

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

Vitaly Feldman, Audra McMillan, and Kunal Talwar

Apple

Cupertino, USA

Email: audra.mcmillan@apple.com, ktalwar@apple.com

Abstract—Recent work of Erlingsson, Feldman, Mironov, Raghunathan, Talwar, and Thakurta [1] 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 [2] and has lead to significant interest in the shuffle model of privacy [3], [1].

We give a characterization of the privacy guarantee of the random shuffling of n data records input to epsilon-differentially private local randomizers that significantly improves over previous work and achieves the asymptotically optimal dependence in epsilon. 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. Importantly, our work also yields an algorithm for deriving tighter bounds on the resulting epsilon and delta as well as Rényi differential privacy guarantees. We show numerically that our algorithm gets to within a small constant factor of the optimal bound. As a direct corollary of our analysis we derive a simple and nearly optimal algorithm for frequency 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.

Keywords—differential privacy; privacy-preserving machine learning

I. 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 [2]. The authors of [1] 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 [3], who also formalized a related *shuffle model of privacy*.

The analysis in [1] 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 [1] 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 [1] 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 [5] 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 $\tilde{O}(e^{\varepsilon_0/2}/\sqrt{n})$ [3]. 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 paper² 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

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

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 [3]. 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 [4].

Importantly, our proof is also simpler than the rather delicate approaches in prior work [1], [5]. The proof reduces the problem of analyzing the shuffling privacy guarantee of an adaptive series of local algorithms to analyzing the shuffling

¹The dependence on ε_0 was recently sharpened to $e^{2.5\varepsilon_0}$ in [4].

²A full version of this work can be found at [6]

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 [7]. Another important property of our proof is that it provides a simple and efficient method for numerically computing an amplification bound that is tighter than our closed-form bounds³. Specifically, we show that it is sufficient to analyze the appropriate notion of divergence between two specific distributions over 3 values. As in the case of binary randomized response, the constant 1/2 in the exponent arises naturally from analyzing the divergence between binomial distributions. In Section V we show that this approach leads to numerical bounds that significantly improve on the state of the art. Our approach can be similarly used to numerically compute the privacy amplification guarantee in terms of other notions of privacy. In particular, we use it to compute the privacy amplification guarantee in terms of Rényi differential privacy. In the full version of this paper we demonstrate that this results in tighter privacy guarantees when composing shuffled mechanisms.

Our proof extends naturally to approximate differential privacy. The extension to approximate DP involves an additional technical lemma (Lemma III.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 $\text{TV}(\mathcal{A}(x), \mathcal{A}'(x)) \leq \delta_0$ 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 take existing and existing LDP and immediately obtain an algorithm that achieves (nearly) 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 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 [8]. 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 the full version of this paper, we show an example of such an application for the problem of building a histogram (or estimating the frequency distribution) over an alphabet of size k . For a given level of accuracy, this algorithm simultaneously achieves nearly optimal central as well as local differential privacy bounds. The algorithm is based on applying our results to the algorithm from [9] and has low communication and server-side decoding time. 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 [5] 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 the full version 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. A closed form amplification bound tailored to k -ary randomized response was also given in [5]. Their bound matches the bound presented in this work when $k = O(e^{\varepsilon_0})$ and $\varepsilon_0 > 1$, but is worse than our bound when $k = \Omega(e^{\varepsilon_0})$. To the best of our knowledge, the bound presented in this work is the first closed form result that shows that the privacy amplification, for a fixed ε_0 , improves with k . This corroborates empirical results from [5].

Follow-up work: A natural direction for future work is to analyze privacy amplification by shuffling in terms of other notions of differential privacy, such as Rényi differential privacy (RDP) [10]. RDP guarantees are particularly useful in multi-step algorithms where the composition properties of differential privacy need to be used. A bound on RDP guarantees is implied by the proof technique in [1] but, just as in the case of approximate DP, this bound is asymptotically suboptimal when $\varepsilon_0 > 1$. Specifically, for sufficiently small order α , the α -RDP divergence is $O(\alpha e^{6\varepsilon_0})$ in this regime.

Subsequent work of Girgis, Data, Diggavi, Kairouz, and Suresh [11], (implicitly) relies on the idea proposed in this work: it views each data point as creating a clone of the first element of the first dataset (in the given pair of

³The resulting code is available at <https://github.com/apple/ml-shuffling-amplification>.

Algorithm/Setting	Previous results	This work
Binary randomized response	$\frac{e^{\varepsilon_0/2}}{\sqrt{n}}$ [3]	$\frac{e^{\varepsilon_0/2}}{\sqrt{n}}$ [Thm. III.1]
Non-adaptive shuffle	$\frac{e^{\varepsilon_0}}{\sqrt{n}}$ [5]	
Adaptive shuffle	$\frac{e^{3\varepsilon_0}}{\sqrt{n}}$ [1] $\frac{e^{2.5\varepsilon_0}}{\sqrt{n}}$ [4]	
Adaptive shuffle for $(\varepsilon_0, \delta_0)$ -DP	$\frac{e^{20\varepsilon_0}}{\sqrt{n}}$ [4]	$\left(\frac{e^{\varepsilon_0/2}}{\sqrt{n}}, \delta + n\delta_0\right)$ [Thm. III.8]
k -Randomized response	$\max\left\{\frac{e^{\varepsilon_0}+k}{n}, \sqrt{\frac{e^{\varepsilon_0}+k}{n}}\right\}$ [5]	$\frac{e^{\varepsilon_0}}{\sqrt{n(e^{\varepsilon_0}+k)}}$ [Cor. IV.2]

Table I: 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.

datasets) and thereby reduces the problem to analysis of divergence between a pair of datasets that have just two different elements. The divergence is then analyzed using both analytic and numerical tools. Their resulting bound on the order α RDP is $O(\alpha e^{2\varepsilon_0})$ (when $\varepsilon_0 > 1$). This is a significant improvement on the results in [1] but is still asymptotically suboptimal.

In the full version, we evaluate numerically the Rényi divergence between the pair of distributions given in our reduction and demonstrate that our approach leads to significantly better bounds⁴. We leave the computation of a closed-form bound on the Rényi divergence between the pair of distributions over three values that emerge from our reduction for future work.

Another subsequent work [12] motivated by the problem of obtaining better guarantees for the composition of multiple shuffled protocols relies on a different approach to composition referred to as Fourier accountant [13]. The approach requires computation of the Fast Fourier Transform of the entire privacy loss variable and Koskela, Jälkö, and Honkela [12] use the pair of distributions given in our work as the input to their analysis.

II. 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:

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

⁴The numerical evaluation has not been included in the earlier version of this work and thus not available to the authors of [11].

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 [14].

We will consider several models of differentially private data analysis. In the central model, introduced in [15], 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 II.1 (Differential privacy in the central model). An algorithm $\mathcal{A} : \mathcal{D}^n \rightarrow \mathcal{S}$ 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 [16], 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 [17], [18], [19], [20]. 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 II.2 (Local randomizer). An algorithm $\mathcal{R}: \mathcal{D} \rightarrow \mathcal{S}$ 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 $\mathcal{R}^{(i)}: \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) 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 $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$. 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 $1, \dots, i-1$.

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 [20]). 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 [2], who designed and implemented a principled systems architecture for shuffling. The intuition was formalized in [1], [3]. Erlingsson et al. [1] showed that for arbitrary ε_0 -DP local randomizers random shuffling of reports amplifies the privacy guarantees. Cheu et al. [3] 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 [1], 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 [16]. 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 $(\log(1 + \frac{m}{n}(e^\varepsilon - 1)), \frac{m}{n}\delta)$ -DP [16], [21]. The following lemma is a slightly more general and abstract version of this result.

Lemma II.3 (Advanced joint convexity, [21]). *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

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

III. 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 III.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 III.1.

Theorem III.1. *For any domain $\mathcal{D}$, let $\mathcal{R}^{(i)}: \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an ε_0 -DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_s: \mathcal{D}^n \rightarrow \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$ 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 $\varepsilon_0 \leq \log(\frac{n}{16 \log(2/\delta)})$, $\mathcal{A}_s$ is (ε, δ) -DP, where*

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

Note that when $\varepsilon_0 > 1$, $\varepsilon = O\left(\frac{\sqrt{e^{\varepsilon_0} \log(1/\delta)}}{\sqrt{n}}\right)$ and when $\varepsilon_0 \leq 1$, $\varepsilon = O\left(\varepsilon_0 \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}}\right)$.

A natural question is whether the given bound also holds when $\varepsilon_0 > \log(\frac{n}{16 \log(2/\delta)})$ or, equivalently, $\log(2/\delta) \leq \frac{ne^{\varepsilon_0}}{16}$. In the full version we give numerical evidence that there is no privacy amplification when $\log(1/\delta) = \Omega(ne^{-\varepsilon_0})$.

Our proof relies on a more general result showing that in order to analyze the privacy amplification by shuffling it suffices to upper bound divergence between a specific pair of distributions.

Theorem III.2. For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an ε_0 -DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_s : \mathcal{D}^n \rightarrow \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$ 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 and X_1 be two arbitrary neighboring datasets in $\mathcal{D}^n$. Let $C \sim \text{Bin}(n-1, e^{-\varepsilon_0})$, $A \sim \text{Bin}(C, 1/2)$ and $\Delta \sim \text{Bern}\left(\frac{e^{-\varepsilon_0}}{e^{-\varepsilon_0}+1}\right)$, where $\text{Bern}(p)$ denotes a Bernoulli random variable with bias p . Then there exists a randomized postprocessing algorithm Φ such that $\mathcal{A}_s(X_0)$ is distributed identically to $\Phi(A + \Delta, C - A + 1 - \Delta)$ and $\mathcal{A}_s(X_1)$ is distributed identically to $\Phi(A + 1 - \Delta, C - A + \Delta)$.

In addition to giving our closed-form upper bound, Theorem III.2, provides an efficient method for numerically computing a tighter upper bound. We describe this numerical approach in Section V.

Now let us turn to the proof of Theorems III.1 and III.2. 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 neighboring 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 III.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 III.3, which captures just the privacy amplification due to the generation of clones. The following lemma analyzes the divergence between the distributions on the number of clones that result from this reduction.

For notational brevity, we will frequently use the same symbol to refer to both a random variable and its 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 III.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 $\text{LO}(x)$ then we can apply our reduction. We will then conclude the proof of Theorem III.1 by observing that that every differentially private randomizer has this property.

Lemma III.3. For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an ε_0 -DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_s : \mathcal{D}^n \rightarrow \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$ 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 $X_0 = (x_1^0, x_2, \dots, x_n)$ and $X_1 = (x_1^1, x_2, \dots, x_n)$ 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 $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$, there exists a distribution $\text{LO}^{(i)}(z_{1:i-1}, x)$ such that

$$\begin{aligned} \mathcal{R}^{(i)}(z_{1:i-1}, x) &= \frac{p}{2} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) \\ &\quad + (1-p) \text{LO}^{(i)}(z_{1:i-1}, x). \end{aligned}$$

Then there exists a randomized postprocessing algorithm f such that $\mathcal{A}_s(X_0)$ is distributed identically to $f(A+1, C-A)$ and $\mathcal{A}_s(X_1)$ is distributed identically to $f(A, C-A+1)$, where $C \sim \text{Bin}(n-1, p)$, $A \sim \text{Bin}(C, 1/2)$.

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.

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 $2, \dots, n$ 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 $x_1^0, x_1^1, x_2, \dots, x_n$) such that for y sampled from ρ_b , $f(y)$ is distributed identically to $\mathcal{A}_s(X_b)$. 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 $\mathcal{S} = \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$ is the output space of $\mathcal{A}_s$. 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,

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

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 $\mathcal{A}_s(X_b)$ 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 ρ_0 and ρ_1 are 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, if $C \sim \text{Bin}(n-1, p)$ and $A \sim \text{Bin}(C, 1/2)$, 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 $(A, C - A + 1)$. ■

We can now complete the proof of Theorem III.1. To do this we will use privacy amplification provided by the clones (Lemma III.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 [22] (see also [23]).

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 $\begin{cases} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) & \text{if } y_i = 0; \\ \mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) & \text{if } y_i = 1; \\ \text{LO}^{(i)}(z_{1:i-1}, x_{j_i}) & \text{if } y_i = 2. \end{cases}$
return $z_1, \dots, z_n$

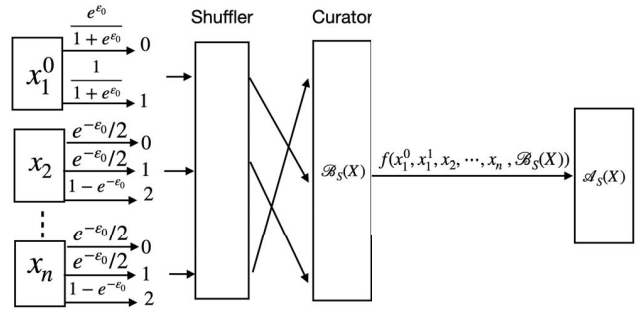


Figure 1: Graphical representation of the reduction in the proof of Theorem III.1

Lemma III.4 ([22]). *Let $\mathcal{R} : \mathcal{D} \rightarrow \mathcal{S}$ be an ε_0 -DP local randomizer and $x_0, x_1 \in \mathcal{D}$. Then there exists a randomized algorithm $\mathcal{Q} : \{0, 1\} \rightarrow \mathcal{S}$ 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 III.2: 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, we can assume without loss of generality that for all $j \neq 1$, $x_j \notin \{x_1^0, x_1^1\}$. This is true since we can add elements x_1^0 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 $\mathcal{A}_s(x_1^b, x_2, \dots, x_n)$ is identical to that of $\mathcal{A}_s(X_b)$.

By Lemma III.4 we have that for every i and value of

auxiliary input $z_{1:i-1}$ there exists a randomized algorithm $\mathcal{Q}^{(i)}(z_{1:i-1}, \cdot): \{x_1^0, x_1^1\} \rightarrow \mathcal{S}^{(i)}$ such that

$$\begin{aligned} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) \\ = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0} + 1} \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \frac{1}{e^{\varepsilon_0} + 1} \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) \end{aligned} \quad (2)$$

and

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

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 $\mathcal{R}: \mathcal{D} \rightarrow \mathcal{S}$ and any input $x_0 \in \mathcal{D}$, there exists a randomized algorithm $\text{LO}: \mathcal{D} \rightarrow \mathcal{S}$ such that $\mathcal{R}(x)$ can be decomposed as

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

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

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

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 $\text{LO}^{(i)}$ such that for $p = e^{-\varepsilon_0}$ we have:

$$\begin{aligned} \mathcal{R}^{(i)}(z_{1:i-1}, x) &= \frac{p}{2} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2} \mathcal{R}^{(i)}(z_{1:i-1}, x_1^1) \\ &\quad + (1 - p) \text{LO}^{(i)}(z_{1:i-1}, x). \end{aligned}$$

Note that

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

and therefore we also obtain that

$$\begin{aligned} \mathcal{R}^{(i)}(z_{1:i-1}, x) &= \frac{p}{2} \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2} \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^1) \\ &\quad + (1 - p) \text{LO}^{(i)}(z_{1:i-1}, x), \end{aligned} \quad (4)$$

Next we consider the algorithm $\mathcal{A}_Q$ which is defined in the same way as $\mathcal{A}_s$, 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 $\mathcal{R}_Q^{(i)}$ as follows: For all $x \in \mathcal{D}$, $i \in [n]$ and values of $z_{1:i-1}$ we let

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

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

Equations (2) and (3), allow us to decompose $\mathcal{A}_s(X_0)$ and $\mathcal{A}_s(X_1)$ into the mixture of two components as follows:

$$\mathcal{A}_s(X_0) = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0} + 1} \mathcal{A}_Q(X_0) + \frac{1}{e^{\varepsilon_0} + 1} \mathcal{A}_Q(X_1) \quad (5)$$

and

$$\mathcal{A}_s(X_1) = \frac{1}{e^{\varepsilon_0} + 1} \mathcal{A}_Q(X_0) + \frac{e^{\varepsilon_0}}{e^{\varepsilon_0} + 1} \mathcal{A}_Q(X_1).$$

Note that by eq. (4), for all $x \notin \{x_1^0, x_1^1\}$,

$$\begin{aligned} \mathcal{R}_Q^{(i)}(z_{1:i-1}, x) &= \frac{p}{2} \mathcal{R}_Q^{(i)}(z_{1:i-1}, x_1^0) + \frac{p}{2} \mathcal{R}_Q^{(i)}(z_{1:i-1}, x_1^1) \\ &\quad + (1 - p) \text{LO}^{(i)}(z_{1:i-1}, x). \end{aligned}$$

Therefore, by Lemma III.3, there exists a postprocessing function f such that $\mathcal{A}_Q(X_0)$ is distributed identically to $f(A + 1, C - A)$ and $\mathcal{A}_Q(X_1)$ is distributed identically to $f(A, C - A + 1)$, where $C \sim \text{Bin}(n - 1, e^{-\varepsilon_0})$, $A \sim \text{Bin}(C, 1/2)$. Now, the decomposition in eq. (4) implies that $\mathcal{A}_s(X_0)$ is distributed identically to $f(A + \Delta, C - A + 1 - \Delta)$ and $\mathcal{A}_s(X_1)$ is distributed identically to $f(A + 1 - \Delta, C - A + \Delta)$, $\Delta \sim \text{Bern}\left(\frac{e^{\varepsilon_0}}{e^{\varepsilon_0} + 1}\right)$. ■

To complete the proof of Theorem III.1, all we need is the following lemma on the hockey-stick divergence between the distributions resulting from Theorem III.2 and the post-processing inequality for the hockey-stick divergence.

Lemma III.5. *Let $n \in \mathbb{N}$, $\delta \in (0, 1]$ and $\varepsilon_0 \leq \log\left(\frac{n}{16 \log(2/\delta)}\right)$. Consider the process where we sample $C \sim \text{Bin}(n - 1, e^{-\varepsilon_0})$, $A \sim \text{Bin}(C, 1/2)$ and $\Delta \sim \text{Bern}\left(\frac{e^{\varepsilon_0}}{e^{\varepsilon_0} + 1}\right)$. Let $P = (A + \Delta, C - A + 1 - \Delta)$ and $Q = (A + 1 - \Delta, C - A + \Delta)$, then P and Q are (ε, δ) -indistinguishable for*

$$\varepsilon = \log \left(1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1} \left(\frac{8\sqrt{e^{\varepsilon_0} \log(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_0}}{n} \right) \right). \quad (6)$$

The proof of Lemma III.5 can be found in the full version of the paper.

A. Generalization to Approximate Differential Privacy

In order to analyze 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 in total variation distance to the output of n ε'_0 -DP randomizers. A fairly involved reduction of this type is given in [3] (building on the results from [24]) 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. [4] 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 [25]. Its tightest form is best stated in the add/delete variant of differential privacy which was defined for local randomizers in [20].

Definition III.6 (Local Deletion Differential Privacy). An algorithm $\mathcal{R}: \mathcal{D} \rightarrow \mathcal{S}$ 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 II.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 III.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 , $\text{TV}(\mathcal{R}(x), \mathcal{R}'(x)) \leq \delta$. In particular, $\mathcal{R}'$ is a (replacement) 2ε -DP local randomizer.

The proof can be found in the full version of this paper. 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 III.7 directly in our proof of the amplification bound. This leads to the following resulting bound which we prove in full version of this paper.

Theorem III.8. For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)}: \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a $(\varepsilon_0, \delta_0)$ -DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_s: \mathcal{D}^n \rightarrow \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$ 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 $\varepsilon_0 \leq \log(\frac{n}{16 \log(2/\delta)})$, $\mathcal{A}_s$ is $(\varepsilon, \delta + (e^\varepsilon + 1)(1 + e^{-\varepsilon_0}/2)n\delta_0)$ -DP, where ε is as in Equation (1).

Notice that the bound on ε in Theorem III.8 matches the bound in Theorem III.1. A natural question is if the dependence of δ on n and ε_0 is optimal. We leave this question for future work.

IV. 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 IV.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 [5]. As with Theorem III.1, Theorem IV.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. We will use $\text{MultNom}(n; p_1, \dots, p_k)$ to denote the multinomial distribution over k -tuples with n samples and probabilities of each of the outcomes being $p_1, \dots, p_k$ (where $\sum_{i \in [k]} p_k = 1$).

Theorem IV.1. For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)}: \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a ε_0 -DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_s: \mathcal{D}^n \rightarrow \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$ 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 $X_0 = (x_1^0, x_2, \dots, x_n)$ and $X_1 = (x_1^1, x_2, \dots, x_n)$ 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 $p \in (0, 1/3]$ and $q \in (0, 1)$ such that for all $i \in [n]$ and $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$, 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 $b \in \{0, 1\}$,

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

and for all $x \in \mathcal{D} \setminus \{x_1^0, x_1^1\}$, there exists a distribution $\text{LO}(i)(z_{1:i-1}, x)$ such that

$$\begin{aligned} \mathcal{R}^{(i)}(z_{1:i-1}, x) &= p\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^0) + p\mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^1) \\ &\quad + p\mathcal{Q}_0^{(i)}(z_{1:i-1}) + (1 - 3p)\text{LO}_b^{(i)}(z_{1:i-1}, x). \end{aligned} \quad (8)$$

Let $(A, B, C, D) \sim \text{MultNom}(n - 1; p, p, p, 1 - 3p)$ and $\Gamma \sim \text{Bern}(q)$. There exists a randomized postprocessing algorithm Φ such that $\mathcal{A}_s(X_0)$ is distributed identically to $\Phi(A + \Gamma, B, C + 1 - \Gamma)$ and $\mathcal{A}_s(X_1)$ is distributed identically to $\Phi(A, B + \Gamma, C + 1 - \Gamma)$. In particular, for any $\delta \in [0, 1]$ such that $p \geq \frac{8 \ln(2/\delta)}{n}$, $\mathcal{A}_s(X_0)$ is $(\varepsilon, q\delta)$ -indistinguishable

from $\mathcal{A}_s(X_1)$, where

$$\varepsilon \leq \ln \left(1 + q \left(\frac{4\sqrt{2\log(4/\delta)}}{\sqrt{pn}} + \frac{4}{pn} \right) \right). \quad (9)$$

The reduction to analysis of the divergence between a pair of multinomial distributions in the proof of Theorem IV.1 is achieved using the same argument as the one we used to prove Theorem III.2. The analysis of the resulting divergence relies on decomposing each of the resulting multinomial distributions into a mixture of two distributions. We then show that distributions we analyzed when proving Lemma III.5 can be postprocessed into pairs of components of these mixtures. This allows us to show that all components of these mixtures are (ε', δ') -indistinguishable. Joint convexity can then be used to obtain the bound. The details of the proof appear in full version of this paper.

The main strength of Theorem IV.1 is that it allows us to prove tighter bounds than Theorem III.1 for specific local randomizers. In the next section, we will show how Theorem IV.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 III.1, although with a slightly worse constant factor. The decomposition in Lemma III.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 neighboring datasets X_0 and X_1 such that $x_0^i \neq x_1^i$, there exist distributions $\mathcal{Q}^{(i)}(z_{1:i-1}, x_0^i)$ and $\mathcal{Q}^{(i)}(z_{1:i-1}, x_1^i)$ that satisfy equations (2) and (3). Thus, by setting

$$\mathcal{Q}_0^{(i)}(z_{1:i-1}) = \frac{1}{2} \left(\mathcal{Q}^{(i)}(z_{1:i-1}, x_0^i) + \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^i) \right)$$

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

$$\begin{aligned} \mathcal{R}^{(i)}(z_{1:i-1}, x) &= \frac{1}{2e^{\varepsilon_0}} \mathcal{Q}^{(i)}(z_{1:i-1}, x_0^i) + \frac{1}{2e^{\varepsilon_0}} \mathcal{Q}^{(i)}(z_{1:i-1}, x_1^i) \\ &\quad + \left(1 - \frac{1}{e^{\varepsilon_0}} \right) \text{LO}^{(i)}(z_{1:i-1}, x) \\ &= \frac{1}{3e^{\varepsilon_0}} \mathcal{Q}_0^{(i)}(z_{1:i-1}) + \frac{1}{3e^{\varepsilon_0}} \mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^b) \\ &\quad + \frac{1}{3e^{\varepsilon_0}} \mathcal{Q}_1^{(i)}(z_{1:i-1}, x_1^{1-b}) \\ &\quad + \left(1 - \frac{1}{e^{\varepsilon_0}} \right) \text{LO}^{(i)}(z_{1:i-1}, x). \end{aligned}$$

Letting $q = \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1}$ and $p = e^{-\varepsilon_0}/3$ and assuming $\varepsilon_0 \leq \ln \left(\frac{n}{24 \ln(2/\delta)} \right)$ (which implies $p \geq \frac{8 \ln(2/\delta)}{n}$),

Equation (9) gives the bound

$$\varepsilon \leq \ln \left(1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1} \left(\frac{4\sqrt{6e^{\varepsilon_0} \log(4/\delta)}}{\sqrt{n}} + \frac{12e^{\varepsilon_0}}{n} \right) \right),$$

which matches the bound stated in Equation (1) of Theorem III.1 up to a factor of $\sqrt{3/2}$.

A. k -Randomized Response

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

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

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 IV.2. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)}: \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow \mathcal{S}^{(i)}$ for $i \in [n]$ (where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$) be a sequence of algorithms such that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is a ε_0 -DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_s: \mathcal{D}^n \rightarrow \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(n)}$ 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 $f^{(i)}: \mathcal{S}^{(1)} \times \dots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \rightarrow [k]$ such that $\mathcal{R}^{(i)}(z_{1:i-1}, x) = \text{kRR}(f^{(i)}(z_{1:i-1}, x))$. Then for $\delta \in [0, 1]$ such that $\varepsilon_0 \leq \log \left(\frac{n}{16 \log(2/\delta)} \right)$, $\mathcal{A}_s$ is (ε, δ) -DP where*

$$\varepsilon \leq \log \left(1 + (e^{\varepsilon_0} - 1) \left(\frac{4\sqrt{2(k+1)\log(4/\delta)}}{\sqrt{(e^{\varepsilon_0} + k - 1)kn}} + \frac{4(k+1)}{kn} \right) \right) \quad (10)$$

Notice that when k is small, this bound matches that given in Theorem III.1, but when k and ε_0 are large, ε scales like $\frac{e^{\varepsilon_0}}{\sqrt{kn}}$. The proof of Corollary IV.2 can be found in the full version of this paper. The key observation is that if we let ν be the uniform distribution on $[k]$, then for any $x \in [k]$,

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

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

V. NUMERICAL RESULTS

Theorem III.2 provides an efficient method for numerically computing an amplification bound that is tighter than our closed-form bound. Our implementation is outlined in the full version of this paper, but the main component is numerically computing the indistinguishability bound for multinomial random variables from Lemma III.5. 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 III.1 and

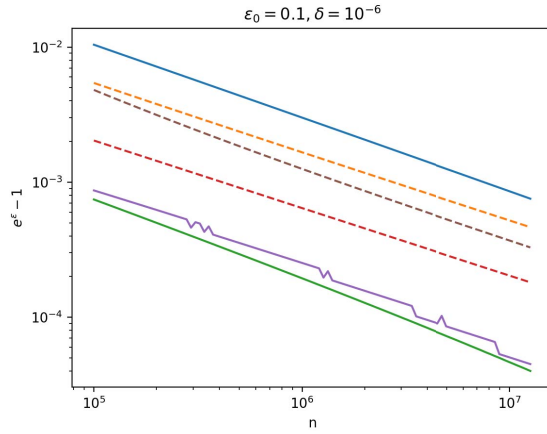
Clones is the numerical version derived from Theorem III.2. Also, BBGN'19 is the privacy amplification bound for general algorithms given in [5]⁵ and BKMTT'20 is the bound proven in [4]. Finally, CSUZZ'19, 2RR is the closed form amplification bound for binary randomized response proved in [3] and 2RR, 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 the full version of this paper. We do not include the bounds from [1] in the comparison since [4] gives a tighter bound for the same analysis.

In all parameter regimes tested, Clones 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 closed-form bounds 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 closely tracks 2RR, lower bound, particularly for small ϵ_0 . We note in all three graphs in Figure 2, Clones 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 full version of this paper.

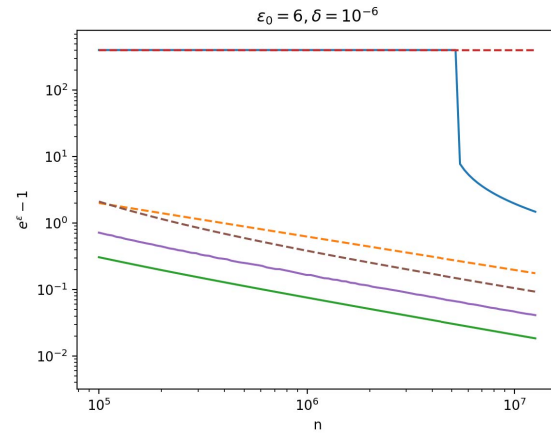
REFERENCES

- [1] 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*, ser. SODA '19. USA: Society for Industrial and Applied Mathematics, 2019, p. 2468–2479.
- [2] 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*, ser. SOSP '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 441–459. [Online]. Available: <https://doi.org/10.1145/3132747.3132769>
- [3] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Advances in Cryptology – EUROCRYPT 2019*, Y. Ishai and V. Rijmen, Eds. Cham: Springer International Publishing, 2019, pp. 375–403.
- [4] B. Balle, P. Kairouz, H. McMahan, O. Thakkar, and A. Thakurta, "Privacy amplification via random check-ins," *ArXiv*, vol. abs/2007.06605, 2020.
- [5] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," in *Advances in Cryptology – CRYPTO 2019*, A. Boldyreva and D. Micciancio, Eds. Cham: Springer International Publishing, 2019, pp. 638–667.
- [6] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," *CoRR*, vol. abs/2012.12803, 2020. [Online]. Available: <https://arxiv.org/abs/2012.12803>
- [7] P. Kairouz, S. Oh, and P. Viswanath, "Differentially private multi-party computation," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 128–132.
- [8] 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)*, ser. Lecture Notes in Computer Science, vol. 4004. Springer Verlag, May 2006, pp. 486–503. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/our-data-ourselves-privacy-via-distributed-noise-generation/>
- [9] J. Acharya, Z. Sun, and H. Zhang, "Hadamard response: Estimating distributions privately, efficiently, and with little communication," ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 1120–1129.
- [10] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 2017, pp. 263–275.
- [11] A. M. Girgis, D. Data, S. Diggavi, A. T. Suresh, and P. Kairouz, "On the renyi differential privacy of the shuffle model," 2021.
- [12] A. Koskela, M. A. Heikkilä, and A. Honkela, "Tight accounting in the shuffle model of differential privacy," 2021.
- [13] A. Koskela, J. Jälkö, and A. Honkela, "Computing tight differential privacy guarantees using fft," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2560–2569.
- [14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, Aug. 2014.
- [15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.
- [16] 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.
- [17] 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*, ser. CCS '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1054–1067.
- [18] Apple's Differential Privacy Team, "Learning with privacy at scale," *Apple Machine Learning Journal*, vol. 1, no. 9, 2017.

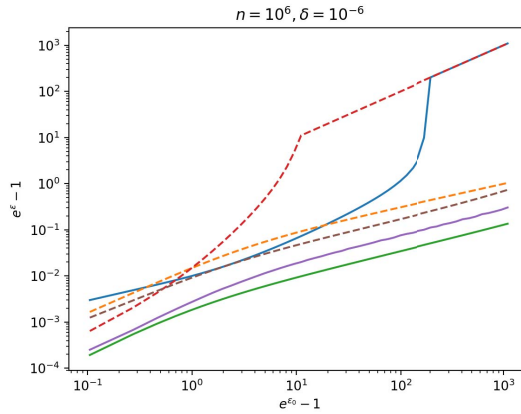
⁵BBGN'19 curves were produced using open source code released by Balle, Bell, Gascón, and Nissim [5], they correspond to the curve (Bennett, Generic) in [5].



(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.

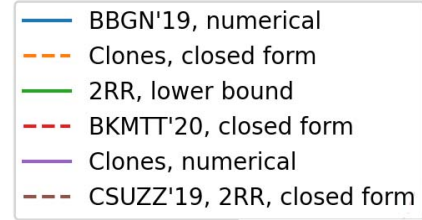


Figure 2: Comparison of new privacy amplification by shuffling bounds given in this work to bounds given in [5] and [4] and to bounds specific to 2RR given in [3]. Dotted lines represent closed-form bounds, while solid lines represent bounds that were computed numerically.

- [19] B. Ding, J. Kulkarni, and S. Yekhanin, “Collecting telemetry data privately,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 3574–3583.
- [20] 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.
- [21] B. Balle, G. Barthe, and M. Gaboardi, “Privacy profiles and amplification by subsampling,” *Journal of Privacy and Confidentiality*, vol. 10, no. 1, Jan. 2020.
- [22] P. Kairouz, S. Oh, and P. Viswanath, “The composition theorem for differential privacy,” ser. *Proceedings of Machine Learning Research*, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1376–1385.
- [23] 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.
- [24] M. Bun, J. Nelson, and U. Stemmer, “Heavy hitters and the structure of local privacy,” *ACM Transactions on Algorithms (TALG)*, vol. 15, no. 4, pp. 1–40, 2019.
- [25] A. Cheu and J. R. Ullman, “The limits of pan privacy and shuffle privacy for learning and estimation,” *CoRR*, vol. abs/2009.08000, 2020. [Online]. Available: <https://arxiv.org/abs/2009.08000>