truncation as in several previous works that use distributed noise addition for a similar goal.

In Section 5.1, we show an example of such an algorithm for the problem of estimating a distribution over an alphabet of size k, from samples. For a given level of accuracy, this algorithm simultaneously achieves optimal central as well as local differential privacy bounds. This algorithm also enjoys low communication. We remark that getting the correct bound here requires taking ε_0 up to $\ln k$. Thus using the $e^{\varepsilon_0}/\sqrt{n}$ amplification bound from [BBGN19] would give a result that is off by a factor of $\sqrt{k}$ in terms of the central privacy-utility tradeoff. In particular, the correct dependence on ε_0 in the exponent is crucial to get this bound.

A second example in Section 5.2 relies on the fact the our results holds for an adaptive sequence of local queries. We analyze the privacy of noisy stochastic gradient descent when run on a random permutation of the data. This bridges a disconnect between the prior theoretical analyses that assumed random sampling with replacement, and the practical implementations that use permutations.

Finally, we give an alternative analysis of privacy amplification by shuffling that can be tighter for specific LDP randomizers. In particular, we can use this to give a tighter analysis of k-ary randomized response. To the best of our knowledge, this is the first theoretical result that shows that the privacy amplification improves with k, corroborating empirical results from [BBGN19].

2 Background and Preliminaries

Differential privacy (DP) is a measure of stability of a randomized algorithm. It bounds the change in the distribution on the outputs when one of the inputs is replaced with an arbitrary other element. The most common way to measure the change in the output distribution is (ε, δ)-indistinguishability. Two random variables P and Q over some probability space are (ε, δ)-*indistinguishable* if for all events E over that probability space,

$$e^{-\varepsilon}(\Pr(Q \in E) - \delta) \leq \Pr(P \in E) \leq e^{\varepsilon} \Pr(Q \in E) + \delta.$$

The following *hockey-stick* divergence can be used to characterize (ε, δ)-indistinguishability:

$$D_{e^\varepsilon}(P\|Q) = \int \max\{0, P(x) - e^{\varepsilon}Q(x)\}dx,$$

where we use the notation P and Q to refer to both the random variables and their probability density functions. So P and Q are (ε, δ)-indistinguishable if $\max\{D_{e^\varepsilon}(P\|Q), D_{e^\varepsilon}(Q\|P)\} \leq \delta$. We will rely on several standard properties of the hockey-stick divergence such as convexity and preservation under post-processing [DR14].

We will consider several models of differentially private data analysis. In the central model, introduced in [DMNS06], the data of the individuals is held by the curator. The curator is then trusted to perform data analysis whose output does not disclose too much about any particular individual's data. While this model requires a higher level of trust than the local model, it is possible to design significantly more accurate algorithms. We say that two databases are neighboring if they differ on the data of a single individual.

Definition 2.1 (Differential privacy in the central model)**.** An algorithm $\mathcal{A} : \mathcal{D}^n \to \mathcal{S}$ is (ε, δ)-*differentially private* if for all neighboring databases X and X', $\mathcal{A}(X)$ and $\mathcal{A}(X')$ are (ε, δ)-indistinguishable.

In the local model, formally introduced in [KLNRS08], each individual (or client) randomizes their data before sending it to data curator (or server). This means that individuals are not required to trust the curator. Due to this minimal trust model, most current industrial deployments of differential privacy rely on local differential privacy [EPK14; App17; DKY17; EFMRSTT20]. More formally, in the general model of local DP clients holding their data can communicate with the server in an arbitrary order with multiple rounds of interaction. The protocol is said to satisfy local (ε, δ)-differential privacy if the transcripts of the protocol on any pair of neighboring datasets are (ε, δ)-indistinguishable. We will not require this general definition as we will only be considering protocols in which each client receives at most one message from the server and sends a single message to the server (which may depend on the message from the server). In this setting, the condition reduces to requiring that the algorithm the client uses to respond to the server satisfies differential privacy with respect to the input of the client, such an algorithm is often referred to as a *local randomizer*.

Definition 2.2 (Local randomizer)**.** An algorithm $\mathcal{R} \colon \mathcal{D} \to \mathcal{S}$ is (ε, δ)-DP *local randomizer* if for all pairs $x, x' \in \mathcal{D}$, $\mathcal{R}(x)$ and $\mathcal{R}(x')$ are (ε, δ)-indistinguishable.

Let $\mathcal{D}$ denote the domain. A single pass ε_0-DP local protocol can be equivalently described as a sequence of algorithms $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) where the i-th client returns $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_i)$, and $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an (ε, δ)-DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. The dependence on $z_{1:i-1}$ models the fact that the server can communicate to the client i any information based on the messages from clients $1, \ldots, i-1$.

The shuffle model of privacy is a distributed model of computation in which, as in the local model, clients hold their data and a server communicates with the clients to perform data analysis. In addition, the model includes a shuffler (also referred to as a mixnet). The shuffler collects all the reports sent by clients and anonymizes them by applying a random permutation to all the reports. The shuffled reports are then released to the server. Note that the reports are typically encrypted in a way that they can be decrypted by the server but not by the shuffler (a more detailed discussion of the trust assumptions can be found in [EFMRSTT20]). A server can also communicate back to the clients and the protocol may consist of multiple rounds of communication via a shuffler. Such a protocol is said to satisfy (ε, δ)-differential privacy in the shuffle model if the outputs of the shuffler on any pair of neighboring datasets are (ε, δ)-indistinguishable. We remark, that such protocols do not necessarily satisfy local differential privacy.

The anonymization of local reports to improve differential privacy was proposed in [BEMMRLRKTS17], who designed and implemented a principled systems architecture for shuffling. The intuition was formalized in [EFMRTT19; CSUZZ19]. Erlingsson et al. [EFMRTT19] showed that for arbitrary ε_0-DP local randomizers random shuffling of reports amplifies the privacy guarantees. Cheu et al. [CSUZZ19] formally defined the shuffle model of computation and analyzed the privacy guarantees of the binary randomized response in this model. It is important to note that, as in [EFMRTT19], we analyze privacy amplification that results from shuffling of the data before applying local randomizers. In contrast, in the shuffle model the shuffler is applied to reports from the clients. To apply our analysis to the shuffle model, it suffices to observe that for a fixed local randomizer, shuffling of the randomized responses is distributed in the same way as randomized reports on shuffled data. In particular, it enjoys the privacy guarantees that we establish.

In all models, if $\delta = 0$ then we will refer to an algorithm as ε-differentially private. Further, we will occasionally refer to $\delta = 0$ as pure differentially private and $\delta > 0$ as approximate differential privacy.

Subsampling is a well known technique for amplifying privacy guarantees [KLNRS08]. That is, suppose that $\mathcal{A}$ is an (ε, δ)-DP algorithm over datasets of size m and define an algorithm $\mathcal{A}'$ over a dataset X of size $n > m$ by picking a random and uniform subset of m elements from X and running $\mathcal{A}$ on the resulting dataset. Privacy amplification by subsampling states that $\mathcal{A}'$ is $\left(\log(1 + \frac{m}{n}(e^\varepsilon - 1)), \frac{m}{n}\delta\right)$-DP [KLNRS08; BBG20]. The following lemma is a slightly more general and abstract version of this result.

Lemma 2.3 (Advanced joint convexity, [BBG20]). *Let P and Q be distributions satisfying $P = (1-q)P_0 + qP_1$ and $Q = (1-q)P_0 + qQ_1$ for some $q \in [0,1]$ and distributions P_0, P_1 and Q_1. Given $\alpha \geq 1$, let $\alpha' = 1 + q(\alpha - 1)$ and $\theta = \alpha'/\alpha$. Then the following holds*

$$D_{\alpha'}(P\|Q) = qD_{\alpha}(P_1\|(1-\theta)P_0 + \theta Q_1).$$

In particular, for any $\varepsilon > 0$, if $\varepsilon' = \log(1 + q(e^{\varepsilon} - 1))$ then

$$D_{e^{\varepsilon'}}(P\|Q) \leq q \max\{D_{e^{\varepsilon}}(P_1\|P_0), D_{e^{\varepsilon}}(P_1\|Q_1)\}.$$

3 Improved Guarantees for Privacy Amplification by Shuffling

In this section we present our main theorem, an improved upper bound for privacy amplification via shuffling for adaptive ε_0-DP randomizers. Theorem 3.1 provides a clean statement of our improved privacy amplification guarantee, improving on previous bounds in the high ε_0 regime. The main technical statement is contained in Theorem 3.1.

Theorem 3.1. *For any domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an ε_0-DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}} : \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a uniform random permutation π over $[n]$, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Then for any $\delta \in [0,1]$ such that $\varepsilon_0 \leq \log(\frac{n}{16\log(2/\delta)})$, $\mathcal{A}_{\mathrm{s}}$ is (ε, δ)-DP, where*

$$\varepsilon \leq \log\left(1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1}\left(\frac{8\sqrt{e^{\varepsilon_0}\log(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_0}}{n}\right)\right) \tag{1}$$

Note that when $\varepsilon_0 > 1$, $\varepsilon = O\left(\frac{\sqrt{e^{\varepsilon_0}\log(1/\delta)}}{\sqrt{n}}\right)$ and when $\varepsilon_0 \leq 1$, $\varepsilon = O\left(\varepsilon_0\frac{\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$. In addition to the theoretical upper bound given in Theorem 3.1, our proof also provides an efficient method for empirically computing a tighter upper bound. We will demonstrate this empirical approach in Section 6.

Now let us turn to the proof of Theorem 3.1. As mentioned earlier, our proof involves reducing the problem of analyzing the privacy guarantee of an adaptive series of local randomizers applied to shuffled data to analyzing the privacy guarantee of a simple non-adaptive local algorithm with three outputs. Suppose that X_0 and X_1 are neighbouring databases that differ on the first datapoint, $x_1^0 \neq x_1^1$. The reduction is outlined in Figure 1, which shows the behavior of the simpler algorithm on data set X_0. The key observation is that for any ε_0-DP local randomizer $\mathcal{R}$ and data point x, $\mathcal{R}(x)$ can be seen as sampling from the same distribution as $\mathcal{R}(x_1^0)$ with probability at least $e^{-\varepsilon_0}/2$ and sampling from the same distribution as $\mathcal{R}(x_1^1)$ with probability at least $e^{-\varepsilon_0}/2$. That is, with probability $e^{-\varepsilon_0}$ each data point can create a clone of the output of $\mathcal{R}(x_1^0)$ or a clone of $\mathcal{R}(x_1^1)$ with equal probability. Thus $n-1$ data elements effectively produce a random number of clones of both x_1^0 and x_1^1. These clones make distinguishing whether the original data set contains x_1^0 or x_1^1 as its first element much harder.

The proof of Theorem 3.1 will require an additional step where we observe that if $\mathcal{R}$ is ε_0-DP then $\mathcal{R}(x_1^0)$ and $\mathcal{R}(x_1^1)$ are similar, and thus privacy is further amplified. Let us start by proving Lemma 3.3, which captures just the privacy amplification due to the generation of clones. The following lemma analyses the divergence between the distributions on the number of clones that result from this reduction.

Lemma 3.2. *Let $p \in [0,1]$ and $n \in \mathbb{N}$ be such that $p \geq \frac{16\ln(2/\delta)}{n}$. Consider the process where we first sample $C \sim \mathrm{Bin}(n-1, p)$ then $A \sim \mathrm{Bin}(C, 1/2)$. Then if $P = (A+1, C-A)$ and $Q = (A, C-A+1)$, then P and Q are $(\log(1 + \frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}), \delta)$- indistinguishable.*

Intuitively, when C is fixed, we are bounding the hockey-stick divergence between a binomial random variable A, and $A+1$. Since C is likely to be around pn, we should expect a dependence like the one above. We defer the actual proof to Appendix A.

For notational brevity, we will frequently use the same symbol to refer to both a random variable and it's probability density function. In particular, if X and Y are random variables and $\alpha \in [0,1]$ then $Z = \alpha X + (1-\alpha)Y$ denotes the random variable that samples from X with probability α and Y with probability $1-\alpha$. Consequently, for a randomized algorithm $\mathcal{A}$ and input x, we treat the output $\mathcal{A}(x)$ as a random variable and also use $\mathcal{A}(x)$ to refer to the probability density function of this random variable.

Lemma 3.3 describes our reduction from the original problem to analysis of shuffling for a simple non-adaptive protocol. Specifically, it shows that if each local randomizer $\mathcal{R}$ we apply can be decomposed into a mixture of $\mathcal{R}(x_1^0)$, $\mathcal{R}(x_1^1)$ and some "left-over" distribution $\mathtt{LO}(x)$ then we can apply our reduction. We will then conclude the proof of Theorem 3.1 by observing that that every differentially private randomizer has this property.

Lemma 3.3. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an ε_0-DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}} : \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a permutation π uniformly at random, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Let $X_0 = (x_1^0, x_2, \ldots, x_n)$ and $X_1 = (x_1^1, x_2, \ldots, x_n)$ be two neighboring datasets such that for all $j \neq 1$, $x_j \notin \{x_1^0, x_1^1\}$. Suppose that there exists a positive value $p \in (0,1]$ such that for all $i \in [n]$, $x \in \mathcal{D} \setminus \{x_1^0, x_1^1\}$ and $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$, there exists a distribution $\mathtt{LO}^{(i)}(z_{1:i-1}, x)$ such that*

$$\mathcal{R}^{(i)}(z_{1:i-1}, x) = \frac{p}{2}\mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2}\mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) + (1-p)\mathtt{LO}^{(i)}(z_{1:i-1}, x). \tag{2}$$

Then for any $\delta \in [0,1]$ such that $\varepsilon_0 \leq \log(\frac{n}{16\log(2/\delta)})$, $\mathcal{A}_{\mathrm{s}}(X_0)$ is (ε, δ)-indistinguishable from $\mathcal{A}_{\mathrm{s}}(X_1)$, where

$$\varepsilon \leq \log\left(1 + \frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right)$$

Proof. The proof of the above lemma relies on a decomposition of the algorithm that shuffles the data and then applies the local randomizers, to an algorithm in which each client first reports which component of the mixture it will sample from, then applies shuffling to these reports and finally applies a post-processing step in which randomizers are applied according to the shuffled mixture component indices. The first part of the algorithm is non-adaptive and its analysis reduces to the bound on the divergence we proved in Lemma 3.2. The result then follows by using the fact that post-processing does not increase the hockey-stick divergence.
Formally, define a random variable Y as follows

$$Y = \begin{cases} 0 & \text{w.p. } p/2 \\ 1 & \text{w.p. } p/2 \\ 2 & \text{w.p. } 1-p \end{cases}.$$

Given a dataset X_b for $b \in \{0,1\}$ we generate n samples from $\{0,1,2\}$ in the following way. Client number one (holding the first element of the dataset) reports b. Clients $2, \ldots, n$ each report an independent sample from Y. We then shuffle the reports randomly. Let ρ_b denote the resulting distribution over $\{0,1,2\}^n$.

We claim that there exists a post-processing function f (that depends on $x_1^0, x_1^1, x_2, \ldots, x_n$) such that for y sampled from ρ_b, $f(y)$ is distributed identically to $\mathcal{A}_{\mathrm{s}}(X_b)$. To see this, consider the following process that takes a $b \in \{0,1\}$ as an input and defines a jointly distributed pair of random variables (z, y) over $\mathcal{S} \times \{0,1,2\}^n$, where $\mathcal{S} = \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ is the output space of $\mathcal{A}_{\mathrm{s}}$. Let π be a randomly and uniformly chosen permutation of $[n]$. For every $i \in [n]$, if $\pi(i) \neq 1$ then we first sample y_i according to Y, and if $i = 1$, we set $y_i = b$. We then use the y_i's to generate z_i's. Specifically,

$$z_i \sim \begin{cases} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) & \text{if } y_i = 0; \\ \mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) & \text{if } y_i = 1; \\ \mathtt{LO}^{(i)}(z_{1:i-1}, x_{\pi(i)}) & \text{if } y_i = 2. \end{cases}$$

By our assumption, this produces a sample z_i from $\mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$. It is easy to see that the resulting random variable (z, y) has the property that for input $b \in \{0,1\}$ its marginal distribution over $\mathcal{S}$ is the same as $\mathcal{A}_{\mathrm{s}}(X_b)$ and marginal distribution over $\{0,1,2\}^n$ is ρ_b.

We can now show that for every fixed value $v \in \{0,1,2\}^n$ and $b \in \{0,1\}$, one can generate a sample from the distribution of z conditioned on $y = v$ without knowing b from which the pair (z, y) was generated. First observe that the above construction of (y, z, π) has the property that for any permutation σ, conditioned on $(y = v, \pi = \sigma)$, the random variable z does not depend on b. A natural approach to sample from $z|_{y=v}$ would be to sample a permutation σ from the distribution of π conditioned on $y = v$, and then sample $z|_{(y,\pi)=(v,\sigma)}$. The conditional distribution of $\pi|_{y=v}$ however is not independent of b. Let $T = \pi(\{i : y_i = 2\})$ be the indices corresponding to $v_i = 2$. First observe that $T|_{y=v}$ is independent of b, since this distribution is uniform over subsets of $\{2, \dots, n\}$ of the appropriate size. Finally, note that the sampling of z given y only needs T. Thus we can sample from $z|_{(y,T)=(v,J)}$ without knowing b. This conditional sampling is exactly the post-processing step that we claimed. We include a formal description of the post-processing step in Algorithm 1.

We now analyze the divergence between ρ_0 and ρ_1. First, the shuffling step implies that $p(X)$ is symmetric. That is, the probability of any sequence $y \in \{0,1,2\}^n$ depends only on the total number of 0s and 1s in the sequence. This implies that the divergence between ρ_0 and ρ_1 is equal to the divergence between the distribution of the counts of 0's and 1's. Let $(c_0, c_1) = (\sum_{i\in[n]} \mathbb{1}_{y_i=0}, \sum_{i\in[n]} \mathbb{1}_{y_i=1})$. It now suffices to observe that, for ρ_0 and ρ_1 the counts are distributed exactly as the random variables we studied in Lemma 3.2; if $C \sim \mathrm{Bin}(n-1, p)$ and $A \sim \mathrm{Bin}(C, 1/2)$ then the counts for $y \sim \rho_0$ are distributed as $(1 + A, C - A)$ and the count for $y \sim \rho_1$ is distributed as $(A, C - A + 1)$. Therefore, by Lemma 3.2 and the post-processing inequality, $\mathcal{A}_{\mathrm{s}}(X_0)$ and $\mathcal{A}_{\mathrm{s}}(X_1)$ are

$$\left(\log\left(1 + \frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right), \delta\right)\text{-indistinguishable.}$$

Algorithm 1: Post-processing function, f

Input: $x_1^0, x_1^1, x_2, \dots, x_n$; $y \in \{0,1,2\}^n$
$J := \emptyset$
for $i = 1, \dots, n$ **do**
  if $y_i = 2$ **then**
    Let j_i be a randomly and uniformly chosen element of $[2:n] \setminus J$
    $J := J \cup \{j_i\}$
  Sample z_i from $\begin{cases} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) & \text{if } y_i = 0; \\ \mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) & \text{if } y_i = 1; \\ \mathtt{LO}^{(i)}(z_{1:i-1}, x_{j_i}) & \text{if } y_i = 2. \end{cases}$
return $z_1, \dots, z_n$

□

We can now complete the proof of Theorem 3.1. To do this we will use privacy amplification provided by the clones (Lemma 3.3) together with the observation that for any ε_0-DP randomizer $\mathcal{R}$, $\mathcal{R}(x_1^0)$ and $\mathcal{R}(x_1^1)$ are ε_0-indistinguishable. To fit this into our analysis we will use the fact that $\mathcal{R}(x_1^0)$ and $\mathcal{R}(x_1^1)$ can be viewed as post-processing of a binary randomized response with parameter ε_0 [KOV15] (see also [MV16]).

Lemma 3.4 ([KOV15])**.** *Let $\mathcal{R}\colon \mathcal{D} \to \mathcal{S}$ be an ε_0-DP local randomizer and $x_0, x_1 \in \mathcal{D}$. Then there exists a randomized algorithm $\mathcal{Q}\colon \{0,1\} \to \mathcal{S}$ such that $\mathcal{R}(x_0) = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{Q}(0) + \frac{1}{e^{\varepsilon_0}+1}\mathcal{Q}(1)$ and $\mathcal{R}(x_1) = \frac{1}{e^{\varepsilon_0}+1}\mathcal{Q}(0) + \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{Q}(1)$.*

Binary randomized response can be seen as an algorithm that outputs its input with probability $\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}$ and outputs an unbiased coin flip with probability $\frac{2}{e^{\varepsilon_0}+1}$. Thus it can be seen as providing privacy amplification by subsampling with probability $\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}$. Note that for $\varepsilon_0 < 1$ this probability is $O(\varepsilon_0)$. This will ensure that our bound is accurate in the regime $\varepsilon_0 < 1$.

Proof of Theorem 3.1. Let X_0 and X_1 be neighboring datasets. We can assume without loss of generality that $X_0 = (x_1^0, x_2, \dots, x_n)$ and $X_1 = (x_1^1, x_2, \dots, x_n)$ for $x_1^0 \neq x_1^1$ (since the shuffler applies a random permutation). Further,

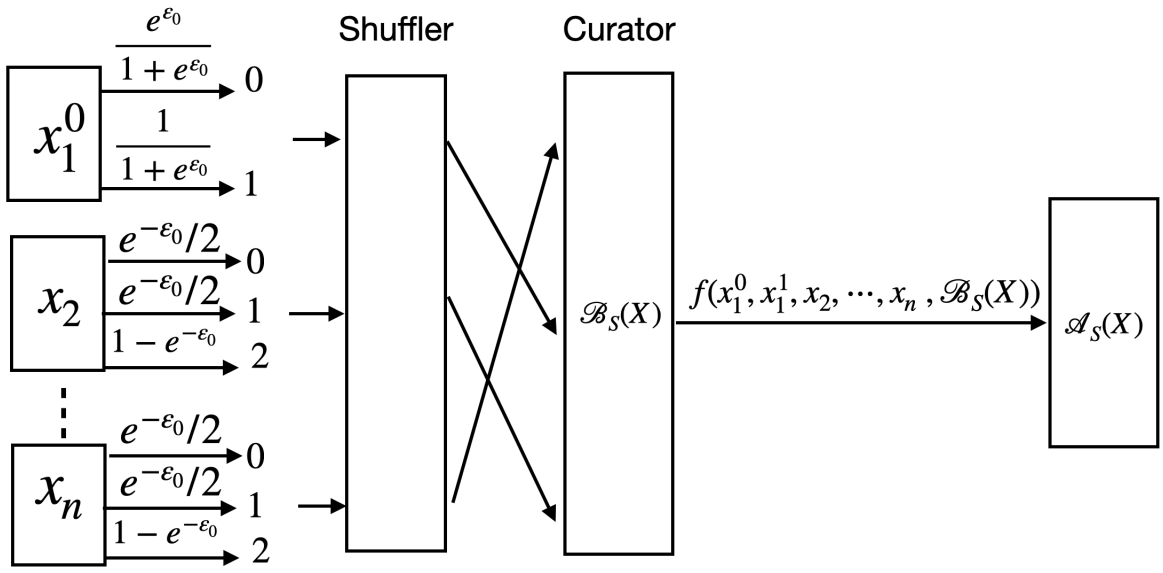


Figure 1: Graphical representation of the reduction in the proof of Theorem 3.1

we can assume without loss of generality that for all $j \neq 1$, $x_j \notin \{x_1^0, x_1^1\}$. This true since we can add elements x'^0_1 and x'^1_1 to the domain $\mathcal{D}$ and define $\mathcal{R}^{(i)}$ on them in the same way as on x_1^0 and x_1^1, respectively. With this definition, for $b \in \{0,1\}$, the output distribution of $\mathcal{A}_{\mathrm{s}}(x'^b_1, x_2, \ldots, x_n)$ is identical to that of $\mathcal{A}_{\mathrm{s}}(X_b)$.

By Lemma 3.4 we have that for every i and value of auxiliary input $z_{1:i-1}$ there exists a randomized algorithm $\mathcal{Q}^{(i)}(z_{1:i-1}, \cdot)\colon \{x_1^0, x_1^1\} \to \mathcal{S}^{(i)}$ such that

$$\mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \frac{1}{e^{\varepsilon_0}+1}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) \tag{3}$$

and

$$\mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) = \frac{1}{e^{\varepsilon_0}+1}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) \tag{4}$$

where again for notational brevity we use the same notation for a random variable and its probability density function.

Next we observe that the definition of ε_0-DP directly implies that for an ε_0-DP randomizer $\mathcal{R}\colon \mathcal{D} \to \mathcal{S}$ and any input $x_0 \in \mathcal{D}$, there exists a randomized algorithm $\mathtt{LO}\colon \mathcal{D} \to \mathcal{S}$ such that $\mathcal{R}(x)$ can be decomposed as

$$\mathcal{R}(x) = \frac{1}{e^{\varepsilon_0}}\mathcal{R}(x_0) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}(x).$$

Applying this argument twice we obtain that, for any pair of inputs $x_0, x_1 \in \mathcal{D}$ there exists a decomposition

$$\mathcal{R}(x) = \frac{1}{2e^{\varepsilon_0}}\mathcal{R}(x_0) + \frac{1}{2e^{\varepsilon_0}}\mathcal{R}(x_1) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}'(x).$$

Applying this observation to $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$, for all $i \in [n]$ and values of $z_{1:i-1}$, we have that there exists a randomized algorithm $\mathtt{LO}^{(i)}$ such that for $p = e^{-\varepsilon_0}$ we have:

$$\mathcal{R}^{(i)}(z_{1:i-1}, x) = \frac{p}{2}\mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2}\mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) + (1-p)\mathtt{LO}^{(i)}(z_{1:i-1}, x).$$

Note that

$$\mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) + \mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) = \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1)$$

and therefore we also obtain that

$$\mathcal{R}^{(i)}(z_{1:i-1}, x) = \frac{p}{2}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) + (1-p)\mathtt{LO}^{(i)}(z_{1:i-1}, x), \tag{5}$$

Next we consider the algorithm $\mathcal{A}_\mathcal{Q}$ which is defined in the same way as $\mathcal{A}_\mathrm{s}$, except $\mathcal{R}^{(i)}(z_{1:i-1}, x_1^b)$ is replaced with $\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^b)$. Formally, we define a randomizer $\mathcal{R}_\mathcal{Q}^{(i)}$ as follows: For all $x \in \mathcal{D}$, $i \in [n]$ and values of $z_{1:i-1}$ we let

$$\mathcal{R}_\mathcal{Q}^{(i)}(z_{1:i-1}, x) = \begin{cases} \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) & \text{if } x = x_1^0; \\ \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) & \text{if } x = x_1^1; \\ \mathcal{R}^{(i)}(z_{1:i-1}, x) & \text{otherwise.} \end{cases}$$

Let $\mathcal{A}_\mathcal{Q}$ be defined in the same way as $\mathcal{A}_\mathrm{s}$, except $\mathcal{R}^{(i)}$ is replaced with $\mathcal{R}_\mathcal{Q}^{(i)}$.

Equations (3) and (4), allow us to decompose $\mathcal{A}_\mathrm{s}(X_0)$ and $\mathcal{A}_\mathrm{s}(X_1)$ into the mixture of two components as follows:

$$\mathcal{A}_\mathrm{s}(X_0) = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{A}_\mathcal{Q}(X_0) + \frac{1}{e^{\varepsilon_0}+1}\mathcal{A}_\mathcal{Q}(X_1) \text{ and } \mathcal{A}_\mathrm{s}(X_1) = \frac{1}{e^{\varepsilon_0}+1}\mathcal{A}_\mathcal{Q}(X_0) + \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{A}_\mathcal{Q}(X_1). \tag{6}$$

To compute the divergence between $\mathcal{A}_\mathcal{Q}(X_0)$ and $\mathcal{A}_\mathcal{Q}(X_1)$ we note that by eq. (5), for all $x \notin \{x_1^0, x_1^1\}$,

$$\mathcal{R}_\mathcal{Q}^{(i)}(z_{1:i-1}, x) = \frac{p}{2}\mathcal{R}_\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2}\mathcal{R}_\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) + (1-p)\mathtt{LO}^{(i)}(z_{1:i-1}, x).$$

Therefore, by Lemma 3.3, $\mathcal{A}_\mathcal{Q}(X_0)$ and $\mathcal{A}_\mathcal{Q}(X_1)$ are

$$\left(\log\left(1 + \frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right), \delta\right)\text{-indistinguishable.}$$

Now, note that if we let $P_0 = \frac{1}{2}(\mathcal{A}_\mathcal{Q}(X_0) + \mathcal{A}_\mathcal{Q}(X_1))$ then we can rewrite Equation (6) as

$$\mathcal{A}_\mathrm{s}(X_0) = \frac{2}{e^{\varepsilon_0}+1}P_0 + \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\mathcal{A}_\mathcal{Q}(X_0) \text{ and } \mathcal{A}_\mathrm{s}(X_1) = \frac{2}{e^{\varepsilon_0}+1}P_0 + \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\mathcal{A}_\mathcal{Q}(X_1).$$

Further, for any $\varepsilon > 0$, by the convexity of the hockey-stick divergence,

$$\max\{D_{e^\varepsilon}(P_0 \| \mathcal{A}_\mathcal{Q}(X_0))), D_{e^\varepsilon}(\mathcal{A}_\mathcal{Q}(X_1)) \| P_0))\} \le D_{e^\varepsilon}(\mathcal{A}_\mathcal{Q}(X_0) \| \mathcal{A}_\mathcal{Q}(X_1))).$$

Thus, by Lemma 2.3, $\mathcal{A}_\mathrm{s}(X_0)$ and $\mathcal{A}_\mathrm{s}(X_1)$ are $(\varepsilon, p\delta)$-indistinguishable where

$$\varepsilon \le \log\left(1 + \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\left(\frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right)\right)$$

□

Remark 3.5. *An equivalent way to summarize our reduction is as showing that for any fixed pair of neighboring datasets X_0 and X_1 the output distributions $\mathcal{A}_\mathrm{s}(X_0)$ and $\mathcal{A}_\mathrm{s}(X_1)$ can be seen as post-processing of two distributions over counts. Specifically, let $C \sim \mathrm{Bin}(n-1, e^{-\varepsilon_0})$, $A \sim \mathrm{Bin}(C, 1/2)$ and $\Delta \sim \mathrm{Bern}\left(\frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\right)$ (where, $\mathrm{Bern}(p)$ denotes a Bernoulli random variable with bias p. Then $\mathcal{A}_\mathrm{s}(X_0)$ and $\mathcal{A}_\mathrm{s}(X_1)$ can be seen as postprocessing of $(A + \Delta, C - A)$ and $(A, C - A + \Delta)$. This makes it straightforward to compute the upper bound on the hockey-stick divergence analytically and empirically (which we do in Section E). This also gives an approach for bounding other notions of privacy of $\mathcal{A}_\mathrm{s}$ (for example Rényi DP [Mir17]) which we leave for future work.*

3.1 Generalization to Approximate Differential Privacy

In order to analyse the approximate DP case, we can use a reduction from $(\varepsilon_0, \delta_0)$-DP local randomizers to ε_0'-DP local randomizers. We use a black box argument showing that for every $(\varepsilon_0, \delta_0)$-DP local randomizer $\mathcal{A}$ there exists an ε_0'-DP local randomizer $\mathcal{A}$ that is δ_0' close in total variation distance to $\mathcal{A}$. The output of n $(\varepsilon_0, \delta_0)$-DP randomizers on shuffled data is then $n\delta_0'$ close is total variation distance to the output of n ε_0'-DP randomizers. A fairly involved reduction of this type is given in [CSUZZ19] (building on the results from [BNS19]) and achieves $\varepsilon_0' = 8\varepsilon_0$ and $\delta_0' = \tilde{O}(\delta e^{\varepsilon_0}/(1 - e^{-\varepsilon_0}))$. Applying this reduction to their shuffling amplification bound for pure DP, Balle et al. [BKMTT20] obtain a resulting ε of order $\Theta\left(\frac{e^{30\varepsilon_0}}{n} + \frac{e^{20\varepsilon_0}\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$ in the high ε_0 regime.

Our bound is based on a simpler and stronger reduction of this type that was also independently shown in [CU20]. Its tightest form is best stated in the add/delete variant of differential privacy which was defined for local randomizers in [EFMRSTT20].

Definition 3.6 (Local Deletion Differential Privacy)**.** An algorithm $\mathcal{R}\colon \mathcal{D} \to \mathcal{S}$ is a *deletion* (ε, δ)*-DP local randomizer* if there exists a reference distribution ρ such that for all data points $x \in \mathcal{D}$, $\mathcal{R}(x)$ and ρ are (ε, δ)-indistinguishable.

We will occasionally refer to a function that satisfies Definition 2.2 as a *replacement* (ε, δ)-DP local randomizer. It is easy to show that a replacement (ε, δ)-DP algorithm is also a deletion (ε, δ)-DP algorithm, and that a deletion (ε, δ)-DP algorithm is also a replacement $(2\varepsilon, 2\delta)$-DP algorithm. The following Lemma allows us to convert (ε, δ)-DP local randomizers to ε-DP local randomizers that are within δ in total variation distance.

Lemma 3.7. *Suppose $\mathcal{R}$ is a deletion (ε, δ)-DP local randomizer with reference distribution ρ. Then there exists a randomizer $\mathcal{R}'$ that is a deletion ε-DP local randomizer with reference distribution ρ, and for all inputs x, $\mathrm{TV}(\mathcal{R}(x), \mathcal{R}'(x)) \le \delta$. In particular, $\mathcal{R}'$ is a (replacement) 2ε-DP local randomizer.*

The proof can be found in Appendix B. We note that when this transformation is applied to an (ε, δ)-DP local randomizer that is also deletion $(\varepsilon/2, \delta)$-DP (such as, for example, addition of isotropic Gaussian noise to a vector from a Euclidean ball of bounded radius) then the extra factor 2 can be avoided. To avoid incurring this factor of 2 in our amplification by shuffling bound, we apply the core observation from the proof of Lemma 3.7 directly in our proof of the amplification bound. This leads to the following resulting bound which we prove in Appendix B.

Theorem 3.8. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)}\colon f \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a $(\varepsilon_0, \delta_0)$-DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}}\colon \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a uniformly random permutation π, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Then for any $\delta \in [0, 1]$ such that $\varepsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$, $\mathcal{A}_{\mathrm{s}}$ is $(\varepsilon, \delta + (e^{\varepsilon} + 1)(1 + e^{-\varepsilon_0}/2)n\delta_0)$-DP, where ε is as in Equation (1).*

4 A Tighter Analysis for Specific Randomizers

In this section we give a more refined analysis of the amplification by shuffling that can exploit additional properties of the local randomizer and lead to improved privacy amplification results for specific randomizers. In particular, Theorem 4.1 will immediately imply a privacy amplification by shuffling result for k randomized response (`kRR`). To our knowledge, this is the first theoretical result that shows that privacy amplification improves as k increases, corroborating empirical results from [BBGN19]. As with Theorem 3.1, Theorem 4.1 also gives us a method for empirically computing the privacy guarantee.

We will show the amplification bound is stronger if, in addition, $\mathcal{R}^{(i)}(z_{1:i-1}, x_1^0)$ and $\mathcal{R}^{(i)}(z_{1:i-1}, x_1^1)$ are close in total variation distance, specifically when the total variation distance is less than $2/(e^{\varepsilon_0} + 1)$ (which is implied by $\mathcal{R}^{(i)}$ being ε_0-DP).

We first state the most general form of this amplification theorem.

Theorem 4.1. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a ε_0-DP local randomizer for all values of*

auxiliary inputs $z_{1,i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. *Let* $\mathcal{A}_{\mathrm{s}} : \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ *be the algorithm that given a dataset* $x_{1:n} \in \mathcal{D}^n$*, samples a uniformly random permutation* π*, then sequentially computes* $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ *for* $i \in [n]$ *and outputs* $z_{1:n}$*. Let* $X_0 = (x_1^0, x_2, \ldots, x_n)$ *and* $X_1 = (x_1^1, x_2, \ldots, x_n)$ *be two neighboring datasets such that for all* $j \neq 1$*,* $x_j \notin \{x_1^0, x_1^1\}$*. Suppose that there exist positive values* $p, q \in (0, 1]$ *such that for all* $i \in [n]$ *and* $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$*, there exists distributions* $\mathcal{Q}_0^{(i)}(z_{1:i-1})$*,* $\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^0)$ *and* $\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^1)$ *such that for all* $b \in \{0, 1\}$*,*

$$\mathcal{R}^{(i)}(z_{1:i-1}, x_1^b) = (1-q)\mathcal{Q}_0^{(i)}(z_{1:i-1}) + q\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b) \tag{7}$$

and for all $x \in \mathcal{D} \backslash \{x_1^0, x_1^1\}$*, there exists a distributions* $\mathtt{LO}_0^{(i)}(z_{1:i-1}, x)$ *and* $\mathtt{LO}_1^{(i)}(z_{1:i-1}, x)$ *such that for all* $b \in \{0, 1\}$

$$\mathcal{R}^{(i)}(z_{1:i-1}, x) = p\mathcal{Q}_0^{(i)}(z_{1:i-1}) + p\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b) + (1-2p)\mathtt{LO}_b^{(i)}(z_{1:i-1}, x). \tag{8}$$

Then for any $\delta \in [0, 1]$ *such that* $\varepsilon_0 \leq \log(\frac{n}{16\log(2/\delta)})$*,* $\mathcal{A}_{\mathrm{s}}(X_0)$ *is* $(\varepsilon, q\delta)$*-indistinguishable from* $\mathcal{A}_{\mathrm{s}}(X_1)$*, where*

$$\varepsilon \leq \log\left(1 + q\left(\frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right)\right). \tag{9}$$

The proof of Theorem 4.1 relies on decomposing $\mathcal{A}_{\mathrm{s}}$ into a mixture of two distributions based on the decomposition in eq.(7). We then apply Lemma 3.3 twice to argue that the components of the mixture are (ε', δ')-indistinguishable and also that the second component of the mixture for $\mathcal{A}_{\mathrm{s}}(X_0)$ is $(\varepsilon'', \delta'')$-indistinguishable from the second component of the mixture for $\mathcal{A}_{\mathrm{s}}(X_1)$. Joint convexity can then be used to obtain the bound. The details of the proof appear in Appendix C.

The main strength of Theorem 4.1 is that it allows us to prove tighter bounds than Theorem 3.1 for specific local randomizers. In the next section, we will show how Theorem 4.1 can be used to provide a tighter analysis of k randomized response. However, we can also use it to rederive the bound from Theorem 3.1, although with a slightly worse constant factor. The decomposition in Lemma 3.4 implies that for any ε_0-DP local randomizer Equation (7) holds with $q = \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}$. That is, given a sequence of $\mathcal{R}^{(i)}$ of ε_0-DP local randomizers, and neighbouring datasets X_0 and X_1 such that $x_0^0 \neq x_1^1$, there exist distributions $\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0)$ and $\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1)$ that satisfy equations (3) and (4). Thus, by setting

$$\mathcal{Q}_0^{(i)}(z_{1:i-1}) = \frac{1}{2}\Big(\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1)\Big) \text{ and } \mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b) = \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^b)$$

we get that for $b \in \{0, 1\}$ and $q = \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}$,

$$\mathcal{R}^{(i)}(z_{1:i-1}, x_1^b) = (1-q)\mathcal{Q}_0^{(i)}(z_{1:i-1}) + q\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b).$$

Further, Equation (5) implies that for any $x \in \mathcal{D}$ and $b \in \{0, 1\}$

$$\begin{aligned}
\mathcal{R}^{(i)}(z_{1:i-1}, x) &= \frac{1}{2e^{\varepsilon_0}}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \frac{1}{2e^{\varepsilon_0}}\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}^{(i)}(z_{1:i-1}, x) \\
&= \frac{1}{3e^{\varepsilon_0}}\mathcal{Q}_0^{(i)}(z_{1:i-1}) + \frac{1}{3e^{\varepsilon_0}}\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b) + \frac{1}{3e^{\varepsilon_0}}\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^{1-b}) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}^{(i)}(z_{1:i-1}, x).
\end{aligned}$$

Thus by setting

$$\mathtt{LO}_b^{(i)}(z_{1:i-1}, x) = \frac{1}{1 - \frac{2}{3e^{\varepsilon_0}}}\left(\frac{1}{3e^{\varepsilon_0}}\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^{1-b}) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}^{(i)}(z_{1:i-1}, x)\right)$$

we obtain that Equation (8) holds with $p = e^{-\varepsilon_0}/3$. Plugging $q = \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}$ and $p = e^{-\varepsilon_0}/3$ in Equation (9) gives a bound that matches the bound stated in Equation (1) of Theorem 3.1 up to a factor of $\sqrt{3/2}$.

4.1 k-Randomized Response

For any $k \in \mathbb{N}$ and $\varepsilon_0 > 0$, the k-randomized response $\mathtt{kRR}\colon [k] \to [k]$ is defined as

$$\mathtt{kRR}(x) = \begin{cases} x & \text{with probability } \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+k-1} \\ y \sim \mathcal{U}_{[k]} & \text{with probability } \frac{k}{e^{\varepsilon_0}+k-1} \end{cases},$$

where $\mathcal{U}_{[k]}$ is the uniform distribution over $[k]$. That is, with probability $\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+k-1}$ the true data point is reported, and otherwise a random value is reported.

Corollary 4.2. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a ε_0-DP local randomizer for all values of auxiliary inputs $z_{1,i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}} : \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a uniformly random permutation π, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Assume that for some $k \in \mathbb{N}$ we have that for all $i \in [n]$, there exists a function $f^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to [k]$ such that $\mathcal{R}^{(i)}(z_{1:i-1}, x) = kRR(f^{(i)}(z_{1:i-1}, x))$. Then for $\delta \in [0, 1]$ such that such that $\varepsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$, $\mathcal{A}_{\mathrm{s}}$ is (ε, δ)-DP where*

$$\varepsilon \le \log\left(1 + (e^{\varepsilon_0} - 1)\left(\frac{8\sqrt{(k+1)\log(4/\delta)}}{\sqrt{(e^{\varepsilon_0}+k-1)kn}} + \frac{8(k+1)}{kn}\right)\right) \tag{10}$$

Notice that when k is small, this bound matches that given in Theorem 3.1, but when k and ε_0 are large, ε scales like $\frac{e^{\varepsilon_0}}{\sqrt{kn}}$. The proof of Corollary 4.2 can be found in Appendix D. The key observation is that if we let ν be the uniform distribution on $[k]$, then for any $x \in [k]$,

$$\mathtt{kRR}(x) = \frac{k}{e^{\varepsilon_0}+k-1}\nu + \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+k-1}\mathbb{1}_x,$$

where $\mathbb{1}_x$ is the distribution that always outputs x.

5 Applications

5.1 Distribution Estimation

In this section we show how our amplification result immediately implies an algorithm for distribution estimation in the shuffle model with nearly optimal utility/privacy trade-off in both the local and the central models and low communication. For an integer k a discrete distribution over $[k]$ is described by a vector p such that $\sum_{i\in[k]} p_i = 1$ and $p_i \ge 0$ for all $i \in [k]$. We also write $x \sim p$ to say that x is drawn from the distribution defined by p. We say that an algorithm $\mathcal{A}$ achieves ℓ_2 error of α for distribution estimation if for all distributions over p, the algorithm given n i.i.d. samples from p outputs a vector $\hat{p}$ such that

$$\mathbb{E}[\|p - \hat{p}\|_2],$$

where the expectation is over the randomness of samples and the algorithm.

The problem of distribution estimation is closely related to the problem of frequency estimation or a histogram with essentially the only difference being the additional statistical error of $O(\sqrt{1/n})$. In the latter problem the goal is to estimate the empirical frequencies of elements in the dataset. It is one of the most well-studied problems in private data analysis and the optimal utility/privacy trade-offs for this problem are well-understood [DMNS06] (although the focus is typically on the ℓ_∞ error in the estimation). In particular, there exist an algorithm achieving error of $O(\sqrt{1/n} + \sqrt{k}/(\varepsilon n))$ and this is asymptotically optimal for $\varepsilon \le 1$ [Vad17, Theorem 5.14].

Frequency estimation in the local model was studied in several recent works. In particular, Acharya et al. [ASZ19] give an efficient, low communication ($\log k + 2$ bits per user) ε_0-DP local algorithm that is asymptotically optimal for all ε_0.

Theorem 5.1 ([ASZ19]). *For every positive integer k and $\varepsilon_0 > 0$, there exists an ε_0-LDP protocol for distribution estimation that outputs a vector of frequencies $\hat{p}$ such that*

$$\mathbb{E}[\|p - \hat{p}\|_2] = \begin{cases} O\left(\frac{\sqrt{k}}{\varepsilon_0 \sqrt{n}}\right) & \text{for } \varepsilon_0 \in (0,1); \\ O\left(\frac{\sqrt{k}}{\sqrt{e^{\varepsilon_0} n}} + \frac{1}{\sqrt{n}}\right) & \text{for } \varepsilon_0 \geq 1. \end{cases}.$$

The protocol runs in time $\tilde{O}(n + k)$ and uses a local randomizer that sends a single message of length $\log k + 2$.

Moreover, the bounds achieved by this algorithm are asymptotically optimal [YB18].

Using the fact that our amplification bound has optimal dependence on ε_0, we are able to leverage the local DP result of Acharya, Sun, and Zhang [ASZ19] to immediately obtain a low communication, single round algorithm in the shuffle model whose error matches the optimal error in the central model (up to a $O(\sqrt{\log(1/\delta)})$ factor), as well as the local model.

Theorem 5.2. *For every positive integer k and $\varepsilon, \delta \in (0,1)$, there exists an (ε, δ)-DP protocol for distribution estimation in the shuffle model that outputs a vector $\hat{p}$ of frequencies such that*

$$\mathbb{E}[\|p - \hat{p}\|_2] = O\left(\frac{\sqrt{k \log(1/\delta)}}{\varepsilon n} + \frac{1}{\sqrt{n}}\right).$$

The protocol runs in time $\tilde{O}(n + k)$ and uses a local randomizer that sends a single message of length $\log k + 2$ and satisfies ε_0-LDP for

$$\varepsilon_0 = \begin{cases} O\left(\frac{\varepsilon \sqrt{n}}{\sqrt{\log(1/\delta)}}\right) & \text{for } \varepsilon \leq \sqrt{\log(1/\delta)/n}; \\ \log\left(\frac{\varepsilon^2 n}{\log(1/\delta)}\right) + O(1) & \text{for } \varepsilon \in (\sqrt{\log(1/\delta)/n}, 1). \end{cases}.$$

Proof. For a local randomizer $\mathcal{A}_{\mathrm{ldp}}$, let $\mathcal{A}_{\mathrm{s}}(\mathcal{A}_{\mathrm{ldp}})$ be the algorithm that given a data set $x_{1:n} \in \mathcal{D}^n$, samples a uniform random permutation π over $[n]$, then outputs $(\mathcal{A}_{\mathrm{ldp}}(x_{\pi(1)}), \cdots, \mathcal{A}_{\mathrm{ldp}}(x_{\pi(n)}))$.

By Theorem 3.1, setting

$$\varepsilon_0 = \begin{cases} \frac{\varepsilon \sqrt{n}}{16\sqrt{\log(1/\delta)}} & \text{for } \varepsilon \leq \sqrt{\log(1/\delta)/n}; \\ \log\left(\frac{\varepsilon^2 n}{100 \log(1/\delta)}\right) & \text{for } \varepsilon \in (\sqrt{\log(1/\delta)/n}, 1). \end{cases}$$

ensure that for a local randomizer $\mathcal{A}_{\mathrm{ldp}}$ that is ε_0-DP, the shuffled output $\mathcal{A}_{\mathrm{s}}(\mathcal{A}_{\mathrm{ldp}})$ is (ε, δ)-DP.

According to [ASZ19], there exists a ε_0-DP local randomizer $\mathcal{A}_{\mathrm{ldp}}$ and algorithm f such that $f \circ \mathcal{A}_{\mathrm{ldp}}$ has error

$$\mathbb{E}[\|p - \hat{p}\|_2] = \begin{cases} O\left(\frac{\sqrt{k}}{\varepsilon_0 \sqrt{n}}\right) & \text{for } \varepsilon_0 \in (0,1); \\ O\left(\frac{\sqrt{k}}{\sqrt{e^{\varepsilon_0} n}} + \frac{1}{\sqrt{n}}\right) & \text{for } \varepsilon_0 \geq 1. \end{cases}.$$

Substituting the value of ε_0 gives the claimed result. □

We note that for the frequency estimation problem in the shuffle model the only algorithm that achieves communication cost that is $o(k)$ while being asymptotically nearly optimal for the central model was given in [GGKPV19]. Their relatively involved multi-message protocol has communication cost of $O(\log k \cdot \log n \cdot \log(1/(\varepsilon\delta))/\varepsilon^2)$. It also does not have local DP guarantees and is relatively inefficient computationally (requires $\tilde{O}(n)$ time to query the frequency of each element).

5.2 Privacy Analysis of Private Stochastic Gradient Descent

In this section we show how our amplification result can be used to analyse private stochastic gradient descent (SGD), an algorithm for empirical risk minimization (ERM). Suppose there exists a loss function ℓ and each user has a data point x_i. The goal is to minimize

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \ell(\theta, x_i),$$

where $\theta \in \mathbb{R}^d$. Gradient descent is a popular algorithm for solving ERM by taking iterative steps in the negative direction of the gradient. In *stochastic* gradient descent, rather than compute each gradient of $\mathcal{L}$ at each step, a single data point is chosen, and the gradient $\nabla\ell(\cdot, x_i)$ is used to update the estimate. A differentially private version of stochastic gradient descent has been studied in [SCS13; BST14], where the gradients are projected onto a bounded ℓ_2 ball and zero-mean isotropic Gaussian noise is added at each iteration to preserve privacy. Bassily, Smith, and Thakurta [BST14] analysed a version of private SGD where the data point at each round was chosen using sampling *with* replacement. Our amplification result allows us to achieve almost the same utility using sampling *without* replacement. Without replacement sampling is much more common and typically more efficient in practice. It also admits a simpler analysis of the generalization error (for example via an online-to-batch conversion [CbCG02]).

Algorithm 2: Differentially Private Stochastic Gradient Descent

Input: X, privacy parameters ε_0, δ_0 and learning rate $\eta_{1:n}$
Choose a random permutation π of $[n]$
Let $\sigma = \frac{1+\sqrt{2\log(1/\delta_0)}}{\varepsilon_0}$
Choose arbitrary initial point $\widetilde{\theta_0}$
for $i = 1 : n$ **do**
  $b_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_d)$
  $g_i = \nabla\ell(\widetilde{\theta}_{i-1}, x_{\pi(i)})$
  $\tilde{g}_i = g_i / \max(1, \|g_i\|_2)$ `// Clip to norm 1`
  $\widetilde{\theta_i} = \widetilde{\theta}_{i-1} - \eta_i \cdot (g_i + b_i)$
return $\widetilde{\theta_n}$

Proposition 5.3. *For any $\delta \in [0,1]$ such that $\varepsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$, Algorithm 2 is $(\varepsilon, \delta + O(e^{\varepsilon}\delta_0 n))$-DP where*

$$\varepsilon = O\left((1 - e^{-\varepsilon_0})\left(\frac{\sqrt{e^{\varepsilon_0}\log(1/\delta)}}{\sqrt{n}} + \frac{e^{\varepsilon_0}}{n}\right)\right). \tag{11}$$

Proof. Note that if we let

$$\mathcal{R}^{(i)}(z_{1:i-1}, x) = \widetilde{\theta_i} = \widetilde{\theta_{i-1}}(z_{1:i-1}) - \eta_t \cdot (\nabla\ell(\widetilde{\theta_{i-1}}(z_{1:i-1}), x_{\pi(i)}) + b_i),$$

then the output of Algorithm 2 can be obtained by post-processing of the shuffled output $\mathcal{A}_{\mathrm{s}}(X)$. Since each $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a $(\varepsilon_0, \delta_0)$-DP local randomizer and $\varepsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$, Theorem 3.8 implies that Algorithm 2 is $(\varepsilon, \delta + (e^{\varepsilon}+1)(1+e^{-\varepsilon_0}/2)n\delta_0)$-DP, where ε is as given in Equation (11). □

For a single pass over the data and $\varepsilon_0 > 1$, the privacy analysis in Proposition 5.3 is tighter than that given by [BST14], who analyse sampling with replacement. For any $\delta > 0$, their analysis, which relies on privacy amplification by subsampling and advanced composition of differentially private algorithms, gives a privacy guarantee of $(\varepsilon, n\delta + \frac{\delta_0}{n})$ where

$$\varepsilon \;=\; O\left(\sqrt{\log(1/\delta)n}\log(1 + \frac{e^{\varepsilon_0}-1}{n}) + (e^{\varepsilon_0}-1)\log(1 + \frac{e^{\varepsilon_0}-1}{n})\right) \approx \frac{\sqrt{\log(1/\delta)}(e^{\varepsilon_0}-1)}{\sqrt{n}}. \tag{12}$$

For large ε_0, our result above (Proposition 5.3) improves on this by a factor of $\Theta(\sqrt{e^{\varepsilon_0}})$. We note that for multiple passes over the data, our analysis suffers an extra $\sqrt{\log(1/\delta)}$ factor in the privacy loss compared to Equation (12). Our analysis technique for Algorithm 2 is also the basis of the analysis of the optimization algorithms in [BKMTT20]. We also note that Proposition 5.3 can be easily extended to the batched version of SGD by viewing each batch of size b as a single data element in $\mathcal{D}^b$.

6 Numerical Results

The proof of Theorem 3.1 provides an efficient method for empirically computing an amplification bound that is tighter than our theoretical bound. Our implementation is outlined in Appendix E, but the main component is empirically computing the indistinguishability bound for multinomial random variables from Lemma 3.2. In this section, we provide numerical evaluations of the privacy amplification bound in a variety of parameter regimes. In Figure 2, `Clones, theoretical` is the bound presented in Theorem 3.1 and `Clones, empirical` is an empirical version derived from the proof of Theorem 3.1. Also, `BBGN'19` is the privacy amplification bound for general algorithms given in [BBGN19][2] and `BKMTT'20` is the bound proven in [BKMTT20]. Finally, `CSUZZ'19, 2RR theoretical` is the theoretical amplification bound for binary randomized response proved in [CSUZZ19] and `2RR, lower bound` is a lower bound on the shuffling privacy guarantee of binary randomized response that we compute directly. A description of our implementation of `2RR, lower bound` is found in Appendix F. We do not include the bounds from [EFMRTT19] in the comparison since [BKMTT20] gives a tighter bound for the same analysis.

In all parameter regimes tested, `Clones, empirical` gives a tighter bound compared to `BBGN'19` as well as `BKMTT'20`. We can see in Figure 2c that this effect is particularly pronounced when ε_0 is large, which is the parameter regime where our theoretical results asymptotically improve over prior work. Being a lower bound for a particular algorithm, `2RR, lower bound` provides a lower bound on any general privacy amplification result. We can see in Figure 2c that for large n, `Clones, empirical` closely tracks `2RR, lower bound`, particularly for small ε_0. We note in all three graphs in Figure 2, `Clones, empirical` should be monotone. The slight non-monotonicity results from some optimizations that speed up the computation at the cost of slightly looser upper bound. We defer details to Appendix E.

References

[App17] Apple's Differential Privacy Team. "Learning with privacy at scale". In: *Apple Machine Learning Journal* 1.9 (2017).

[ASZ19] J. Acharya, Z. Sun, and H. Zhang. "Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication". In: ed. by K. Chaudhuri and M. Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1120–1129.

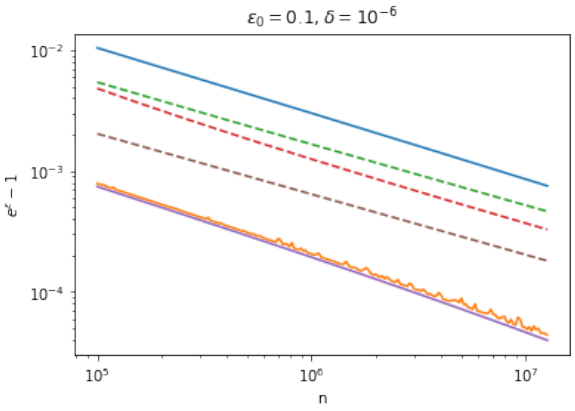
[BBG20] B. Balle, G. Barthe, and M. Gaboardi. "Privacy Profiles and Amplification by Subsampling". In: *Journal of Privacy and Confidentiality* 10.1 (2020).

[BBGN19] B. Balle, J. Bell, A. Gascón, and K. Nissim. "The Privacy Blanket of the Shuffle Model". In: *Advances in Cryptology – CRYPTO 2019*. Ed. by A. Boldyreva and D. Micciancio. Cham: Springer International Publishing, 2019, pp. 638–667.

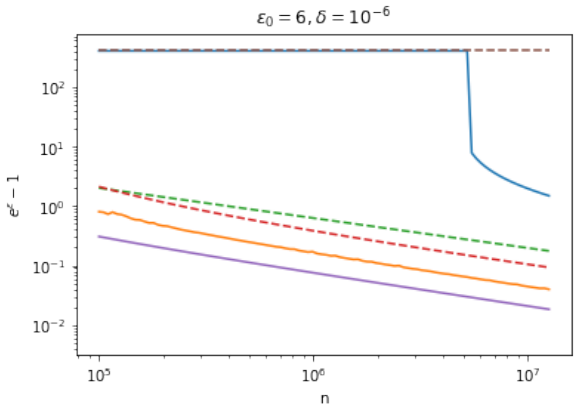
[BEMMRLRKTS17] A. Bittau, U. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld. "Prochlo: Strong Privacy for Analytics in the Crowd". In: *Proceedings of the 26th Symposium on Operating Systems Principles*. SOSP '17. Shanghai, China: Association for Computing Machinery, 2017, 441–459. ISBN: 9781450350853. URL: https://doi.org/10.1145/3132747.3132769.

[BKMTT20] B. Balle, P. Kairouz, H. McMahan, O. Thakkar, and A. Thakurta. "Privacy Amplification via Random Check-Ins". In: *ArXiv* abs/2007.06605 (2020).

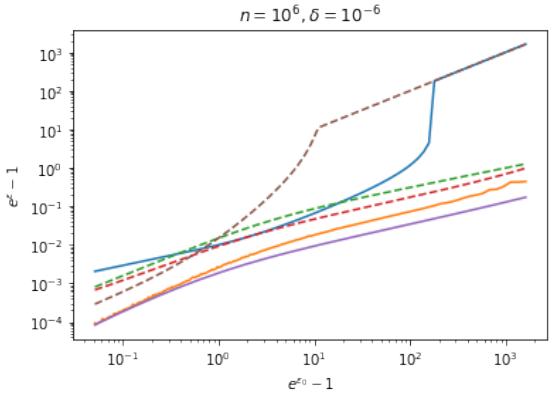
[2] `BBGN'19` curves were produced using open source code released by Balle, Bell, Gascón, and Nissim [BBGN19].



(a) Comparison of central ε for fixed $\varepsilon_0 = 0.1$ and $\delta = 10^{-6}$ and n ranging between 10^5 and 10^7.



(b) Comparison of central ε for fixed $\varepsilon_0 = 6$ and $\delta = 10^{-6}$ and n ranging between 10^5 and 10^7.



(c) Comparison of central ε for fixed $n = 10^6$ and $\delta = 10^{-6}$ and ε_0 ranging between 0.01 and 8.

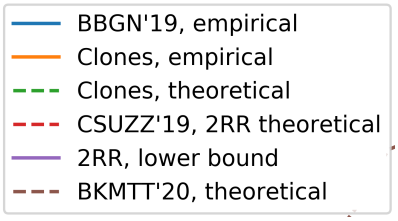


Figure 2: Comparison of new privacy amplification by shuffling bounds given in this work to bounds given in [BBGN19] and [BKMTT20] and to bounds specific to 2RR given in [CSUZZ19]. Dotted lines represent theoretical bounds, while solid lines represent bounds that were computed empirically.

[BNS19] M. Bun, J. Nelson, and U. Stemmer. "Heavy hitters and the structure of local privacy". In: *ACM Transactions on Algorithms (TALG)* 15.4 (2019), pp. 1–40.

[BST14] R. Bassily, A. Smith, and A. Thakurta. "Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds". In: *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. FOCS '14. USA: IEEE Computer Society, 2014, 464–473.

[CbCG02] N. Cesa-bianchi, A. Conconi, and C. Gentile. "On the Generalization Ability of On-Line Learning Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2002, pp. 359–366.

[CSUZZ19] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. "Distributed Differential Privacy via Shuffling". In: *Advances in Cryptology – EUROCRYPT 2019*. Ed. by Y. Ishai and V. Rijmen. Cham: Springer International Publishing, 2019, pp. 375–403. ISBN: 978-3-030-17653-2.

[CU20] A. Cheu and J. R. Ullman. "The Limits of Pan Privacy and Shuffle Privacy for Learning and Estimation". In: *CoRR* abs/2009.08000 (2020). arXiv: 2009.08000. URL: https://arxiv.org/abs/2009.08000.

[DKMMN06] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. "Our Data, Ourselves: Privacy Via Distributed Noise Generation". In: *Advances in Cryptology (EUROCRYPT 2006)*. Vol. 4004. Lecture Notes in Computer Science. Springer Verlag, 2006, pp. 486–503. URL: https://www.microsoft.com/en-us/research/publication/our-data-ourselves-privacy-via-distributed-noise-generation/.

[DKY17] B. Ding, J. Kulkarni, and S. Yekhanin. "Collecting Telemetry Data Privately". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 3574–3583. ISBN: 9781510860964.

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*. Ed. by S. Halevi and T. Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[DR14] C. Dwork and A. Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* 9.3–4 (Aug. 2014), pp. 211–407.

[EFMRSTT20] Úlfar Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, S. Song, K. Talwar, and A. Thakurta. *Encode, Shuffle, Analyze Privacy Revisited: Formalizations and Empirical Evaluation*. 2020. arXiv: 2001.03618 [cs.CR].

[EFMRTT19] U. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. "Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '19. San Diego, California: Society for Industrial and Applied Mathematics, 2019, 2468–2479.

[EPK14] U. Erlingsson, V. Pihur, and A. Korolova. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, 1054–1067. ISBN: 9781450329576.

[GGKPV19] B. Ghazi, N. Golowich, R. Kumar, R. Pagh, and A. Velingker. "On the Power of Multiple Anonymous Messages". In: *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 1382.

[KLNRS08] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. "What Can We Learn Privately?" In: *2008 49th Annual IEEE Symposium on Foundations of Computer Science*. 2008, pp. 531–540.

[KOV15] P. Kairouz, S. Oh, and P. Viswanath. "The Composition Theorem for Differential Privacy". In: ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 1376–1385.

[KOV16] P. Kairouz, S. Oh, and P. Viswanath. "Differentially private multi-party computation". In: *2016 Annual Conference on Information Science and Systems (CISS)*. 2016, pp. 128–132.

[Mir17] I. Mironov. "Rényi differential privacy". In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 263–275.

[MV16] J. Murtagh and S. Vadhan. "The complexity of computing the optimal composition of differential privacy". In: *Theory of Cryptography Conference*. Springer. 2016, pp. 157–175.

[SCS13] S. Song, K. Chaudhuri, and A. D. Sarwate. "Stochastic gradient descent with differentially private updates". In: *IEEE Global Conference on Signal and Information Processing*. 2013.

[Vad17] S. Vadhan. "The Complexity of Differential Privacy". In: Apr. 2017, pp. 347–450. ISBN: 978-3-319-57047-1.

[YB18] M. Ye and A. Barg. "Optimal schemes for discrete distribution estimation under locally differential privacy". In: *IEEE Transactions on Information Theory* 64.8 (2018), pp. 5662–5676.

A Proof of Lemma 3.2

Lemma 3.2. *Let $p \in [0,1]$ and $n \in \mathbb{N}$ be such that $p \geq \frac{16\ln(2/\delta)}{n}$. Consider the process where we first sample $C \sim \mathrm{Bin}(n-1,p)$ then $A \sim \mathrm{Bin}(C,1/2)$. Then if $P = (A+1, C-A)$ and $Q = (A, C-A+1)$, then P and Q are $(\log(1 + \frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}), \delta)$- indistinguishable.*

Proof of Lemma 3.2. Using a Chernoff bound and Hoeffding's inequality, with probability δ since $p \geq \frac{3\log(4/\delta)}{n}$ both

$$|C - pn| \leq \sqrt{3pn\log(4/\delta)} \quad \text{and} \quad |A - C/2| \leq \sqrt{\frac{C}{2}\log(4/\delta)}. \tag{13}$$

Now, note that $p \geq \frac{16\ln(4/\delta)}{n}$ implies that $\sqrt{3pn\log(4/\delta)} \leq pn/2$ and, $\sqrt{\frac{C}{2}\log(4/\delta)} \leq C/4$. Given a specific output (a,b),

$$\begin{aligned}
\Pr(P=(a,b)) &= \Pr(C = a+b-1) \cdot \Pr(A = a-1 | C = a+b-1) \\
&= \Pr(C = a+b-1) \cdot \frac{a}{b} \cdot \Pr(A = a | C = a+b-1) \\
&= \frac{a}{b} \cdot \Pr(Q = (a,b))
\end{aligned}$$

Now, if A and C satisfy equation (13) then

$$\begin{aligned}
\frac{a}{b} &= \frac{A+1}{C-A} \\
&\leq \frac{C/2 + \sqrt{C/2\log(4/\delta)} + 1}{C/2 - \sqrt{C/2\log(4/\delta)}} \\
&\leq 1 + \frac{\sqrt{32\log(4/\delta)}}{\sqrt{C}} + \frac{4}{C} \\
&\leq 1 + \frac{\sqrt{32\log(4/\delta)}}{\sqrt{pn - \sqrt{3pn\log(4/\delta)}}} + \frac{8}{pn} \\
&\leq 1 + \frac{\sqrt{64\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}
\end{aligned}$$

Therefore, setting $\varepsilon = \log(1 + \frac{\sqrt{64\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn})$ gives that P and Q are (ε, δ) indistinguishable.

□

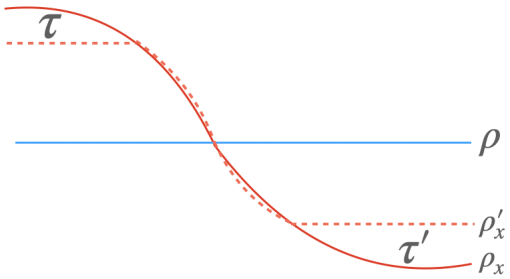


Figure 3: Graphical representation of ρ, ρ_x and ρ'_x from the proof of Lemma 3.7.

B Proof of Proposition 3.8

Lemma 3.7. *Suppose $\mathcal{R}$ is a deletion (ε, δ)-DP local randomizer with reference distribution ρ. Then there exists a randomizer $\mathcal{R}'$ that is a deletion ε-DP local randomizer with reference distribution ρ, and for all inputs x, $\mathrm{TV}(\mathcal{R}(x), \mathcal{R}'(x)) \leq \delta$. In particular, $\mathcal{R}'$ is a (replacement) 2ε-DP local randomizer.*

Proof of Lemma 3.7. Let $x \in \mathcal{D}$, ρ be the reference distribution and ρ_x be the probability density function of $\mathcal{R}(x)$. Thus ρ_x is (ε, δ)-indistinguishable from ρ. For $a, b, c \in \mathbb{R}$ such that $b < c$, let $[a]_b^c = \max\{\min\{a, c\}, b\}$ be a projected onto the interval $[b, c]$. Define a function ρ'_x by

$$\rho'_x(y) = [\rho_x(y)]_{e^{-\varepsilon}\rho(y)}^{e^{\varepsilon}\rho(y)}$$

so that for all y,

$$\ln \frac{\rho(y)}{\rho'_x(y)} \in [-\varepsilon, \varepsilon]. \tag{14}$$

Further, by the definition of (ε, δ)-indistinguishability,

$$\max\left\{\int_y \max\{\rho_x(y) - \rho'_x(y), 0\}dy, \int_y \max\{\rho'_x(y) - \rho_x(y), 0\}dy\right\} \leq \delta. \tag{15}$$

Now, ρ'_x is not necessarily a distribution since $\int_y \rho'_x(y)dy$ is not necessarily 1. Thus, our goal is to define a distribution ρ''_x that preserves Equations (14) and (15). Let $\tau = \int_y \max\{\rho_x(y) - \rho'_x(y), 0\}dy$ and $\tau' = \int_y \max\{\rho'_x(y) - \rho_x(y), 0\}dy$, as in Figure 3. Note that $\tau = \tau'$ if and only if $\int_y \rho'_x(y)dy = 1$. Thus there are two ways that ρ'_x can fail to be a distribution.

Suppose first that $\tau > \tau'$, then intuitively, we can convert ρ'_x into a distribution by moving ρ'_x closer to ρ only in the region where $\rho > \rho_x$. That is, for all $z > 0$, define

$$\rho_x^z(y) = [\rho_x(y)]_{e^{-z}\rho(y)}^{e^{\varepsilon}\rho(y)}$$

and define a function $f(z)$ by

$$f(z) = \int_y \max\{\rho_x^z(y) - \rho_x(y), 0\}dy = \int_{\rho(y) > \rho_x(y)} \max\{\rho_x(y), e^{-z}\rho(y)\} - \rho_x(y)dy.$$

Now, $f(\varepsilon) = \tau' < \tau$ and $f(0) = \mathrm{TV}(\rho, \rho_x) \geq \tau$. Since f is continuous in z, by the intermediate value theorem, there exists $0 \leq \varepsilon' < \varepsilon$ such that $f(\varepsilon') = \tau$. Since the distributions ρ'_x and $\rho_x^{\varepsilon'}$ agree on the region where $\rho_x > \rho'_x > \rho$, we have

$$\int_y \max\{\rho_x(y) - \rho_x^{\varepsilon'}(y), 0\}dy = \int_y \max\{\rho_x(y) - \rho'_x(y), 0\}dy = \tau = \int_y \max\{\rho_x^{\varepsilon'}(y) - \rho_x(y), 0\}dy.$$

This implies that $\rho_x^{\varepsilon'}$ is a distribution. Equation (14) still holds with $\rho_x^{\varepsilon'}$ in place of ρ'_x since $\varepsilon' \le \varepsilon$, and

$$\int_y \max\{\rho_x(y) - \rho_x^{\varepsilon'}(y), 0\}dy = \int_y \max\{\rho_x(y) - \rho_x^{\varepsilon'}(y), 0\}dy = \int_y \max\{\rho_x(y) - \rho'_x(y), 0\}dy \le \delta.$$

We can perform a similar operation if $\tau < \tau'$. Now, we let the output of $\mathcal{R}'(x)$ be distributed according to $\rho_x^{\varepsilon'}$. □

For the proof of Theorem 3.8 we will need the following simple lemma about the hockey-stick divergence.

Lemma B.1. *[DR14, Lemma 3.17] Given random variables P, Q, P' and Q', if $D_{e^\varepsilon}(P', Q') \le \delta$, $\mathrm{TV}(P, P') \le \delta'$ and $\mathrm{TV}(Q, Q') \le \delta'$ then $D_{e^\varepsilon}(P, Q) \le \delta + (e^\varepsilon + 1)\delta'$.*

Theorem 3.8. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} \colon f \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a $(\varepsilon_0, \delta_0)$-DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}} \colon \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a uniformly random permutation π, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Then for any $\delta \in [0, 1]$ such that $\varepsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$, $\mathcal{A}_{\mathrm{s}}$ is $(\varepsilon, \delta + (e^\varepsilon + 1)(1 + e^{-\varepsilon_0}/2)n\delta_0)$-DP, where ε is as in Equation (1).*

Proof of Theorem 3.8. Let X_0 and X_1 be neighboring datasets of size n such that $x_1^0 \ne x_1^1$. As in the proof of Theorem 3.1 we can assume without loss of generality that for all $j \in [2 : n]$, $x_j \notin \{x_1^0, x_1^1\}$.

Now fixing $i \in [n]$ and auxiliary input $z_{1:i-1}$, for $x \in \mathcal{D}$, let $\mathcal{R}(x) = \mathcal{R}^{(i)}(z_{1:i-1}, x)$. If we define $\rho = \mathcal{R}(x_1^0)$ then $\mathcal{R}$ is a deletion $(\varepsilon_0, \delta_0)$-DP local randomizer with reference distribution ρ. By Lemma 3.7, there exists a randomizer $\mathcal{R}'$ that is a deletion ε_0-DP local randomizer with the same reference distribution. In particular, $\mathcal{R}(x_1^0)$ and $\mathcal{R}'(x_1^1)$ are ε_0-indistinguishable, and $\mathrm{TV}(\mathcal{R}(x_1^1), \mathcal{R}'(x_1^1)) \le \delta_0$. Now, by Lemma 3.4, there exist distributions $\mathcal{Q}(x_1^0)$ and $\mathcal{Q}(x_1^1)$ so that

$$\mathcal{R}(x_1^0) = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0} + 1}\mathcal{Q}(x_1^0) + \frac{1}{e^{\varepsilon_0} + 1}\mathcal{Q}(x_1^1), \tag{16}$$

and

$$\mathcal{R}'(x_1^1) = \frac{1}{e^{\varepsilon_0} + 1}\mathcal{Q}(x_1^0) + \frac{e^{\varepsilon_0}}{e^{\varepsilon_0} + 1}\mathcal{Q}(x_1^1). \tag{17}$$

Our next goal is to decompose $\mathcal{R}(x)$ in terms of $\mathcal{Q}(x_1^0)$ and $\mathcal{Q}(x_1^1)$ for all $x \in \mathcal{D}$. By convexity of the hockey-stick divergence, for any $x \in \mathcal{D}$, $\mathcal{R}(x)$ is $(\varepsilon_0, \delta_0)$-indistinguishable from $\frac{1}{2}(\mathcal{R}(x_1^0) + \mathcal{R}(x_1^1))$. That is, $\mathcal{R}$ is a $(\varepsilon_0, \delta_0)$ deletion DP local randomizer with reference distribution $\frac{1}{2}(\mathcal{R}(x_1^0) + \mathcal{R}(x_1^1))$. Therefore, by Lemma 3.7, we can define $\mathcal{R}''$ such that for all $x \in \mathcal{D}$, $\mathcal{R}''(x)$ and $\frac{1}{2}(\mathcal{R}(x_1^0) + \mathcal{R}(x_1^1))$ are ε_0-indistinguishable and $\mathrm{TV}(\mathcal{R}''(x), \mathcal{R}(x)) \le \delta_0$. This implies that there exists a randomized algorithm $\mathtt{LO} \colon \mathcal{D} \to \mathcal{S}$ such that we can decompose $\mathcal{R}''(x)$ as

$$\mathcal{R}''(x) = \frac{1}{2e^{\varepsilon_0}}\mathcal{R}(x_1^0) + \frac{1}{2e^{\varepsilon_0}}\mathcal{R}(x_1^1) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}(x).$$

Next we define a randomizer $\mathcal{L}$ by $\mathcal{L}(x_1^0) = \mathcal{R}(x_1^0)$, $\mathcal{L}(x_1^1) = \mathcal{R}'(x_1^1)$ and for other $x \in \mathcal{D}$,

$$\begin{aligned}\mathcal{L}(x) &= \frac{1}{2e^{\varepsilon_0}}\mathcal{R}(x_1^0) + \frac{1}{2e^{\varepsilon_0}}\mathcal{R}'(x_1^1) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}(x) \\ &= \frac{1}{2e^{\varepsilon_0}}\mathcal{Q}(x_1^0) + \frac{1}{2e^{\varepsilon_0}}\mathcal{Q}(x_1^1) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}(x).\end{aligned} \tag{18}$$

Note that $\mathrm{TV}(\mathcal{R}(x_1^0), \mathcal{L}(x_1^0)) = 0$, $\mathrm{TV}(\mathcal{R}(x_1^1), \mathcal{L}(x_1^1)) \le \delta_0$ and for all $x \in \mathcal{D} \setminus \{x_1^0, x_1^1\}$,

$$\mathrm{TV}(\mathcal{R}(x), \mathcal{L}(x)) \le \mathrm{TV}(\mathcal{R}(x), \mathcal{R}''(x)) + \mathrm{TV}(\mathcal{R}''(x), \mathcal{L}(x)) \le \delta_0 + \frac{1}{2e^{\varepsilon_0}}\mathrm{TV}(\mathcal{R}'(x_1^1), \mathcal{R}(x_1^1)) \le \left(1 + \frac{1}{2e^{\varepsilon_0}}\right)\delta_0.$$

For all $i \in [n]$ and $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$, let $\mathcal{L}^{(i)}(z_{1:i-1}, \cdot)$, $\mathcal{Q}^{(i)}(z_{1:i-1}, \cdot)$ and $\mathtt{LO}^{(i)}(z_{1:i-1}, \cdot)$ denote the result of this transformation applied to $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$. Let $\mathcal{A}_{\mathcal{L}}$ be the same algorithm as $\mathcal{A}_{\mathrm{s}}$ except $\mathcal{R}^{(i)}(z_{1:i-1}, x)$ is replaced by $\mathcal{L}^{(i)}(z_{1:i-1}, x)$. Note that, $\mathcal{A}_{\mathrm{s}}$ applies each randomizer exactly once and hence, by the union bound,

$$\mathrm{TV}(\mathcal{A}_{\mathrm{s}}(X_0), \mathcal{A}_{\mathcal{L}}(X_0)) \le n(1 + e^{-\varepsilon_0}/2)\delta_0 \quad \text{and} \quad \mathrm{TV}(\mathcal{A}_{\mathrm{s}}(X_1), \mathcal{A}_{\mathcal{L}}(X_1)) \le n(1 + e^{-\varepsilon_0}/2)\delta_0.$$

Thus, by Lemma B.1, for any $\varepsilon > 0$, if $\mathcal{A}_{\mathcal{L}}(X_0)$ and $\mathcal{A}_{\mathcal{L}}(X_1)$ are (ε, δ)-indistinguishable then $\mathcal{A}_{\mathrm{s}}(X_0)$ and $\mathcal{A}_{\mathrm{s}}(X_1)$ are $(\varepsilon, \delta + (e^{\varepsilon}+1)(1+e^{-\varepsilon_0}/2)n\delta_0)$-indistinguishable. So, all we need is to analyze the hockey-stick divergence between $\mathcal{A}_{\mathcal{L}}(X_0)$ and $\mathcal{A}_{\mathcal{L}}(X_1)$.

Equations (16), (17), and (18) mirror Equations (3), (4) and (5) in the proof of Theorem 3.1, so the proof will proceed similarly. We define a randomizer $\mathcal{L}_{\mathcal{Q}}^{(i)}$ as follows: For all $x \in \mathcal{D}$, $i \in [n]$ and values of $z_{1:i-1}$ we let

$$\mathcal{L}_{\mathcal{Q}}^{(i)}(z_{1:i-1}, x) = \begin{cases} \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) & \text{if } x = x_1^0; \\ \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) & \text{if } x = x_1^1; \\ \mathcal{L}^{(i)}(z_{1:i-1}, x) & \text{otherwise.} \end{cases}$$

Let $\mathcal{A}_{\mathcal{Q}}$ be defined in the same way as $\mathcal{A}_{\mathrm{s}}$, except $\mathcal{R}^{(i)}$ is replaced with $\mathcal{L}_{\mathcal{Q}}^{(i)}$. Equations (16) and (17), allow us to decompose $\mathcal{A}_{\mathcal{L}}(X_0)$ and $\mathcal{A}_{\mathcal{L}}(X_1)$ into the mixture of two components as follows:

$$\mathcal{A}_{\mathcal{L}}(X_0) = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{A}_{\mathcal{Q}}(X_0) + \frac{1}{e^{\varepsilon_0}+1}\mathcal{A}_{\mathcal{Q}}(X_1) \text{ and } \mathcal{A}_{\mathcal{L}}(X_1) = \frac{1}{e^{\varepsilon_0}+1}\mathcal{A}_{\mathcal{Q}}(X_0) + \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\mathcal{A}_{\mathcal{Q}}(X_1). \tag{19}$$

To compute the divergence between $\mathcal{A}_{\mathcal{Q}}(X_0)$ and $\mathcal{A}_{\mathcal{Q}}(X_1)$ we note that by Equation (18), for all $x \notin \{x_1^0, x_1^1\}$,

$$\mathcal{L}_{\mathcal{Q}}^{(i)}(z_{1:i-1}, x) = \frac{1}{2e^{\varepsilon_0}}\mathcal{L}_{\mathcal{Q}}^{(i)}(z_{1:i-1}, x_1^0) + \frac{1}{2e^{\varepsilon_0}}\mathcal{L}_{\mathcal{Q}}^{(i)}(z_{1:i-1}, x_1^1) + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\mathtt{LO}^{(i)}(z_{1:i-1}, x).$$

Therefore, by Lemma 3.3, $\mathcal{A}_{\mathcal{Q}}(X_0)$ and $\mathcal{A}_{\mathcal{Q}}(X_1)$ are

$$\left(\log\left(1 + \frac{8\sqrt{e^{\varepsilon_0}\log(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_0}}{n}\right), \delta\right)\text{-indistinguishable.}$$

Finally, as before, Equation (19) and Lemma 2.3 together imply that $\mathcal{A}_{\mathcal{L}}(X_0)$ and $\mathcal{A}_{\mathcal{L}}(X_1)$ are (ε, δ)-indistinguishable where

$$\varepsilon \le \log\left(1 + \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\left(\frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right)\right).$$

□

C Proof of Theorem 4.1

Theorem 4.1. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a ε_0-DP local randomizer for all values of auxiliary inputs $z_{1,i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}} : \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a uniformly random permutation π, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Let $X_0 = (x_1^0, x_2, \ldots, x_n)$ and $X_1 = (x_1^1, x_2, \ldots, x_n)$ be two neighboring datasets such that for all $j \neq 1$, $x_j \notin \{x_1^0, x_1^1\}$. Suppose that there exist positive values $p, q \in (0, 1]$ such that for all $i \in [n]$ and $z_{1:i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$, there exists distributions $\mathcal{Q}_0^{(i)}(z_{1:i-1})$, $\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^0)$ and $\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^1)$ such that for all $b \in \{0, 1\}$,*

$$\mathcal{R}^{(i)}(z_{1:i-1}, x_1^b) = (1-q)\mathcal{Q}_0^{(i)}(z_{1:i-1}) + q\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b) \tag{7}$$

and for all $x \in \mathcal{D}\backslash\{x_1^0, x_1^1\}$, there exists a distributions $\mathtt{LO}_0^{(i)}(z_{1:i-1}, x)$ and $\mathtt{LO}_1^{(i)}(z_{1:i-1}, x)$ such that for all $b \in \{0, 1\}$

$$\mathcal{R}^{(i)}(z_{1:i-1}, x) = p\mathcal{Q}_0^{(i)}(z_{1:i-1}) + p\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b) + (1-2p)\mathtt{LO}_b^{(i)}(z_{1:i-1}, x). \tag{8}$$

Then for any $\delta \in [0, 1]$ such that $\varepsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$, $\mathcal{A}_{\mathrm{s}}(X_0)$ is $(\varepsilon, q\delta)$-indistinguishable from $\mathcal{A}_{\mathrm{s}}(X_1)$, where

$$\varepsilon \le \log\left(1 + q\left(\frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right)\right). \tag{9}$$

Proof of Theorem 4.1. Using the decomposition of $\mathcal{R}^{(i)}(z_{1:i-1}, x)$ in Equation 8 we decompose $\mathcal{A}_{\mathrm{s}}$ into the mixture of two algorithms, which we analyze separately. For this purpose we define two randomizers $\mathcal{R}_0^{(i)}$ and $\mathcal{R}_1^{(i)}$ as follows:

For all $i \in [n]$, $x \in \mathcal{D}$, and values of $z_{1:i-1}$ we let

$$\mathcal{R}_0^{(i)}(z_{1:i-1}, x) = \begin{cases} \mathcal{Q}_0^{(i)}(z_{1:i-1}) & \text{if } x = x_1^0; \\ \mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^0) & \text{if } x = x_1^1; \\ \mathcal{R}^{(i)}(z_{1:i-1}, x) & \text{otherwise.} \end{cases}$$

and

$$\mathcal{R}_1^{(i)}(z_{1:i-1}, x) = \begin{cases} \mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^0) & \text{if } x = x_1^0; \\ \mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^1) & \text{if } x = x_1^1; \\ \mathcal{R}^{(i)}(z_{1:i-1}, x) & \text{otherwise.} \end{cases}$$

Now, let $\mathcal{A}_{\mathrm{s},0}$, and $\mathcal{A}_{\mathrm{s},1}$ be defined in the same way as $\mathcal{A}_{\mathrm{s}}$ with $\mathcal{R}^{(i)}$ replaced with $\mathcal{R}_0^{(i)}$ and $\mathcal{R}_1^{(i)}$, respectively. Equation (7) implies that

$$\mathcal{A}_{\mathrm{s}}(X_0) = (1-q)\mathcal{A}_{\mathrm{s},0}(X_0) + q\mathcal{A}_{\mathrm{s},1}(X_0) \text{ and } \mathcal{A}_{\mathrm{s}}(X_1) = (1-q)\mathcal{A}_{\mathrm{s},0}(X_0) + q\mathcal{A}_{\mathrm{s},1}(X_1),$$

Let us first compute the divergence between $\mathcal{A}_{\mathrm{s},0}(X_0)$ and $\mathcal{A}_{\mathrm{s},0}(X_1) = \mathcal{A}_{\mathrm{s},1}(X_0)$.

By Equation (8), for any $x \in \mathcal{D} \setminus \{x_1^0, x_1^1\}$,

$$\mathcal{R}_0^{(i)}(z_{1:i-1}, x) = p\mathcal{R}_0^{(i)}(z_{1:i-1}, x_1^0) + p\mathcal{R}_0^{(i)}(z_{1:i-1}, x_1^1) + (1-2p)\mathtt{LO}^{(i)}(z_{1:i-1}, x).$$

Thus, by Lemma 3.3, $\mathcal{A}_{\mathrm{s},0}(X_0)$ and $\mathcal{A}_{\mathrm{s},0}(X_1)$ are

$$\left(\log\left(1 + \frac{4\sqrt{2\log(4/\delta)}}{\sqrt{pn}} + \frac{4}{pn}\right), \delta\right)\text{-indistinguishable.}$$

Now, for any $x \in \mathcal{D} \setminus \{x_1^0, x_1^1\}$, if we take the average of Equation (8) with $b = 0$ and $b = 1$, and set

$$\mathtt{LO}_+^{(i)}(z_{1:i-1}, x) = \frac{p}{1-p}\mathcal{Q}_0^{(i)}(z_{1:i-1}) + \frac{1-2p}{2(1-p)}\left(\mathtt{LO}_0^{(i)}(z_{1:i-1}, x) + \mathtt{LO}_1^{(i)}(z_{1:i-1}, x)\right)$$

then we have that

$$\mathcal{R}_1^{(i)}(z_{1:i-1}, x) = \frac{p}{2}\mathcal{R}_1^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2}\mathcal{R}_1^{(i)}(z_{1:i-1}, x_1^1) + (1-p)\mathtt{LO}_+^{(i)}(z_{1:i-1}, x).$$

Therefore, again by Lemma 3.3, $\mathcal{A}_{\mathrm{s},1}(X_0)$ and $\mathcal{A}_{\mathrm{s},1}(X_1)$ are

$$\left(\log\left(1 + \frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right), \delta\right)\text{-indistinguishable.}$$

Now, by Lemma 2.3, $\mathcal{A}_{\mathrm{s}}(X_0)$ and $\mathcal{A}_{\mathrm{s}}(X_1)$ are

$$\left(\log\left(1 + q\left(\frac{8\sqrt{\log(4/\delta)}}{\sqrt{pn}} + \frac{8}{pn}\right)\right), p\delta\right)\text{-indistinguishable.}$$

□

D Proof of Corollary 4.2

Corollary 4.2. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a ε_0-DP local randomizer for all values of auxiliary inputs $z_{1,i-1} \in \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_{\mathrm{s}} : \mathcal{D}^n \to \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, samples a uniformly random permutation π, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi(i)})$ for $i \in [n]$ and outputs $z_{1:n}$. Assume that for some $k \in \mathbb{N}$ we have that for all $i \in [n]$, there exists a function $f^{(i)} : \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to [k]$ such that $\mathcal{R}^{(i)}(z_{1:i-1}, x) = kRR(f^{(i)}(z_{1:i-1}, x))$. Then for $\delta \in [0, 1]$ such that such that $\varepsilon_0 \le \log(\frac{n}{16\log(2/\delta)})$, $\mathcal{A}_{\mathrm{s}}$ is (ε, δ)-DP where*

$$\varepsilon \le \log\left(1 + (e^{\varepsilon_0} - 1)\left(\frac{8\sqrt{(k+1)\log(4/\delta)}}{\sqrt{(e^{\varepsilon_0} + k - 1)kn}} + \frac{8(k+1)}{kn}\right)\right) \tag{10}$$

Proof of Corollary 4.2. To show that $\mathcal{A}_{\mathrm{s}}$ is (ε, δ)-DP, it is sufficient to show that for any pair of neighboring datasets X_0 and X_1, $\mathcal{A}_{\mathrm{s}}(X_0)$ and $\mathcal{A}_{\mathrm{s}}(X_1)$ are (ε, δ)-indistinguishable. Let X_0 and X_1 be a pair of neighboring datasets with $x_1^0 \neq x_1^1$. Let $\mathcal{Q}_0^{(i)}(z_{1:i-1})$ be the uniform distribution on $[k]$, and for any $j \in [k]$, let $\mathbb{1}_j$ is the distribution that always outputs j. So for any $x \in \mathcal{D}$,

$$\mathcal{R}^{(i)}(z_{1:i-1}, x) = \mathtt{kRR}(f^{(i)}(z_{1:i-1}, x)) = \frac{k}{e^{\varepsilon_0} + k - 1}\mathcal{Q}_0^{(i)}(z_{1:i-1}) + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + k - 1}\mathbb{1}_{f^{(i)}(z_{1:i-1}, x)}.$$

Let $p = \frac{k}{(k+1)(e^{\varepsilon_0}+k-1)}$. Now, note that $\mathcal{Q}_0^{(i)}(z_{1:i-1}) = \frac{1}{k+1}\mathcal{Q}_0^{(i)}(z_{1:i-1}) + \frac{1}{k+1}\sum_{j=1}^k \mathbb{1}_j$. So, for any x, if we let

$$\mathtt{LO}_b^{(i)}(z_{1:i-1}, x) = \frac{1}{(1-2p)}\left(\frac{k}{(k+1)(e^{\varepsilon_0} + k - 1)} \sum_{j \in [k],\, j \neq f^{(i)}(z_{1:i-1}, x_1^b)} \mathbb{1}_j + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + k - 1}\mathbb{1}_{f^{(i)}(z_{1:i-1}, x)}\right)$$

then

$$\begin{aligned}
\mathcal{R}^{(i)}(z_{1:i-1}, x) &= \frac{k}{e^{\varepsilon_0} + k - 1}\mathcal{Q}_0^{(i)}(z_{1:i-1}) + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + k - 1}\mathbb{1}_{f^{(i)}(z_{1:i-1}, x)} \\
&= \frac{k}{e^{\varepsilon_0} + k - 1}\left(\frac{1}{k+1}\mathcal{Q}_0^{(i)}(z_{1:i-1}) + \frac{1}{k+1}\sum_{j=1}^k \mathbb{1}_j\right) + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + k - 1}\mathbb{1}_{f^{(i)}(z_{1:i-1}, x)} \\
&= p\mathcal{Q}_0^{(i)}(z_{1:i-1}) + p\mathbb{1}_{f^{(i)}(z_{1:i-1}, x_1^b)} + (1-2p)\mathtt{LO}_b^{(i)}(z_{1:i-1}, x)
\end{aligned}$$

Therefore, $\mathcal{R}^{(i)}(z_{1:i-1}, x)$ satisfies the conditions of Theorem 4.1 with $q = \frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+k-1}$ and $p = \frac{k}{(k+1)(e^{\varepsilon_0}+k-1)}$. Theorem 4.1 implies that $\mathcal{A}_{\mathrm{s}}(X_0)$ and $\mathcal{A}_{\mathrm{s}}(X_1)$ are (ε, δ)-indistinguishable where

$$\begin{aligned}
\varepsilon &\le \log\left(1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + k - 1}\left(\frac{8\sqrt{(k+1)(e^{\varepsilon_0} + k - 1)\log(4/\delta)}}{\sqrt{kn}} + \frac{8(k+1)(e^{\varepsilon_0} + k - 1)}{kn}\right)\right) \\
&\le \log\left(1 + (e^{\varepsilon_0} - 1)\left(\frac{8\sqrt{(k+1)\log(4/\delta)}}{\sqrt{(e^{\varepsilon_0} + k - 1)kn}} + \frac{8(k+1)}{kn}\right)\right)
\end{aligned}$$

□

E Implementation of `Clones, empirical`

In this section we outline our implementation of the proof of Theorem 3.1 to compute the amplification bound for general local randomizers. Let $q = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}$ and $p = e^{-\varepsilon_0}$. Recall that the shuffled ε is upper bounded by the divergence

between the random variables

$$P = \begin{cases} (A, C) & \text{w.p. } q \\ (A+1, C) & \text{w.p. } 1-q \end{cases} \quad \text{and} \quad Q = \begin{cases} (A, C) & \text{w.p. } 1-q \\ (A+1, C) & \text{w.p. } q \end{cases}, \tag{20}$$

where $C \sim \mathrm{Bin}(n-1, p)$ and $A \sim \mathrm{Bin}(C, 1/2)$. For a given δ, our goal is to compute an approximately minimal ε such that P and Q are (ε, δ)-indistinguishable. It is computationally easier to compute an approximately minimal δ for a given ε than the converse, so for a given δ we'll use binary search to find such an ε. Algorithm 3 takes as input a function M that computes an approximation to the smallest δ such that P and Q are (ε, δ)-indistinguishable for a given ε.

Algorithm 3: Binary Search, `BinS`

Input: $\varepsilon_0, \delta, T, M$
$\varepsilon^L = 0$
$\varepsilon^R = \varepsilon_0$
for $t \in [T]$ **do**
  $\varepsilon_t = \frac{\varepsilon^L + \varepsilon^R}{2}$
  $\delta_t = M(\varepsilon_t)$ `// M(ε_t) = the smallest value such that P and Q are (ε_t, δ_t)-indistinguishable`
  if $\delta_t < \delta$ **then**
    $\varepsilon^R = \varepsilon_t$
  else
    $\varepsilon^L = \varepsilon_t$
return ε^R

Next, we need to design the algorithm M. Note that for a given ε, the minimal δ is given by the equation

$$\delta = D_{e^\varepsilon}(P, Q) = \max\left\{ \int_{(a,c)} \max\{0, P(a,c) - e^\varepsilon Q(a,c)\} d(a,c), \int_{(a,c)} \max\{0, Q(a,c) - e^\varepsilon P(a,c)\} d(a,c) \right\},$$

where for ease of notation we use P and Q for both the random variables and their probability density functions (pdf). We present an algorithm M that upper bounds this integral. Let us first look at an important subroutine: for a given c, Algorithm 4 computes $\int_a \max\{0, P(a,c) - e^\varepsilon Q(a,c)\} da$ if $b = +$ and $\int_a \max\{0, Q(a,c) - e^\varepsilon P(a,c)\} da$ if $b = -$.

Lemma E.1. *For $c \in \mathbb{Z}$, and with P and Q as in Equation (20), if $\varepsilon < \varepsilon_0$ then*

$$\Pr(\mathrm{Bin}(n-1, p) = c) \cdot \mathcal{B}(c, \epsilon, \epsilon_0, +) = \int_{\mathbb{Z}} \max\{0, P(a,c) - e^\varepsilon Q(a,c)\} da$$

and

$$\Pr(\mathrm{Bin}(n-1, p) = c) \cdot \mathcal{B}(c, \epsilon, \epsilon_0, -) = \int_{\mathbb{Z}} \max\{0, Q(a,c) - e^\varepsilon P(a,c)\} da.$$

Proof. First, let us characterize when $P(a,c) - e^\varepsilon Q(a,c) > 0$. We will let C denote both the random variable $\mathrm{Bin}(n-1, p)$ and its pdf. Similarly, A_c denotes the random variable $\mathrm{Bin}(c, 1/2)$ and its pdf.

$$\begin{aligned} P(a,c) - e^\varepsilon Q(a,c) > 0 &\iff C(c)((qA_c(a) + (1-q)A_c(a-1)) - e^\varepsilon((1-q)A_c(a) + qA_c(a-1))) > 0 \\ &\iff (q - e^\varepsilon(1-q))A_c(a) > (e^\varepsilon q - (1-q))A_c(a-1) \end{aligned}$$

Algorithm 4: $\mathcal{B}$

Input: $c, \varepsilon, \varepsilon_0, b$
$q = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}$
$\varepsilon_{q,\varepsilon} = \ln\left(\frac{(e^{\varepsilon}+1)q-1}{(e^{\varepsilon}+1)q-e^{\varepsilon}}\right)$
if $b = +$ **then**
  $\beta = \frac{1}{e^{\varepsilon_{q,\varepsilon}}+1}$
else
  $\beta = \frac{1}{e^{-\varepsilon_{q,\varepsilon}}+1}$
$\tau = \beta(c+1)$
$\gamma_P = q * \Pr(\mathrm{Bin}(c, 0.5) \le \tau) + (1-q)\Pr(\mathrm{Bin}(c, 0.5) \le \tau - 1)$
$\gamma_Q = (1-q) * \Pr(\mathrm{Bin}(c, 0.5) \le \tau) + q\Pr(\mathrm{Bin}(c, 0.5) \le \tau - 1)$
if $b = +$ **then**
  return $\gamma_P - e^{\varepsilon}\gamma_Q$
else
  return $(1-\gamma_Q) - e^{\varepsilon}(1-\gamma_P)$

Now, if $\varepsilon < \varepsilon_0$ then $q - e^{\varepsilon}(1-q) > 0$. Let $\varepsilon_{q,\varepsilon} = \ln\left(\frac{e^{\varepsilon}q-(1-q)}{q-e^{\varepsilon}(1-q)}\right)$. So,

$$\begin{aligned} P(a,c) - e^{\varepsilon}Q(a,c) > 0 &\iff \binom{c}{a} > e^{\varepsilon_{q,\varepsilon}}\binom{c}{a-1} \\ &\iff \frac{c+1-a}{a} > e^{\varepsilon_{q,\varepsilon}} \\ &\iff (e^{\varepsilon_{q,\varepsilon}}+1)a < c+1 \\ &\iff a < \frac{c+1}{e^{\varepsilon_{q,\varepsilon}}+1} \end{aligned}$$

If $b = +$ then $\tau = \frac{c+1}{e^{\varepsilon_{q,\varepsilon}}+1}$ so $P(a,c) - e^{\varepsilon}Q(a,c) > 0 \iff a < \tau$. Note that $\gamma_P = \Pr(P \le \tau)$ and $\gamma_Q = \Pr(Q \le \tau)$. Therefore,

$$\int_{\mathbb{Z}} \max\{0, P(a,c) - e^{\varepsilon}Q(a,c)\}da = \int_{-\infty}^{\tau} (P(a,c) - e^{\varepsilon}Q(a,c))da = C(c)(\gamma_P - e^{\varepsilon}\gamma_Q)$$

Now, for the second integral,

$$\begin{aligned} Q(a,c) - e^{\varepsilon}P(a,c) > 0 &\iff C(c)(((1-q)A_c(a) + qA_c(a-1)) - e^{\varepsilon}(qA_c(a) + (1-q)A_c(a-1))) > 0 \\ &\iff ((1-q) - e^{\varepsilon}q)A_c(a) > (e^{\varepsilon}(1-q) - q)A_c(a-1) \end{aligned}$$

For any ε, $(1-q) - e^{\varepsilon}q < 0$ so,

$$\begin{aligned} Q(a,c) - e^{\varepsilon}P(a,c) > 0 &\iff A_c(a) < e^{-\varepsilon_{q,\varepsilon}}A_c(a-1) \\ &\iff \frac{c+1-a}{a} < e^{-\varepsilon_{q,\varepsilon}} \\ &\iff (e^{-\varepsilon_{q,\varepsilon}}+1)a > c+1 \\ &\iff a > \frac{c+1}{e^{-\varepsilon_{q,\varepsilon}}+1} \end{aligned}$$

If $b = -$ then $\tau = \frac{c+1}{e^{-\varepsilon_{q,\varepsilon}}+1}$ so $Q(a,c) - e^{\varepsilon}P(a,c) > 0 \iff a > \tau$. Noting that $1 - \gamma_P = \Pr(P \ge \tau)$ and $1 - \gamma_Q = \Pr(Q \ge \tau)$ we have,

$$\int_{\mathbb{Z}} \max\{0, Q(a,c) - e^{\varepsilon}P(a,c)\}da = \int_{\tau}^{\infty} Q(a,c) - e^{\varepsilon}P(a,c)da = C(c)((1-\gamma_Q) - e^{\varepsilon}(1-\gamma_P).$$

□

Lemma E.1 shows that for a fixed c, $\mathcal{B}$ computes the integral $\mathcal{B}(c,\varepsilon,\varepsilon_0,+) = \int_a \max\{0, P(a,c) - e^{\varepsilon}Q(a,c)\}da$. Recall, our goal is to estimate

$$\int_c \int_a \max\{0, P(a,c) - e^{\varepsilon}Q(a,c)\}dadc = \int_c \mathcal{B}(c,\varepsilon,\varepsilon_0,+)dc.$$

We could compute $\mathcal{B}(c,\varepsilon,\varepsilon_0,+)$ for every c, but in order to make the computation more efficient, instead of computing $\mathcal{B}(c,\varepsilon,\varepsilon_0,+)$ for every c, Algorithm 5 defines a parameter S, and only computes it for every Sth value of c. For values between c and $c+S$, we can leverage the fact that $\mathcal{B}(c,\varepsilon,\varepsilon_0,+)$ is monotone to bound their contribution to the integral.

Lemma E.2. *For any $\varepsilon < \varepsilon_0$ and $b \in \{+,-\}$, if $c > c'$ then $\mathcal{B}(c,\varepsilon,\varepsilon_0,b) < \mathcal{B}(c',\varepsilon,\varepsilon_0,b)$.*

Proof. Let us prove the result for $b = +$, the proof for $b = -$ is identical. Recall $q = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}$ and $c > 0$, let

$$P_c = \begin{cases} \mathrm{Bin}(c,1/2) & \text{w.p. } q \\ \mathrm{Bin}(c,1/2)+1 & \text{w.p. } 1-q \end{cases} \quad \text{and} \quad Q_c = \begin{cases} \mathrm{Bin}(c,1/2) & \text{w.p. } 1-q \\ \mathrm{Bin}(c,1/2)+1 & \text{w.p. } q \end{cases}.$$

Now, suppose that $c > c'$ then $P_c = P_{c'} + \mathrm{Bin}(c-c',1/2)$ and $Q_c = Q_{c'} + \mathrm{Bin}(c-c',1/2)$, where we are summing the random variables (not the pdfs). That is, P_c and Q_c can be obtained by post-processing $P_{c'}$ and $Q_{c'}$. Also $\int_{\mathbb{Z}} \max\{0, P_c(a) - e^{\varepsilon}Q_c(a)\}da$ is the hockey-stick divergence between P_c and Q_c. Therefore, since the hockey-stick divergence satisfies the data processing inequality and by Lemma E.1,

$$\mathcal{B}(c,\varepsilon,\varepsilon_0,b) = \int_{\mathbb{Z}} \max\{0, P_c(a) - e^{\varepsilon}Q_c(a)\}da \le \int_{\mathbb{Z}} \max\{0, P_{c'}(a) - e^{\varepsilon}Q_{c'}(a)\}da = \mathcal{B}(c',\varepsilon,\varepsilon_0,b).$$

□

Additionally, the algorithm takes in a value δ^U (set to be the target δ when called from binary search) so that if the current guess for ε is too small, then we may save on computation by aborting as soon as we have established that $\delta < \delta^U$. We remark that while we have stated the algorithm as iterating over the values of t starting at zero, the proof does not require that the values $\{0,\dots,T\}$ be processed in increasing order. In an actual implementation, we process these in increasing order of $|t - T/2|$ so as to process the c's which have large probability mass first.

Proposition E.3. *For any $n \in \mathbb{N}, \varepsilon_0 > 0, \delta \in [0,1], T \in \mathbb{N}$, let $\varepsilon = \texttt{BinS}(\varepsilon_0,\delta,T,M_1(n,\varepsilon_0,\delta,S,\cdot))$ be the output of Algorithm 3 where M_1 is given by Algorithm 5. Then P and Q, as given by Equation (20), are (ε,δ)-indistinguishable.*

Proof. Note that since P and Q are (ε_0,δ)-indistinguishable and Algorithm 3 outputs the largest value inside the final range $[\varepsilon_L,\varepsilon_R]$, it suffices to show that at each iteration,

$$\delta_t = M_1(n,\varepsilon_0,\delta,S,\varepsilon_t) \ge \min\left\{\delta, \max\left\{\int_{\mathbb{Z}}\int_{\mathbb{Z}} P(a,c) - e^{\varepsilon_t}Q(a,c)d(a,c), \int_{\mathbb{Z}}\int_{\mathbb{Z}} Q(a,c) - e^{\varepsilon_t}P(a,c)d(a,c)\right\}\right\}.$$

That is, that Algorithm 3 makes conservative choices at each iteration. Let us then focus on showing that for any choices of $n,\varepsilon_0,\varepsilon,\delta^U,S$ such that $\varepsilon_0 > \varepsilon$,

$$M_1(n,\varepsilon_0,\delta^U,S,\varepsilon_t) \ge \min\left\{\delta^U, \max\left\{\int_{\mathbb{Z}}\int_{\mathbb{Z}} P(a,c) - e^{\varepsilon_t}Q(a,c)d(a,c), \int_{\mathbb{Z}}\int_{\mathbb{Z}} Q(a,c) - e^{\varepsilon_t}P(a,c)d(a,c)\right\}\right\}. \tag{21}$$

Let $t \in \{0,\cdots,T\}$ and $B^t = t * S$. Since $\mathcal{B}(\cdot,\varepsilon,\varepsilon_0,b)$ is monotone, $\mathcal{B}(B^t,\varepsilon,\varepsilon_0,b) \ge \mathcal{B}(D,\varepsilon,\varepsilon_0,b)$ for any $D \in [B^t, B^t+S)$. Thus,

$$Pr_{[C_{\min},C_{\max}]} \cdot \max\{\mathcal{B}(B^t,\varepsilon,\varepsilon_0,+), \mathcal{B}(B^t+S-1,\varepsilon,\varepsilon_0,+)\} \ge \int_{B^t}^{B^t+S-1} \int_{\mathbb{Z}} \max\{0, P(a,c) - e^{\varepsilon}Q(a,c)\}d(a,c)$$

Algorithm 5: M_1

Input: $n, \varepsilon_0, \delta^U, S, \varepsilon$

$\delta_P^0 = 0$ // will keep track of $\int \max\{P(x) - e^{\varepsilon} Q(x), 0\} dx$

$\delta_Q^0 = 0$ // will keep track of $\int \max\{Q(x) - e^{\varepsilon} P(x), 0\} dx$

$\zeta_C^0 = 0$ // will keep track of the probability mass of C covered

$p = e^{-\varepsilon_0}$

$T = \lfloor n/S \rfloor$

for $t = \{0, \cdots, T\}$ **do**

$B^t = t * S$

Step 1: if $\max(\delta_P^t, \delta_Q^t) > \delta^U$ then P and Q are not (ε, δ^U)-indistinguishable so exit.

if $\max(\delta_P^t, \delta_Q^t) > \delta^U$ **then**

return δ^U

Step 2: any further contribution to either δ_P or δ_Q will not exceed $1 - \zeta_C$, so if this is small we exit.

else if $1 - \zeta_C^t < \delta_P^t$ *and* $1 - \zeta_C^t < \delta_Q^t$ **then**

return $\max\{\delta_P^t + 1 - \zeta_C^t, \delta_Q^t + 1 - \zeta_C^t\}$

Step 3: we estimate the contribution to δ_P and δ_Q from the next interval $[B^t, B^t + S)$

else

$C_{\max} = B^t + S - 1$

$C_{\min} = B^t$

$Pr_{[C_{\min}, C_{\max}]} = \Pr(\mathrm{Bin}(n-1, p) \in [C_{\min}, C_{\max}])$

Step 3a: Compute contribution to δ_P

$c_{P,\max,\delta} = \mathcal{B}(C_{\max}, n, p, \varepsilon, \varepsilon_0, +)$

$c_{P,\min,\delta} = \mathcal{B}(C_{\min}, n, p, \varepsilon, \varepsilon_0, +)$

$\delta_P^{t+1} = \delta_P^t + Pr_{[C_{\min}, C_{\max}]} \cdot \max\{c_{P,\max,\delta}, c_{P,\min,\delta}\}$

Step 3b: Compute contribution to δ_Q

$c_{Q,\max,\delta} = \mathcal{B}(C_{\max}, n, p, \varepsilon, \varepsilon_0, -)$

$c_{Q,\min,\delta} = \mathcal{B}(C_{\min}, n, p, \varepsilon, \varepsilon_0, -)$

$\delta_Q^{t+1} = \delta_Q^t + \Pr_{[C_{\min}, C_{\max}]} \cdot \max\{c_{Q,\max,\delta}, c_{Q,\min,\delta}\}$

Step 3c: Compute contribution to ζ_C

$\zeta_C^{t+1} = \zeta_C^t + \Pr_{[C_{\min}, C_{\max}]}$

return $\max\{\delta_P^{T+1}, \delta_Q^{T+1}\}$

and

$$Pr_{[C_{\min},C_{\max}]} \cdot \max\{\mathcal{B}(B^t,\varepsilon,\varepsilon_0,-),\mathcal{B}(B^t+S-1,\varepsilon,\varepsilon_0,-)\} \geq \int_{B^t}^{B^t+S-1}\int_{\mathbb{Z}} \max\{0,Q(a,c)-e^{\varepsilon}P(a,c)\}d(a,c).$$

Let $\mathcal{R}_C^t = [0,B^t)$ be the range of values of C that have already been covered by the first $t-1$ iterations. So, the above equations imply that we always have

$$\delta_P^t \geq \int_{\mathcal{R}_C^t}\int_{\mathbb{Z}} \max\{0,P(a,c)-e^{\varepsilon}Q(a,c)\}d(a,c) \text{ and } \delta_Q^t \geq \int_{\mathcal{R}_C^t}\int_{\mathbb{Z}} \max\{0,Q(a,c)-e^{\varepsilon}P(a,c)\}d(a,c).$$

Now, there are three ways Algorithm 5 can terminate.

Case 1: If at round t, $\max\{\delta_P^t,\delta_Q^t\} > \delta^U$ then $M_1(n,\varepsilon_0,\delta,S,\varepsilon_t) = \delta^U$ and Equation (21) holds.

Case 2: If at round t, $1-\zeta_C^t < \delta_P^t$ and $1-\zeta_C^t < \delta_Q^t$ then

$$\begin{aligned}
&\int_{\mathbb{Z}}\int_{\mathbb{Z}} \max\{0,P(a,c)-e^{\varepsilon}Q(a,c)\}d(a,c)\\
&\quad= \int_{\mathcal{R}_C^t}\int_{\mathbb{Z}} \max\{0,P(a,c)-e^{\varepsilon}Q(a,c)\}d(a,c) + \int_{\mathbb{Z}\setminus\mathcal{R}_C^t}\int_{\mathbb{Z}} \max\{0,P(a,c)-e^{\varepsilon}Q(a,c)\}d(a,c)\\
&\quad\leq \delta_P + \Pr(\mathrm{Bin}(n-1,p)\in\mathbb{Z}\setminus\mathcal{R}_C^t)\\
&\quad\leq \delta_P^t + 1 - \zeta_C^t,
\end{aligned}$$

and similarly

$$\int_{\mathbb{Z}}\int_{\mathbb{Z}} \max\{0,Q(a,c)-e^{\varepsilon}P(a,c)\}d(a,c) \leq \delta_Q^t + 1 - \zeta_C^t.$$

Since $M_1(n,\varepsilon_0,\delta,S,\varepsilon_t) = \max\{\delta_P^t+1-\zeta_C^t,\delta_Q^t+1-\zeta_C^t\}$, this implies Equation (21) holds.

Case 3: If the algorithm doesn't terminate early, then $\mathcal{R}_C^{T+1} = \mathbb{Z}$ so

$$\int_{\mathbb{Z}}\int_{\mathbb{Z}} \max\{0,P(a,c)-e^{\varepsilon}Q(a,c)\}d(a,c) \leq \delta_P^{T+1}$$

and

$$\int_{\mathbb{Z}}\int_{\mathbb{Z}} \max\{0,Q(a,c)-e^{\varepsilon}P(a,c)\}d(a,c) \leq \delta_Q^{T+1}.$$

Since $M_1(n,\varepsilon_0,\delta,S,\varepsilon_t) = \max\{\delta_P^{T+1},\delta_Q^{T+1}\}$ this implies Equation (21) holds. □

F Implementation of `2RR, lower bound`

For any $\varepsilon_0 > 0$, binary randomized response $\mathtt{2RR}\colon \{0,1\}\to\{0,1\}$ is defined as

$$\mathtt{2RR}(x) = \begin{cases} x & \text{with probability } \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1} \\ 1-x & \text{with probability } \frac{1}{e^{\varepsilon_0}+1} \end{cases}.$$

Let $\mathcal{A}_{\mathrm{s}} : \{0,1\}^n \to \{0,1\}^n$ be the algorithm that given a dataset $x_{1:n}\in\{0,1\}^n$, samples a uniformly random permutation π, computes $z_i = \mathtt{2RR}(x_{\pi(i)})$ for all $i\in[n]$, then outputs $z_{1:n}$. That is, each client simply reports their value using 2RR, and the reports are permuted.

For any $\delta\in[0,1]$, and random variables P and Q, let $D_\infty^\delta(P,Q)$ be the minimal ε such that P and Q are (ε,δ)-indistinguishable. Let ε_δ be the minimal ε such that $\mathcal{A}_{\mathrm{s}}$ is (ε,δ)-DP so

$$\varepsilon_\delta = \max_{X_0,X_1} D_\infty^\delta(\mathcal{A}_{\mathrm{s}}(X_0),\mathcal{A}_{\mathrm{s}}(X_1)),$$

where the maximum is over all possible pairs of neigboring datasets $X_0, X_1 \in \{0,1\}^n$. In the implementation of `2RR, lower bound`, we set

$$X_0 = (0, \ldots, 0) \text{ and } X_1 = (1, 0, \ldots, 0)$$

and compute a lower bound on $D_\infty^\delta(\mathcal{A}_\mathrm{s}(X_0), \mathcal{A}_\mathrm{s}(X_1))$, which in turn, gives a lower bound on ε_δ.

Again, since it is computationally easier to compute the minimal δ for a given ε, we use binary search to find a lower bound on the minimal ε for a given δ. Since we want a lower bound, we use Algorithm 6, which is the same as Algorithm 3, except at the final stage it outputs ε_L rather than ε_R. The final component we need to describe is the function M, which given ε, computes $D_{e^\varepsilon}(\mathcal{A}_\mathrm{s}(X_0), \mathcal{A}_\mathrm{s}(X_1))$. Note that the output of $\mathcal{A}_\mathrm{s}$ is characterized by simply the number of 0s and 1s in the local reports. Thus, the divergence between $\mathcal{A}_\mathrm{s}(X_0) = z_{1:n}^0$ and $\mathcal{A}_\mathrm{s}(X_1) = z_{1:n}^1$ is the same as the divergence between

$$c^0 = \sum_{i=1}^n z_i^0 = \mathrm{Bin}\left(n, \frac{1}{e^{\varepsilon_0}+1}\right) \text{ and } c^1 = \sum_{i=1}^n z_i^1 = \mathrm{Bin}\left(n-1, \frac{1}{e^{\varepsilon_0}+1}\right) + \mathrm{Bern}\left(\frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\right),$$

where $\mathrm{Bern}\left(\frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}\right)$ denotes a Bernoulli random variable, and we are summing the random variables (NOT the pdfs). Since c^0 and c^1 are discrete, provided n is not too large, we can efficiently compute

$$M(\varepsilon) = D_{e^\varepsilon}(c_0, c_1) = \max\left\{\sum_{c=0}^n \max\{c_0(c) - e^\varepsilon c_1(c), 0\}, \sum_{c=0}^n \max\{c_1(c) - e^\varepsilon c_0(c), 0\}\right\}, \tag{22}$$

by explicitly computing the pdfs of c^0 and c^1. Now, since this computation is exact (up to numerical precision), for all $t \in [T]$, Algorithm 6 moves in the right direction at every iterate. This implies that for the duration of the algorithm $D_\infty^\delta(c_0, c_1) \in [\varepsilon^L, \varepsilon^R]$, before finally outputting the lower bound ε^L.

Proposition F.1. *For any $n \in \mathbb{N}, \varepsilon_0 > 0, \delta \in [0,1], T \in \mathbb{N}$, let $\varepsilon =$ `BinSLower`$(\varepsilon_0, \delta, T, M(\cdot))$ be the output of Algorithm 6 where M is given by Equation (22). Then $\varepsilon_\delta \geq \varepsilon$.*

Algorithm 6: Binary Search, `BinSLower`

Input: $\varepsilon_0, \delta, T, M$
$\varepsilon^L = 0$
$\varepsilon^R = \varepsilon_0$
for $t \in [T]$ **do**
  $\varepsilon_t = \frac{\varepsilon^L + \varepsilon^R}{2}$
  $\delta_t = M(\varepsilon_t)$ `// M(ε_t) = the smallest value such that P and Q are (ε_t, δ_t)-indistinguishable`
  if $\delta_t < \delta$ **then**
    $\varepsilon^R = \varepsilon_t$
  else
    $\varepsilon^L = \varepsilon_t$
return ε^L